

# USER SIMULATIONS IN SEARCH SESSIONS

by

Mustafa Zengin

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Summer 2019

© 2019 Mustafa Zengin  
All Rights Reserved

## USER SIMULATIONS IN SEARCH SESSIONS

by

Mustafa Zengin

Approved: \_\_\_\_\_  
Kathleen McCoy, Ph.D.  
Chair of the Department of Computer and Information Sciences

Approved: \_\_\_\_\_  
Levi T. Thompson, Ph.D.  
Dean of the College of Engineering

Approved: \_\_\_\_\_  
Douglas J. Doren, Ph.D.  
Interim Vice Provost for Graduate and Professional Education and  
Dean of the Graduate College

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: \_\_\_\_\_  
Benjamin A. Carterette, Ph.D.  
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: \_\_\_\_\_  
Evangelos Kanoulas, Ph.D.  
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: \_\_\_\_\_  
Hui Fang, Ph.D.  
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: \_\_\_\_\_  
Vijay Shanker, Ph.D.  
Member of dissertation committee

## ACKNOWLEDGEMENTS

Working as a Ph.D. student, publishing research papers and completing a doctoral dissertation were magnificent as well as challenging experiences to me. During this journey I met, discussed and exchanged ideas with many invaluable people. All this hard work could not be accomplished without precious help and support of these personalities. In here I would like to express my gratitude and appreciation to them.

First of all, I am deeply grateful to my advisor, Dr. Ben Carterette, for his support, patience and encouragement during this long Ph.D. work. His guidance in approaching a research problem, asking correct questions and solving them have helped me to become a better scientist. I am very thankful for his feedbacks on my ideas, suggestions and the most importantly the freedom he provided in my research. This dissertation thesis would not be possible without his support.

I am thankful to my committee members Dr. Hui Fang, Dr. Vijay Shanker and Dr. Evangelos Kanoulas for their invaluable time and their feedback throughout this dissertation thesis.

I would also like to thank my former lab-mates Karankumar Shabnani, Ashraf Bah, Praveen Chandar, Ashwani Rao, Mohammad Alsulmi, Dongqing Zhu with whom I exchanged ideas.

Finally, I would like to acknowledge with gratitude, the support and love of my family; my parents Elmas and Mehmet and my sister Meliha and brother in-law Evrim.

## TABLE OF CONTENTS

<b>LIST OF TABLES</b> . . . . .	<b>ix</b>
<b>LIST OF FIGURES</b> . . . . .	<b>xi</b>
<b>ABSTRACT</b> . . . . .	<b>xiii</b>
 <b>Chapter</b>	
<b>1 INTRODUCTION</b> . . . . .	<b>1</b>
1.1 User Click Prediction . . . . .	3
1.2 User Dwell Time Prediction . . . . .	3
1.3 Session Abandonment Prediction . . . . .	4
1.4 A New Session Dataset . . . . .	4
1.5 Query Generation and Complete Session Simulation . . . . .	4
<b>2 RELATED WORK</b> . . . . .	<b>6</b>
2.1 Traditional Ad-hoc Information Retrieval . . . . .	6
2.1.1 Boolean Model . . . . .	6
2.1.2 Vector Space Model . . . . .	7
2.1.3 BM25 . . . . .	8
2.1.4 Language Model . . . . .	9
2.2 Learning To Rank . . . . .	10
2.2.1 Pointwise approach . . . . .	11
2.2.2 Pairwise approach . . . . .	11
2.2.3 Listwise approach . . . . .	12
2.2.4 LETOR collection and feature set . . . . .	12
2.2.5 Reading level document features . . . . .	13
2.3 Clueweb12 Collection . . . . .	13
2.4 The New York Times Collection . . . . .	15

2.5	User Logs and the TREC Session Track . . . . .	15
2.6	User Click Detection . . . . .	16
2.7	Dwell Time Prediction . . . . .	18
2.8	Session Abandonment Prediction . . . . .	19
2.9	Query Generation and Session Simulation . . . . .	20
<b>3</b>	<b>PREDICTING USER CLICKS IN IDEAL SESSIONS . . . . .</b>	<b>23</b>
3.1	Experiment Data . . . . .	24
3.2	Methods . . . . .	24
3.2.1	Session Features . . . . .	24
3.2.2	Other Session Features . . . . .	25
3.3	Models . . . . .	26
3.4	Experiments . . . . .	27
3.5	Results . . . . .	28
3.6	Conclusions . . . . .	32
<b>4</b>	<b>USER DWELL TIME PREDICTION . . . . .</b>	<b>33</b>
4.1	Experiment Data . . . . .	33
4.2	Models . . . . .	34
4.3	Experiments . . . . .	34
4.3.1	Results . . . . .	35
4.4	Conclusions . . . . .	38
<b>5</b>	<b>REFORMULATE OR QUIT: PREDICTING USER ABANDONMENT IN IDEAL SESSIONS . . . . .</b>	<b>39</b>
5.1	Experiment Data . . . . .	39
5.2	Methods . . . . .	40
5.2.1	LETOR Features . . . . .	41
5.2.2	Session and interaction features . . . . .	42
5.2.2.1	Query approximation . . . . .	42
5.2.2.2	Relevance related features . . . . .	43
5.2.2.3	Duration-based features . . . . .	43
5.2.2.4	Other sessions based features . . . . .	43

5.2.2.5	Other features . . . . .	44
5.3	Experiments . . . . .	44
5.4	Results . . . . .	45
5.5	Conclusions . . . . .	47
<b>6</b>	<b>EVALUATION OF MODELS WITH A NEW DATASET . . . . .</b>	<b>48</b>
6.1	Session Data Collection . . . . .	48
6.1.1	Corpus . . . . .	48
6.1.2	Topics . . . . .	49
6.1.3	Sessions . . . . .	49
6.2	Evaluation of User Click Models . . . . .	51
6.2.1	Features . . . . .	51
6.2.2	Models . . . . .	52
6.2.3	Experiments . . . . .	52
6.2.4	Results . . . . .	53
6.3	Evaluation of Session Abandonment Prediction . . . . .	54
6.3.1	Features . . . . .	56
6.3.2	Experiments . . . . .	56
6.3.3	Results . . . . .	56
6.4	Evaluation of Dwell Time Prediction . . . . .	57
6.4.1	Features . . . . .	57
6.4.2	Models . . . . .	58
6.4.3	Experiments . . . . .	58
6.4.4	Results . . . . .	58
6.5	Conclusions . . . . .	59

<b>7</b>	<b>QUERY SIMULATION</b>	<b>62</b>
7.1	Simulating Queries	62
7.1.1	Topical Language Model	63
7.1.1.1	Other Models	64
7.1.2	Sampling Queries	66
7.1.3	Scoring Sampled Queries	66
7.1.4	Summary	68
7.2	Framework and Data	69
7.3	Session Abandonment	69
7.4	Evaluation	70
7.4.1	Ad hoc evaluation using simulated queries	71
7.4.2	Session evaluation using simulated queries	71
7.4.3	Ranking systems using simulated queries	73
7.4.4	Actual sessions vs simulated sessions	74
7.5	Conclusions	74
<b>8</b>	<b>A COMPLETE SESSION SIMULATION</b>	<b>77</b>
8.1	User Search Session Generation	77
8.1.1	Data and Framework	78
8.1.2	Query Generation	78
8.1.3	Click Generation	79
8.1.4	Dwell Time Generation	80
8.1.5	Session Length Generation	80
8.2	User Experiment	80
8.3	Results	82
8.4	Conclusions	85
<b>9</b>	<b>CONCLUSIONS AND FUTURE WORK</b>	<b>87</b>
	<b>BIBLIOGRAPHY</b>	<b>91</b>

## LIST OF TABLES

2.1	List of features in the LETOR collection. . . . .	14
3.1	Corpus statistics . . . . .	24
3.2	List of session related features. . . . .	25
3.3	List of features that are retrieved from other user sessions. . . . .	26
3.4	Nine different click models derived from combining feature sets from two different groups: the search results page and the session history. s.h. stands for session history and c.a.s stands for clicks across sessions. . . . .	26
3.5	Precision/recall of our nine models, c.a.s. stands for clicks across sessions . . . . .	29
3.6	AUC of our nine models, c.a.s. stands for clicks across sessions. . .	29
3.7	NYT-Precision/recall of our nine models, c.a.s. stands for clicks across sessions . . . . .	29
3.8	NYT-AUC of our nine models, c.a.s. stands for clicks across sessions.	30
4.1	RMSE of dwell time estimation of methods and their comparisons to mean and median dwell times. . . . .	35
5.1	List of session and interaction features. . . . .	44
5.2	Precision/recall/AUC/accuracy of our four models . . . . .	45
6.1	Precision/recall of our nine models, c.a.s. stands for clicks across sessions . . . . .	54
6.2	AUC of our nine models, c.a.s. stands for clicks across sessions. . .	55

6.3	Precision/recall/AUC/accuracy of our four models . . . . .	57
6.4	RMSE of dwell time estimation of methods and their comparisons to mean and median dwell times. . . . .	59
7.1	Top 4 most frequent terms appearing in queries for three topics with their binomial occurrence model probability $P(w \in Q)$ , their multinomial language model probability $P(w Q)$ , and their frequency in queries sampled using the procedure in Section 6.1.1. . . . .	67
7.2	Overall precision@10 averaged across all sessions and all rounds in each session for each of our six systems. . . . .	73
7.3	Examples of sequences of queries for six topics in our set. For each topic we show an actual user session of queries and a simulated session. . . . .	75
8.1	Session review satisfaction rating agreement between participants. . . . .	83
8.2	Session type agreement between participants. . . . .	83
8.3	Session type prediction decision table. . . . .	85

## LIST OF FIGURES

3.1	ROC curves for different click detection methods . . . . .	30
3.2	Features ranked by mean raw importance for the positive class for independence+session experiment. Yellow represents other session related features, green represents query-URL related features, grey represents query-snippet related features, black represents document related features, blue represents URL related features, red represents session related features, purple represents snippet related features, pink represents query-title related features, salmon represents title related features and orange represents search engine related features.	31
4.1	Density of actual dwell times by relevance class. . . . .	35
4.2	Density of actual dwell times generated by independence model by relevance class. . . . .	36
4.3	Density of actual dwell times generated by dependence model by relevance class. . . . .	36
4.4	Density of actual dwell times generated by dependence model along with click information and reading level features by relevance class.	37
5.1	Histogram of number of interactions per session . . . . .	41
5.2	Feature importance based on mean decrease in accuracy. Key: blue: duration features; green: features extracted using other sessions with same topic; black: interaction id; purple: query approximation features; red: relevance features; yellow: duration and relevance features . . . . .	46
5.3	ROC curves for different session abandonment prediction methods .	46
5.4	Top 10 features according to mean decrease in accuracy . . . . .	47
6.1	User interface of The New York Times data search engine . . . . .	51

6.2	Histogram of number of interactions per session . . . . .	52
6.3	ROC curves of models from left to right and top to bottom . . . . .	55
6.4	Density of actual dwell times by relevance class. . . . .	60
6.5	Density of actual dwell times generated by dependence model by relevance class. . . . .	60
7.1	Distributions of query lengths for a selection of topics (given as a topic number from the TREC 2014 Session track along with a brief topic description). Each bar plot shows the number of queries of lengths 1 through 10. . . . .	63
7.2	Comparison of six retrieval systems across six rounds of a session using non-simulated queries . . . . .	72
7.3	Comparison of six retrieval systems across six rounds of a session using simulated queries . . . . .	72
8.1	User interface of The New York Times data search engine . . . . .	78
8.2	User interface of The New York Times data search engine . . . . .	81
8.3	Average satisfaction rates of actual and generated sessions by topic	83

## ABSTRACT

When users interact with the search engine to satisfy their information need, in most cases they reformulate their queries many times before abandoning the search due to their evolving information need, or low quality search results, or broad information need that needs many reformulations or some other unclear reasons we do not know. During the search process, users inspect the listed snippets on search result pages, click to the documents that they find relevant and attractive, spend time on the clicked documents and eventually abandon the search.

User models have been a vital part of information retrieval research for understanding users' behavior on search process and assisting them in retrieving useful information for their information need. In this thesis, we present several efforts for building user models in information search scenario. We first introduce models that predict a single click on a search engine result page that make use of other snippets that are listed along with the snippet we predict user click, users' search history and other users' search histories. Then we propose models that estimate user dwell time on clicked documents using document, snippet and session features.

Next, we model user search session abandonment by using users' defined information needs, submitted queries, session features, document features and features extracted from other users' sessions. We show that session and query features have major impact on predicting users' session abandonment decision.

Our next endeavor consists in generating user search queries when information needs are clearly defined. We use a two-phase process by which we first generate queries by sampling from a language model, and then score them based on their discriminative power among topics. Evaluation of the query generation methods is the hardest part but for this task we care most about whether the queries we generate were "good" for

evaluating retrieval systems, and in particular, whether they were good for evaluating systems that use features derived from session history. We found that generated search queries create user search sessions very similar to actual search sessions.

Finally, our last effort consists in modeling complete artificial user search sessions by simulating query reformulations, user click decisions, user dwell times and session abandonments. This phase is the most important phase because we put all our machine learned user models together to create artificial search sessions and search answers to following questions; (1) "Can artificially created search sessions satisfy user information needs?" and (2) "Can users classify artificial and actual user sessions?" with a user experiment. For user experiment, we pick 48 actual sessions for 8 topics randomly from a user search session corpus and merge these actual search sessions with the generated 48 search sessions in a corpus. Then we ask users to follow the search sessions and provide their search satisfaction in terms of scores 1 to 5 such that 1 means no satisfaction and 5 means complete satisfaction. We also ask users to predict the provided session type. They answer to this question by selecting "C" for computer-generated session and "A" for actual user search session. Our findings show that search session type has no significant importance on user search satisfaction and users are barely better than random in predicting search session type.

## Chapter 1

### INTRODUCTION

An information retrieval (IR) system's main duty is assisting users in retrieving useful information for their information need. The sought information can be in many forms such as web documents, images, videos, audio files or music playlists. Most of the time the information need is expressed by users as queries in text, vocal form or selecting an option such as "*more like this*" from a provided user interface. Apart from the systems that suggest information, users usually form queries in two or three words [36]. This allows users to spend less effort in typing and allows systems to respond quickly, but it fails to fully capture users' real information needs. As a result, users express their needs with very ambiguous queries, and different information needs are often mapped into similar search queries. Thus, users' specific information needs are lost [8].

Swanson et al. [66] claims that document retrieval is a trial and error process and an initial search request is just a prediction of attributes a desired document expected to have. Response of the system is used to correct the initial request for another try. Prior work supports this claim and has shown that effective query reformulations improves overall search results [59] and 52% of users reformulate their queries [35]. Other than improving the search result quality, users that tackle on complex tasks may adopt a divide and conquer strategy and reformulate queries many times in order to gather information on different facets of the information need [6]. Sometimes, users do not have clear goals and their needs are not well-defined in their minds. Their search process is open-ended and exploratory in nature which requires many reformulations [72]. For these and many other reasons, users perform query reformulations and interact with the search engine many times.

Kanoulas et al. [39] defines the word "*session*" to mean a sequence of query reformulations along with user interactions with the retrieved search results in order to satisfy an information need. We use the term "*session*" in the same meaning in this document.

Since information searching behaviour is complex and consists of many interactions with the search engine for different motivations, understanding the user information need and predicting user actions in search sessions has become an important area in information retrieval applications and research to improve overall search satisfaction. For instance, predicting the documents that users are likely to click give opportunities to search engines to show those results on higher ranks.

In our thesis, we conduct several studies aimed at simulating and modeling user actions on search engines in session context. The user actions we target are creating queries, clicking to documents, inspecting clicked documents, abandoning search sessions and finally complete information searching when information need is clearly provided. We demonstrate how to incorporate the context of the full session into predictions, in order to capture the user's evolving needs as they interact with the system. This leads us towards full simulations of user sessions with the ultimate goal of obtaining a better understanding of how search engines can help users make better use of search engines.

This dissertation is organized as follows: after reviewing previous work in Chapter 2, we begin by investigating models for predicting clicks (Chapter 3), dwell times (Chapter 4), and session abandonment (Chapter 5) using features of search sessions. Chapter 6 applies our previous models to an alternative search setting. In Chapter 7 we turn our attention to simulating actual user queries for a given information need. Finally, Chapter 8 incorporates all of these models to produce full simulations of user sessions.

## 1.1 User Click Prediction

User click models predict whether a user click is going to happen or not on a given document that is listed on a search engine result page (SERP). They have been used widely in information retrieval as a way to improve document ranking (e.g., by inferring document relevance from clicks predicted by a click model), to improve evaluation metrics (e.g., model-based metrics) and to better understand users by inspecting the parameters of click models [17]. In Chapter 3, we introduce several machine learned models that predict a single user click on a snippet that is listed on a SERP in several different scenarios. We consider two different main settings: (1) only considering SERP; (2) considering session history, (3) considering session history with sessions of other users along with the SERP. For each of these main settings we consider three different settings (1) only considering a snippet by itself; (2) considering all snippets on a results page along with the snippet that we make click prediction; and (3) considering the clicks in a SERP. As a result we test nine different models derived from combining feature sets from two different groups. We found that a dependency model which uses features of all snippets in a results page is a better click model over one that uses only features of a target document, and furthermore that using features of the session history and of other sessions on the same topic uniformly improve precision and area under the receiver operating characteristic curve (AUC).

## 1.2 User Dwell Time Prediction

Dwell time on a web page is the actual duration that a user spends on a page after clicking its URL that is listed on a SERP. Dwell time on web pages has been used extensively for various information retrieval tasks, research, marketing and advertising [74][1][45][76]. Research on users revealed that dwell time is generally the most significant indicator of document relevance besides clickthrough data [40][41]. In Chapter 4, we introduce machine learned models that estimate user dwell time on clicked documents based on the snippet and document features.

### 1.3 Session Abandonment Prediction

All search engine users eventually abandon a search. This may be due to obtaining the information they sought throughout the search session or due to dissatisfaction with the results shown by the search engine or simply due to the amount of effort spent on searching. Detecting when and why a user is close to abandoning a search session is an important problem in search for information needs in order for a system to be able to take precautions and respond to users' information needs. In Chapter 5, we present a comparison of different feature sets for detecting session abandonment. We have two groups. The first group of features are extracted solely from a SERP: title of a snippet, text of a snippet, URL text. The second group of features are extracted from the session and interaction. We develop machine learning models and tested them with low noise: fully-segmented sessions on a single topic, with the topic description available as well as sessions by other users for the same topic. We found that the latter set is far superior to the first set, which actually degrades effectiveness at predicting abandonment.

### 1.4 A New Session Dataset

In Chapters 3, 4, and 5 we train and test click prediction, dwell time prediction and session abandonment prediction models on user web search session data that is collected from a web corpus. In Chapter 6, we explain the user experiment that we set up to collect a new session dataset that is based on news data which consists of news passages in *xml* format. We train and test the machine learned models that are introduced in previous chapters on this new dataset and present the results. These experiments demonstrate the generalizability of our approaches.

### 1.5 Query Generation and Complete Session Simulation

User search session history has been used to improve overall search results in terms of providing relevant documents to users in higher ranks [1], predicting user

clicks [78], query auto-completion [64] and in many areas in information retrieval research and applications. In this thesis, we introduce many machine learned models that utilize user search sessions to accomplish a task or improve performances of simpler models. However, creating a user search session corpus is time consuming, expensive and not always possible due to certain privacy concerns.

In Chapter 7 we propose a model for query generation when information need is clearly defined. Finally in Chapter 8, we model complete user search session by using the models that we introduce in previous chapters. We believe that our models will help: (1) to create artificial session test collections, (2) enrich existing session test collections.

## Chapter 2

### RELATED WORK

In this chapter we start by presenting related work on basic ranking models and their use in learning to rank in information retrieval. We then describe work on the TREC Session track, which ran from 2011—2014, and additional work specific to the topics of this dissertation.

#### 2.1 Traditional Ad-hoc Information Retrieval

Ad-hoc information retrieval models retrieve and rank documents from a text based corpus by calculating a score of a document given a query. The model is ad-hoc because it treats queries as if they are single faceted.

In this section, we briefly describe the ad-hoc retrieval models and their usage in feature calculations for machine learning models.

##### 2.1.1 Boolean Model

The boolean retrieval model is a model for information retrieval in which documents are represented as set of terms and queries are represented as boolean expressions over terms which are combined with the operators AND, OR, and NOT[50]. The retrieval system evaluates the boolean expression on an inverted document index and retrieves the documents that satisfy the boolean expression. The classical boolean model cannot produce a ranking of documents. A very simple search scenario example could be as follows; consider a simple conjunctive query: *caesar* AND *republic*. The system first locates *caesar* in the dictionary and retrieves its postings. The system then locates *republic* in the dictionary and retrieves its postings. Then a new retrieval set is constructed from merging these two sets. In this case the logical operator is **AND**.

Therefore the merging algorithm is intersection. This step is also known as merging posting lists[50].

Besides document retrieval and ranking, boolean model can be used for feature calculations for machine learning models for user simulations in search tasks. Different fields of web documents such as header, body, complete text and snippets such as URL, title, snippet text are evaluated with boolean model with queries. Results of these evaluations which are *true* or *false* are used as features.

### 2.1.2 Vector Space Model

The vector space model is a commonly used method in information retrieval that represents documents and queries as vectors of identifiers, for example, terms. Each distinct term in vector space model is represented as a vector dimension and all dimensions are assumed to be linearly independent from each other. This assumption also brings a disadvantage to the vector space model. Terms in documents and queries are not independent from each other.

The magnitude of vectors could be represented in different ways such as; binary representation in which terms existence sets a dimension's value to 1 and non-existence sets the value to 0, term frequency in which the number of terms' appearance in a document or a query is the value set for a dimension, multiplication of term frequency and inverse document frequency in which the number of terms' appearance in a document or a query is multiplied by logarithm of total number of documents in a corpus divided by number of documents that contain the term is the value set for a dimension.

The vector space model ranks the documents in the collection according to the vector-space similarity between a document vector and a query vector. There are many ways to calculate the vector-space similarity such as cosine and jaccard similarities.

For feature calculations, vector space model calculates a score between a document field or a snippet field and a query. This score is normalized and used as a feature.

### 2.1.3 BM25

In boolean and vector space retrieval models document retrieval is based on defined calculus operations without emphasizing the uncertainty of information need of users. Probability based retrieval models are build around the notion of relevance which is uncertain and usually hidden in the sense of not observable[60]. These models estimate the probability of relevance of document query pairs and rank documents in descending order according to the estimated probability. One of the best achieving models is BM25 which has developed in stages over the years. There are many different implementations of BM25 but in the context of this document we will only mention Okapi BM25 model. The Okapi BM25 model scoring is as follows;

$$score(Q, D) = \sum_{i=1}^n \log \frac{N t_{fi}(k_1 + 1)}{d_{fi} t_f + k_1 B} \quad (2.1)$$

$$B = 1 - b + b \frac{|D|}{avgdl} \quad (2.2)$$

where D is the document and Q is the query that consists of n terms, N is the number of documents in the collection,  $t_{fi}$  is the term frequency of  $q_i$  in document D,  $d_{fi}$  is the document frequency of term  $q_i$  in the collection,  $|D|$  is the number of terms in document D, avgdl is the average document length in the text collection, B is the document length normalization component, b and k are parameters. Values of b is between 0 and 1 and setting b to 0 will switch normalization off and setting b to 1 will perform full document length normalization.  $k_1$  parameter is used to adjust the term frequency scaling such that setting  $k_1$  to 0 sets the system to binary model and setting  $k_1$  to a large number sets the system to term frequency model.

Feature calculation with Okapi BM25 model is done in a similar way to other ad-hoc retrieval models. Average length of document and snippet fields are calculated, document frequencies are calculated among related fields, and calculated scores are normalized to be used as features.

### 2.1.4 Language Model

One of the best ways to come up with a good search query is considering the terms that are likely to occur commonly in relevant documents and choosing the terms that discriminate the relevant documents from non-relevant documents[50]. The language model is based on a similar idea such that the documents are ranked according to their probability to generate given query model. There are many ways to generate models in language models, but we only cover query likelihood model which is the most common and the most basic method. In query likelihood method, document ( $M_d$ ) and query models ( $M_q$ ) are generated and documents are ranked according to  $P(d|q)$  which could be interpreted as likelihood of document  $d$  is relevant to query  $q$ . By using bayes theorem  $P(d|q)$  could be written as follows:

$$P(d|q) = \frac{P(q|d)P(d)}{P(q)} \quad (2.3)$$

Since  $P(q)$  is equal for every document for a given query, it has no influence on ranking of documents. Therefore it's ignored.  $P(d)$  is assumed to be uniform across documents. It is ignored as well. This assumption has been taken in many existing works[7][65][34][56]. Therefore the documents in a collection could be ranked by  $P(q|d)$  which could be interpreted as probability of generating query from the given document model.

The  $P(q|d)$  is estimated by using maximum likelihood and unigram assumption is as follows;

$$P(q|M_d) = \prod_{t \in q} P_m l e(t|M_d) = \prod_{t \in q} \frac{t f_t d}{L_d} \quad (2.4)$$

where  $M_d$  is the language model of document  $d$ ,  $t$  is the term in query  $q$ ,  $t f_t d$  is the term frequency of term  $t$  in document  $d$ ,  $L_d$  is the number of terms in document  $d$ .

The classic problem with language models is estimating probability of generating a query model over a document model when the query terms are sparse or even do not

exist in the document. If a query term do not exist in the document the  $P(q|M_d)$  becomes 0. The solution to the problem is smoothing which refers to the adjustment of maximum likelihood estimator for more accurate language model. There are many proposed smoothing methods which are mostly in speech recognition context but in the context of this document we will only mention about two smoothing methods; Jelinek-Mercer smoothing and Bayesian smoothing with Dirichlet priors.

The Jelinek-Mercer method uses a  $\lambda$  parameter to linearly interpolate the maximum likelihood model with the collection model. The  $\lambda$  controls the influence of each models.

$$P_\lambda(t|M_d) = (1 - \lambda)P_{mle}(t|M_d) + \lambda P(t|C) \quad (2.5)$$

The bayesian smoothing with dirichlet priors is given by

$$P_\mu(t|M_d) = \frac{tf_d + \mu P(t|C)}{\sum_t tf + \mu} \quad (2.6)$$

For details of smoothing methods, readers are encouraged to see Zhai and Lafferty[79].

For feature calculations, language model scores of document and snippet fields are calculated with different smoothing methods. Scores are normalized and used as features.

## 2.2 Learning To Rank

Learning to rank for information retrieval refers to machine learning techniques that learn ranking models from training data, usually arranged as a set of queries, each of which contains documents represented as a label (relevance judgment) with a vector of feature values. Learning to rank methods have been used in various information retrieval specific tasks such as document retrieval, question answering, multimedia retrieval and online advertising. In the scope of this document, we will only focus on learning to rank for document retrieval.

The main problem of IR in document retrieval is ranking a set of documents according to a given information need. Therefore, using learning to rank methods when there is massive amount of user log data for document retrieval is available is a natural and wise choice. Learning to rank in the scope of document retrieval works as follows. Assume there's a collection of documents. The learning to rank model learns from a perfect ranking of documents and queries and creates a ranking model. In retrieval, the ranking function ranks the documents according to the model that is learned.

There are three main learning to rank methods according to their approach to learning to rank problem; (1) pointwise approach (2) pairwise approach (3) list-wise approach. These different approaches implement different input/output spaces, use different loss functions and hypotheses. In this section, we are going to summarize learning to rank methods and introduce the LETOR collection[57] and features extracted from documents for LETOR collection.

### **2.2.1 Pointwise approach**

The pointwise approach as its name suggests learns from one query document pair at a time. The input space consists of only single document's features. The output space is one dimensional and contains relevance degree of a single document. The hypothesis space of pointwise approach contains one scoring function that takes the feature vector of a single document and outputs a relevancy score which could be used to rank list of documents. The loss function of this model compares the prediction with label assigned to the document. Keep in mind that in this model, in each step each document isolated from other documents.

### **2.2.2 Pairwise approach**

The pairwise approach learns from a pair of documents at a time. The input space consists of a pair of document's features. The output space is the relative ordering of two documents. The hypothesis space of pairwise approach contains a preference function that tries to learn from ordering of two documents. The ground truth could be

given in preferences or in relevancy scores. The loss function of this model compares the prediction with the given ground truth and tries to minimize the inversions in document orderings. In general pairwise approach performs better than pointwise approaches. This could be due to judgements of relative orderings is easier than graded relevance judgements. Some very popular learning to rank algorithms like RankNet[9], FRank[68], RankBoost[27], Ranking SVM[33][37] are pairwise algorithms.

### 2.2.3 Listwise approach

The listwise approach learns from a list of document for a query at a time. The input space consists of list of documents' features which are related to a query. The output space is either ranking of list these documents, or relevancy scores of each document similar to pointwise approach. The hypothesis space of listwise approach contains a ranking function that takes feature vector of list of documents and predicts either a ranking of documents or relevancy scores of documents. There are two types of loss functions. If the ground truth is given as a score of an evaluation measure, the loss function measures the difference between the ground truth and score achieved by the ranking. If the ground truth is given as ranking of documents, the loss function measures the difference between two rankings.

### 2.2.4 LETOR collection and feature set

LETOR is a benchmark collection for learning to rank research released by Microsoft Research Asia[57]. The LETOR collection contains indexed documents, queries for training and testing, feature vectors for every document, and implementation of baseline algorithms. It also contains a hyperlink graph, similarity relationships and sitemap of documents.

LETOR features consist of retrieval model scores (from the Boolean model (Sec. 2.1.1), vector space model (Sec. 2.1.2), BM25 (Sec. 2.1.3), and LM (Sec. 2.1.4)) between the query and various fields of web pages, and between the query and the page snippet; statistics about query term frequencies (normalized and non-normalized) in

fields of document and snippet; URL length and depth; number of inlinks and outlinks. The complete list is given in Table 2.1. There are five fields in a document: body, anchor, title, URL, and their union. There are three fields in a snippet: title, URL, and their union with snippet text.

### 2.2.5 Reading level document features

There are many readability metrics such as Flesch-Kincaid [46], FOG [29] and SMOG [53]. They all estimate the hardness of a text in order to understand the minimal age group that the text can be comprehended. The Flesch-Kincaid readability metric is defined as:

$$\text{Score} = 0.39 \frac{\text{number of words}}{\text{number of sentences}} + 11.8 \frac{\text{number of syllables}}{\text{number of words}} - 15.59 \quad (2.7)$$

The SMOG readability metric is defined as:

$$\text{Score} = 3 + \sqrt{\text{number of polysyllable words}} \quad (2.8)$$

The FOG readability metric is defined as:

$$\text{Score} = 3.0680 + 0.877(\text{average sentence length}) + 0.984(\text{percentage of monosyllables}) \quad (2.9)$$

## 2.3 Clueweb12 Collection

The Clueweb12 dataset consists of 733,019,372 English web pages that was collected between February 10, 2012 and May 10, 2012. This dataset is successor of Clueweb09 web dataset. Initial crawling of Clueweb12 started with 10 million Clueweb09 urls that has highest pagerank [55] scores and is not in top 90% of pages ranked with Waterloo spam score.

**Table 2.1:** List of features in the LETOR collection.

Feature	Fields
covered query term number	URL, Title, Snippet
covered query term ratio	URL, Title, Snippet
stream length of the field	URL, Title, Snippet
sum of term frequency of the field	URL, Title, Snippet
min of term frequency of the field	URL, Title, Snippet
max of term frequency of the field	URL, Title, Snippet
mean of term frequency of the field	URL, Title, Snippet
variance of term frequency of the field	URL, Title, Snippet
sum of stream length normalized term frequency	URL, Title, Snippet
min of stream length normalized term frequency	URL, Title, Snippet
max of stream length normalized term frequency	URL, Title, Snippet
mean of stream length normalized term frequency	URL, Title, Snippet
variance of stream length normalized term frequency	URL, Title, Snippet
sum of term frequency * inverse document frequency	URL, Title, Snippet
min of term frequency * inverse document frequency	URL, Title, Snippet
max of term frequency * inverse document frequency	URL, Title, Snippet
mean of term frequency * inverse document frequency	URL, Title, Snippet
variance of term frequency * inverse document frequency	URL, Title, Snippet
boolean model	URL, Title, Snippet
vector space model	URL, Title, Snippet
bm25(okapi) score	Snippet
LM with Dirichlet smoothing	URL, Title, Snippet
LM with Jelinek-Mercer smoothing	URL, Title, Snippet
LM with two-stage smoothing	URL, Title, Snippet
number of slashes in URL	URL
length of URL	URL
Waterloo spam score	—
number of inlinks to the web page	—

## 2.4 The New York Times Collection

The New York Times Annotated corpus consists of over 1.8 million articles that were written and published by the New York Times between January 1, 1987 and June 19, 2017<sup>1</sup>. Along with the articles, corpus also includes over 650,000 article summaries provided by library scientists, over 1,500,000 articles manually tagged by library scientists, over 270,000 algorithmically-tagged articles that were verified by nytimes.com online production staff and tools written in *java* to parse documents.

## 2.5 User Logs and the TREC Session Track

A user session means a sequence of query reformulations along with interactions made by user in order to satisfy an information need. A session starts with an initial query which is often ill-specified, and will likely need to be reformulated several times before a user finds the satisfactory information: early studies on web search query logs showed that half of all search engine users reformulated their initial query: 52% of the users in 1997 Excite data set, 45% of the users in the 2001 Excite dataset[73]. Thus investigation on such cases play an important role in improving user search experience.

User interaction data has been released several times by companies. AOL released user search sessions of 650,000 AOL users along with 20 million web queries<sup>2</sup>. However this data was retracted by AOL due to massive amount of private data which was made public despite the AOL usernames changed to random ID numbers. The data included personal names, social security numbers, addresses, and many personal data that should be kept private<sup>3 4</sup>.

In 2013 Yandex released dataset for "Yandex Personalized Web Search Challenge" hosted by Kaggle[62]. The dataset consists of 34 million fully anonymized user

---

<sup>1</sup> <https://catalog.ldc.upenn.edu/LDC2008T19>

<sup>2</sup> <https://techcrunch.com/2006/08/06/aol-proudly-releases-massive-amounts-of-user-search-data/>

<sup>3</sup> <https://www.cnet.com/news/aol-apologizes-for-release-of-user-search-data/>

<sup>4</sup> <http://www.nytimes.com/2006/08/23/technology/23search.html>

sessions of 5.7 million users which include user ids, queries, query terms, URLs, their domains, URL rankings and 64 million user clicks<sup>5</sup>. Since this data is fully anonymized by hashing user ID/queries/URLs to random strings, its usage in full scale search session research is limited.

Researchers in academia have limited access to user interaction data due to commercial and privacy concerns. thus, The TREC Session track was organized by collaboration between the University of Sheffield and University of Delaware to run as a competition in the TREC conference organized by NIST in order to provide test collections and evaluation measures for studying information retrieval over user sessions rather than one time queries[13]. The TREC Session track’s main task is improving the system performance using previously submitted queries and interaction data in session. The data contains query reformulations, results returned from search engine along with user clicks and dwell time on clicked pages.

In our experiments we used TREC session track 2013 and 2014 data. During collection of TREC session track 2013 and 2014 data, Clueweb12 collection is used. Clueweb12 consists of roughly 730 million English language web pages, comprising approximately 5TB of compressed data that crawled from the Web during February and March 2012[13][12].

## 2.6 User Click Detection

User click models have been used widely in information retrieval as a way to improve document ranking (e.g., by inferring document relevance from clicks predicted by a click model), to improve evaluation metrics (e.g., model-based metrics) and to better understand users by inspecting the parameters of click models. Craswell et al. [23] introduced the *cascade model*, a model based on position bias, or the fact that the probability of click is influenced by a document’s position in the results page[28][49][38]. The cascade model’s main assumption is that users scan documents on a SERP top to bottom and click the first relevant snippet they find. The assumption discards the

---

<sup>5</sup> <https://www.kaggle.com/c/yandex-personalized-web-search-challenge/data>

snippets below the clicked snippet and probability of a click to a snippet is dependent to other snippets above it. This model can only predict one click on a given SERP. Richardson et al. [58] proposed a model based on examination hypothesis and position bias for clicks on ranked list of ads on a SERP. The model assumes probability of a click on an ad depends on its position and its probability of that the user thinks snippet is relevant enough to click. Craswell et al. [23] adapted this model to a general click model and named it as examination model. Dupret and Piwowarski [25] introduced the user browsing model (UBM) which is an extension of examination model and has some similarity to the cascade model. Unlike the cascade model this model does not assume user scans all the snippets all the way down to the clicked snippet and abandons the search. The main idea of this model is that the probability of user’s examination of a snippet depends on the distance from the last click action as well as on the position of the snippet in the ranking.

Chapelle and Zhang [15] introduced the dynamic Bayesian network click model (DBN) for web search ranking by tackling position bias. These studies only made predictions on a single click in the results page; Guo [31] presented the multiple click model which attempts to predict more than one click at a time. Shen et al. [63] claimed previous click models assume all users act in the same way and proposed three click models: (1) a matrix factorization click model which relates documents and queries, (2) a personalized click model which uses indiscriminate tensor factorization to utilize the relationship between users, documents and queries, (3) a hybrid personalized click model that combines previous two models. Guo et al. [30] proposed *click chain model* which assumes users starts examining from top to bottom and at each position user may choose to click or skip to the next result according to perceived relevance and probability of examining the next document depends on the user action done on the previous document. This model is based on bayesian modeling and inference of user perceived relevance. Several other methods have been presented based on these methods.

Our dependency models differ from these models in many ways. Our baseline,

independency model assumes a user investigates a single snippet and makes a click decision. We extend the baseline model with various features that are extracted from user’s own session history and other users’ session histories. We also take clicks made by a user on current SERP into consideration and investigate their influence on click decision on a snippet. The dependency model assumes a user scans all the results, compares them and makes click a decision on a given snippet. We train and test the dependency model in many different scenarios such as considering only the snippets on the given SERP, user’s own session history, and all clicked results on the current SERP. We also investigate the benefit of using other people’s session histories as feature sources.

## 2.7 Dwell Time Prediction

Dwell time is the duration that starts with a user click on a result that is listed on a SERP and ends with returning to the results or abandoning the search. Dwell time is the one of the most useful implicit feedback that can be used to estimate document relevance or result-level satisfaction [10] [21] [26] and it has been used in many applications such as selecting relevant words for query expansion [10], re-ranking the search results [1] and for implicit relevance feedback [71].

One of the first studies on understanding and modeling the dwell time was done by Morita and Shinoda [54]. They measured the relationship between dwell time and user interest and found that user preference on an article is the dominating factor that affects time to spend reading it. Their study also showed that there is a very low correlation, less than 0.08, between length of the article and a time to read the article. Liu et al. [48] proposed a model that predicts the Weibull distribution of dwell time by using multiple additive regression trees with low-level page features such as features related to page formatting which are hidden from users, features related to document content and dynamic features which are related to rendering and loading of the web page. Yi et al. [77] used support vector regression model with topical category of the article and the context in which the article shown as features to predict user dwell

time. They interpreted the weights of the trained support vector model and claimed that readers spend more time on documents on desktop than mobile or tablet devices, longer articles and more serious topics lead to higher user dwell time. Kim et al. [45] proposed a model that predicts gamma distributions of satisfied and unsatisfied click dwell times using features that were extracted from query such as labels assigned to queries from a Open Directory Project (ODP) classifier, labels assigned to query from a search intent classifier and web document such as ODP category of the document and reading level attributes. Xu et al. [74] proposed a re-ranking algorithm based on predicting user dwell time on web documents by creating mappings between concept terms and dwell times and treating a document as a collection of concept terms. Wang et al. [69] adopted a factorization model that captures the interaction between users and webpages to create a model that predicts the dwell time of a user on certain depth of a document.

In our models we do not discriminate satisfied or unsatisfied clicks and try to estimate the user dwell time on a clicked document. We investigate effectiveness of LETOR features extracted from query and snippets and session and interaction features on estimating user dwell times. Our dwell time prediction models are not novel but are they part of a complete user search simulation.

## 2.8 Session Abandonment Prediction

Session abandonment has been studied extensively in several forms such as search session satisfaction prediction and session abandonment type prediction. Kelly defines satisfaction as the fulfillment of a special desire or goal [42]. Hassan et al.[32] proposed a model that predicts search satisfaction by examining the query reformulations and showed that a model based on query features outperforms a model based on click features. They also showed that the combination of two outperforms both models. The relationship between implicit and explicit measures of user satisfaction was studied by Fox [26]. They used Bayesian modeling techniques and found that combination of certain implicit measures such as clickthrough data, dwell time and

search exit type help to predict explicit judgements of user search satisfaction. Kim defined three different dwell time types, i.e., server-side which starts with a click to a search result and ends with new query or new click, client-side which starts with a user click and ends with user returning to SERP or abandoning the search, and trail dwell time which includes the dwell time on pages that are clicked after landing page and examined their effectiveness on predicting user search satisfaction [44].

Li et al.[47] introduced and made the first distinction between good and bad abandonment. They defined good abandonment as an abandoned query for which the searcher’s information need was successfully satisfied, without needing to clickthrough to additional pages. Chuklin and Serdyukov examined query extensions [19] and editorial and click metrics [18] for their relationship to good or bad abandonment. In [20] they developed machine learned models to predict good abandonment by using topical, linguistic, and historic features. Diriye [24] studied abandonment rationales and developed a model to predict abandonment rationale. Beyond the SERP (search engine result page), White and Dumais [70] studied aspects of search engine switching behavior, and develop and evaluate predictive models of switching behavior using features of the active query, the current session, and user search history.

Our work differs from these in terms of the problem definition and the method. We predict an abandonment apart from its rationale on a given information need, session and interactions. We use features extracted from user queries, statistics from query term snippet terms, and overall search session with other users’ sessions on the same information need.

## 2.9 Query Generation and Session Simulation

Simulations in search sessions have been used in various forms to study independent parts of the search process such as query reformulation and suggestion, browsing behavior, stopping models and strategies, and performance over sessions in information retrieval research. Even though there are vast number of works on user simulations

in information retrieval search, in this section we only mention about the works that target creating search sessions and session test collections.

One of the first works that question effectiveness of Cranfield style IR experiments and costs of query reformulations was done by Keskustalo [43]. In this work Keskustalo compares four different query strategies in terms of search success which are defined as finding one relevant document in top ten results, and reformulating the query in case of failure for three models and finding one relevant document in top fifty results for one model. The first three models are different variations of query reformulation and the last model is single verbose query. Keskustalo used TREC 7-8 collection with 41 topics and selected user generated queries randomly according to the model and found that short query reformulations are as successful as using one single long verbose query. They suggested that future evaluations in IR should model processes where the user submits many queries for a topic and use broader costs and benefits than evaluating the search results by their quality for single query.

Baskaya et al.[4] performed user simulations to study effects of relevance feedback on IR performance when users provide short queries and erroneous relevance feedback. They showed that fallible relevance feedbacks acquired from simulation models are as good as actual correct relevance feedbacks. In follow up studies [5] [3], Baskaya simulated user sessions with various query modification strategies such as initially creating a one word query and repeatedly replacing the word with a new word from a dictionary, creating an initial two-word query and changing the query with updating the second term, creating an initial three-word query and changing the query with updating the third word, creating an initial one-word query and updating the query with adding new terms, and creating an initial two-word query and updating the query with adding new terms. They created sessions with various combinations of these query modification and result scanning methods.

Maxwell et al.[51] proposed an open-source toolkit, SimIIR, for building and conducting interactive information retrieval experiments. Toolkit consists of high level configurable components such as query generation methods and stopping strategies (the

depth which user examines snippets and documents). SimIIR models the interactive information retrieval process using *Complex Searcher Model* which was proposed by Maxwell et. al.[52]. In summary the model simulates the searching process as follows: The simulated searcher begins by reading the given topic, generates a series of queries, issues a query from generated list, proceeds to examine the snippets in the SERP. At this point searcher can return to issue a new query or continue to examining the results. If the examined snippet is found relevant, the searcher proceeds to reading the document. If the document is found relevant, the document is marked as relevant.

Our models differ from these models in terms of session creation and evaluation of the created search sessions. In query generation, we generate queries based on probability of terms appearing in a query and select queries on their probability of generating the given information need. We evaluate the query generation models by creating sessions using the queries and comparing retrieval models' performances on the generated sessions and actual user search sessions. In session simulation, we generate sessions by using machine learning models that we introduce and evaluate the sessions with a user experiment.

## Chapter 3

### PREDICTING USER CLICKS IN IDEAL SESSIONS

User click models predict whether a user click is going to happen or not on a given document that is listed on a search engine result page (SERP). They have been used widely in information retrieval as a way to improve document ranking (e.g., by inferring document relevance from clicks predicted by a click model), to improve evaluation metrics (e.g., model-based metrics) and to better understand users by inspecting the parameters of click models [17].

In this chapter, we introduce several machine learned models that predict a single user click on a snippet that is listed on a SERP in several different scenarios. We consider two different main settings: (1) only considering SERP; (2) considering session history, (3) considering session history with sessions of other users along with the SERP. For each of these main settings we consider three different settings (1) only considering a snippet by itself; (2) considering all snippets on a results page along with the snippet that we make click prediction; and (3) considering the clicks in a SERP. As a result we test nine different models derived from combining feature sets from two different groups. We found that a dependency model which uses features of all snippets in a results page is a better click model over one that uses only features of a target document, and furthermore that using features of the session history and of other sessions on the same topic uniformly improve precision and area under the receiver operating characteristic curve (AUC) <sup>1</sup>.

---

<sup>1</sup> This work was previously published in the proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval. ACM, 2017.

**Table 3.1:** Corpus statistics

# sessions	1253
# average number of interactions per session	3.16
# average number of clicks per interaction	0.514
# average number of clicks per session	1.624
# snippets	39570

### 3.1 Experiment Data

For this study, we used the 2013 and 2014 TREC Session track data for training and testing our models.

### 3.2 Methods

Our main assumption for a proposed click prediction model is that users base their clicks primarily on the features of the submitted query and the snippets on the SERP and their search experience through the session. The features that are related to snippet and query-snippet are the features of document titles, URLs, and snippet texts on the current search result page and their scores on retrieval models. We explain these features in detail in section 2.2.4. Besides we added Alexa ranking of the web document and rank of the document in a SERP as features to this set. The second set of features are created from user’s own session. The third set of features are the features that are created from other users’ sessions with the same information need.

#### 3.2.1 Session Features

Besides features that are extracted from snippets and documents on a SERP, we make use of session based features. We explain the features and their extraction methods in this section.

Since sessions can have varying length, and therefore varying numbers of snippets seen during the course of the session, it is not necessarily straightforward to include features for previously-seen documents (as it is to include features for other documents appear in the same results page). Furthermore, the length of time passed between

**Table 3.2:** List of session related features.

Feature	Notes
number of times snippet appears in previous interactions	Session
distance of current interaction to previous interaction that the snippet appeared	Session
number of clicks on the snippet in current session	Session
distance of snippet to its previous click in previous interactions	Session
maximum score of each query dependent feature across previously submitted query in session	Session

seeing two snippets may affect the likelihood of a user click as well. For the first set of features that we created from search session we used previous queries submitted to the search engine during the course of the session. We compute all of the textual query-document features listed section 2.2.4 for every snippet ranked for every query in the session. We then take the maximum value of the feature over the session history to be included in the model.

For the second set of features we used clicks and snippets appeared in previous interactions in the same session.

These features are number of times the snippet has been seen prior to current result page, the distance between the current result page and the result page on which the snippet was most recently clicked, the distance between the current result page and the result page on which the snippet most recently appeared, and the number of times the snippet was clicked in the current session were also calculated. If the snippet has never appeared in any prior interactions, the distance for both click and appearance distances is set to zero. All features are listed in Table 3.2.

### 3.2.2 Other Session Features

TREC data provides topics along with the sessions. Even though this is not easily computed in real query logs, we made use of this information. We calculated a snippet’s appearance and click count in other sessions. We also calculated the same

**Table 3.3:** List of features that are retrieved from other user sessions.

Feature	Notes
session appearance count	Other sessions
session click count	Other sessions
same query/different session appearance count	Other sessions
same query/different session click count	Other sessions

**Table 3.4:** Nine different click models derived from combining feature sets from two different groups: the search results page and the session history. s.h. stands for session history and c.a.s stands for clicks across sessions.

	session features		
ranking features	none	s.h.	s.h. + c.a.s.
target snippet	M1	M2	M3
target snippet + all others	M4	M5	M6
target snippet + all others + clicks	M7	M8	M9

information with the same user query in other sessions. These features are listed in Table 3.3.

### 3.3 Models

We test nine different models that offer refinements using incrementally-sized feature sets, laid out in a  $3 \times 3$  table (Table 3.4). These models build on each other in two directions: first, using more and more features of a single search result page; second, using more and more features of the session and of other users’ activity in similar sessions. For example, M1 only uses features of the snippet for which we are trying to make a prediction—specifically, features in Table 2.1. M4 builds on that by using features of the target snippet, but also including the same features for the other nine snippets ranked alongside it, while M2 builds on M1 by adding features of the session history, specifically features in Table 3.2. M7 builds on M4 by including as a feature the same user’s click decision on each of the other nine snippets, while M3 builds on M2 by including features of other users’ activity in other sessions on the same topic—those in Table 3.3.

For each of the nine models, we will train a machine-learned model using 0/1 click decisions as the label and a feature vector consisting of one or more of the following feature sets:

1. target result URL/title/snippet. See Table 2.1 for complete list of features. This set appears in all nine models.
2. other results' URLs/titles/snippets from the same result page (for the fourth to ninth models)
3. other clicks by the same user on other results in the same result page (seventh, eighth and ninth models)
4. historical results' (from earlier in the session) URLs/titles/snippets calculated for corresponding historical queries, along with other information related to previously seen results (for the second, third, fifth, sixth, eighth, and ninth models). See Table 3.2 for complete list of features.
5. User clicks across all topically-related sessions (third, sixth, ninth models). See Table 3.3 for complete list of features.

### 3.4 Experiments

We evaluate our click detection models by their effectiveness at predicting actual user clicks, using standard classification evaluation measures like precision, recall, and AUC.

We trained and tested random forest models using all sessions from the TREC 2013 and 2014 Session track. For each ranked result that appears in the Session track data, we have one instance for training/testing: the 0/1 label indicating whether there was a click on that result or not, the features derived from that result, and, depending on the model, other results ranked with it, other results from the session history, or other features of the session. In total there are 39,570 instances in the data, of which 37,535 are no-clicks and 2,035 are clicks. Since the class distribution is so skewed, we rebalanced the training data with SMOTE, which creates artificial data for the under-represented class[16]. We trained and tested using four-fold cross-validation and report micro-averaged evaluation measures aggregated across all four testing splits. Note that only training data, not testing data, in each fold is rebalanced with SMOTE.

### 3.5 Results

Table 3.7, Table 3.8 and Figure 3.1 summarize performance of the nine models. Table 3.7 and Table 3.8 is organized in a way that moving from left to right or top to bottom adds new features to a model, from the current results page or the session, respectively. The top left cell is the baseline. Moving to the left, just adding session related features to the baseline increases AUC by 1.2% and precision by 8.6%, but decreases recall by 5.2%. Using clicks from other sessions along with the session data results in an 6.2% increase in AUC and 24.6% increase in precision, and 4.6% decrease in recall.

Moving down from the top left cell increases precision by 25.1% and AUC by 3.6%. From there, moving left to add session-related features has less influence on the dependence model. Only adding these features increases the precision by 5.5% and AUC by 0.8%, while decreasing recall by 5.4%. Using other click decisions in an interaction has positive influence on all measures.

Finally, moving down to the last row, we see a 2.4% improvement in AUC when we move from M4 to M7. Similar improvements as above are achieved in adding session data to produce the M8 and M9 models. The general tendency of the performances shown on the table demonstrate that moving from top to bottom in models increases AUC and precision, and decreases recall. Moving from left to right mostly increases AUC and precision.

There is clearly benefit in moving in both directions. Result page features alone improve the baseline by up to 39% for precision and 6% for AUC, with a 4% drop in recall. Session features alone can account for up to 25% improvement in precision and 6% in AUC, again with a 4% drop in recall. When both result page features and session features are considered, the improvement over the baseline is 43% for precision and 8% for AUC, though this improvement comes with a 10% drop in recall.

Figure 3.1 compares the ROC curves for related groups of models. Reading figures from left-to-right/top-to-bottom, we see that adding information about the

**Table 3.5:** Precision/recall of our nine models, c.a.s. stands for clicks across sessions

ranking features \ session features	none	session history	history + c.a.s.
	target snippet	0.175/0.408	0.190/0.388
target snippet + all others	0.219/0.391	0.231/0.371	0.249/0.379
target snippet + all others + clicks	0.243/0.390	0.252/0.362	0.251/0.369

**Table 3.6:** AUC of our nine models, c.a.s. stands for clicks across sessions.

ranking features \ session features	none	session history	history + c.a.s.
	target snippet	0.753	0.762
target snippet + all others	0.780	0.786	0.798
target snippet + all others + clicks	0.799	0.797	0.811

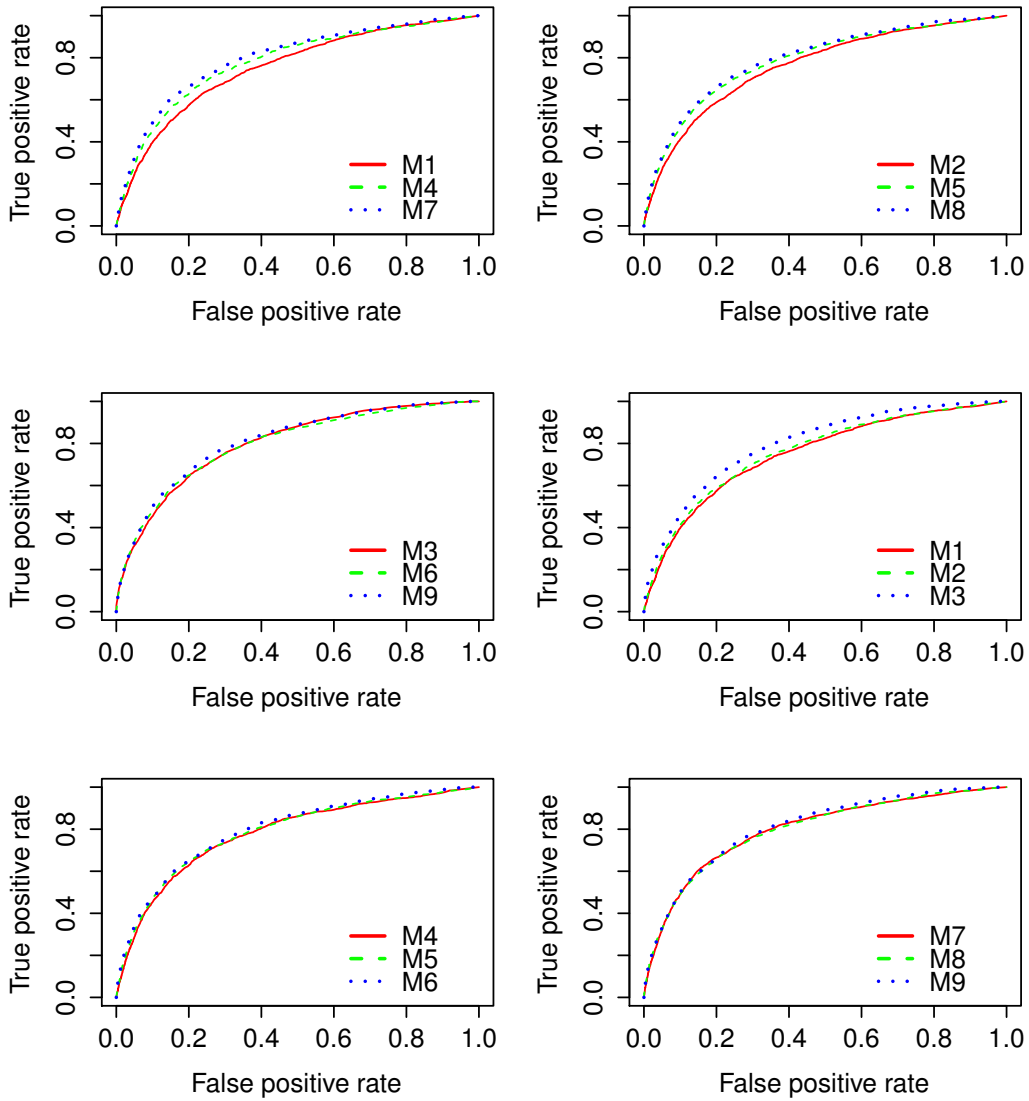
results page with no session context (going from M1 to M4 to M7) pushes the ROC curve out, and adding that information with session context (M2 to M5 to M8) pushes it out further. When both session context and other sessions are used, the results page does not contribute much to the ROC curve (M3 to M6 to M9).

Similarly, the next figure shows a clear impact from using session history when no information about the results page is used (M1 to M2 to M3). But as more information about the results page is taken into account, the session history seems to provide less impact on the ROC curve (M4 to M5 to M6 and M7 to M8 to M9). Nevertheless, when both are considered together, the gain is clear.

Figure 3.2 shows mean raw importance of all features for the click class in the independence model that uses session information and clicks across sessions (M3). The

**Table 3.7:** NYT-Precision/recall of our nine models, c.a.s. stands for clicks across sessions

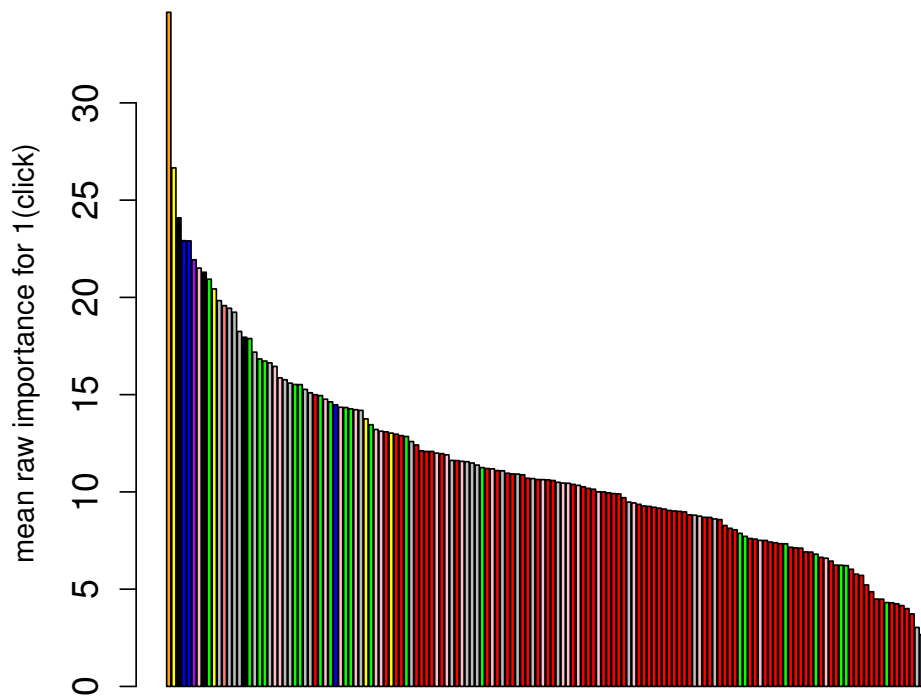
ranking features \ session features	none	session history	history + c.a.s.
	target snippet	0.175/0.408	0.190/0.388
target snippet + all others	0.219/0.391	0.231/0.371	0.249/0.379
target snippet + all others + clicks	0.243/0.390	0.252/0.362	0.251/0.369



**Figure 3.1:** ROC curves for different click detection methods

**Table 3.8:** NYT-AUC of our nine models, c.a.s. stands for clicks across sessions.

ranking features \ session features	none	session history	history + c.a.s.
	target snippet	0.753	0.762
target snippet + all others	0.780	0.786	0.798
target snippet + all others + clicks	0.799	0.797	0.811



**Figure 3.2:** Features ranked by mean raw importance for the positive class for independence+session experiment. Yellow represents other session related features, green represents query-URL related features, grey represents query-snippet related features, black represents document related features, blue represents URL related features, red represents session related features, purple represents snippet related features, pink represents query-title related features, salmon represents title related features and orange represents search engine related features.

top three features in click decisions are the rank of the snippet, the number of clicks on the same snippet in across sessions, and the URL’s spam score. Note that session-related features (colored red and yellow) plays a major role in click decisions: the most important session history feature is maximum LM/Jelinek-Mercer score of the URL field, ranked 31st out of 152 features. These features have an overall mean important of 18.5, compared to 11.8 for query-URL features.

### **3.6 Conclusions**

We have presented nine different user click models based on increasing information about the overall results page and the session history. We found that a dependency model which uses features of all snippets in a results page is a better click model over one that uses only features of a target document, and furthermore that using features of the session history and of other sessions on the same topic uniformly improve precision and AUC. Still, our predictions frequently do not match reality; we intend to perform a user study to further evaluate our predicted clicks in greater detail.

## Chapter 4

### USER DWELL TIME PREDICTION

Dwell time is the duration that starts with a user click on a result that is listed on a SERP and ends with returning to the results or abandoning the search. Dwell time is one of the most useful implicit feedback that can be used to estimate document relevance or result-level satisfaction [10] [21] [26] and it has been used in many applications such as selecting relevant words for query expansion [10], re-ranking the search results [1] and for implicit relevance feedback [71]. However dwell time is not always present in test collections and it requires client-side applications to accurately measure.

As a complementary work to user click detection and simulation, in this chapter, we introduce three machine learned models that estimate user dwell time on web documents in user search session scenarios. Our models estimate the user dwell time on the assumption that the amount that users spend on documents is based on document features, query-document features and features of the snippets that the document is listed along with.

#### 4.1 Experiment Data

We use TREC 2013 and TREC 2014 Session track data to train and test models that predict user dwell time on web documents. Since only clicked documents have non-zero dwell time, we will have 2095 possible training and testing instances from total of 1253 user search sessions. Of those, some were clear outliers in which a user stayed on a page for a very long time. In order to not bias towards such outliers, we excluded 39 instances with dwell times greater than 120 seconds. We also excluded the

clicks occurred other than the first page of the search engine result page due to those clicks do not fit to our models that we will explain in next section.

## 4.2 Models

The methods we use for simulating dwell times are similar to those we use for simulating clicks: we start from the assumption that the main basis of dwell time can be modeled primarily by features of the full document the user is looking at, then refined that with other data from ranked results. Our three dwell time models will be:

1. the length of time a user stays on a result is based on features of that document;
2. the length of time a user stays on a result is based on features of the document as well as features of URLs/titles/snippets of other ranked results;
3. the length of time a user stays on a result is based on features of the document, features of URLs/titles/ snippets of other ranked results, and clicks the user made on those results.

The first model is the most basic model that solely depend on LETOR features of clicked document. The second model augments the first model based on the assumption that a user’s dwell time on a web document may be influenced by the LETOR features of other results displayed on SERP. The third model extends the second model by including user clicks on other documents listed on SERP and several reading level features of clicked document such as Flesch-Kincaid, SMOG, Coleman-Liau, and so on. For details of LETOR and reading level features on documents and snippets, readers are encouraged to see sections [2.2.4](#) and [2.2.5](#).

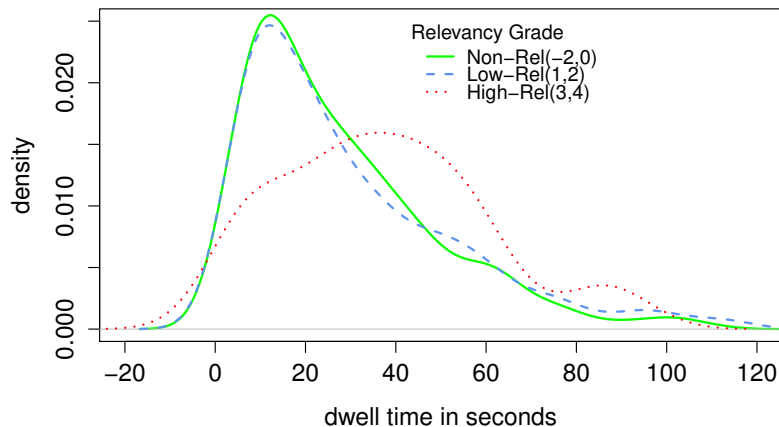
Since dwell time is (for all intents and purposes) a continuous number, we will use a regression model rather than a classification model used in previous chapters.

## 4.3 Experiments

We trained and tested regression random forest models using all sessions from TREC 2013 and TREC 2014 Session track. For each clicked ranked result that appears in the Session track data, we have one instance for training/testing: the dwell time

**Table 4.1:** RMSE of dwell time estimation of methods and their comparisons to mean and median dwell times.

method	RMSE	$\% \Delta_{mean}$	$\% \Delta_{median}$
mean	22.51	-	-
median	23.41	-	-
1. independence	23.20	3.00%	-0.89%
2. dependence	22.47	-0.17%	-4.00%
3. dependence+clicks	22.35	-0.71%	-4.50%

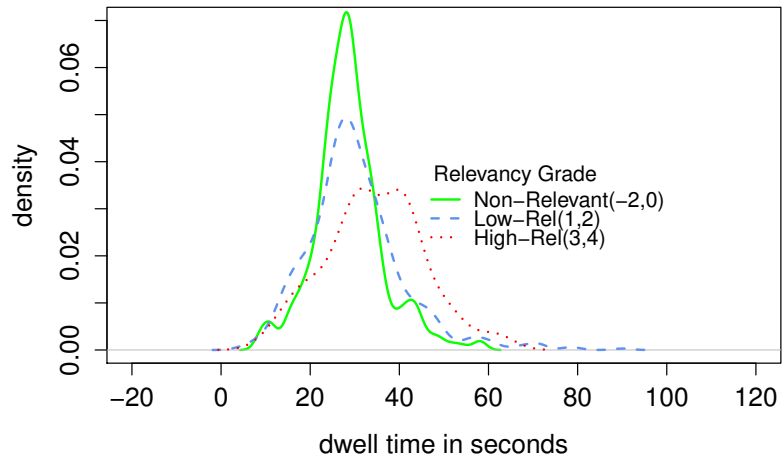


**Figure 4.1:** Density of actual dwell times by relevance class.

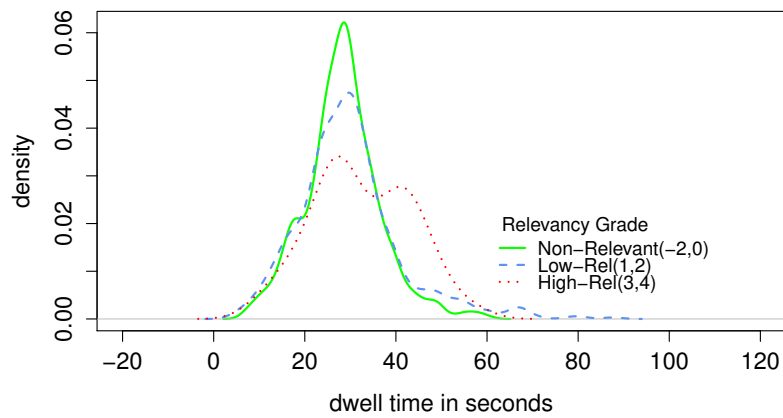
indicating the duration that a user spent on a web document, the features derived from that web document, and, depending on the model, snippet features of other results ranked with it, clicks done on other results, reading level features.

### 4.3.1 Results

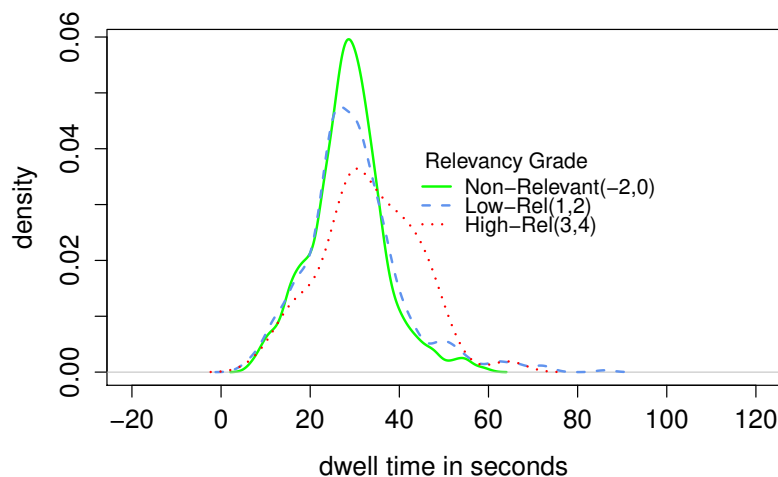
We perform four-fold cross validation on our dwell time estimation models and evaluate their effectiveness at predicting actual user dwell times that are extracted from TREC Session track data. Since the dwell time prediction is a regression task, we use root mean squared error (RMSE) to summarize results. We define mean and median of dwell times as baselines. Performance of the three methods along with performance of



**Figure 4.2:** Density of actual dwell times generated by independence model by relevance class.



**Figure 4.3:** Density of actual dwell times generated by dependence model by relevance class.



**Figure 4.4:** Density of actual dwell times generated by dependence model along with click information and reading level features by relevance class.

baselines are shown on Table 6.4. RMSE of mean and median mean are RMSE values if we used mean or median of dwell times as estimated dwell times instead of actually predicting it. Performance of all the three models are close to mean and median and unlike user click models, taking into account other rank results along with the result that we are trying to estimate dwell time does not help to improve the prediction performance.

We binned the dwell times of documents into three classes. First class consists of dwell time estimation of documents that were judged non-relevant or spam. The second class consists of dwell time estimations of documents that were judged relevant or highly relevant and the third class consists of dwell time estimations of documents that were judged key or navigational document. Figure 4.1 shows density distribution of actual dwell times by document class. Figure 4.2, Figure 4.3 and Figure 4.4 show density distributions of dwell time predictions of independence, dependence and dependence with user click models. In all our models we see clear separation of relevancy classes even though relevancy grade was not used as a feature. It also clear that our models

capture and model the actual dwell time distribution.

#### **4.4 Conclusions**

In this section, we created models that estimate user dwell time on web documents. We introduced three models from simple to complex that base dwell time estimations to features of clicked documents, features of the snippets that are listed along with the clicked document, and user clicks on current SERP, document reading level features. We tested all the models on TREC 2013 and TREC 2014 Session track data and found out our models do not do better than mean or median dwell time as estimated dwell times. However the models captured the actual dwell time distributions by relevance levels.

## Chapter 5

### REFORMULATE OR QUIT: PREDICTING USER ABANDONMENT IN IDEAL SESSIONS

All search engine users eventually abandon a search. This may be due to obtaining the information they sought throughout the search session or due to dissatisfaction with the results shown by the search engine or simply due to the amount of effort spent on searching. Detecting when and why a user is close to abandoning a search session is an important problem in search for information needs in order for a system to be able to take precautions and respond to users' information needs.

In this chapter, we present a comparison of different feature sets for detecting session abandonment. We have two groups. The first group of features are extracted solely from a SERP: title of a snippet, text of a snippet, URL text. The second group of features are extracted from the session and interaction. We develop machine learning models and tested them with low noise: fully-segmented sessions on a single topic, with the topic description available as well as sessions by other users for the same topic. We found that the latter set is far superior to the first set, which actually degrades effectiveness at predicting abandonment <sup>1</sup>.

#### 5.1 Experiment Data

For our experiments, we use the 2014 TREC Session track data [13], which consists of a large amount of logged user actions in the course of full search sessions for pre-defined topical information needs, for training and testing our models. The

---

<sup>1</sup> This work previously published in the proceedings of the 12th Asia Information Retrieval Societies Conference, AIRS

experiment data was collected from workers on Amazon’s Mechanical Turk. In total there are 1257 unique sessions.

As ensured by the track protocol, sessions in this data have clearly marked start and end points, and are entirely related to a given topic. Thus this data is much cleaner than standard search engine log data. From this data we used 1063 full sessions, each of which is made up of a sequence of interactions. We removed sessions that have interactions more than 10 search results, sessions that have only 1 interaction and sessions that have consecutive interactions with the same user query. Thus sessions in our training and testing data have interactions consist of one query which does not repeated in the following interaction, 10 ranked results for that query from a search engine, and user clicks and dwell times on those results.

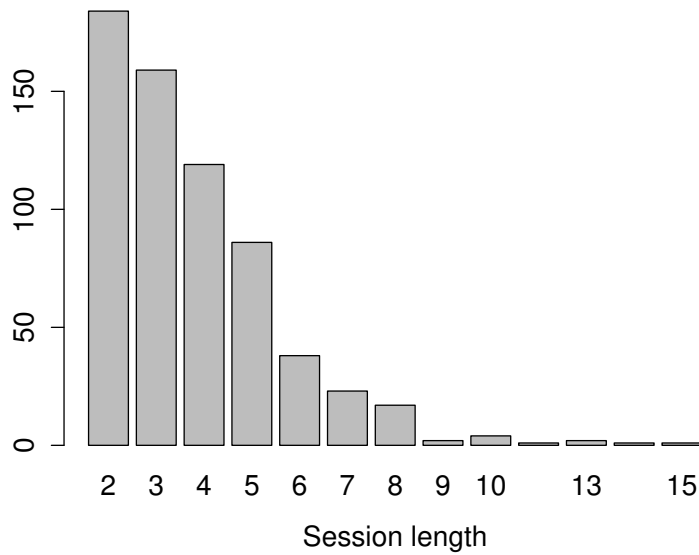
In total there are 2400 instances in the data (2400 interactions across 1063 sessions), of which 1763 are non-abandonment and 637 are abandonment. The average number of interactions per session is 3. Figure 6.2 shows the histogram of interactions per session.

## 5.2 Methods

Our main assumption for a proposed session abandonment prediction model is that users abandonment decision base primarily on three factors:

1. Information satisfaction: User gathered all the information he/she needed through out the search session.
2. Query coverage: All possible queries were submitted to the search engine.
3. Cost of search process: User spent too much time on the interactions and on the session.

Thus, we formulate the complete search experience through the features of document titles, URLs, and snippets on the current search result page, the queries submitted to the search engine in the history of the users’ current session, the relevancy of seen documents on the current search result page and through out the current search



**Figure 5.1:** Histogram of number of interactions per session

session, dwell time on relevant, non-relevant and on whole clicked documents, total duration of the interaction and the session.

Additionally, since each topic is provided to many different users, there are sessions with the same information need that were completed by different users in the experiment data. We extracted features related to other users' sessions in predicting the session abandonment of a user with the same information need.

### 5.2.1 LETOR Features

The first set of features is derived from those that have been used in LETOR [67] and other datasets [14] [61] for learning to rank. For details of this features, readers are encouraged to see section 2.2.4.

## 5.2.2 Session and interaction features

The second set of features includes features based on document relevance, durations of actions, similarity between queries and topic statement, and similarities between a user’s actions and those of other users working on the same topic. All features in this set are summarized in Table 5.1. We describe them in more detail below.

### 5.2.2.1 Query approximation

TREC Session track experiment participants are provided with a topic description for use to guide their interactions with the search engine. We extract key-phrases from the topic description using the Alchemy API extraction tool. We also combined all queries that were submitted by other users working on the same topic. For each interaction, by using the previously submitted queries in prior interactions by the user, we calculate an approximation of submitted user queries to newly formed Alchemy topic key-phrases and other users’ combined query.

The approximation of query  $Q_i$  to  $Q_j$  is computed as follows:

$$\text{Approximation}(Q_i, Q_j) = |Q_i \cap Q_j| / |Q_j| \quad (5.1)$$

We de-case all letters, replace all white space with one space, remove all leading and trailing white spaces, replace punctuation, remove duplicate terms, and remove stop words. Thus each query is represented as a bag of non-stopwords.

In order to calculate  $|Q_i \cap Q_j|$ , we consider two terms equivalent if any one of the following four criteria are met:

1. The two terms match exactly.
2. The Levenshtein edit distance between the two terms is less than two.
3. The stemmed roots of the two terms match exactly.
4. The WordNet Wu and Palmer measure is greater than 0.5.

The full steps of feature generation are summarized as follows.

1. For each session S with topic t:

- (a) Combine and form a vector of non-stopwords from queries from other sessions with the same topic  $t$ .
- (b) For each interaction  $i$ :
  - i. Combine and form a vector of non-stopwords from queries from previous interactions.
  - ii. Calculate query approximation between the vector that is formed from extracted key-phrases of topic description and vector formed in i.
  - iii. Calculate query approximation between vector formed in (a) and ii.

#### **5.2.2.2 Relevance related features**

The relevancy judgements of documents according to the topic description is used as a feature and as a source to extract some other features. Features we use include relevancy grade of each document in the interaction, total relevant and non-relevant document counts in the interaction, and total relevant and non-relevant document counts appeared in the session up to the current interaction. High positive scores of relevancy indicates that the document is more relevant to the topic.

#### **5.2.2.3 Duration-based features**

Duration-based features extracted from the session data include start time of each interaction, time spent in each interaction, time spent in each clicked document, total time spent in clicked relevant documents which have relevance judgement of 1, total time spent in highly relevant clicked documents which have relevance judgement of 2 or higher, total time spent by user on scanning and reading documents including the last interaction.

#### **5.2.2.4 Other sessions based features**

Other users' sessions on the same topic are also used to extract features. Their queries, clicked documents, documents appeared on sessions are collected. The features related with other user sessions are, total number of other users' clicked documents appearance in current and previous interactions, its ratio, total number of other users' sessions' documents appearance in current and previous interactions and its ratio.

**Table 5.1:** List of session and interaction features.

Feature	Notes
int_order_num	Interaction number in a session
int_start_time	interaction start time
int_duration	interaction duration
int_duration_nr	dwel time on NR documents in interaction
int_duration_r	dwel time on R documents in interaction
int_duration_hr	dwel time on HR documents in interaction
search_duration	time spent on current and previous interactions
doc_rank_n_rl	relevancy of document at rank n
doc_duration_n	time spent on document at rank n
topic_alchemy_app	query approximation to alchemy keywords
other_queries_app	query approximation to other users' queries
non_rel_count_ses	number of NR documents appeared in session
rel_count_ses	number of R and HR documents appeared in session
non_rel_count_int	number of NR documents appeared in interaction
rel_count_in	number of R documents appeared in interaction
seen_clicked_docs_count_ses	number of seen clicked documents from other sessions
seen_clicked_docs_ratio_ses	ratio of seen clicked documents from other sessions
seen_clicked_docs_count_int	number of seen clicked documents in the interaction from other sessions
seen_clicked_docs_ratio_int	ratio of seen clicked documents from other sessions
seen_doc_ratio	ratio of seen documents from other sessions through out the current session

### 5.2.2.5 Other features

Interaction order number is the the number of the interaction within a session. Actions including the first query and any clicks on results retrieved for that query are associated with interaction order number 1; actions starting from the second query up to just before the third query are associated with interaction order number 2; and so on.

## 5.3 Experiments

We compare our session abandonment feature sets by their effectiveness at predicting actual user abandonments, using standard classification evaluation measures like precision, recall, AUC, and classification accuracy. We trained and tested random forest models using all selected sessions from the TREC 2014 Session track. For each interaction that appears in the Session track data, we have one instance for training/testing: the 0/1 label indicating whether the session was abandoned immediately after that interaction or not.

In total there are 2400 instances in the data (2400 interactions across 1063

**Table 5.2:** Precision/recall/AUC/accuracy of our four models

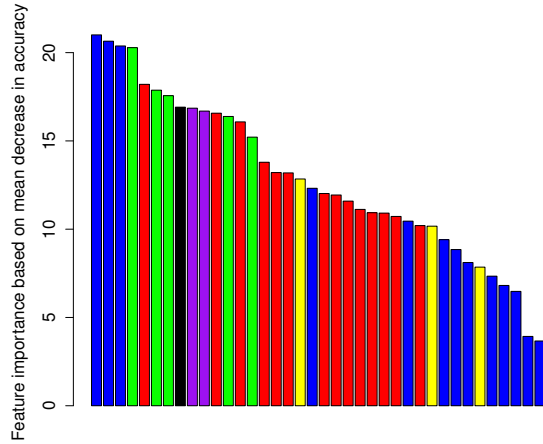
method	precision	recall	AUC	accuracy
baseline	0.438	0.320	0.718	0.710
LETOR	0.273	0.267	0.529	0.617
Session+Interaction	0.536	0.752	0.841	0.762
LETOR+Session+Interaction	0.452	0.457	0.681	0.708

sessions), of which 1763 are non-abandonment and 637 are abandonment. Since the class distribution is so skewed, we re-balanced the training data with SMOTE, which creates artificial data for the under-represented class [16]. We trained and tested using four-fold cross-validation and report micro-averaged evaluation measures aggregated across all four testing splits. Note that only training data, not testing data, in each fold is rebalanced with SMOTE.

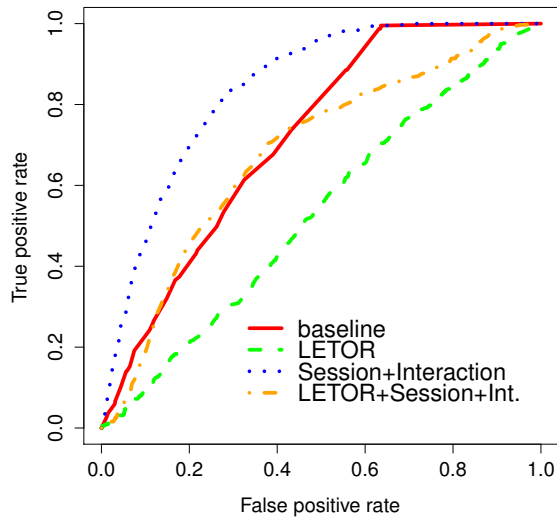
## 5.4 Results

We tested 4 models. The first model, a simple baseline, uses only the interaction order number. The second model uses the LETOR features listed in Table 1 for every document appeared in the interaction. The third model uses the session and interaction features listed in Table 2, and the fourth model uses all features. Table 6.4 and Fig 5.3 summarizes the performance of the four models. The baseline using only interaction order number for training performs better than model 2 (which only uses LETOR features) in all measures, *and* model 4 (which uses all features) in AUC and accuracy (though that model scores slightly higher in precision and substantially higher recall). The best achieving model is the model that uses only session and interaction features. It improves precision over the baseline by 22%, recall by 135%, AUC by 17% and accuracy by 7%.

Figure 5.2 shows the feature importance of the random forest that is trained with session and interaction based features. Mean importance of the features that are extracted by using the other sessions with the same topic is 17.46, only interaction id is 16.91, query approximation features is 16.77, relevancy based features is 12.89,

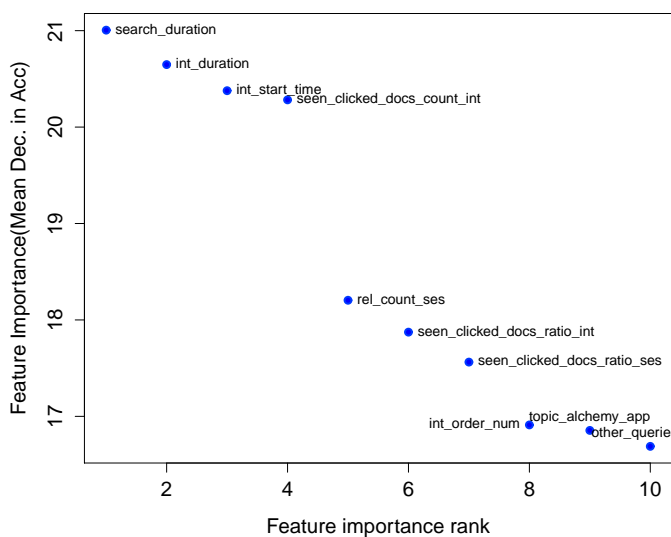


**Figure 5.2:** Feature importance based on mean decrease in accuracy. Key: blue: duration features; green: features extracted using other sessions with same topic; black: interaction id; purple: query approximation features; red: relevance features; yellow: duration and relevance features



**Figure 5.3:** ROC curves for different session abandonment prediction methods

duration based features is 12.89 and lastly duration and relevancy based features is 10.29.



**Figure 5.4:** Top 10 features according to mean decrease in accuracy

Figure 5.4 shows the top 10 features based on the mean decrease in accuracy. The first three features are unsurprisingly based on session and interaction duration. The fourth feature is the number of clicked documents appeared in the current interaction. The fifth is the number of relevant documents seen in session including the current interaction. The last two features are query approximations.

## 5.5 Conclusions

We have compared two different sets of features for predicting abandonment: one set is more like LETOR, including many features related to query/document similarity; the other includes a collection of features derived from the session and topic. We tested them in a setting with low noise: fully-segmented sessions on a single topic, with the topic description available as well as sessions by other users for the same topic. We found that the second set is far superior to the first set, which actually degrades effectiveness at predicting abandonment. We also found that the inclusion of durability-based features in the second set is not the sole reason they perform better.

## Chapter 6

### EVALUATION OF MODELS WITH A NEW DATASET

In previous chapters we introduced machine learned models that predict and simulate different user actions on search sessions. In chapter 3, we introduced several user click models that predicts user clicks on search session. As a complementary work to user click models, in chapter 4, we estimated user dwell time on clicked documents. Finally in chapter 5, we predict session abandonment. We trained and tested all of our models on TREC 2013 Session track and TREC 2014 Session track data. These two session test collections consist of web user search sessions that were collected by using Clueweb12.

In this chapter, we first create a new session test collection from a news collection and we evaluate our models on newly created session data which has different properties than web session data. We see that even the data type is different, our machine learned models perform similar improvements on similar feature sets. These experiments demonstrate the generalizability of our approaches.

#### 6.1 Session Data Collection

In this section, we explain how we created new session test collection from a news collection.

##### 6.1.1 Corpus

We used New York Times annotated corpus that is published by Evan Sandhaus in 2008 under LDC Catalog No. LDC2008T19. For details of this collection, readers are encouraged to see section [2.4](#).

### 6.1.2 Topics

We used a subset of 2017 TREC Dynamic Domain track [75] topics and rewrote them to make them easier to understand and work on. The 8 topics we used are as follows;

1. The abortion pill RU-486 was approved for distribution to women in the United states under the name of mifepristone. Find news stories covering the history of the abortion drug RU-486, known as mifepristone in the United States, from its development in France to approval by the Food and Drug Administration for termination of early pregnancy.
2. After years of stops and starts in a nuclear program, North Korea announced that it has developed nuclear weapons. Find news stories reporting on North Korea's progress in developing nuclear weapons, including its pledges to stop development and subsequent backtracks.
3. Benazir Bhutto, Pakistan's prime minister, was convicted of corruption and misuse of power. Find news stories reporting on her convictions, trials, penalties, and sentencing, and her actions in response.
4. Find information about health benefits, both positive and negative, from antioxidants in supplements and foods and beverages.
5. Find information on the Melissa virus's impact on automated systems, its methods of infection, and particularly its financial damage including its effect on the stock market.
6. Drug testing reveals that professional athletes are using illegal drugs to enhance performance. Find news stories reporting on the start of periods when drug tests uncovered performance-enhancing substances in the blood serum of pro athletes.
7. After the U.S. surgeon general declared nicotine an addictive substance the federal government took action to regulate sale of cigarettes to young people. Find news stories reporting on the dangers of nicotine and the actions taken by the federal government and tobacco companies to address the problem.
8. Find articles reporting on modern-day efforts, on or off ship, to prevent ships being boarded or captured by pirates specifically for monetary gain not to prevent smuggling, pollution, terrorism, or for political reasons.

### 6.1.3 Sessions

As described in previous sections, a session is a series of actions including user queries, their results, and user clicks as a result of an information seeking behavior.

For this task, we employed a crowdsourcing platform, Amazon Mechanical Turk, from September to October 2018. We only employed workers with master title and have work acceptance rate of 95%.

During the experiment, we presented topics to users which describe the information need clearly. Users were provided a fully-functional search engine to perform news searches and satisfy the assigned information need. See Figure 6.1 for screenshot of search engine. Users were asked to perform search on the provided search engine for 3 minutes for each topic and they were paid \$0.5 per session. At the end of the sessions, users were directed to a questionnaire. The questionnaire consisted of two questions. The first question was *"Please rate your satisfaction with the search results returned from 1 to 5"* and the second question was *"Are you sufficiently informed? Rate from 1 to 5"*. Each submitted session was reviewed and sessions with unrelated queries removed.

The search system used an indri index of The New York Times annotated corpus which was indexed using the Krovetz stemmer and no stopping. Apart from the complete document text index, the title and body fields were indexed as fields.

Each user query was formulated with an indri query language template and submitted to the search engine. The template for the query *"benazir bhutto conviction"* was: `combine(combine(benazir.body bhutto.body convict.body)) 50 combine(combine(benazir.body bhutto.body convict.body))`. The search engine retrieved 50 results and 10 results were shown per page. The retrieved results were cached to speed up the retrieving process.

The system recorded user's interactions with the search system, including issued queries, reformulation of those queries, clicks on the displayed results with their timestamps along with unique user ids. When data collection phase was done, we reviewed the acquired data and selected a subset of it.

The final dataset consists of 122 user sessions with an average length of 3.24 interactions. 94 of these sessions have at least one reformulation, 75 have at least two, 49 have at least three, 27 have at least four, 15 have at least five, and 1 has at least 10

After the U.S. surgeon general declared nicotine an addictive substance the federal government took action to regulate sale of cigarettes to young people. Find news stories reporting on the dangers of nicotine and the actions taken by the federal government and tobacco companies to address the problem.

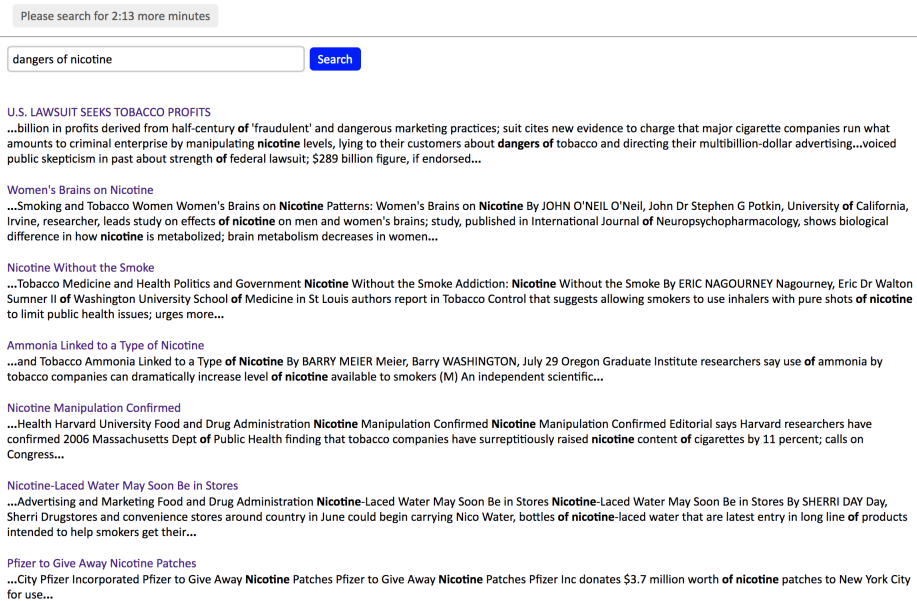


Figure 6.1: User interface of The New York Times data search engine

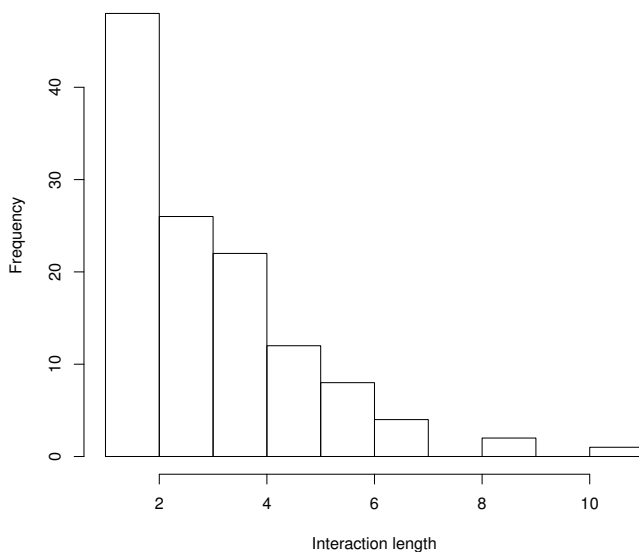
reformulations. Figure 6.2 shows counts of sessions per session length.

## 6.2 Evaluation of User Click Models

In this section we revisit the user click models that were presented in Chapter 3, evaluate their effectiveness at predicting actual user clicks, using standard classification measures like precision, recall, and AUC on The New York Times session data and compare them with the results that we showed in section 3.5.

### 6.2.1 Features

We use the same features that we presented in sections 3.2.1 and 3.2.2 except features specific to web documents such as URL length, number of slashes in URL, URL based LETOR features and web page popularity.



**Figure 6.2:** Histogram of number of interactions per session

### 6.2.2 Models

We use the same models that we presented in section 3.3. The models are summarized in Table 3.4. M1 is the independence model which only uses LETOR features of a snippet to predict user click. M4 is the dependence model which uses LETOR features of the snippet that we try to make prediction along with the other 9 snippets’ LETOR features that are listed in the same SERP. M7 extends M4 by including user clicks on the current SERP. Moving horizontally from left to right in Table 3.4 adds session features and other session features to the models and moving vertically from top to bottom changes models from independence to dependence and adds user click decisions in the current SERP.

### 6.2.3 Experiments

In total there are 3,717 instances in the data, of which 3,294 are no-clicks and 423 are clicks. Since the class distribution is so skewed, we rebalanced the training

data with SMOTE, which creates artificial data for the under-represented class [16]. We trained and tested using four-fold cross-validation and report micro-averaged evaluation measures aggregated across all four testing splits. Note that only training data, not testing data, in each fold is rebalanced with SMOTE.

#### 6.2.4 Results

Table 6.1, Table 6.2 and Figure 6.3 summarize performance of the nine models. Table 6.1 and Table 6.2 are organized in the same way with Table 3.4 in which going from left to right extends feature sets with session related features and clicks from other sessions respectively. Going from top to bottom changes the models from independence to dependence and dependence to dependence with user clicks on the same SERP.

The top left cell is the baseline. Moving to the right, just adding session related features to the baseline increases AUC by %2.3 and precision by %14.0, but decreases recall by %5.4. Using clicks from other sessions along with the session data results in an 10.1% increase in AUC and 20.6% increase in precision, and 1.9% increase in recall. The changes in AUC and precision when moving from left to right are similar to the results that we presented in section 3.5. However we see a slight difference in recall value changes. Recall values drop significantly from M1 to M2 and M2 to M3 in models that were trained and tested in 2013, 2014 TREC Session track data.

Moving down from the top left cell increases precision by 7.9% and AUC by 7.1%. From there, moving right to add session-related features has less influence on the dependence model. Only adding these features increases the precision by 7.9% and AUC by 3.9%, while decreasing recall by 16.8% at first, and recovers from that drop at M6. Using other click decisions in an interaction has positive influence on all measures. Finally, moving down to the last row, we see a 0.6% improvement in AUC when we move from M4 to M7. Similar improvements as above are achieved in adding session data to produce the M8 and M9 models. The general tendency of the performances shown on the table demonstrate that moving from top to bottom in models increases AUC and precision, and decreases recall. Moving from left to right mostly increases

**Table 6.1:** Precision/recall of our nine models, c.a.s. stands for clicks across sessions

ranking features	session features		
	none	session history	history + c.a.s.
target snippet	0.300/0.352	0.342/0.333	0.362/0.359
target snippet + all others	0.326/0.354	0.314/0.303	0.352/0.359
target snippet + all others + clicks	0.327/0.342	0.335/0.343	0.354/0.336

AUC and precision, decreases recall at first and recovers from that drop at the last column.

Figure 6.3 compares the ROC curves for related groups of models. Reading figures from left-to-right/top-to-bottom, we see that changing the model from independence to dependence and adding user click info in the same SERP that we make prediction (going from M1 to M4 to M7) pushes the ROC curve out, and adding that information with session context (M2 to M5 to M8) pushes it out further. When both session information and features from other sessions are used, changing independence to dependence model and user click information does not contribute much to the ROC curve (M3 to M6 to M9).

The next figure shows a clear improvement by pushing the ROC curve to left from using session history when independence model is used (M1 to M2 to M3). But as we change the independence model to dependence such that we add more information about the results page, the session history seems to provide less impact on the ROC curve (M4 to M5 to M6 and M7 to M8 to M9).

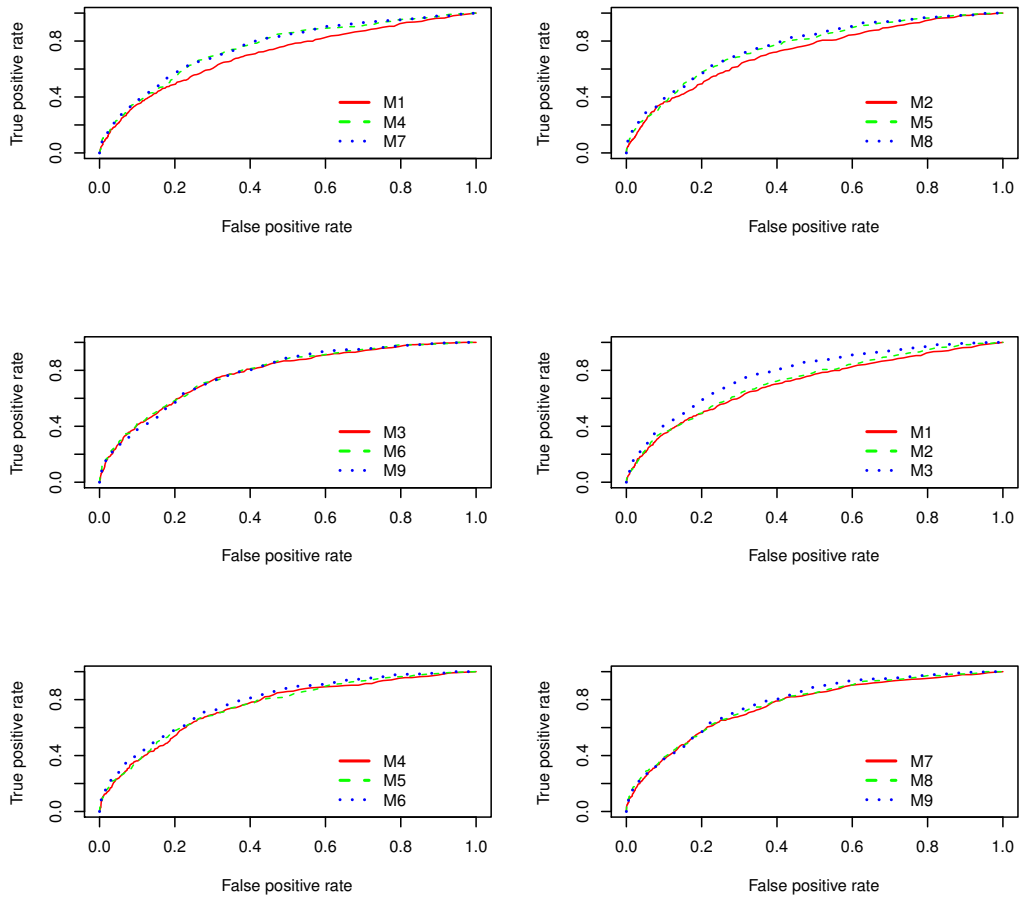
Nevertheless, when Table 6.1, Table 6.2 and Figure 6.3 are considered together, the behavior of our models in both 2013, 2014 TREC Session track and The New York Times session data is very similar.

### 6.3 Evaluation of Session Abandonment Prediction

In this section we evaluate the session abandonment prediction models that we presented in section 5.2 with the New York Times session data. We present the results and compare them with the results shown in section 5.4.

**Table 6.2:** AUC of our nine models, c.a.s. stands for clicks across sessions.

ranking features \ session features	none	session history	history + c.a.s.
	target snippet	0.705	0.721
target snippet + all others	0.753	0.759	0.783
target snippet + all others + clicks	0.758	0.768	0.780



**Figure 6.3:** ROC curves of models from left to right and top to bottom

### 6.3.1 Features

We used two sets of features for the experiments in Chapter 5 which are LETOR features and session and interaction features. LETOR features extracted from fields of snippets such as title, URL and snippet text and their relation with the user query. For more detail readers are encouraged to see section 2.2.4. Since we use the New York Times session data which consists of news articles that lack URL and web related features, LETOR features related to URL, Waterloo spam score and web page popularity are missing. We use the same session and interaction related features that are listed in section 5.2.2.

### 6.3.2 Experiments

We compare our session abandonment feature sets by their effectiveness at predicting actual user abandonments, using standard classification evaluation measures like precision, recall, AUC, and classification accuracy. We trained and tested random forest models using all selected sessions from the New York Times session data. For each interaction that appears in data, we have one instance for training/testing: the 0/1 label indicating whether the session was abandoned immediately after that interaction or not.

In total there are 381 instances in the data (381 interactions across 122 sessions), of which 258 are non-abandonment and 123 are abandonment. Since the class distribution is so skewed, we re-balanced the training data with SMOTE, which creates artificial data for the under-represented class [16]. We trained and tested using four-fold cross-validation and report micro-averaged evaluation measures aggregated across all four testing splits. Note that only training data, not testing data, in each fold is rebalanced with SMOTE.

### 6.3.3 Results

We tested 4 models. The first model, a simple baseline, uses only the interaction order number. The second model uses the LETOR features other than URL related

**Table 6.3:** Precision/recall/AUC/accuracy of our four models

method	precision	recall	AUC	accuracy
baseline	0.474	0.146	0.570	0.670
LETOR	0.167	0.049	0.490	0.610
Session+Interaction	0.603	0.382	0.724	0.720
LETOR+Session+Interaction	0.277	0.106	0.530	0.620

features, Waterloo spam score and alexa ranking listed in Table 2.1 for every document appeared in the interaction. The third model uses the session and interaction features listed in Table 5.1, and the fourth model uses all features. Table 6.3 summarizes the performance of the four models. The baseline using only interaction order number for training performs better than model 2 which only uses LETOR features in all measures, and model 4 (which uses all features) in AUC and accuracy (though that model scores slightly higher in precision and substantially higher recall). The best achieving model is the model that uses only session and interaction features. It improves precision over the baseline by 27.2%, recall by 161%, AUC by 27% and accuracy by 7.5%.

The performances of the models in precision, recall and AUC that are summarized in Table 6.3 are similar to performances of the models in 2014 TREC Session track data that is shown in Table 5.4.

## 6.4 Evaluation of Dwell Time Prediction

In this section we evaluate the dwell time prediction models that we introduced in section 4.2 on the New York Times session data and compare the performance of our models with their performances in TREC 2013 and TREC 2014 session track data.

### 6.4.1 Features

We use LETOR features of the clicked document that we are trying to predict the dwell time and snippets that are displayed along with the clicked document on a SERP. All LETOR features are listed in Table 2.1. Since the New York Times corpus

consists of news articles, our feature set does not include web document related LETOR features that depend on URL, Alexa ranking, and Waterloo spam score.

#### **6.4.2 Models**

We use the 3 models; independence, dependence and dependence with user clicks that are presented in section 4.2.

#### **6.4.3 Experiments**

We trained and tested our regression random forest models using all clicked documents on the New York Times session data. For each clicked ranked result that appears in the New York Times session data, we have one instance for training/testing: the dwell time indicating the duration that a user spent on a web document, the features derived from that web document, and, depending on the model, snippet features of other results ranked with it, clicks done on other results and reading level features. In total we have 276 training/testing instances. We performed four-fold cross validation and report the micro averaged results.

#### **6.4.4 Results**

We define RMSE values of mean and median of dwell times of clicked documents in the New York Times session data as our baselines. The mean and median of dwell times are 20.59 seconds and 12.33 seconds respectively. We report the performances of our models in RMSE in Table 6.4. None of our models perform better than mean baseline. All three models are close to the mean/median, though unlike the click models, taking into account other ranked results does not help prediction performance. User clicks in other snippets ranked along with the clicked document that we are trying to estimate dwell time improve RMSE slightly. These results correlate with the results we saw in section 4.3.1. Our best achieving model is close to performance of mean dwell time in both TREC 2013 and 2014 Session Track data and The New York Times session data.

**Table 6.4:** RMSE of dwell time estimation of methods and their comparisons to mean and median dwell times.

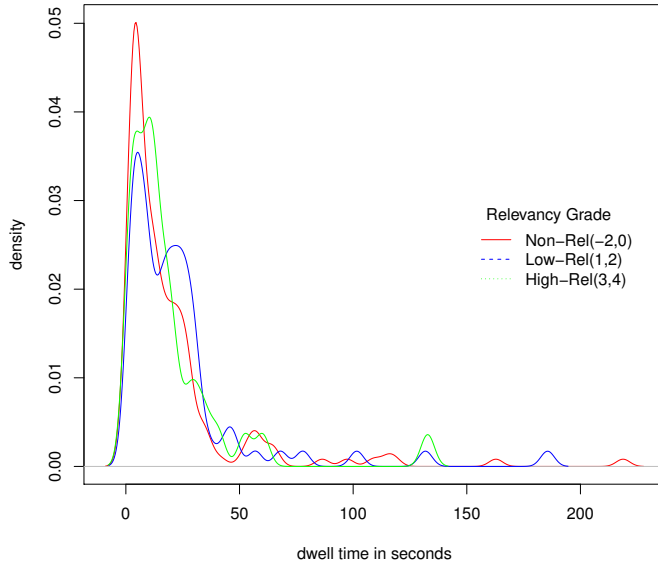
method	RMSE	$\% \Delta_{mean}$	$\% \Delta_{median}$
mean	28.02	-	-
median	29.21	-	-
1. independence	30.05	-7.2%	-2.87%
2. dependence	29.94	-6.85%	-2.49%
3. dependence+clicks	28.95	-3.31%	-0.89%

We also looked at the density graph of actual and predicted dwell times grouped by relevance class. For documents with relevance judgments, we binned into three classes: those that were judged non-relevant (0) or those that were judged relevant (1,2) or highly relevant (3,4). Then we plotted the density of dwell times within those classes. Figure 6.4 shows results for actual dwell times and Figure 6.5 shows results of our second model which is similar to density graphs' of other two models. Our three models all produce distributions with lower peaks and less separation in terms of peaks than actual dwell times, but the separation of relevant class seems (by visual inspection) to be similar. Even though our models do not do better than mean, their predicted dwell time density graphs show models can capture some properties of dwell times.

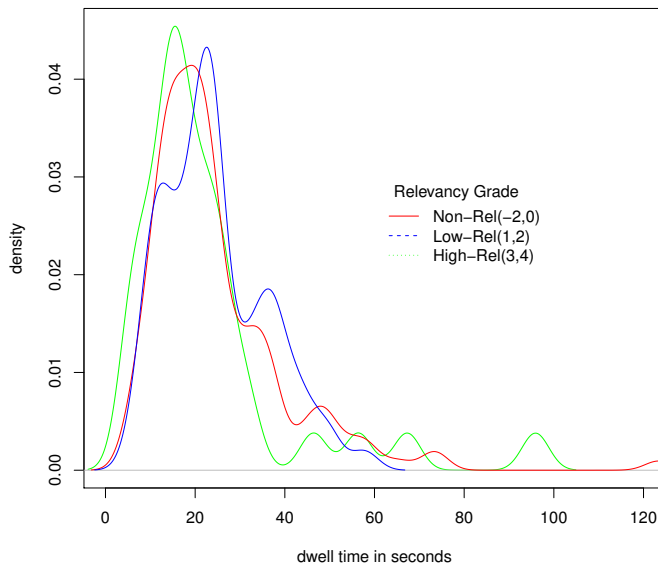
## 6.5 Conclusions

In this chapter we created a news search session data and trained and tested our models that were introduced in previous chapters on the collected data. For creation of new session data, we used the New York Times annotated corpus and a subset of 2017 TREC Dynamic Domain track [75] topics. We employed a crowdsourcing platform, Amazon Mechanical Turk, with workers satisfying certain quality standards. For this purpose we built a search engine with web interface and provided it to experiment participants. We collected 122 user news search sessions for 8 topics. Average number of query reformulations is 3.24.

After creating a decent size news search session corpus, we trained and tested our



**Figure 6.4:** Density of actual dwell times by relevance class.



**Figure 6.5:** Density of actual dwell times generated by dependence model by relevance class.

click models that was first introduced in chapter 3. Even though we did not see same results in terms of AUC, precision and recall with the news session data as it was in TREC Session track data, the transition between models showed similar performance changes. Adding session history and other sessions' data resulted with increase in precision and AUC. Changing the model from independence to dependence and using other clicks on the same SERP resulted with slight increase in precision and AUC. These were the similar behaviors we saw in TREC Session track data. This shows that our click models' performances are independent of data.

We then trained and tested our dwell time estimation regression models with the news session data. Our regression models performed around mean and median dwell times which were defined as baselines. This poor performance was also seen in TREC Session track dataset. However in both of these datasets, our models showed a similar dwell time density distribution in terms of relevancy grade with the actual density distributions.

Lastly, we trained and tested our session abandonment prediction methods. In both TREC Session track dataset and news session dataset we experienced the positive performance effect of using session and interaction related features and negative effect of using LETOR features.

## Chapter 7

### QUERY SIMULATION

In previous chapters we introduced simulation models of various user actions on a search session such as clicking to documents, abandoning sessions and spending time on clicked documents. In this chapter we introduce query generation models and evaluate the models in different settings.

In a typical search scenario, a user starts to a search session with an initial query with a specific information need in-mind or not and continues to reformulate and refine their queries until their information need is satisfied or completely abandon the search. In order to simulate the queries in a search session we use predefined information needs and user's session history as basis for query generation. <sup>1</sup>.

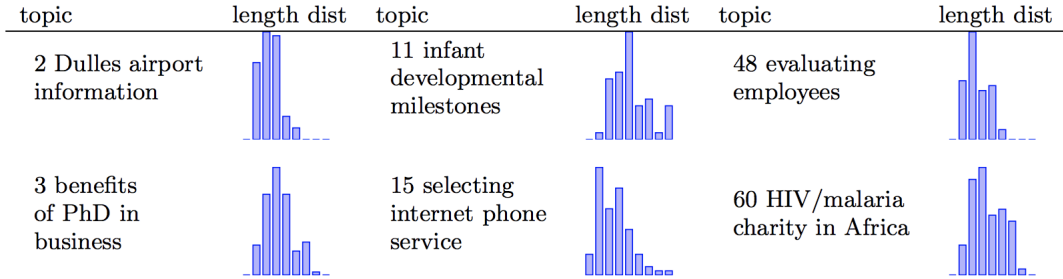
#### 7.1 Simulating Queries

Simulating queries is the most difficult aspect of simulating users. We use a two-phase process by which we first generate queries by sampling from a language model, then score them based on their discriminative power among topics. The language model we use is actually a series of binomial models with parameters  $P(w \in Q|T, i, S_{1..i1})$ , i.e. the probability of a term  $w$  being in a query  $Q$  given information about a topic  $T$ , the current point in the session  $i$  (a discrete number from  $i_1$  to  $i_{max}$ ), and the history of the session up to that point  $S_{1..i-1}$ .

A key component of our model is that it models query length conditional on topic. Some topics lend themselves more naturally to longer queries. Topic 2, for instance, is looking for information about Dulles International Airport in Washington,

---

<sup>1</sup> This work was previously published in the proceedings of the 2015 international conference on the theory of information retrieval. ACM, 2015.



**Figure 7.1:** Distributions of query lengths for a selection of topics (given as a topic number from the TREC 2014 Session track along with a brief topic description). Each bar plot shows the number of queries of lengths 1 through 10.

DC. *Dulles* is an extremely important term for queries on this topic; it appears in almost every real user query. *Airport* is also an important term appearing in most user queries. Because most queries will contain those two terms by default, queries on this topic should include 3–4 terms to find the specific information requested in the topic description (nearby hotels, parking, shuttles to the airport, buses, etc). For topic 10, on arguments for instituting a tax on “junk food”, most queries include the phrase “junk food tax” and therefore have minimum length of 3. Figure 7.1 shows distributions of query lengths for a selection of topics. Therefore our model starts by marginalizing over query lengths:

$$P(w \in Q|T, i, S_{1..i-1}) = \sum_{\ell=1}^{\ell_{max}} P(w \in Q|T, i, S, \ell)P(\ell|T) \quad (7.1)$$

### 7.1.1 Topical Language Model

The basis of query generation model is formed on a binomial model for term presence in queries of length  $\ell$ . In simplest form, the model’s maximum likelihood estimate is the number of queries that contain the given term divided by the total number of queries in a given topic in the data. However, this binomial model can only generate queries with terms that exist in the data. In order to add new terms to the

vocabulary, the model’s maximum likelihood estimate is smoothed with additional text data such as topic descriptions and snippet texts of previous interaction depending on the availability. For the first interaction, the smoothed maximum likelihood estimate of this probability is defined:

$$P(w \in Q|T, \ell, i == 1) = \frac{qf_{w,\ell,T} + \mu \frac{to_{w,T}}{|T_{desc}|}}{qc_{\ell,T} + \mu} \quad (7.2)$$

In this equation,  $qf_{w,\ell,T}$  stands for the number of queries in topic  $T$  that contains term  $w$  with query length  $\ell$ . The term  $qc_{\ell,T}$  stands for number of queries with length  $\ell$  in topic  $T$ . In this function  $to_{w,T}$  is set to 1 if the term appears in topic description of  $T$  and to 0 otherwise.  $|T_{desc}|$  stands for number of unique words in topic description.

The MLE function is modified for other interactions as follows;

$$P(w \in Q|T, \ell, i) = \frac{qf_{w,\ell,T} + \mu \frac{sf_{w,T}}{|S|}}{qc_{\ell,T} + \mu} \quad (7.3)$$

The term  $sf_{w,T}$  is set to 1, if the term appears in the topic description, or in the title of the retrieved documents in the previous interaction, or in the snippet of the retrieved documents in the previous interaction.

### 7.1.1.1 Other Models

The models that we will introduce follow the same steps of the baseline method that is just explained. Our sole modification to the existing method will be to the source of data for smoothing of the maximum likelihood model. Smoothing is important not only because adding new terms to the query generation vocabulary other than existing user queries in the data, but also making use of user’s search experience by adding the terms that user reads on SERP and documents to the model.

In the baseline model, the smoothing data is title and snippet of the documents retrieved by previous query and the topic description, and there is an implicit assumption that the user has scanned all ten blue links. For our models, we smooth

the maximum likelihood model with clicked documents, snippets from previous interactions as well as external sources such as suggested queries by commercial search engines and user search query repositories. We summarize the models in the following lines.

For this model, we smooth the maximum likelihood model with all snippets in previous interactions with decreasing weights. The closer the interaction to the current interaction  $i$ , the more weight it have.

$$P(w \in Q|T, \ell, i) = \frac{qf_{w,\ell,T} + \sum_{j=1..(i-1), \mu_j \in C} \mu_j \frac{sf_{i,w,T}}{|S_j|}}{qC_{\ell,T} + \sum_{i=1..c, \mu \in C} \mu_j} \quad (7.4)$$

$C$  is set of weights for session interactions such that  $\mu_0 < \mu_1 < \dots < \mu_i$ . The term  $sf_{i,w,T}$  is set to 1, if the term appears in the topic description, or in the title of the retrieved documents in the  $j$ th interaction, or in the snippet of the retrieved documents in the  $j$ th interaction.  $|S_j|$  stands for number of unique terms in document titles and snippets in interaction  $j$ .

For this model we smooth the maximum likelihood model with snippets and clicked documents in previous interaction.

$$P(w \in Q|T, \ell, i) = \frac{qf_{w,\ell,T} + \mu_{sn} \frac{sf_{w,T}}{|S_{sn}|} + \mu_{doc} \frac{sdf_{w,T}}{|S_{doc}|}}{qC_{\ell,T} + \mu_{sn} + \mu_{doc}} \quad (7.5)$$

The term  $sf_{w,T}$  is set to 1, if the term appears in the topic description, or in the title of the retrieved documents in the  $i$ th interaction, or in the snippet of the retrieved documents in the  $i$ th interaction. Otherwise it is set to 0. The term  $sdf_{w,T}$  is set to 1, if the term appears in one of the relevant document that is clicked by a user or appears in any document that has dwell time over five seconds in the  $i$ th interaction. Otherwise, it is set to 0.  $|S_{sn}|$  stands for number of unique terms in document titles and snippets in previous interaction.  $|S_{doc}|$  stands for number of unique terms in clicked documents in previous interaction.

For this model we use previous interaction with related queries retrieved from

a commercial search engine.

$$P(w \in Q|T, \ell, i) = \frac{q_{f_{w,\ell,T}} + \mu_{sn} \frac{sf_{w,T}}{|S_{sn}|} + \mu_{rel} \frac{sr_{f_{w,T}}}{|S_{rel}|}}{q_{C_{\ell,T}} + \mu_{sn} + \mu_{rel}} \quad (7.6)$$

The term  $sf_{w,T}$  is set to 1, if the term appears in the topic description, or in the title of the retrieved documents in the  $i$ th interaction, or in the snippet of the retrieved documents in the  $i$ th interaction. Otherwise, it is set to 0. The term  $sr_{f_{w,T}}$  is set to 1, if the term appears in one of the related queries that are collected from a commercial search engine. Otherwise, it is set to 0.  $|S_{sn}|$  stands for number of unique terms in document titles and snippets in previous interaction.  $|S_{rel}|$  stands for number of unique terms in related queries.

### 7.1.2 Sampling Queries

We sample a query length  $\ell$  from a distribution  $P(L|T)$  calculated from training data. Then the terms in the language model are iterated from the highest to the lowest probability. In each iteration the term is sampled with 0.5 probability. This process is repeated until a query with length  $\ell$  is sampled. The procedure we follow gives greater probability for the highest-frequency terms in real queries to appear in sampled queries such that we boost the probabilities of the most common terms above what they would be otherwise. This is meant to mitigate a problem with language models, that their probabilities tend to underestimate the importance of common terms while overestimating the importance of rare terms. Table 7.1 shows examples of terms in our topics, their binomial model probabilities, their multinomial language model probabilities, and their frequency of occurrence when sampling using our approach. At each step  $i$  of the session, we generate  $N$  candidate queries. One query will be sampled from the set to be returned as the simulated reformulation.

### 7.1.3 Scoring Sampled Queries

After generating  $N$  candidates, each one is scored according to the probability that each word in the query could generate the topic that the query is meant for. This

**Table 7.1:** Top 4 most frequent terms appearing in queries for three topics with their binomial occurrence model probability  $P(w \in Q)$ , their multinomial language model probability  $P(w|Q)$ , and their frequency in queries sampled using the procedure in Section 6.1.1.

topic	term	$P(w \in Q)$	$P(w Q)$	$P(w Q')$
2	dulles	0.95	0.29	1.00
	airport	0.95	0.29	0.99
	hotel	0.13	0.04	0.17
	metro	0.12	0.04	0.19
11	milestone	0.71	0.14	0.87
	culture	0.66	0.12	0.85
	development	0.61	0.11	0.87
	infant	0.44	0.07	0.55
48	evaluate	0.98	0.28	1.00
	employee	0.93	0.27	0.95
	to	0.30	0.09	0.10
	how	0.19	0.06	0.00

is a way of scoring candidate queries by their ability to discriminate among the topics.

$$P(T|w) = \frac{qt f_{w,T} + st f_{w,T} + \mu \frac{1}{|T|}}{qt f_w + st f_w + \mu} \quad (7.7)$$

where  $qt f_{w,T}$  is the total number of times  $w$  appears in queries on topic  $T$ ,  $qt f_w$  is the total number of times  $w$  appears in all queries on any topic, and  $|T|$  is the total number of unique topics.  $st f_{w,T}$  and  $st f_w$  are the "session frequencies" of the term (in topic and across topics, respectively), and include the counts of the term in the topic description and in titles and snippets of documents retrieved for previous queries.

Each candidate query  $Q_j$  is scored as:

$$P(Q_j|T) = \prod P(w|T) \propto \prod P(T|w)P(w) \quad (7.8)$$

where  $P(w)$  is a prior probability of term  $w$ , which for now we treat as uniform.

The scores are then renormalized into a proper probability distribution, i.e.

$$P_{norm}(Q_j|T) = \frac{P(Q_j|T)}{\sum_{k=1}^N P(Q_k|T)} \quad (7.9)$$

and one candidate is sampled from this distribution to be the simulated reformulation.

#### 7.1.4 Summary

We have described a two-stage approach by which queries are first generated from a series of binomial distributions, then scored using a topical discriminator. The first part generates candidates that are practically guaranteed to contain the most important terms and phrases (as observed in real user queries), while the second part ensures that there will still be variety in other terms included in queries. We use TREC 2013 and 2014 Session track data to fit distributions  $P(i_{max})$ ,  $P(\ell|T)$  for each topic  $T$  in our set,  $P(w \in Q|\ell, T)$  for each query length for each topic  $T$ . The full steps are as follows:

1. For each topic  $T$ :
  - (a) Sample  $i_{max}$  and loop from  $i = 1$  to  $i_{max}$ :
    - i. Update  $P(w \in Q|\ell, T, i)$  using text in the topic description and results for query  $i$
    - ii. For  $j = 1$  to  $N$ :
      - A. Sample a query length  $\ell$
      - B. Sample a query  $Q_j$  by sampling 1/0 from  $P(w \in Q|\ell, T, i)$  until  $\ell$  terms sampled.
      - C. Score query by  $P(T|Q_j)$
    - iii. Sample one query  $Q_{sim}$  from  $P(Q|T)$  to send to client
    - iv. Receive retrieval results from the client
  - (b) Loop back to step (a) to simulate another session for the same topic

We note that instead of using our two-stage process, we could sample queries directly from a traditional multinomial language model. However, the queries such a model generates tend to not look much like real queries: rare terms appear too

frequently, terms that frequently appear together in real queries rarely occur together in sampled queries, etc. Some of these problems could be resolved using n-gram models and better smoothing, but our more heuristic approach seems to work well.

## 7.2 Framework and Data

We have selected a fixed set of 30 topics (a sample of the TREC 2014 Session track topics [13]) for which to generate sessions. We will use the TREC term run, which typically means the ranked results for every topic in a set, for the ranked results for every query in every session on every topic in the set. The simulation server can run multiple user simulations at the same time to generate M sessions per topic, so a final run will consist of ranked results for each query in each of the M sessions on each of the 30 topics. The number of queries per session will vary depending on the simulation model.

The user simulation currently generates the following information:

1. a first query for each session on each topic—with a fixed set of 30 topics, there will always be  $30 \cdot M$  queries for the first round of retrieval.
2. after receiving ranked results for queries:
  - (a) titles, URLs, and snippets for the top-10 ranked documents for each query (we limit to 10 for now for speed of response);
  - (b) clicks and dwell times on ranked documents;
  - (c) for each session, a decision to stop the session or continue;
  - (d) for each continuing session, a query reformulation.

## 7.3 Session Abandonment

For query reformulation simulation, we do not model "abandonment" in the sense of a user stopping a session due to success or frustration or any other reason real users might decide to stop. We simply sample a session length from a distribution fit to existing user data.

Let  $i_{max}$  be the length of a session (that is, the number of rounds of querying in the session). We will define:

$$P(i_{max}) = \frac{\text{\#of sessions of length } i_{max}}{\text{\#of sessions}} \quad (7.10)$$

To model "abandonment", we simply sample a session length from this distribution, and the session will stop after that many rounds of querying.

## 7.4 Evaluation

Finally we turn to evaluating simulated queries. This is by far the most challenging aspect of evaluation: there is an aspect of human judgment to it (i.e. do the queries look like queries a person would enter?) but for this task we care most about whether the queries we generate are "good" for evaluating retrieval systems, and in particular, whether they are good for evaluating systems that use features derived from session history. To that end, in this section we will evaluate simulated queries indirectly by evaluating retrieval systems that take them as input.

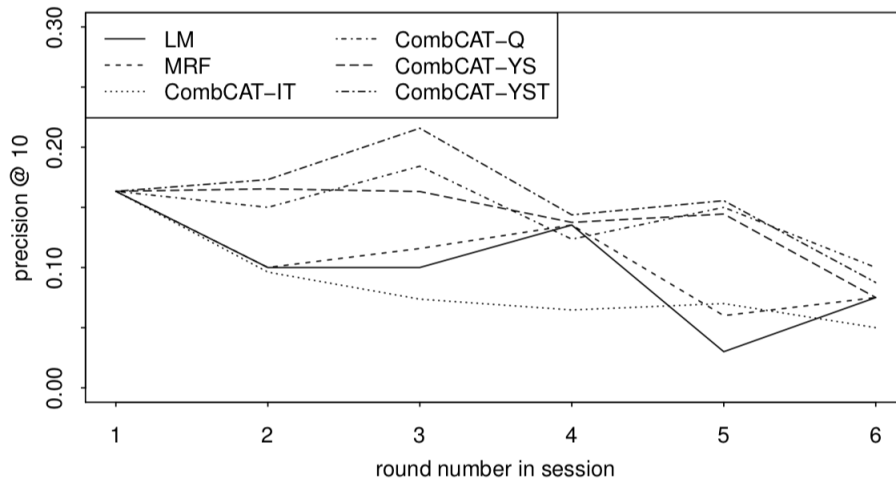
All of our systems are based on Indri and an index of ClueWeb12 that has been filtered using spam scores from the Waterloo spam classifier [22]. Two of them (LM and MRF) are ad hoc systems that treat each query as an independent event and do not make any use of session history. Four of them (CombCAT-\*) fuse different sources of data derived from the session history; these use the CombCAT fusion method that has been successful in the TREC Session track [2]. This method takes strings of text from various sources in the session history, uses those strings as a query, then fuses the results from all strings based on the frequency with which documents occur. One of the query simulation modules was providing user queries sampled uniformly from those submitted for the Session track; since they were being sampled randomly they are not based on session history. We refer to this as *non-sim*. Another model uses terms from previously-ranked documents to generate queries. We refer to this as *sim*. We evaluate all runs by precision@10 using TREC Session track relevance judgments.

### 7.4.1 Ad hoc evaluation using simulated queries

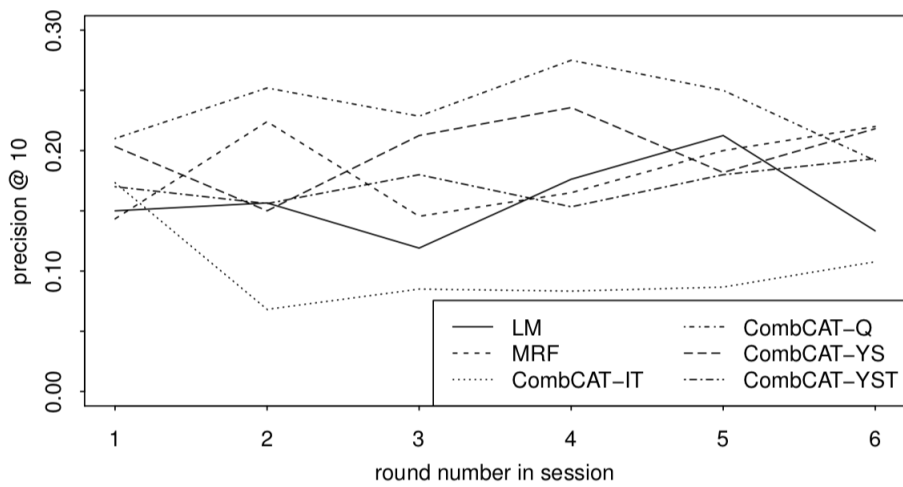
We first investigate the suitability of simulated queries for ad hoc evaluation, that is, the extent to which the queries provided in standard static IR test collections could be replaced with simulated queries. Averaged over all runs, sessions, and topics, the average precision@10 for non-simulated queries is 0.134 (0.016 confidence interval), while for simulated queries the mean is  $0.174 \pm 0.020$ . This difference is statistically significant by an unpaired two-sided t-test ( $p \approx 0.002$ ). To investigate this more closely, we matched evaluation measures as best we could: for each run, round number within that run, and topic number within that round, we paired the precision at 10 from the non-simulated case and the precision at 10 from the simulated case if both were available. There is still a difference in P10, but it is much smaller and not statistically significant:  $0.144 \pm 0.010$  for non-simulated queries versus  $0.152 \pm 0.011$  for simulated queries. Among this group, the average percent difference in P10 is about 5%. From this we conclude that simulated queries are not an unacceptable substitute for real user queries for ad hoc evaluation: the effectiveness measure values they produce are on the same “scale” as those produced by actual queries, though a bit higher on average. We will need to keep this in mind for the next sections, when we compare the ability of the two cases to evaluate systems.

### 7.4.2 Session evaluation using simulated queries

Next we looked at evaluation over the entirety of the session. Again we compare two cases: one in which sessions consist entirely of real user queries, the other in which sessions consist entirely of simulated queries. This time, our goal is to determine how ad hoc systems that make no use of session history compare to systems that do make use of session history. There is no widely accepted evaluation measure for a full session. We will look at mean precision@10 for each run, averaged over sessions in the non-simulated and simulated cases, for each of the first 6 rounds of the session. Figure 7.2 and Figure 7.3 shows the results. We note the following: the fact that simulated queries produce higher precision@10 is evident in these two plots;



**Figure 7.2:** Comparison of six retrieval systems across six rounds of a session using non-simulated queries



**Figure 7.3:** Comparison of six retrieval systems across six rounds of a session using simulated queries

**Table 7.2:** Overall precision@10 averaged across all sessions and all rounds in each session for each of our six systems.

rank	run	non-sim P10	run	sim P10
1	CombCAT-YST	0.1566	CombCAT-Q	0.2345
2	CombCAT-Q	0.1452	CombCAT-YS	0.2003
3	CombCAT-YS	0.1415	MRF	0.1830
4	MRF	0.1082	CombCAT-YST	0.1721
5	LM	0.1001	LM	0.1579
6	CombCAT-IT	0.0863	CombCAT-IT	0.1001

1. the differences between systems are more pronounced with simulated queries than with non-simulated queries;
2. the same system is more-or-less consistently at the bottom in both cases: CombCAT-IT, which uses titles of previously-ranked documents as queries, then fuses results for the current round.
3. the baseline LM ad hoc system is consistently among the lower-performing systems in both cases;
4. the top system is not the same: for non-simulated queries, CombCAT-YST appears better, while for simulated queries CombCAT-Q performs better.
5. Overall, our conclusion is that non-simulated queries and simulated queries provide similar overall session evaluation results.

### 7.4.3 Ranking systems using simulated queries

Next we look at the ability of simulated queries to rank systems by relative effectiveness. An important question about simulated queries is whether they are necessary: if simple non-simulated queries can evaluate systems effectively without taking any session history into account (as the previous evaluation suggests), are simulated queries needed at all? Thus our goal is to determine whether simulated queries can distinguish between systems of different effectiveness over the session more efficiently than non-simulated queries. Table 7.2 presents averages of the precision@10 numbers shown in Figure 7.2 and Figure 7.3 across all six rounds in the session. The rank correlation between the two is 0.6, mainly because the CombCAT-YST drops from rank 1 with non-simulated queries to rank 4 with simulated queries. It is difficult to say

that one ranking is more "correct" than the other, but we note that the differences between systems are larger with simulated queries than with non-simulated queries. This suggests that even if non-simulated and simulated queries agree on the relative ordering of systems, using simulated queries magnifies the differences.

#### **7.4.4 Actual sessions vs simulated sessions**

Finally, we return to the question of whether our simulated sessions "look like" real user sessions. We show some examples of real user sessions and simulated sessions for a random selection of our 30 topics in Table 7.3. Note that simulated queries often use very similar terms and phrases as actual user queries (though sometimes the ordering of terms in a phrase is lost in the simulated queries). Topic 11 shows an example of a user trying different possible features of a query language such as phrasing with quotes, boolean OR, and looking for a term in the title; none of this can be captured by the simulation as it currently exists. We do not draw any broad conclusions from this table; we only show it to give a sense of what our simulated queries and sessions look like compared to actual sessions.

### **7.5 Conclusions**

In this chapter we tried to model users' query generation and reformulation behavior by using their predefined information needs and session histories. We assumed users reformulate their queries based on the terms in their information needs and terms and phrases that they encounter during search processes. We evaluated the query generation method in a way that whether the queries we generate are "good" for evaluating retrieval systems, and in particular, whether they are good for evaluating systems that use features derived from session history. We evaluated simulated queries in different settings. We concluded that simulated queries are not unacceptable substitute for real user queries for ad hoc evaluation, non-simulated and simulated queries provide similar overall session evaluation results and non-simulated and simulated queries agree on the

**Table 7.3:** Examples of sequences of queries for six topics in our set. For each topic we show an actual user session of queries and a simulated session.

topic	query type	query
2	non-sim	dulles airport → dulles airport location → dulles hotels
2	sim	airport dulles hotels stop → airport dulles park → airport cheap dulles stop → airport dulles hotels metro → airport dulles metro near → airport dulles hotels metro
3	non-sim	jobs from business phds → business phd
3	sim	benefits business cost master → benefits business phd → business mba phd → business cost master phd → benefits business phd worth → business doctoral phd
11	non-sim	infants development culture → infant development "cultural effects" → infant OR child development intitle:culture → infant OR child development milestones → infant OR child development milestones research
11	sim	culture developmental infant milestones → culture infant milestones → infant milestones → culture developmental infant milestones → culture developmental milestones
15	non-sim	internet phone services → internet phone services review → guide internet phone services → voip providers
15	sim	providers reviews voip → services voip → cheapest internet phone services → providers reviews voip → features voip → providers reviews voip
48	non-sim	employee evaluation → evaluate employees
48	sim	employee evaluation → employee evaluation performance → employee evaluation guide → employee evaluation
60	non-sim	malaria impact on economy africa → malaria economy africa → malaria economy → malaria donations
60	sim	africa aids charity hiv malaria → charity hiv → aids charity → africa aids charity → charity hiv → africa aids charity fight hiv

relative ordering of the systems when we compare the performance of the systems on collections that are created by both of these query types.

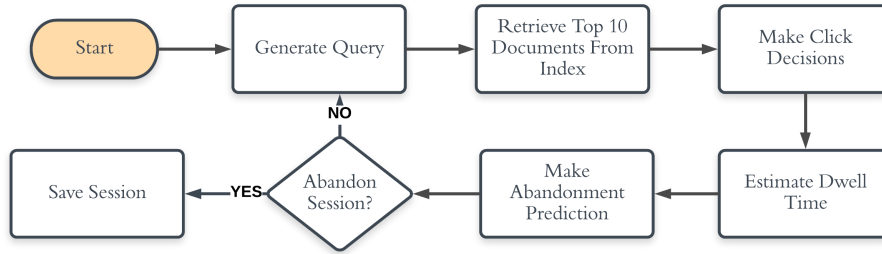
## Chapter 8

### A COMPLETE SESSION SIMULATION

An ideal user search session consists of a predefined information need, a clear beginning point, user queries, search engine result pages, list of clicked documents on those pages, time spent on those documents and an abandonment decision. In previous chapters we introduced machine learned models that generate queries, predict user clicks, estimate dwell time on clicked documents and make abandonment decisions. Thus we have all necessary models to create artificial search sessions. In this chapter we will generate artificial user search sessions by using the methods in previous chapters and we will ask following research questions; "*Can artificially created search sessions satisfy user information needs?*" and "*Can users classify artificial and actual user sessions?*". We will try to find answers to these questions with a user experiment.

#### 8.1 User Search Session Generation

In this section we explain how we generated user search sessions that contains queries, SERPs and clicked documents. The complete session generation method is summarized in Figure 8.1. The process starts with a generated query. Then top 10 documents related to that query are retrieved from document index, click prediction algorithm is run on the retrieved documents, dwell times on the clicked documents are estimated and session abandonment decision is made. If session abandonment is decided, the session is saved to sessions corpus. Otherwise, a new query is generated and process is repeated. In the following lines, we go over each of these steps.



**Figure 8.1:** User interface of The New York Times data search engine

### 8.1.1 Data and Framework

The search system we employ uses an indri index of The New York Times annotated corpus which is indexed using the Krovetz stemmer and no stopping. Apart from the complete document text index, the title and body fields are indexed as fields.

Each generated query is formulated with an indri query language template and submitted to the search engine. The template for the query *"benazir bhutto conviction"* is: "combine(combine(benazir.body bhutto.body convict.body)) 50 combine(combine(benazir.body bhutto.body convict.body))". For each query the search engine retrieves 50 results and 10 results are shown per page. The retrieved results are cached to speed up the retrieving process.

We use The New York Times session data that's explained in section 6.1.3 for training models in query generation, user click generation, abandonment prediction and dwell time estimation.

### 8.1.2 Query Generation

In Chapter 7, we explain several query generation models that assume users reformulate their queries based on the terms that they encounter during search process, terms that exist in their predefined information need or terms that exist in suggested queries that are retrieved from a commercial search engine. Two models that have

different smoothing sources are selected for query generation in order to create variability in generated queries. The first model we use smooths the maximum likelihood of terms with terms that appear in all snippets in previous interactions with decreasing weights and predefined information need. The closer the interaction to the interaction we generate query, the more weight it has. This method is shown in equation 7.4. The second model we use smooths the maximum likelihood of terms with terms that appear in previous interaction, predefined information need and suggested queries that are collected from a commercial search engine. This method is shown in equation 7.6. We do not select the model that uses clicked documents as smoothing term source because the clicked documents are sparse in the data, this phenomenon happened in many cases. Note that only the maximum likelihood functions are different in these models and query sampling and scoring the sampled query methods are same.

20 queries for each topic are generated for each model and the top 3 queries are selected. In total we generated 24 user search sessions for each model.

### 8.1.3 Click Generation

In Chapter 3, we introduce several machine learned models that predict a single user click on a snippet that is listed on a SERP in several different scenarios. We consider two different main settings: (1) only considering SERP; (2) considering session history, (3) considering session history with sessions of other users along with the SERP. For each of these main settings we consider three different settings (1) only considering a snippet by itself; (2) considering all snippets on a results page along with the snippet that we make click prediction; and (3) considering the clicks in a SERP. As a result we test nine different models derived from combining feature sets from two different groups.

For generating clicks on top 10 documents that are retrieved as a result of search with the generated query on The New York Times annotated corpus index, we use a model that considers all snippets on a result page along with the snippet that we make click prediction. See M4 in Table 3.4. Even though M4 is not the best achieving model

in terms of precision, recall or AUC, it is better than defined baseline and easy to implement comparing to other models.

#### 8.1.4 Dwell Time Generation

Dwell time is the duration that starts with a user click on a result that is listed on a SERP and ends with returning to the results or abandoning the search. Even though dwell time is not used in the our user experiments directly, it is used for calculating the session length. We use a model that bases the time spent on a clicked document on features of the document, features of URLs/titles/ snippets of other ranked results along with reading level features of the document. See model three in section 4.2 for detailed information.

#### 8.1.5 Session Length Generation

For session length generation, we use one of our session abandonment prediction methods that uses session and interaction features which is the best performing feature set in both 2014 TREC Session track data and the New York Times session data. See section 5.2.2 for detailed information. Since we do not simulate session duration and interaction duration, features related to those are not used for training and prediction.

### 8.2 User Experiment

In order to evaluate how generated sessions are perceived by users, we set up a user experiment and collected user judgements on performance of the simulations. In the experiment we present a participant a search session along with a topic description. The participant reads the topic description, follows the provided session, clicks on the snippets that are written in red, examines the clicked document and at the end of the session she is directed to a questionnaire that consist of following questions question; *'Are you satisfied with the provided information in terms of your information need that is given in the topic description?'* by selecting a score between 1-5 such that 1 means no satisfaction and 5 means complete satisfaction, *'Is this search session done by a human or a computer?'* by selecting H for human and C for computer.

After the U.S. surgeon general declared nicotine an addictive substance the federal government took action to regulate sale of cigarettes to young people. Find news stories reporting on the dangers of nicotine and the actions taken by the federal government and tobacco companies to address the problem.

### [U.S. LAWSUIT SEEKS TOBACCO PROFITS](#)

...billion in profits derived from half-century of 'fraudulent' and dangerous marketing practices; suit cites new evidence to charge that major cigarette companies run what amounts to criminal enterprise by manipulating **nicotine** levels, lying to their customers about **dangers of** tobacco and directing their multibillion-dollar advertising...voiced public skepticism in past about strength of federal lawsuit; \$289 billion figure, if endorsed...

### [Women's Brains on Nicotine](#)

...Smoking and Tobacco Women Women's Brains on **Nicotine** Patterns: Women's Brains on **Nicotine** By JOHN O'NEIL O'Neil, John Dr Stephen G Potkin, University of California, Irvine, researcher, leads study on effects of **nicotine** on men and women's brains; study, published in International Journal of Neuropsychopharmacology, shows biological difference in how **nicotine** is metabolized; brain metabolism decreases in women...

### [Nicotine Without the Smoke](#)

...Tobacco Medicine and Health Politics and Government **Nicotine** Without the Smoke Addiction: **Nicotine** Without the Smoke By ERIC NAGOURNEY Nagourney, Eric Dr Walton Sumner II of Washington University School of Medicine in St Louis authors report in Tobacco Control that suggests allowing smokers to use inhalers with pure shots of **nicotine** to limit public health issues; urges more...

### [Ammonia Linked to a Type of Nicotine](#)

...and Tobacco Ammonia Linked to a Type of **Nicotine** By BARRY MEIER Meier, Barry WASHINGTON, July 29 Oregon Graduate Institute researchers say use of ammonia by tobacco companies can dramatically increase level of **nicotine** available to smokers (M) An independent scientific...

### [Nicotine Manipulation Confirmed](#)

...Health Harvard University Food and Drug Administration **Nicotine** Manipulation Confirmed **Nicotine** Manipulation Confirmed Editorial says Harvard researchers have confirmed 2006 Massachusetts Dept of Public Health finding that tobacco companies have surreptitiously raised **nicotine** content of cigarettes by 11 percent; calls on Congress...

### [Nicotine-Laced Water May Soon Be in Stores](#)

...Advertising and Marketing Food and Drug Administration **Nicotine**-Laced Water May Soon Be in Stores **Nicotine**-Laced Water May Soon Be in Stores By SHERRI DAY Day, Sherri Drugstores and convenience stores around country in June could begin carrying Nico Water, bottles of **nicotine**-laced water that are latest entry in long line of products intended to help smokers get their...

### [Pfizer to Give Away Nicotine Patches](#)

...City Pfizer Incorporated Pfizer to Give Away **Nicotine** Patches Pfizer to Give Away **Nicotine** Patches Pfizer Inc donates \$3.7 million worth of **nicotine** patches to New York City for use...

### [Metro Briefing | New York: Manhattan: No More Free Nicotine Patches](#)

...Briefing | New York: Manhattan: No More Free **Nicotine** Patches By Nichole M. Christian (NYT) (Compiled by Daisy Hernı́andez) Christian, Nichole M New York City runs out of free **nicotine** patches for smokers interested in quitting (S...

Figure 8.2: User interface of The New York Times data search engine

The session that is provided to a user starts with an initial query which is not shown but the terms are highlighted on the search results and an interaction that consists of a SERP, and clickable documents. The clickable documents are only the documents that are predicted to be clicked by a selected user click generation algorithm and they are written in red color. Figure 8.2 shows a screenshot of an interaction page for query "*dangers of nicotine*".

In total we provided 48 generated sessions and 48 user sessions that were randomly selected from The New York Times session data to 12 participants who are University of Delaware graduate students. Each participant reviewed 16 sessions and they were paid 10 USD for approximately 1 hour of work.

### 8.3 Results

We collected 192 satisfaction scores and session type predictions for 96 search sessions. We first investigated the users' score agreements on satisfaction rates and session type predictions. Then we investigated whether users can separate generated search sessions from user search sessions and satisfaction rate differences between these two type of search sessions.

Users scored their satisfaction rates in 1 to 5 range, such that 1 meant *no satisfaction* and 5 meant *complete satisfaction*. We mapped these ratings to 1 to 3 range. We mapped 1 to *no satisfaction*, 2 and 3 to *low satisfaction* and 4 and 5 to *high satisfaction*. Table 8.1 summarizes users agreement on satisfaction ratings of sessions. Overall agreement, calculated as the sum of diagonals divided by the total count, is about 52%, which is over than measured human agreement about relevance [11], [?]. Table 8.2 summarizes users agreement on session type predictions which is 53% and is on par with measured session satisfaction rating agreement. Both of these results show us, users more or less agree on their satisfaction rates and predictions on session types.

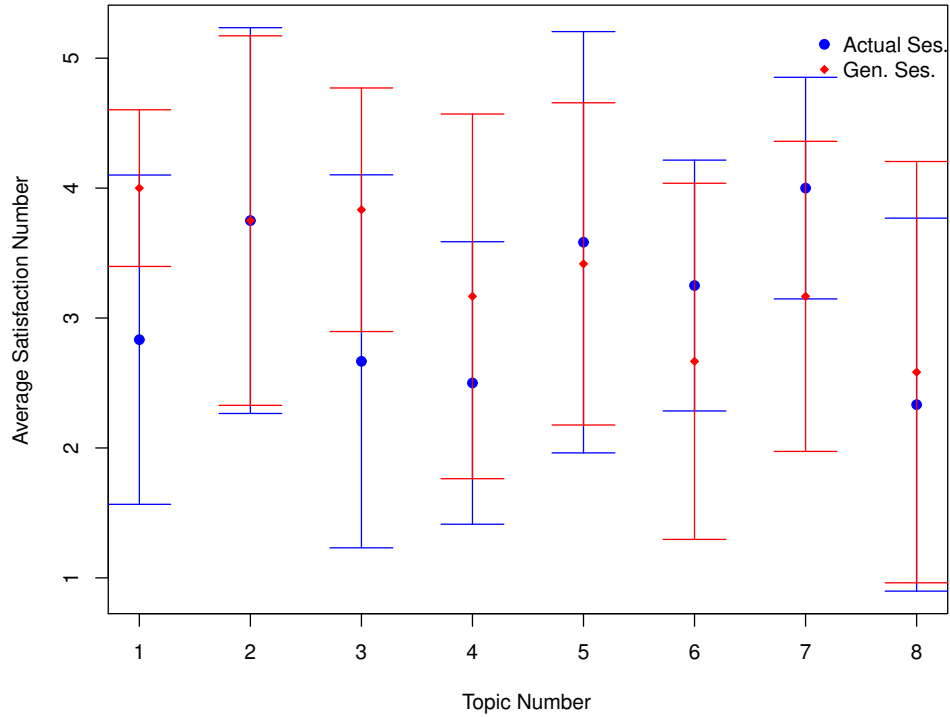
We calculated average satisfaction ratings for each topic for actual user sessions and generated sessions. They are summarized in Figure 8.3. For topics 1, 2, 3, and 4 generated user search sessions have higher or equal satisfaction rates to actual user

**Table 8.1:** Session review satisfaction rating agreement between participants.

P-1 \ P-2	1	Low(2,3)	High(4,5)
	1	6	8
Low(2,3)	4	18	10
High(4,5)	2	20	26

**Table 8.2:** Session type agreement between participants.

P-1 \ P-2	User Session	Generated Session
	User Session	22
Generated Session	26	29



**Figure 8.3:** Average satisfaction rates of actual and generated sessions by topic

search sessions. For topics 5 and 8 satisfaction rates are very close to each other. On average user search sessions' satisfaction rate is 3.11 and generated search sessions' satisfaction rate is 3.32.

A one-way between sessions ANOVA was conducted to compare the effect of session type on satisfaction ratings. Session type was not found significant,  $p = 0.12 > 0.05$ , on session satisfaction ratings. This result shows that session type is not a factor on the user satisfaction independent of the user, topic or query. From this result we can also argue that users gather similar amount of information from provided actual user search sessions and simulated search sessions.

The overall correlation between average satisfaction ratings of actual user search sessions and simulated search sessions by topic is 0.182 by Kendall's  $\tau$  rank correlation, 0.216 by Spearman correlation, and 0.150 by Pearson correlation. Even though these correlation values are small, they show a positive correlation between average satisfaction rates between user search sessions and generated search sessions.

We counted the satisfaction ratings for each topic for actual and simulated sessions. So for each topic we have  $\vec{U}$  and  $\vec{G}$  such that  $U[i]$  is number number of user search sessions that have satisfaction rating of  $i + 1$  and  $G[i]$  is number of generated search sessions that have rating of  $i + 1$ . We combined all  $\vec{U}$ s and  $\vec{G}$ s from all topics and calculated the correlation between them. The values are 0.16 for Kendall's  $\tau$  rank correlation, 0.27 for Pearson correlation and 0.22 for Spearman correlation. These results are similar to average satisfaction rate correlations and show that even though correlation values are small, there is a clear association between these two session types in terms of user search satisfactions.

Lastly, we analyzed how users are successful to separate actual user search sessions from generated search sessions. Table 8.3 shows the confusion matrix of all predictions. The precision, recall and accuracy values are 0.55, 0.59 and 0.56 respectively. These results show that users are barely better than random in predicting session type.

**Table 8.3:** Session type prediction decision table.

Prediction \ Actual	User Session	Generated Session
	User Session	50
Generated Session	46	57

## 8.4 Conclusions

In this chapter we explained artificial user search session generation process when information need is clearly defined and a search session corpus and a document collection are provided. We used machine learned models that simulate different parts of information searching behavior on a search engine such as generating queries, clicking to documents, spending time on documents, and abandoning the search. We generated 48 user search sessions for 8 topics and evaluated generated search sessions with a user experiment. The user experiment sought answers to following questions; (1) "*Can artificially created search sessions satisfy user information needs?*" and (2) "*Can users classify artificial and actual user sessions?*".

For user experiment, we picked 48 actual sessions for 8 topics randomly from the New York Times session corpus and merged these actual search sessions with the generated search sessions in a corpus. We provided these search sessions to users and asked them to review the sessions. In total we collected 192 satisfaction scores and session type predictions. User agreement on satisfaction scores is 52% which is over than measured human agreement about relevance [11] [?].

We performed a one-way ANOVA to compare the effect of session type on session satisfaction ratings. Session type was not found significant,  $p=0.12>0.05$ , on session satisfaction ratings. This result shows that users' satisfaction from search session is not related to session type. We also investigated the correlation between average satisfaction ratings of actual and simulated search sessions. The correlation values we found were small but positive which shows there's some association between these two session types. We also found that users are slightly better than random in predicting

session type.

## Chapter 9

### CONCLUSIONS AND FUTURE WORK

In this work, we conducted several studies aimed at simulating and modeling user actions on search engines in session context. The user actions we target were creating queries, clicking to documents, inspecting clicked documents, abandoning search sessions and finally complete information searching when information need is clearly provided. We demonstrated how to incorporate the context of the full session into predictions, in order to capture the user’s evolving needs as they interact with the system. This lead us towards full simulations of user sessions with the ultimate goal of obtaining a better understanding of how search engines can help users make better use of search engines.

We started this thesis with modeling user click behavior on search sessions when information need is clearly defined. Our focus was to investigate impacts of results listed along with the result we try to predict the user click, users’ session history, clicks done on other results, clicks of other users on the same topic on user click prediction. We created 9 different models and trained and tested our models on TREC 2013 and TREC 2014 Session track data. We found that a dependency model which uses features of all snippets in a results page is a better click model over one that uses only features of a target document, and furthermore that using features of the session history and of other sessions on the same topic uniformly improve precision and AUC.

As a complementary work to user click detection and simulation, we changed our focus on estimating user dwell time on web documents in search session scenarios. Our models estimate the user dwell time on the assumption that the amount that users spend on on documents is based on document features, query-document features and features of the snippets that the document is listed along with. We trained and tested

our regression models on TREC 2013 and TREC 2014 Session track data. We found that none of our models are doing better than median dwell time which is defined as baseline in terms of RMSE. Then we binned the dwell times of documents into three classes. First class consists of dwell time estimation of documents that were judged non-relevant or spam. The second class consists of dwell time estimations of documents that were judged relevant or highly relevant and the third class consists of dwell time estimations of documents that were judged key or navigational document. We performed the same binning process to estimated dwell times as well. We plot the density distribution of actual and predicted dwell times by document class and found that our models captured the actual dwell time density distribution.

The next user action we modeled was session abandonment behavior on web search session scenarios. We performed a comparison of different feature sets for detecting session abandonment. We had two groups. The first group of features were extracted solely from a SERP: title of a snippet, text of a snippet, URL text. The second group of features were extracted from the session and interaction. We developed machine learning models and tested them with low noise: fully-segmented sessions on a single topic, with the topic description available as well as sessions by other users for the same topic. We found that the latter set is far superior to the first set, which actually degrades effectiveness at predicting abandonment.

In order to demonstrate generalizability of our machine learned models we created a new session test corpus that consists of 122 user search sessions and 8 user topics by using a crowdsourcing platform and trained and tested the user click prediction, dwell time estimation and session abandonment prediction methods. The new session corpus is based on The New York Times annotated corpus which consists of news documents in *xml* format and 2017 TREC Dynamic Domain track topics. All of our machine learned models showed similar performance changes in same feature sets over their baselines with the new session data. As a future work we plan to extend the user news search session corpus with more users and more search topics in order to create a viable test collection.

The last user action we modeled on user search sessions is query generation when information need is clearly defined. We used a two-phase process by which we first generated queries by sampling from a language model, then scored them based on their discriminative power among topics. We used TREC 2013 and TREC 2014 Session track data for query generation process. Evaluation of the query generation methods was the hardest part but for this task we cared most about whether the queries we generate were “good” for evaluating retrieval systems, and in particular, whether they were good for evaluating systems that use features derived from session history. To that end, we evaluated simulated queries indirectly by evaluating retrieval systems that take them as input. We concluded that simulated queries are not unacceptable substitute for real user queries for ad hoc evaluation, non-simulated and simulated queries provide similar overall session evaluation results and non-simulated and simulated queries agree on the relative ordering of the systems when we compare the performance of the systems on collections that are created by both of these query types. For future work, we plan to set up a user experiment and ask users to provide feedback on the generated queries for each interaction after reviewing previous interactions, clicked documents and user actions.

Finally, our last effort consisted in generating artificial user search sessions by simulating query reformulations, user click decisions, user dwell times and session abandonments. This phase was important because we put all our machine learned user models together to create artificial search sessions and sought answers to following questions; (1) *"Can artificially created search sessions satisfy user information needs?"* and (2) *"Can users classify artificial and actual user sessions?"* with a user experiment. For user experiment, we picked 48 actual sessions for 8 topics randomly from the New York Times session corpus and merged these actual search sessions with the generated 48 search sessions in a corpus. We provided these search sessions to users and asked them to review the sessions. Specifically the users were asked to follow the search sessions, read the marked clicked documents and provide a satisfaction score for their assigned information need at the end of the session. In total we collected

192 satisfaction scores and session type predictions. Our analysis on the collected data showed that session type is not an important factor on search session satisfaction and users are barely better than random in predicting session type.

## BIBLIOGRAPHY

- [1] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM, 2006.
- [2] Ashraf Bah, Karankumar Sabhnani, Mustafa Zengin, and Ben Carterette. University of delaware at trec 2014. Technical report, DELAWARE UNIV NEWARK DEPT OF COMPUTER AND INFORMATION SCIENCES, 2014.
- [3] Feza Baskaya. Simulating search sessions in interactive information retrieval evaluation. 2014.
- [4] Feza Baskaya, Heikki Keskustalo, and Kalervo Järvelin. Simulating simple and fallible relevance feedback. In *European Conference on Information Retrieval*, pages 593–604. Springer, 2011.
- [5] Feza Baskaya, Heikki Keskustalo, and Kalervo Järvelin. Time drives interaction: simulating sessions in diverse searching environments. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 105–114. ACM, 2012.
- [6] Marcia J Bates. The design of browsing and berrypicking techniques for the online search interface. *Online review*, 13(5):407–424, 1989.
- [7] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *ACM SIGIR Forum*, volume 51, pages 219–226. ACM, 2017.
- [8] Peter Brusilovsky and Carlo Tasso. Preface to special issue on user modeling for web information retrieval. *User Modeling and User-Adapted Interaction*, 14(2):147–157, 2004.
- [9] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning (ICML-05)*, pages 89–96, 2005.
- [10] Georg Buscher, Ludger Van Elst, and Andreas Dengel. Segment-level display time as implicit feedback: a comparison to eye tracking. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 67–74. ACM, 2009.

- [11] Ben Carterette, Paul N Bennett, David Maxwell Chickering, and Susan T Dumais. Here or there. In *European Conference on Information Retrieval*, pages 16–27. Springer, 2008.
- [12] Ben Carterette, Evangelos Kanoulas, Mark Hall, Ashraf Bah, and Paul Clough. Overview of the trec 2013 session track. Technical report, DELAWARE UNIV NEWARK DEPT OF COMPUTER AND INFORMATION SCIENCES, 2014.
- [13] Ben Carterette, Evangelos Kanoulas, Mark Hall, and Paul Clough. Overview of the trec 2014 session track. Technical report, DELAWARE UNIV NEWARK DEPT OF COMPUTER AND INFORMATION SCIENCES, 2014.
- [14] Olivier Chapelle and Yi Chang. Yahoo! learning to rank challenge overview. In *Proceedings of the Learning to Rank Challenge*, pages 1–24, 2011.
- [15] Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pages 1–10. ACM, 2009.
- [16] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [17] A Chuklin, I Markov, and M de Rijke. Click models for web search. synthesis lectures on information concepts, retrieval, and services, 2015.
- [18] Aleksandr Chuklin and Pavel Serdyukov. Good abandonments in factoid queries. In *Proceedings of the 21st International Conference on World Wide Web*, pages 483–484. ACM, 2012.
- [19] Aleksandr Chuklin and Pavel Serdyukov. How query extensions reflect search result abandonments. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1087–1088. ACM, 2012.
- [20] Aleksandr Chuklin and Pavel Serdyukov. Potential good abandonment prediction. In *Proceedings of the 21st International Conference on World Wide Web*, pages 485–486. ACM, 2012.
- [21] Mark Claypool, Phong Le, Makoto Wased, and David Brown. Implicit interest indicators. In *Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40. ACM, 2001.
- [22] Gordon V Cormack, Mark D Smucker, and Charles LA Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information retrieval*, 14(5):441–465, 2011.

- [23] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 international conference on web search and data mining*, pages 87–94. ACM, 2008.
- [24] Abdigani Diriye, Ryen White, Georg Buscher, and Susan Dumais. Leaving so soon?: understanding and predicting web search abandonment rationales. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1025–1034. ACM, 2012.
- [25] Georges Dupret, Vanessa Murdock, and Benjamin Piwowarski. Web search engine evaluation using clickthrough data and a user model. In *WWW2007 workshop Query Log Analysis: Social and Technological Challenges*, volume 2, 2007.
- [26] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168, 2005.
- [27] Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of machine learning research*, 4(Nov):933–969, 2003.
- [28] Zhiwei Guan and Edward Cutrell. An eye tracking study of the effect of target rank on web search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 417–420. ACM, 2007.
- [29] Robert Gunning. *The technique of clear writing*. 1952.
- [30] Fan Guo, Chao Liu, Anitha Kannan, Tom Minka, Michael Taylor, Yi-Min Wang, and Christos Faloutsos. Click chain model in web search. In *Proceedings of the 18th international conference on World wide web*, pages 11–20. ACM, 2009.
- [31] Fan Guo, Chao Liu, and Yi Min Wang. Efficient multiple-click models in web search. In *Proceedings of the second acm international conference on web search and data mining*, pages 124–131. ACM, 2009.
- [32] Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. Beyond clicks: query reformulation as a predictor of search satisfaction. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2019–2028. ACM, 2013.
- [33] Ralf Herbrich. Large margin rank boundaries for ordinal regression. *Advances in large margin classifiers*, pages 115–132, 2000.
- [34] Djoerd Hiemstra and Wessel Kraaij. Twenty-one at trec-7: Ad-hoc and cross-language track. In *TREC*, pages 174–185, 1998.

- [35] Bernard J Jansen, Amanda Spink, and Jan Pedersen. A temporal comparison of altavista web searching. *Journal of the American Society for Information Science and Technology*, 56(6):559–570, 2005.
- [36] Bernard J Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227, 2000.
- [37] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.
- [38] Thorsten Joachims, Laura A Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Sigir*, volume 5, pages 154–161, 2005.
- [39] Evangelos Kanoulas, Ben Carterette, Mark Hall, Paul Clough, and Mark Sanderson. Overview of the trec 2011 session track. 2011.
- [40] Diane Kelly and Nicholas J Belkin. Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevance feedback. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 408–409. ACM, 2001.
- [41] Diane Kelly and Nicholas J Belkin. Display time as implicit feedback: understanding task effects. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 377–384. ACM, 2004.
- [42] Diane Kelly et al. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends® in Information Retrieval*, 3(1–2):1–224, 2009.
- [43] Heikki Keskustalo, Kalervo Järvelin, Ari Pirkola, Tarun Sharma, and Marianne Lykke. Test collection-based ir evaluation needs extension toward sessions—a case of extremely short queries. In *Asia Information Retrieval Symposium*, pages 63–74. Springer, 2009.
- [44] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. Comparing client and server dwell time estimates for click-level satisfaction prediction. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 895–898. ACM, 2014.
- [45] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 193–202. ACM, 2014.

- [46] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. 1975.
- [47] Jane Li, Scott Huffman, and Akihito Tokuda. Good abandonment in mobile and pc internet search. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50. ACM, 2009.
- [48] Chao Liu, Ryen W White, and Susan Dumais. Understanding web browsing behaviors through weibull analysis of dwell time. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 379–386. ACM, 2010.
- [49] Lori Lorigo, Maya Haridasan, Hrönn Brynjarsdóttir, Ling Xia, Thorsten Joachims, Geri Gay, Laura Granka, Fabio Pellacini, and Bing Pan. Eye tracking and online search: Lessons learned and challenges ahead. *Journal of the American Society for Information Science and Technology*, 59(7):1041–1052, 2008.
- [50] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.
- [51] David Maxwell and Leif Azzopardi. Simulating interactive information retrieval: Simiir: A framework for the simulation of interaction. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1141–1144. ACM, 2016.
- [52] David Maxwell, Leif Azzopardi, Kalervo Järvelin, and Heikki Keskustalo. Searching and stopping: An analysis of stopping rules and strategies. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 313–322. ACM, 2015.
- [53] G Harry Mc Laughlin. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646, 1969.
- [54] Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *SIGIR'94*, pages 272–281. Springer, 1994.
- [55] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.
- [56] Jay Michael Ponte and W Bruce Croft. *A language modeling approach to information retrieval*. PhD thesis, University of Massachusetts at Amherst, 1998.
- [57] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, 2010.

- [58] Matthew Richardson, Ewa Dominowska, and Robert Ragno. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*, pages 521–530. ACM, 2007.
- [59] Soo Young Rieh et al. Analysis of multiple query reformulations on the web: The interactive information retrieval context. *Information Processing & Management*, 42(3):751–768, 2006.
- [60] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.
- [61] Pavel Serdyukov, Georges Dupret, and Nick Craswell. Wscd2013: workshop on web search click data 2013. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 787–788. ACM, 2013.
- [62] Pavel Serdyukov, Georges Dupret, and Nick Craswell. Log-based personalization: The 4th web search click data (wscd) workshop. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 685–686. ACM, 2014.
- [63] Si Shen, Botao Hu, Weizhu Chen, and Qiang Yang. Personalized click model through collaborative filtering. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 323–332. ACM, 2012.
- [64] Milad Shokouhi. Learning to personalize query auto-completion. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 103–112. ACM, 2013.
- [65] Fei Song and W Bruce Croft. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321. ACM, 1999.
- [66] Don R Swanson. Information retrieval as a trial-and-error process. *The Library Quarterly*, 47(2):128–148, 1977.
- [67] Liu Tieyan, Qin Tao, Xu Jun, et al. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of the Workshop on Learning to Rank for Information Retrieval*, pages 137–145, 2007.
- [68] Ming-Feng Tsai, Tie-Yan Liu, Tao Qin, Hsin-Hsi Chen, and Wei-Ying Ma. Frank: a ranking method with fidelity loss. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 383–390. ACM, 2007.

- [69] Chong Wang, Achir Kalra, Cristian Borcea, and Yi Chen. Webpage depth-level dwell time prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1937–1940. ACM, 2016.
- [70] Ryen W White and Susan T Dumais. Characterizing and predicting search engine switching behavior. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 87–96. ACM, 2009.
- [71] Ryen W White and Diane Kelly. A study on the effects of personalization and task information on implicit feedback performance. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 297–306. ACM, 2006.
- [72] Ryen W White and Resa A Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis lectures on information concepts, retrieval, and services*, 1(1):1–98, 2009.
- [73] Dietmar Wolfram, Amanda Spink, Bernard J Jansen, Tefko Saracevic, et al. Vox populi: The public searching of the web. *JASIST*, 52(12):1073–1074, 2001.
- [74] Songhua Xu, Hao Jiang, and Francis Chi-Moon Lau. Mining user dwell time for personalized web search re-ranking. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [75] Grace Hui Yang and Ian Soboroff. Trec 2017 dynamic domain track overview. In *TREC*, 2017.
- [76] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. Beyond clicks: dwell time for personalization. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 113–120. ACM, 2014.
- [77] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. Beyond clicks: dwell time for personalization. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 113–120. ACM, 2014.
- [78] Mustafa Zengin and Ben Carterette. User click detection in ideal sessions. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, pages 261–264. ACM, 2017.
- [79] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.