Retrieval Enhancements for Task-Based Web Search

A dissertation presented by
Michael Völske

to the
Faculty of Computer Science and Media
of the
Bauhaus-Universität Weimar

in partial fulfillment of the requirements for the academic degree of
Dr. rer. nat.

Weimar, Germany
January 2019

Advisor:	Prof. Dr. Benno Stein
Reviewer:	Prof. Dr. Pertti Vakkari
Date of oral exam:	June 27, 2019

Abstract

Retrieval Enhancements for Task-Based Web Search

The task-based view of web search implies that retrieval should take the user perspective into account. Going beyond merely retrieving the most relevant result set for the current query, the retrieval system should aim to surface results that are actually useful to the task that motivated the query.

This dissertation explores how retrieval systems can better understand and support their users' tasks from three main angles: First, we study and quantify search engine user behavior during complex writing tasks, and how task success and behavior are associated in such settings. Second, we investigate search engine queries formulated as questions, and explore patterns in a large query log that may help search engines to better support this increasingly prevalent interaction pattern. Third, we propose a novel approach to reranking the search result lists produced by web search engines, taking into account retrieval axioms that formally specify properties of a good ranking.

Abstract (in German)

Retrieval Enhancements for Task-Based Web Search

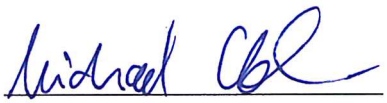
Die Task-basierte Sicht auf Websuche impliziert, dass die Benutzerperspektive berücksichtigt werden sollte. Über das bloße Abrufen der relevantesten Ergebnismenge für die aktuelle Anfrage hinaus, sollten Suchmaschinen Ergebnisse liefern, die tatsächlich für die Aufgabe (Task) nützlich sind, die diese Anfrage motiviert hat.

Diese Dissertation untersucht, wie Retrieval-Systeme die Aufgaben ihrer Benutzer besser verstehen und unterstützen können, und leistet Forschungsbeiträge unter drei Hauptaspekten: Erstens untersuchen und quantifizieren wir das Verhalten von Suchmaschinenbenutzern während komplexer Schreibaufgaben, und wie Aufgabenerfolg und Verhalten in solchen Situationen zusammenhängen. Zweitens untersuchen wir Suchmaschinenanfragen, die als Fragen formuliert sind, und untersuchen ein Suchmaschinenlog mit fast einer Milliarde solcher Anfragen auf Muster, die Suchmaschinen dabei helfen können, diesen zunehmend verbreiteten Anfragentyp besser zu unterstützen. Drittens schlagen wir einen neuen Ansatz vor, um die von Web-Suchmaschinen erstellten Suchergebnislisten neu zu sortieren, wobei Retrieval-Axiome berücksichtigt werden, die die Eigenschaften eines guten Rankings formal beschreiben.

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Teile der Arbeit, die bereits Gegenstand von Prüfungsarbeiten waren, sind ebenfalls unmissverständlich gekennzeichnet.

Weitere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt. Ich versichere, dass ich nach bestem Wissen die reine Wahrheit gesagt und nichts verschwiegen habe.

Weimar, 16. Januar 2019 

Michael Völske

Contents

1 Introduction **1**
- 1.1 Task-based Web Search . . . 2
- 1.2 Context and Retrieval Enhancement . . . 4
- 1.3 Research Questions and Main Contributions . . . 5

2 Background and Related Work **13**
- 2.1 Basics of Web Search Systems . . . 13
- 2.2 Understanding and Supporting Web Search Tasks . . . 20
- 2.3 Context and Relevance Feedback . . . 30
- 2.4 Retrieval Enhancement . . . 32
- 2.5 Axiomatic Ideas in Information Retrieval . . . 35
- 2.6 Summary . . . 40

3 Understanding and Supporting Writing Tasks **41**
- 3.1 The Webis-TRC-12 Dataset . . . 42
- 3.2 Insights into Writing Behavior . . . 52
- 3.3 Insights into Search Behavior . . . 57
- 3.4 Result Usefulness and Retrieval Success . . . 69
- 3.5 Conclusion . . . 78

4 Analysing a Large Question-Query Log **81**
- 4.1 Data Acquisition and Preparation . . . 83
- 4.2 Question Query Classification . . . 91
- 4.3 Experimental Results . . . 95
- 4.4 Conclusion . . . 100

5 Enhancing Result Rankings With Axioms **103**
- 5.1 The Axiomatic Re-Ranking Approach . . . 105
- 5.2 Evaluation on TREC Queries . . . 114
- 5.3 Conclusion . . . 121

6 Conclusion **123**
- 6.1 Main Findings and Implications . . . 123
- 6.2 Open Problems and Future Work . . . 128

References **131**

1 Introduction

When the first modern web search engines appeared in the early nineties, the total number of web sites in existence was in the low hundreds [31]; the web was only a niche application of information retrieval then, which was long established in expert systems and specialized databases like MEDLINE. But over the following decade, web search developed into the number one frontier of information retrieval research.

While retrieval research has brought many advancements to web search, it faces a fundamental limitation in the fact that the queries we type into search interfaces are often vague, speculative, and *short*—the amount of information that can be gleaned from them, about what the person typing those queries wants to find, is limited. The same query can mean different things to different users, or to the same user at different times. As a simple example, consider homonymy: the query `[michael jordan biography]` could be seeking information about the basketball player, the machine learning researcher, the actor, one of the other nine individuals with that name who at the time of this writing have their own English Wikipedia page,[1] or someone else entirely. All of these options could be relevant results for the given query string, but for any one particular occurrence of it, only one or a few will be actually what the searcher wanted.

A plausible, and nowadays common remedy is to seek additional sources of information, beside the query, that may shed light on what information users want. These sources are commonly referred to as context [25], and many different possible sources of context information have been considered. Users' interaction patterns with the retrieval system itself are the most

[1] `https://en.wikipedia.org/wiki/Michael_Jordan_(disambiguation)`

common (cf. Section 2.3), but other sources like location, weather, or time of day have been considered. What all these factors have in common is that they provide information about the *task* behind the user's query, which can then be applied towards enhancing retrieval effectiveness.

This dissertation explores how retrieval systems can better understand and support their users' tasks from three main angles: First, we study and quantify search engine user behavior during complex writing tasks, and how task success and behavior are associated in such settings. Second, we investigate search engine queries formulated as questions, and explore patterns in a large query log that may help search engines to better support this increasingly prevalent interaction pattern. Third, we propose a novel approach to reranking the search result lists produced by web search engines, taking into account so-called retrieval axioms that formally specify properties of a good ranking. Major areas of contribution of this thesis are highlighted in Figure 1.1, alongside the aspects of task-based web search to which they apply.

1.1 Task-based Web Search

All web search is somehow task-based: whenever we enter a query into a search engine, there is some goal behind it, whether it is the pursuit of some simple, isolated fact, or a complex, multi-faceted work task. As such, referring to task-based search is primarily a matter of perspective: our focus is on the user goals behind interacting with the search system, and how those can be better supported. Task-based search thus goes beyond always providing just that set of results that is the best match for the current query, even though in many cases that is still the correct solution.

We broadly classify search tasks into open-ended and closed-ended, and while Section 2.2.1 goes into more depth, the distinction can be summarized thus: closed-ended tasks can be accomplished with one or a few well-defined information items, whereas open-ended tasks require broader exploration of the information space (hence, they are often referred to as exploratory), and evolve along with the user's knowledge level. In this dissertation, Chapter 3 focuses specifically on complex, open-ended tasks; the retrieval experiments in Chapter 5 target rather closed-ended, isolated queries; the question queries studied in Chapter 4 stem from tasks of varying, but generally mid-range complexity (cf. Figure 1.2).

Web search, viewed from the task-based perspective, is a multi-step process initiated by a human interacting with a retrieval system. Figure 1.1

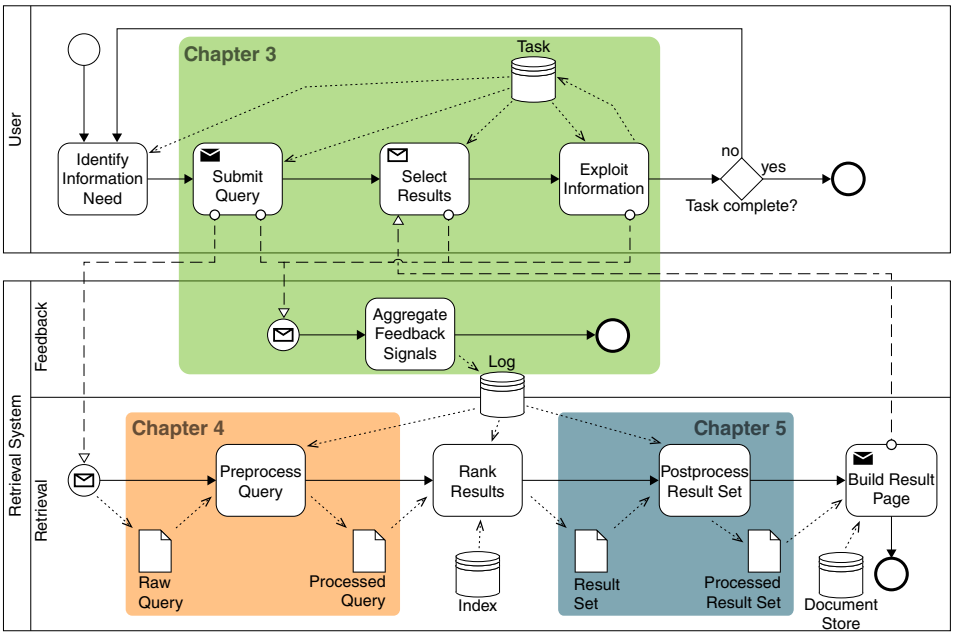


Figure 1.1: An overview of the task-based web search process in BPMN 2.0 notation: User and retrieval system are shown as separate participants to the process; solid arrows represent control flow within subprocesses; dotted arrows show data flow; dashed arrows show message passing between process participants—in this case, the communication pathways between user and system. Areas of investigation of this dissertation are highlighted with the relevant chapter numbers.

shows a diagram illustrating this process: the actions of the retrieval system's *user* are shown in the top swimlane. Informed and motivated by some underlying task, the user undergoes a cycle of (1) identifying some information deficiency with respect to the user's ability to complete her task, (2) formulating a query aimed at rectifying this deficiency, and submitting it to the retrieval system, (3) examining the results returned by the retrieval system and, if they are deemed useful, (4) putting them to use to the task's benefit. This cycle is interrupted by session breaks, or task abandonment (neither of which are shown in the figure for the sake of simplicity), or when the user completes her task.

All user interaction with the retrieval system—query submission, result examination, and information use, by the same user or by others—can potentially be recorded as context information: as shown in the middle swimlane of the diagram, the *feedback* subprocess of the retrieval system aggregates any such context information, such as which queries were submitted, which results were clicked after how much time, and how they were used (e.g., by copy-pasting into a text the user is writing). This aspect of the process is the focus of Chapter 3 of this dissertation, which studies the

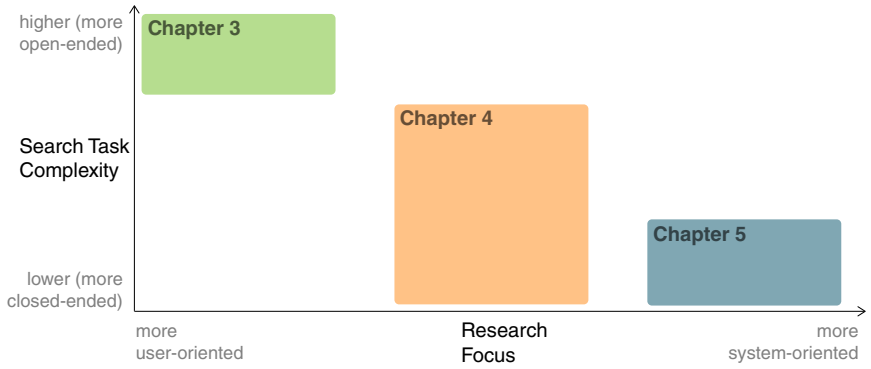


Figure 1.2: The main chapters of this dissertation, organized by whether the retrieval system or its user is the main research focus (horizontal axis), and the complexity of the web search tasks under investigation (vertical axis). Chapter 3 studies high-complexity tasks from a user-oriented perspective; Chapter 5 studies low-complexity tasks from a system-oriented perspective; Chapter 4 occupies a middle ground along both axes, and covers the highest variation in task complexity.

behavior of search engine users engaged in complex writing tasks, and ultimately applies the behavioral information thus derived in a prediction model for the user's success at finding useful search results.

In response to the submitted queries, the retrieval system engages in the *retrieval* subprocess shown in the bottom swimlane in Figure 1.1, wherein (1) the received query is first preprocessed, the processed query then (2) evaluated against an inverted index, and the thus collected set of result documents is then (3) postprocessed to be ultimately assembled into a search result page (SERP) that is returned to the user.

1.2 Context and Retrieval Enhancement

All three of these steps can make use of the context information collected during the aforementioned feedback subprocess, but in this dissertation, we focus on two pathways for retrieval enhancement: query preprocessing (Chapter 4), and result set postprocessing (Chapter 5).

Query preprocessing consists of enriching the query with contextual information before submission to the search index, and can encompass techniques such as query expansion, spelling correction, or categorization (cf. Section 2.4.1). More fundamental query preprocessing techniques like tokenization and normalization, while certainly important for the basic operation of the retrieval system, are taken for granted here—our focus is on those techniques that incorporate context information for retrieval enhancement.

Result set postprocessing techniques aim to enhance the quality of result lists, after they have been retrieved from the inverted index, modifying

the order of search results in the ranking in such a way that some quality measure of the result ranking is optimized (cf. Section 2.4.2). A common approach is learning-to-rank, where a machine learning model exploits a richer feature set on a small top-k result set, than would be possible over the entire collection at the time of initial retrieval.

The complexity of machine learned ranking models can pose a problem in itself: while a ranking model optimized against large amounts of aggregated feedback data can be highly effective, at a certain point it becomes impossible to comprehend how a given ranking came about. Information retrieval researchers have discovered axioms that formally describe the desirable properties of a good result ranking. While these axiomatic ideas promise to make postprocessing more comprehensible, they have not been directly implemented in a retrieval system so far.

1.3 Research Questions and Main Contributions

Figure 1.2 illustrates how this dissertation investigates web search tasks, context, and retrieval enhancement from various angles, and with various degrees of complexity in the web search tasks studied. Each of the main chapters of this dissertation is based on one or more peer-reviewed publications, as summarized in Table 1.1: the findings of four publications form the basis for Chapter 3, while Chapters 4 and 5 are based on one peer-reviewed publication each. Since Chapter 2 introduces common foundations and related work for the three subsequent chapters, all six of the aforementioned publications contribute to Chapter 2. Further publications by the author related to the fields of information retrieval and natural language processing, five of which appear in the bottom part of the table, were not reused in this dissertation in order to maintain a tight topical focus. The first four of these are distantly related to the "Build Result Page" step in Figure 1.1: the first two [77, 199] concern the automatic creation of taxonomies for document collections based on information retrieval principles, which can be applied to present search result sets in a way that is more amenable to exploration. The next two [46, 201] concern automatic summarization, where related approaches have also recently been considered for the creation of search result pages (e.g., by Chen et al. [45]). The final entry [5] does not directly concern the task-based retrieval process, but applies information retrieval principles to the study of text reuse phenomena in large scale datasets.

The following three sections elaborate the contributions of the dissertation at hand, as well as the research questions behind them.

Table 1.1: A selection of peer-reviewed publications by the author and their usage within this dissertation.

Used in	Venue	Type	Pages	Year	Publisher	Ref.
Chap. 3	ACL	conference	10	2013	ACL	[157]
	*Martin Potthast, Matthias Hagen, **Michael Völske**, and Benno Stein. Crowdsourcing Interaction Logs to Understand Text Reuse from the Web.*					
Chap. 3	EuroHCIR	workshop	4	2013	CEUR-WS	[156]
	*Martin Potthast, Matthias Hagen, **Michael Völske**, and Benno Stein. Exploratory Search Missions for TREC Topics.*					
Chap. 3	CHIIR	conference	10	2016	ACM	[81]
	*Matthias Hagen, Martin Potthast, **Michael Völske**, Jakob Gomoll, and Benno Stein. How Writers Search: Analyzing the Search and Writing Logs of Nonfictional Essays.*					
Chap. 3	TPDL	conference	12	2018	Springer	[194]
	*Pertti Vakkari, **Michael Völske**, Matthias Hagen, Martin Potthast, and Benno Stein. Predicting Retrieval Success Based on Information Use for Writing Tasks.*					
Chap. 4	CIKM	conference	10	2015	ACM	[200]
	***Michael Völske**, Pavel Braslavski, Matthias Hagen, Galina Lezina, and Benno Stein. What Users Ask a Search Engine: Analyzing One Billion Russian Question Queries.*					
Chap. 5	CIKM	conference	10	2016	ACM	[82]
	*Matthias Hagen, **Michael Völske**, Steve Göring, and Benno Stein. Axiomatic Result Re-Ranking.*					
–	JCDL	conference	9	2014	ACM/IEEE	[77]
	*Tim Gollub, **Michael Völske**, Matthias Hagen, and Benno Stein. Dynamic Taxonomy Composition via Keyqueries.*					
–	WOSP	workshop	4	2014	CNRI	[199]
	***Michael Völske**, Tim Gollub, Matthias Hagen, and Benno Stein. A Keyquery-Based Classification System for CORE.*					
–	NewSum	workshop	4	2017	ACL	[201]
	***Michael Völske**, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to Learn Automatic Summarization.*					
–	INLG	conference	3	2018	ACL	[46]
	*Shahbaz Syed, **Michael Völske**, Martin Potthast, Nedim Lipka, Benno Stein, and Hinrich Schütze. Task Proposal: The TL;DR Challenge.*					
–	ECIR	conference	6	2019	Springer	[5]
	*Milad Alshomary, **Michael Völske**, Tristan Licht, Henning Wachsmuth, Benno Stein, Matthias Hagen, and Martin Potthas. Wikipedia Text Reuse: Within and Without.*					

1.3.1 Understanding and Supporting Writing Tasks (Chapter 3)

Chapter 3 focuses on the collection of contextual information for a specific class of complex task: we prepare a crowdsourced dataset of 150 long essays written with the support of a search engine, covering an unprecedented level of richness in the context information that such task-oriented search interaction data makes available.

We consider the following retrieval scenario: the user is engaged in a complex writing task while referring to documents from some collection as sources, and reusing content from them; a retrieval system indexing the aforementioned collection is used to retrieve those sources. Chapter 3 reports on a crowdsourcing study with twelve participants that implements this setting, and first aims at better understanding the behavior of writers in such scenarios, by way of Research Questions 1a and 1b.

Research Question 1a (Writing Strategies)**.** *Is it possible to identify distinct writing strategies among users engaged in complex writing tasks that involve the retrieval of sources with a search engine, and potential text reuse from those sources?*

Research Question 1b (Searching Strategies)**.** *In the same setting, can distinct searching strategies be identified?*

Answering these research questions requires a novel kind of integrated search and writing environment for the participants of our study to work in. This environment comprises custom a search engine indexing the ClueWeb09 (a large, static web crawl which aims to constitute a representative sample of the web) which records all queries submitted, search result pages visited, and any links followed from those pages. Simultaneously, participants write their essays in a web-based text editor which records a fine-grained history of the text's development over time. Together, these components enable deeper insights into the search and writing process than have been possible previously.

Our results indicate that study participants follow two diametrically opposed strategies in both cases—the *build-up* and *boil-down* writing strategies, as well as the *querier* and *clicker* searching strategies. Build-up and boil-down writing are differentiated mainly by the way the writers engage with the sources they retrieve while writing: the former add material in an incremental fashion, retrieving only the next source needed at any given time, whereas the latter amass a large collection of material up front, which is then gradually organized and rewritten to produce the final essay. Queriers and clickers explore the information space either by submitting many varied queries or by clicking many search results and following long click trails.

While writing tasks have been used frequently to study task-oriented information retrieval, there is a clear gap in the literature when it comes to objectively measuring the usefulness of search results for a given task. As discussed more extensively in Section 2.2.5, usefulness is typically assessed by way of study participants' subjective usefulness ratings. For the retrieval scenario particular to our study, Research Question 2 aims to objectively quantify the usefulness of search results.

Research Question 2 (Measuring Usefulness)**.** *How can the usefulness of search results to writers be observed and measured?*

In our study, we had writers reuse text to produce their essays, hence we can observe copy-paste events as a direct indicator of search result usefulness—put simply, a search result is useful if it is used. We investigate two concrete measures derived from this notion: one based on the amount of copied text, and the other on the number of times text is copied.

In a follow-up study, we investigate the relationship between these notions of usefulness, and other user behavior during searching and writing, with the aim of predicting how successful a user is, in aggregate, at finding useful sources while working on an essay—such a prediction model can help a retrieval system distinguish successful and struggling users, and better support the latter group, in particular [146]. Research Question 3 aims at predicting retrieval success for the writing task in our study.

Research Question 3 (Predicting Retrieval Success)**.** *Which user behaviors during searching and writing predict success at finding useful sources?*

We develop a set of regression models for the relationship between user behavior and retrieval success. Our results indicate that among the writers in our study, a higher degree of success at finding useful results is associated primarily with (1) fewer queries and more clicks—writers appear to be more successful at finding useful results when following the aforementioned "clicker" search strategy; (2) less editing of the essay text—writers who find more useful sources appear to work less hard integrating those sources into the essay; (3) shorter dwell times—contrary to other search tasks, where long dwell times have often (some of the studies are discussed in Section 2.2.5) been found to positively associate with usefulness or relevance of results. Some particularities of our setting, such as the higher task complexity, lack of a time limit, and the fact that writers were encouraged to reuse text, likely explain this difference.

1.3.2 Analysing a Large Question-Query Log (Chapter 4)

Chapter 4 presents an analysis of a large search engine log—comprising a year's worth, or nearly one billion, Russian-language web search queries—focused on the problem of query pre-processing for question-like queries in particular. Questions are a natural form of expressing information needs—people ask questions when they seek information, help, or advice. In the first two decades of web search, search engines have taught users the "telegram style" of keyword search queries such as `[lose weight]`, but recently—especially with the advent of voice-based search interfaces—the share of natural language questions, for example `[how much exercise should i do to lose 10 pounds]`, in search query logs is increasing [150].

As illustrated in Figure 1.2 on page 4, the complexity of the user tasks behind the question queries in our study varies: while generally lower than in the essay writing tasks that we study in Chapter 3, there is a range in task complexity from simple fact-finding (e.g., `[how high is mount everest]`) to seeking multi-step task instruction (e.g., `[how to make poached eggs]`).

Their increasing prevalence notwithstanding, question queries are longer and rarer than typical search engine queries. Web search engines tend to return results of poorer quality for such long-tail queries, prompting us to study question query preprocessing via topical categorization. Such an approach can improve retrieval performance in various ways—for instance, through query disambiguation or routing to an appropriate vertical search index. However, compared to the documents typically considered in text classification tasks, queries are particularly challenging due to their short length—and still this applies to the slightly longer question queries. As such, any feature representation that can be obtained from search queries is inherently sparse. Common techniques for addressing the sparseness problem from general query classification—such as using the features from the result documents clicked by users—are not available in this setting due to the relative rarity of question queries. Hence, we state Research Question 4 to address the novel problem of question query classification.

Research Question 4 (Question Query Classification)**.** *How can we design an effective classification pipeline for question queries in the face of the dual problem of their sparseness, and their relative rarity in query logs?*

Our proposed approach learns to categorize question queries from Community Question Answering (CQA) data: on CQA sites, users post questions, and label them with categories selected by the author (the author's success at getting an answer can be taken as an indicator of label accuracy).

In a multi-step pipeline that starts with data cleaning to drastically reduce the noise level that's typical for large query logs, we learn to first classify CQA questions by topic, then transfer the learned classifier to the question query log. This transfer learning approach enables the discovery of patterns in the topic distribution of the queries in the log. Research question 5 then targets the study of these patterns, and how they can be exploited to further improve query preprocessing.

Research Question 5 (Question Query Patterns)**.** *Given a basically functioning question query classification pipeline, how can we apply it to discover patterns in the question stream—especially those that bootstrap future question query disambiguation efforts?*

Our experimental study of the year-long question query log yields some interesting insights on question asking behavior across topical categories in a non-English search engine. For instance, the distribution of question query volume by category shows changes over time, such as queries in the "education" category becoming less prevalent in the summer months, while travel-related queries reach their peak.

Such temporal patterns can ultimately serve as context information by themselves: the distribution of topics over time can induce a prior for future query classification schemes, and thus improve the accuracy of classification methods targeting query disambiguation. This idea in itself is not novel—for instance, the current date has been suggested as a contextual ranking signal [101]—but shows promise in its application to question queries in particular.

1.3.3 Enhancing Result Rankings With Axioms (Chapter 5)

Chapter 5 focuses on retrieval enhancement via result set postprocessing, but departs from the contextual retrieval setting of the previous chapters. Rather, it adopts the viewpoint of axiomatic information retrieval, which has long investigated, and formally described, the fundamental properties (referred to as *axioms* in this context) that characterize a good result ranking (refer to Chapter 2, in particular Sections 2.1 and 2.5, for a more thorough introduction). While the axiomatic analysis of information retrieval models has yielded actionable insights on occasion, it has in large part been confined to a more theoretical realm. Inasmuch as this is discernible, commercial search engines do not seem to follow the axiomatic approach, opting for highly complex, empirically optimized ranking models instead. This observation leads to Research Question 6, and an investigation

of whether axiomatic ideas can benefit practical retrieval more directly than they have in the past—it should be possible to improve the performance of existing retrieval models with the help axiomatic ideas, because otherwise the relevance of these ideas must be called into question.

Research Question 6 (Axiomatic Result Reranking)**.** *Is it possible—and how—to seamlessly integrate axioms for ranking preferences into the retrieval process, in order to improve the results of a basis retrieval model?*

In response, we develop a new technique for incorporating axiomatic statements about desirable properties of result rankings directly into the retrieval process, so that rankings can be modified to better satisfy arbitrary axioms. Due to the complexity of the optimization problem involved, we evaluate this system in a comparably rather simple Cranfield-style test environment (cf. Figure 1.2), with a notably lower search task complexity compared to previous chapters.

Our approach proposes a triplet formulation for retrieval axioms, composed of precondition, filter, and conclusion. The precondition checks requirements of whether an axiom can be applied to a given document pair; then, based on filter condition evaluated on properties of the documents under consideration, the conclusion yields a ranking preference. We show that many axioms from the literature can be restated in this triplet form, which allows for a practical implementation.

Based on this foundation, we propose the axiomatic result reranking pipeline. The pipeline begins with top-k retrieval by some basis retrieval model, and then determines each considered axiom's ranking preference matrix over the top-k result set. A machine-learned preference aggregation function derives a combined preference matrix, and a rank aggregation algorithm (we propose to use KwikSort) resolves any conflicts and reranks the top-k set based on the aggregated axioms' preferences.

The preference aggregation function is learned from a set of queries with known relevance judgments, with respect to a particular basis retrieval model. Our study includes a comprehensive empirical evaluation over a set of 16 different basis retrieval models, and 23 different axioms. Among the latter, we propose eight new axioms, where most of them cover aspects of term proximity weighting in the result documents. Through our experiments, we show that axioms can in fact significantly improve the performance of many standard retrieval models.

1.3.4 Thesis Structure

The remainder of this dissertation is organized as follows: Chapter 2 reviews the background and previous work necessary for the subject matter of subsequent chapters; as such, it covers the entirety of the task-based retrieval process depicted in Figure 1.1 on page 3, starting from the basics of retrieval systems and indexing, and then presenting supporting material roughly along the same trajectory as the topical focus of the later chapters as shown in Figure 1.2. Chapter 3 then investigates user behavior during complex writing tasks, and answers Research Questions 1 through 3. The main contribution of Chapter 4 is the analysis of a large query log, focused on the understanding and preprocessing of question-like queries, and the investigation of Research Questions 4 and 5. Chapter 5 contributes a result set postprocessing and re-ranking approach, which targets the integration of information retrieval axioms into the search process and addresses Research Question 6. Finally, Chapter 6 concludes the dissertation, reviews the main findings, and discusses open problems and limitations of the research.

2

Background and Related Work

The following sections will introduce the background and related research that subsequent chapters build upon. As has been shown in Figure 1.1, the contributions of this thesis lie on both the system side and the user side of the task-based information retrieval process. Consequently, Section 2.1 begins by introducing basic concepts of information retrieval systems that are important to this and subsequent chapters; this is followed by a more user-focused overview of the literature on task-based web search in Section 2.2, including the two fundamental kinds of search tasks, and some of the instruments for understanding human information seeking behavior and judging its success, in particular with respect to the usefulness of the retrieved search results. Section 2.2.3 briefly reviews the literature on question information needs, and the corresponding tasks, which are the focus of Chapter 4. Section 2.3 reviews the use of context information to improve search results, and how such context can be inferred from user behavior; Section 2.4 discusses how the gained insights can be integrated into the retrieval process. Section 2.5 reviews the axiomatic ideas that form the foundation for the technique of improving search result rankings discussed in Chapter 5.

2.1 Basics of Web Search Systems

The web search engine as we know it today, with a search box that can be used to query an index of pages kept up-to-date by a web crawler, has only been around for a little more than three decades [31], but the problem of storing and accessing information effectively is as old as civilization it-

self. The idea of using computers to automatically access large amounts of stored information is often attributed (e.g. by Singhal [174]) to Vannevar Bush's article "As We May Think" [34] published in 1945. The SMART system—developed by Gerard Salton's group in the 1960s [171]—is considered by many to be the first practical implementation, and introduced many of the concepts outlined below [131]. A number of comprehensive reference works have emerged over the years—among them Baeza-Yates and Ribeiro-Neto [12], Croft et al. [54], and Manning et al. [131]. This section—while nowhere near their level of sophistication or breadth—is informed by the aforementioned works, and briefly reviews a few basics relevant to the remainder of this chapter and to the dissertation at large.

2.1.1 Document Representations, Indexing, and Ranking

Figure 2.1 recalls the task-based information retrieval process already shown in Figure 1.1 on page 3, but adds additional detail regarding the retrieval system: compared to the previous figure, the retrieval system's indexing process is appended at the bottom. The goal of indexing is that, at retrieval time, a ranked list of documents relevant to the query can be produced as efficiently as possible; to this end, an *inverted index* is created, which allows the rapid evaluation of a document *scoring function* that, at querying time, assigns each document a query-dependent score which determines the position in the result list, and whether the document is included at all. While the indexing process is itself not the focus of the contributions made in this thesis, it precedes querying and retrieval in both causal and temporal order, in that it creates the necessary data structures for efficiently determining a result ranking given a query. On a rather high level of abstraction, Croft et al. [54] summarize the steps of the indexing process as *Text acquisition*, *Text transformation*, and *Index creation*.

The first step involves identifying the documents to be indexed, and making them available to the rest of the process in plaintext, in a unified encoding, and along with metadata describing, e.g., the document type, and its language, title, or length. The documents themselves may either be part of an existing collection that is already present, or they may themselves be acquired via a crawling process. The resulting document store is consulted during the retrieval process, e.g., when assembling the short text snippets on search result pages. Under particular circumstances, the pages visited by search engine users may be served directly from the document store, as well—this was done, for instance, in the crowdsourcing experiments described in Chapter 3 for the sake of reproducibility and control in the study.

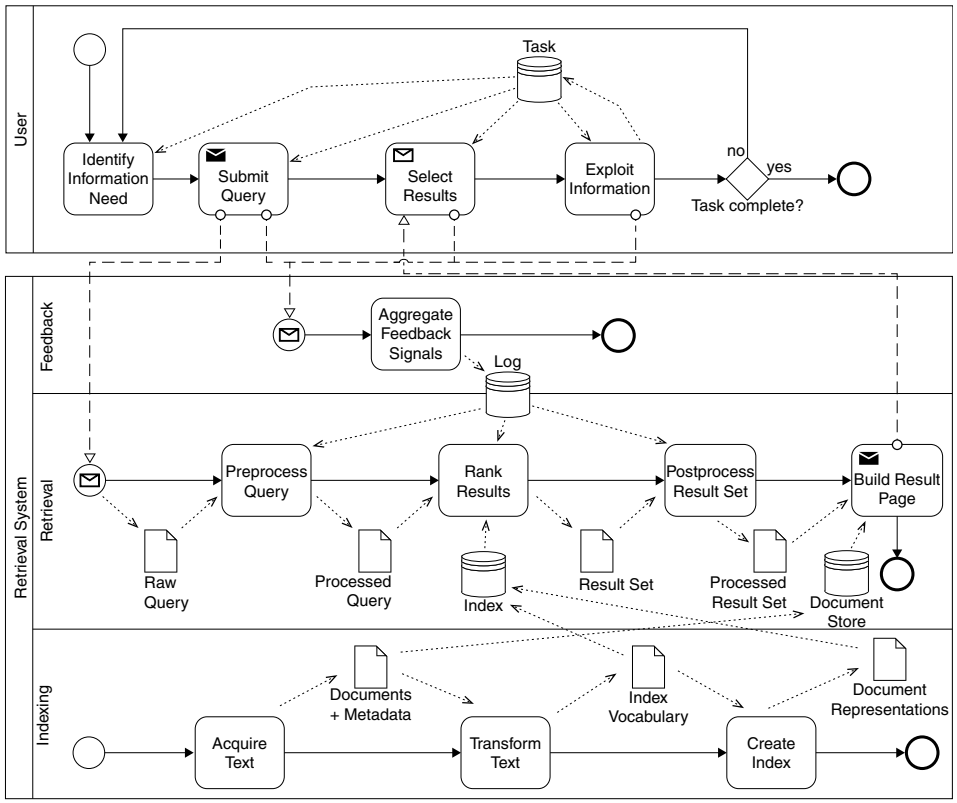


Figure 2.1: User and system in the task-oriented information retrieval process. In addition to Figure 1.1 on page 3, the bottom-most "Retrieval System" swimlane shows the basic steps of the indexing process according to Croft et al. [54].

The second step, text transformation, maps the input texts to *index terms* through tokenization and normalization: the former consists of splitting the text into units (which can be individual words, but also different units of the text like multi-word phrases), and the latter involves, e.g., stop-word removal and stemming. The resulting index vocabulary is one of two key ingredients for the inverted index.

The final index creation step derives a term-document matrix from the (appropriately transformed) input documents, and from that, in turn the inverted index. Each column of the term-document matrix corresponds to the *feature vector* of one of the input documents; each row corresponds to one index term's frequency of occurrence across all documents in the collection, often weighted in some way. For the sake of efficiency, the term document matrix is not stored as a dense array, but in the inverted index, which can be conceived of as essentially a hash table that maps each index term to a *posting list* of documents containing that term. Along with identifiers of included documents, the posting list contains the corresponding

element of the document-term matrix for that particular pair of index term and document (if it is nonzero).

As noted, the inverted index enables the efficient evaluation of the scoring function at retrieval time. How exactly this is done depends on the *retrieval model* used; exactly defined the term refers to the scoring function plus the set of theoretical assumptions on which it is based [54]. However, in practice, the terms retrieval model and scoring function are often used interchangeably, and for the remainder of this dissertation there won't be much need to distinguish between them. Important to note, however, is the fact that a rich diversity of retrieval models (and associated scoring functions) have been proposed: for instance, the Boolean model only makes a binary distinction between relevance and non-relevance of documents to the query; the vector space model is based on the idea of ranking documents by increasing distance (or decreasing similarity) of their vector space representations from that of the query; the bustling class of probabilistic models—named for the underlying probability ranking principle [131, 166]—aims to approximate the probability that a document is relevant to the current query; query likelihood models rank documents by the probability that the query would be generated by a language model of the document [54].

At querying time, most retrieval models evaluate a sum over all query terms for each document found in those terms' posting lists. This can be illustrated with the following extremely basic scoring function, that simply ranks documents based on how often they contain the query terms:

$$\mathrm{TF}(q, d) = \sum_{i=1}^{n} tf(q_i, d)$$

In this, n refers to the number of terms in the query q, and $tf(t, d)$ returns the number of times a term t occurs in document d—i.e., the term frequency. As it turns out, TF is not very effective at ranking, because not all terms in an index vocabulary are equally important to matching the query; in general, the more documents of a collection contain a term t—i.e., the higher its *document frequency*—the less specific t is, and the less useful for distinguishing documents that are relevant to a given query from those that are not. Based on this reasoning, Spärck Jones [178] proposed the *inverse document frequency* (idf) term weighting scheme in 1972, which in its most basic formulation is computed as the logarithm of the inverse fraction of the number of documents in the collection that contain a given term t:

$$idf(t) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

A basic idf weighted scoring function could be operationalized as follows:

$$\text{TF-IDF}(q, d) = \sum_{i=1}^{n} idf(q_i) \cdot tf(q_i, d)$$

Both TF and TF-IDF can be derived from the aforementioned vector space model based on the cosine similarity between the (in the latter case, idf weighted) term frequency vectors of the query and the document. By contrast, the BM25 scoring function, designed by Robertson and Walker [165], implements a probabilistic retrieval model, i.e., it is intended to score documents based on how likely they are to be relevant to the query. The BM25 scoring function still has some practical importance today, and has had a lot of impact on how ranking is done in commercial search engines [54]. Compared to TF-IDF, the most important addition is a document length normalization term for the tf component—this aims to correct for the higher chance that longer documents have to contain a query term by random chance alone. One common formulation of BM25 is given as follows [131]:

$$\text{BM25}(q, d) = \sum_{i=1}^{n} idf(q_i) \cdot \frac{tf(q_i, d) \cdot (k_1 + 1)}{tf(q_i, d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{avgdl}\right)}$$

Here, $|d|$ is the length of the document to be scored, and $avgdl$ is the average document length across the entire collection. The constants k_1 and b are free tuning parameters: k_1 controls the influence of term frequency: at the extreme of $k_1 = 0$, term frequency is ignored, and only (idf weighted) binary term occurrence is considered, higher values increase the influence of term frequency over specificity; the parameter b controls how much the term frequency component is scaled by document length. The empirically determined values $1.2 \leq k_1 \leq 2$ and $b = 0.75$ are often used in practice [131].

Note that the choice of information stored at indexing time controls what retrieval models can be evaluated at querying time; multiple document representations in one index are possible, and often necessary, as are supporting data structures for additional term and collection statistics: for instance, when using only the TF-IDF scoring function, it's possible to store precomputed $tf \cdot idf$ values directly in the index postings. On the other hand, to support also BM25, the postings must contain tf values, in addition to separate data structures storing the idf values of the index terms, as well as the $|d|$ and $avgdl$ values of the documents. The latter index variant can also be used for TF-IDF with only slightly more computation at querying time.

Like BM25, most retrieval functions require some degree of experimentation to identify useful parameter settings, as does the question of how dif-

ferent retrieval models compare to each other. Consequently, a framework for the efficient evaluation of retrieval systems had to be developed.

2.1.2 Retrieval Evaluation

As the previous section has shown, there is a rather large parameter space when it comes to the question of how exactly the documents returned for a query should be ranked. One way to test the goodness of retrieval functions is to have humans use a retrieval system for their daily tasks, and rate the quality of the results. However, this is not practical for at least two reasons: it would be prohibitively costly in terms of time and resources, and the evaluation results for different systems would likely not be comparable, since user's real-world information needs tend to vary. While many kinds of interesting experiments are in fact not possible without a human in the loop (see Chapter 3), the large parameter spaces of retrieval functions can be explored in a productive way with the help of a standardized, automated evaluation procedure.

Such a standard evaluation setting first emerged from the Cranfield experiments in the 1960s [164], and has been firmly established by the Text REtrieval Conference (TREC). It comprises a fixed document collection, a set of pre-defined topics or queries, and a set of relevance judgments (either binary or graded) for a subset of the possible query-document pairs, done by expert assessors. A retrieval system under consideration returns a ranked list of collection documents for each query; based on the relevance judgments, the quality of the ranked list is then computed in terms of an evaluation measure. To keep the relevance judgment effort manageable, typically only documents returned near the top of the ranking by some retrieval system are judged by TREC assessors at all, but those judgments can be reused to benefit future experiments. While the initial creation of the document collection, topics and (especially) relevance judgments is costly, retrieval systems can be quickly re-evaluated without much further human intervention once this infrastructure is in place. Since the early 1990s, the evaluation effort at TREC has yielded several large-scale test environments for this purpose [10].

Given the test environment with collection, queries, and relevance judgments, a wide variety of different evaluation measures have been proposed to actually measure the quality of rankings (cf. Manning et al. [131]). For standard ranked retrieval with a single query evaluated in isolation, the normalized discounted cumulative gain (nDCG) has been widely adopted, and shall serve as an example here. Given a ranked result list, the discounted

cumulative gain at rank k is computed as follows:

$$\mathrm{DCG}_k = \sum_{i=1}^{k} \frac{2^{\mathrm{rel}_i} - 1}{\log_2(i+1)}$$

According to the parameter k, the top k ranks of the ranking under scrutiny are evaluated. Here, $\mathrm{rel_i}$—the relevance judgment for the ith element of the ranked result list—is a numerical value, with higher values representing higher degrees of relevance to the query. Different rating scales have been used, including a binary rating (1 for relevant, 0 for not relevant), and a six-point Likert scale distinguishing spam (-2 and -1, not relevant to any reasonable query), not relevant (0, possibly relevant for a different query), relevant (1–2) and key (3) documents. *Discounted* by the denominator term, which diminishes the contribution of later ranks, these values are summed over the result list up to a maximal considered rank k. In order to allow comparison and aggregation across multiple queries (possibly with different numbers of relevant results), the nDCG_k is obtained as the ratio of this DCG_k to the (ideal) iDCG_k—i.e., the DCG_k of the same set of documents sorted by descending relevance.

Many different document collections with steadily increasing sizes have been used in evaluation efforts over the years; due to their particular relevance to Chapters 3 and 5, we specifically highlight the ClueWeb09 and ClueWeb12 web crawls, which have been used in the TREC web tracks 2009–12, and 2013–14, respectively. ClueWeb09 consists of 1 billion web pages in ten different languages, and ClueWeb12 of 730 million English pages; as such, both provide a realistic, web-scale setting for retrieval experiments.

Having covered the basic background on the retrieval systems side, the following sections will move on to the user side of the information retrieval process, before gradually returning more to systems, and eventually retrieval theory, in Sections 2.4 and 2.5. The remainder of this chapter aspires to tie together a wide range of diverse research reaching back several decades. In support, Figure 2.2 presents a road map of related work since 1990, and places this dissertation's own contributions in context. The vertical axis is arranged by date of publication, with the most recent placed at the top. Conversely, the horizontal axis orders works by their user- or system-oriented focus: works placed to the left of the y-axis make theoretical contributions to understanding user goals and tasks, or more practical observations in concrete experiments; to the right, the figure highlights works focused on retrieval systems, and includes theoretical works on re-

trieval models and axioms, as well as proposals of concrete retrieval system components. An attempt is made to classify each included work in terms of its primary contribution to research, but it should be noted that this (as well as the exact placement along the x-axis) is often subjective.

2.2 Understanding and Supporting Web Search Tasks

Human information needs—and the processes by which they are expressed to retrieval systems—have been the subject of information science research for at least half a century. Inquiries by Taylor [185] and contemporaries brought the concept of the information need—as distinct from and causally preceding the query that is entered into an information system—into focus as early as the 1960s. Attempts to model and formally describe the process of human information seeking abound in the information science literature: for instance, Huurdeman and Kamps [93] mention Ellis's [67] behavioral model of information seeking—formulated in 1989—as an early example. Vakkari's [190] model of the task based information retrieval process is notable in this context, for its particular relevance to the present work, and is itself an extension of Kuhlthau's [111] earlier user-focused model of the search process. Vakkari points out the importance of finding a *focus* in the task-based search process, and thus distinguishes three phases in task based information retrieval—namely the pre-focus, focus-formulation, and post-focus stages, which are characterised by increasingly directed and specific information needs (cf. also the sense-making model [61]).

Recent work by Moshfeghi and Pollick [141] has attempted to identify physical evidence of neural state transitions during information seeking using functional Magnetic Resonance Imaging (fMRI) by directly observing the brain activity of persons engaged in information seeking tasks; since current fMRI scanners impose rather strict limitations on the hardware that can be used within (and thus on the possible retrieval tasks that can be studied), this work is still in an early stage.

The larger tasks motivating searchers' information needs were hardly studied systematically until the 21st century [189], and tasks in the context of information retrieval have since been interpreted at multiple levels of granularity: a person's work function may give rise to several distinct *work tasks*, which can be recursively composed of subtasks in turn, and can be specified either in terms of constraints, or instructions [197]; those work tasks that require information not currently at hand may effect *search tasks* that involve a retrieval system [36].

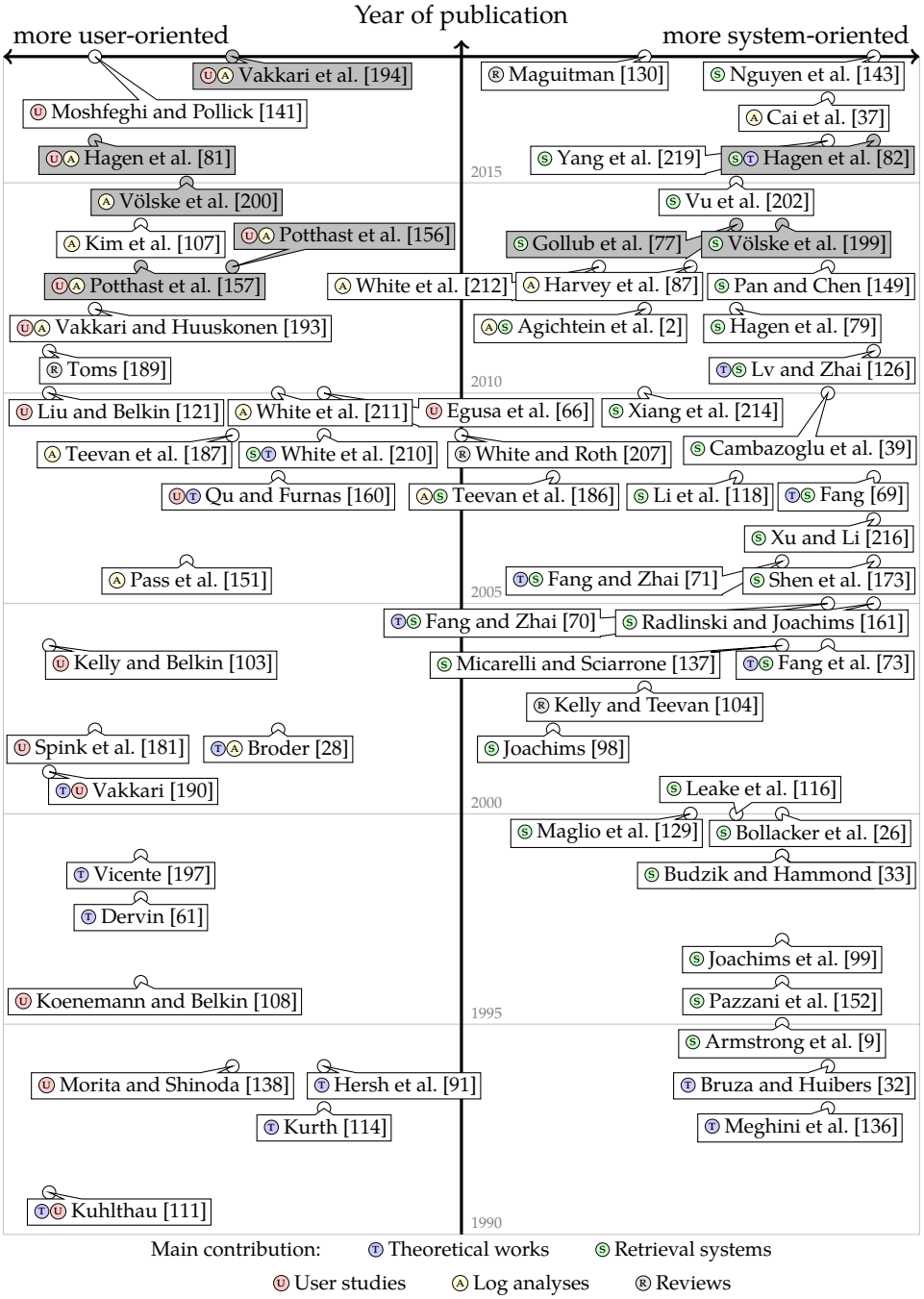


Figure 2.2: A selection of the relevant literature on task-based web search, context, and retrieval enhancement by time of publication and user-oriented versus system-oriented focus. Each entry is annotated with its primary research contribution(s). Own works are highlighted with a dark background.

2.2.1 Closed-ended and Open-ended Search Tasks

Many taxonomies of search tasks have been proposed, such as the popular distinction between informational, transactional, and navigational tasks of Broder [28]. In summarizing nearly a dozen such schemes, Toms [189] identifies a common fundamental distinction between just two core types, corresponding (but not unique) to Marchionini's [134] dichotomy between closed-ended and open-ended tasks. Closed-ended tasks seek a specific fact, item, or information object, whereas open-ended tasks do not necessarily have a clear goal at the outset; open-ended tasks' goals may come into focus, and evolve, over the course of the search. For example, compare the search for a famous person's date of birth with that for the best accommodation for an upcoming trip: in the latter case, the sought hotel rating, accommodation type, or even the destination of the trip may change as the searcher learns about prices and availability. Open-ended tasks are often referred to as *exploratory search* since they do not necessarily lead to only one correct answer, but help to build a mental model of a topic [192].

To date, the most comprehensive overview of research on exploratory search systems is that of White and Roth [207]. More recent contributions not covered in this body of work include the approaches proposed by Morris et al. [139], Bozzon et al. [27], Cartright et al. [43], and Bron et al. [30]. Exploratory search is studied also within contextual IR and interactive IR, as well as across disciplines, including human computer interaction, information visualization, and knowledge management.

White and Roth [207] describe exploratory search as an iterative, multi-tactical process, where the user explores the information space as extensively as necessary to fulfill an open-ended information need. Closed-ended searchers may iteratively refine their queries as well, but they usually zero in on a specific, targeted piece of information. Exploratory searchers, instead, explore the information space extensively; while examining search results, they obtain clues for their next steps [209]. The challenge of exploratory search is to design retrieval models that support users in these tasks. Web search engines are typically tuned towards precision, which limits the chance of finding loosely related information. But exploratory search is more recall-oriented [134]. This can be supported via rapid query refinement in the early phase of a search [205], supporting facets such as search result clustering [182], and leveraging the searcher's context, e.g., via pseudo-relevance feedback [215]. Chapter 3 studies the task of writing an essay on a given topic, which is open-ended and exploratory.

2.2.2 Query Logs for Understanding Search Tasks

The query logs of search engine users are a valuable resource to study their tasks, goals, and the processes by which they achieve them. However, Kurth [114] argued early on that no measure that can be derived from user interactions alone explains the user's intentions. Researchers nevertheless rely on such measures for lack of a better alternative. Typical measures found in the literature include the number of queries submitted by a user, the average number of terms and clicks per query, and the time between query and first click [8]. Log analyses often measure further attributes from a more global context, such as the number of physical sessions to complete a task. Machine learning algorithms then exploit a wide range of such and similar measures [1, 2, 23, 40, 148]. For example, Agichtein et al. [2] predict whether a user is likely to resume a suspended session within the next few days. After determining the dominant topic of the majority of the queries using data from the Open Directory Project, their approach is able to automatically decide for each query whether it is related to the task or not.

While log analyses have been conducted for a long time, exploratory search has shifted into focus only recently: to the best of the author's knowledge, Qu and Furnas [160] were the first to design a corresponding study. Based on the sense-making model [60, 168], they studied the relation between information seeking and construction of a mental representation. In this regard, not only the interactions of their 30 participants with the search system were recorded, but participants were also asked to prepare an outline for a 1-hour talk. Interestingly, Qu and Furnas found that the resulting talk structure strongly correlated with that of the participant's bookmark folders. Human judges rated the topical similarity between consecutive queries and assigned each query to one of the bookmark folders. Qu and Furnas visualized this information on a timeline to show when which query occurred, which folder it referred to, and which web page was bookmarked in this context. The visualizations for all 30 subjects reveal the influence of emerging structure on the following search. Moreover, 14 out of 30 participants used their folder structure as a roadmap for subsequent search. The authors conclude that search engines should support users, for instance, by analyzing the structure of their bookmark folders.

Vakkari and Huuskonen [193] designed a study that concentrates on the search process, especially the effort that users put into the search, and how it is interlinked with the task outcome. Within the scope of a term's course, medical students were asked to find information with a domain-specific search engine in order to write an essay on a medical topic. The search log

interactions were examined with respect to the applied search tactics (narrowing and broadening of queries, use of logical operators, etc.) and effort variables (like number of sessions or the number of read, but not cited articles). The essays' grades as awarded by the course's instructors were used as a performance measure for task outcome. Vakkari and Huuskonen show several interesting relationships between search process, output and outcome variables. They report a negative correlation between diversity of queries, search engine precision, and essay scores: the broader the queries were formulated, the lower the system's precision, yet the higher the essay scores. A similar correlation was observed for search effort: the more sessions a student needed to write the essay, the lower was the system's overall precision because of the larger result set, but the higher was the quality of the essay.

In a similar vein, Liu et al. [123] investigated the association between newspaper article writing and information search in a study with 24 undergraduate students. The participants worked on one of two writing tasks, with intermediate stages of task completion recorded at the end of each of three sessions. A potential source of data for the purpose of assessing current exploratory search behavior is to detect exploratory search tasks within raw search engine logs, such as the 2006 AOL query log [151]. However, most session detection algorithms deal with short term tasks only and the few algorithms that aim to detect longer search missions still have problems when detecting interesting semantic connections of intertwined search tasks [80, 100, 125]. Kules and Capra [112] manually identified exploratory tasks from raw query logs on a small scale; most of the identified tasks turned out to involve writing on a given subject. Inspired in part by this insight, the dataset described in Chapter 3 models exploratory tasks via essay writing on an assigned topic.

Chapter 4 of this dissertation describes extensive experiments with query log data spanning one year. Up to today there are only few studies dealing with query data stretching over such long periods. Richardson [163] explores the long-term dependencies of users' intents and preferences based on a one-year log of millions of users. He concludes that the analysis of user behavior based on such long periods can uncover information not present in shorter logs, and as such be of interest not only for information retrieval, but also for social sciences, psychology, market research, and medical studies. Note that in contrast to Richardson's study, we aim at analyzing question queries in particular, but still the dimensions of the employed log data are comparable. The aforementioned work by Pang and Kumar [150] also draws conclusions based on the analysis of an annual search log, but other studies

used much smaller logs. Beitzel et al. [18] explore the topical structure of a six month query log and Liu et al. [120] track user behavior based on log excerpts spanning two weeks in three subsequent years.

Topical categorization of large query logs can provide high-level insights into the spectrum of user interests and their dynamics. Spink et al. [180] manually label several thousand queries from a search log in an attempt to study user interests. Later, Beitzel et al. [18] automatically match queries against manually compiled topical word lists, classifying 13% of a search engine's query stream. A bootstrapping of this method based on word-category distributions yields an improved recall [19] but still low coverage. In a fully automatic large-scale analysis, Bar-Ilan et al. [15] perform a topical classification of the AOL and MSR logs using an SVM classifier over query word uni- and bigrams.

2.2.3 Questions and Question Answering

Traditionally, the information retrieval subfield of question answering (QA) has dealt with information needs expressed as natural language questions, with emphasis on fact-seeking inquiries, or *factoids* (e.g., `[what is the german population?]`. Automatic question classification within QA generally aims at obtaining the expected answer type (e.g., numeric value, location, person, etc.), rather than the question topic. Implicit topical analysis is left up to the IR-component of a QA system. This is true both for early work [119] and for more recent approaches that employ deeper and more comprehensive analyses [115].

Question queries have been the subject of dedicated search log studies [179] and have been analyzed in the context of long queries [22]. Pang and Kumar [150] draw attention to the phenomenon of question queries in search engine logs, describe their structural and statistical characteristics, and show that the share of these queries grows. Verberne et al. [196] examine how well web search engines answer causal *why*-questions and show room for improvement. Xue et al. [218] analyze question reformulation patterns in search query logs and show that automatic question rewriting can potentially lead to improved search results. A more recent longitudinal study on the evolution of user behavior shows questions as an important part [120]. All authors agree that *how-to*-questions are the most popular, which demonstrates a new direction in comparison to the factoids covered in "classical" question answering. The authors also note that the search results for question queries are usually worse than for the corresponding keyword queries expressing the same information need [11, 22, 150].

The spread of community question answering (CQA) services such as Yahoo! Answers provides a parallel setting in which to study question-asking behavior on the web. CQA sites allow users to pose questions to other community members, to answer questions, rate questions and answers, and receive feedback. The services are quite popular and have collected a vast amount of content in the form of questions and answers that is being indexed by major search engines. The odds are high that a question a user has in mind has already been asked by someone before and can be found through search engines. There are several studies conducted on the intersection of web search and CQA. Weber et al. [204] aim at finding answers (tips) to web queries with *how-to* intent (not necessarily expressed as well-formed *how-to* questions) in Yahoo! Answers archives. Liu et al. [122] evaluate the utility of existing CQA answers in web search scenarios. In a follow-up study [123] the authors track users, who follow up web searching with asking a question on a CQA platform. Two studies [63, 222] propose methods to generate natural-language questions suitable for posting on a CQA service based on keyword queries issued by the user. In contrast to these studies, the study described in Chapter 4 does not aim to develop methods that provide better answers to questions or that recommend CQA items to the users. Instead, we target the topical categorization of questions on the scale of a year; the results might then improve retrieval systems, as is proposed in some studies [41, 64].

The Topical categorization of questions posted on CQA services is the subject of several studies. For instance, Li et al. [117] suggest to use topic information in a question routing task (i.e., delivering newly posted questions to potential answerers). While this use case is rather different from ours, the study of Qu et al. [159] who investigate the contribution of different components to question classification quality (machine learning methods, n-gram features, data fields, and training sample size) is closer to our setting. We incorporate several of their findings in one of our methods using bag-of-words features. Chan et al. [44] apply a set of kernels corresponding to different aspects of questions to hierarchical question classification. Since we are interested in a rather broad, non-nested category scheme that can be used in the actual retrieval process, we do not aim for any hierarchy. Cai et al. [38] propose to enrich CQA questions with Wikipedia entries as a means to counter the sparseness problem discussed above.

2.2.4 Success Criteria: Search Output versus Task Outcome

As noted in Section 2.1.2, retrieval systems are traditionally evaluated in a way that rewards putting known-relevant documents near the top of the result list above all else. Performance metrics like nDCG measure the precision of the top ranks in the result list, and retrieval systems optimized against such metrics hence tend to better support precision-oriented rather than recall-oriented tasks. In this sense, Vakkari [192] differentiates between evaluating *search engine output*—the precision of a result list with regard to the submitted query—and *task outcome*, which describes how well the system supported the user in fulfilling the task. A high precision does not necessarily lead to good overall task performance, as open-ended search tasks tend to be more recall-oriented.

In a 2010 study, Egusa et al. [66] asked 35 undergraduate students to produce a concept map of their understanding of a given topic before and after searching. A concept map is a graph consisting of named entities and labeled connections between them. Egusa et al. analyze the differences between the before- and after-maps to measure the effectiveness of the search. This serves as an example of how task outcome based measures can go beyond traditional IR measures in assessing not only precision but also the *benefit* a user has from a set of search results [207]. At the same time, this example illustrates the much greater difficulty in operationalizing task outcome based measures—concept maps can only be obtained by asking users directly, whereas a large body of literature deals with obtaining relevance feedback implicitly (cf. Section 2.3).

Egusa et al. performed their experiments on only two broad (i.e., open-ended) topics, namely "Politics" and "Media." The task was to find and compare different opinions about these topics. The before- and after-maps were analyzed with respect to the number of kept, discarded and inserted nodes, links and labels. Among other findings, nearly as many deletions as insertions occurred. This indicates that people not only gather new information while exploring a topic but also adjust their existing knowledge. However, the authors conclude that applying descriptive statistics on concept maps cannot serve as a measure for the performance of an exploratory search system. They argue that one has to conduct more qualitative analyses of the described concepts and users' searching behavior.

In general, White and Roth [207] cast doubts on the appropriateness of traditional measures of IR performance based on retrieval accuracy—and the evaluation paradigms discussed in Section 2.1.2—for the evaluation of exploratory search systems. They argue for the inclusion of naturalistic lon-

gitudinal studies in exploratory search evaluation settings, and expect simulations based on interaction logs—while they can bridge the gap to traditional evaluation paradigms—to only work in some cases. The necessity of user studies makes evaluations cumbersome and, above all, expensive. To help overcome the outlined difficulties, Chapter 3 wants to provide part of the solution (a decent corpus) by compiling a solid database of exploratory search behavior, which researchers may use for comparison purposes as well as for bootstrapping simulations.

Regarding standardized resources to evaluate open-ended and exploratory search tasks, hardly any have been published up to now. White et al. [208] dedicated a workshop to evaluating exploratory search systems in which requirements, methodologies, as well as some tools have been proposed. Yet, later on, White and Roth [207] found out that still no "methodological rigor" has been reached—a situation which has not changed much until today. The departure from traditional evaluation methodologies (such as the Cranfield paradigm) and resources (especially those employed at TREC) has lead researchers to devise ad-hoc evaluations which are mostly incomparable across papers and which cannot be reproduced easily.

2.2.5 Usefulness of Search Results

In assessing information retrieval effectiveness, the value of search results to users has gained popularity as a metric of retrieval success. Supplementing established effectiveness indicators like topical relevance [20, 91, 96], the worth [52], utility [91], or usefulness [20] of information depends on the degree to which it contributes to accomplishing a larger task that triggered the use of the search system [52, 91, 191]. Despite the growing interest in information usefulness as a retrieval success indicator, only a handful of studies have emerged so far, and they typically focus on perceived usefulness rather than on the actual usage of information from search results. Even fewer studies explore the associations between user behavior and information usage during task-based search. The lack of contributions towards this important problem arises from the difficulty of measuring the usefulness of a search result for a given task in a laboratory setting.

In order to help their users achieve a favorable task outcome, retrieval systems need to surface search results that are useful, and search results are useful if the information they contain contributes to the user's task [191]. Users are expected to click, scan, and read documents to identify useful pieces of information for immediate or later use [220]. Only a few studies

on the usefulness of search results focus on predicting the usefulness for some task [103, 121, 123, 133]; most others are more interested in comparing expert assessors' topical relevance and usefulness judgments to users' usefulness judgments.

Different studies from this latter group report rather mixed findings, but as a general trend tend to find inconsistencies between usefulness and relevance judgments. Kim et al. [106] compared binary usefulness assessments of results from users searching the web to relevance assessments of the same results by trained assessors. With decreasing relevance, judges classified an increasing proportion of results inconsistently with users. Mao et al. [132] made consistent observations, finding a low correlation between users' usefulness assessments and judges' relevance assessments, as well as between users' and judges' usefulness assessments. By contrast, Jiang et al. [97] obtained topical relevance and usefulness assessments of search results clicked by users for various types of search tasks. They found a high correlation between in situ usefulness assessments on the one hand, and post-session topical relevance and usefulness assessments on the other.

In most cases, usefulness is operationalized as perceived by users, rather than in terms of the actual usage of information. For instance, Kelly and Belkin [103] explored the association between documents' display time and their usefulness during a fourteen-week study on seven PhD students' real-world search tasks. Here, usefulness was operationalized as the degree of users' belief how helpful the document was, as reported via a tailor-made evaluation interface. The study found no association between usefulness and dwell time, regardless of the task type. In a similar vein, Liu and Belkin [121] studied whether the time spent on a clicked document was associated with its perceived usefulness for writing a journalistic article and—contrary to Kelly and Belkin—found a positive association between the dwell time on a document and its usefulness assessment. Users typically moved back and forth between the text they produced and the document informing their writing.

Liu et al. [123] later modeled users' search behavior for predicting the usefulness of documents: they had users assess the usefulness of each saved page for an information gathering task, and employed binary recursive partitioning to identify the most important predictors of usefulness. In an ascending order, dwell time on documents, time to the first click, and the number of visits on a page were the most important predictors—the longer the dwell time, the more visits on a page and the shorter the time to first click, the more useful the page.

Mao et al. [133] recently modeled the usefulness of documents for answering short questions by content, context, and behavioral factors, where usefulness was measured on a four-point scale. They found that behavioral factors were the most important in determining usefulness judgments, followed by content and context factors: the perceived usefulness of documents was positively correlated with dwell time and similarity to the answer, and negatively with the number of previously visited documents.

By comparison, Ahn [3] and He [89] evaluated the actual usefulness of retrieved information by measuring to what extent search systems support finding, collecting and organizing text extracts to help answer questions in intelligence tasks, with experts assessing the utility of each extract. Sakai and Dou [169] proposed a retrieval evaluation measure based on the amount of text read by the user while examining search results, presuming this text is used for some purpose during the search session.

2.3 Context and Relevance Feedback

Aside from information needs expressed explicitly in the form of queries or questions, modern information retrieval systems also take implicit information about the user's task and its context into account. The use of such information has been referred to by various names in the literature—including implicit relevance feedback [104] or contextual search [130].

Kelly and Teevan [104] present a detailed review of the literature (as of 2003) on implicit relevance feedback, along with a classification of the user behaviors that serve as sources of context information, along the axes behavior category and the minimum scope to which the behavior applies. Their categorization is reproduced in Table 2.1 where the more general terminology of the original has been replaced by terms specific to the text retrieval that is the focus of the present work. Kelly and Teevan sorted the vast majority of existing research at the time into the "Examine–Document" cell in the table, as this is where SERP clicks are found. Arguably, clickstream logs are still the most common type of relevance feedback information used today.

A more recent literature review by Maguitman [130] completes the picture; here, the categorization of contextual data is primarily into long- and short-term context, i.e., those relating to a persistent user profile versus to the current task. Uses of contextual data are noted as those related to query expansion, refinement and disambiguation, as well as to filtering and reranking results or routing to specialized search indexes (i.e., vertical search). Maguitman focuses in large part on retrieval systems that have incorporated

Table 2.1: Classification of user behaviors that can serve as a source of context for implicit relevance feedback. Adapted from Kelly and Teevan [104].

Behavior Category	Minimum Scope: Passage	Document	Collection
Examine	View, Listen, Scroll, Find, Query	Select (Click)	Browse
Retain	Print	Bookmark, Save, Delete, Purchase, Email	Subscribe
Reference	Copy-and-Paste, Quote	Forward, Reply, Link, Cite	
Annotate	Mark up	Rate, Publish	Organize
Create	Type, Edit	Author	

contextual information, and presents a historical overview of such systems. The following reproduces a small selection of the works mentioned by Maguitman; Figure 2.2 shows them in context, along with works displaying a more user-oriented or theoretical focus.

Among the earliest known general-purpose context-based retrieval systems are the *WebWatcher* and *Syskill & Webert* systems, presented by Armstrong et al. [9] (with a follow-up work by Joachims et al. [99]) and Pazzani et al. [152], respectively, in the mid-1990s. Those systems and their contemporaries tended to rely on users explicitly specifying preferences, or annotating search results with respect to their usefulness, rather than on observing implicit feedback signals. However, an earlier user study by Morita and Shinoda [138] already put forward the idea of monitoring user behavior in the background, and thus transparently capturing features predicting—in this case—the user's interest in online news items. Contemporary user studies such as the one by Koenemann and Belkin [108] began to systematically test the effectiveness of relevance feedback facilities.

Watson, proposed by Budzik and Hammond [33] in 1999, is an early information system targeting the support of users during writing tasks, by recommending URLs to visit based on the text the user is typing (anticipating an information need before it is stated explicitly); as such, this system is similar in its aim to the research presented in Chapter 3. A range of con-

temporary works from the area of user interface design, including *Context Toolkit* [170], *SUITOR* [129] and *CALVIN* [116], explored ways in which context information can be unobtrusively collected in the background while the user is performing some information intensive task. At least tangentially related are information recommendations systems like *CiteSeer*, proposed by Bollacker et al. [26]. A wide variety of subsequent works build upon this group of early adaptive systems; such as the *HUMOS/WIFS* system presented by Micarelli and Sciarrone [137].

Kelly and Belkin [103] study the effectiveness of using display time (or dwell time) as an implicit relevance feedback signal in a longitudinal user study. While dwell time had been identified as a useful implicit feedback signal in prior decades, and was actively being exploited by commercial search engines at this point, this was the first study to take the user's current information seeking task into account. Kelly and Belkin found a more complex relationship between display time and relevance than previously assumed, which is mediated by task effects.

White et al. [210] present a model of user interest to improve website suggestion, which allows aggregating various categories of context information—e.g., recent and long-term interactions of the user, interactions of other users, or properties of the collection—into a unified model that predicts future user interests. Follow-up work extends this to short-term interests, i.e. related to the current task or session [211].

2.4 Retrieval Enhancement

There are several stages in the process of generating a retrieval system's answer to a task-based query where contextual information about the user's task can be integrated (refer to the lower half of Figure 1.1 on page 3). In the following, two contextual retrieval enhancement strategies—query preprocessing, and result set postprocessing—will be briefly reviewed. Later on, Chapters 4 and 5 will present own contributions to the respective problem areas.

2.4.1 Query Preprocessing and Categorization

Retrieval systems tend to perform poorly for long-tail queries: while a rich set of implicit relevance feedback data can be called upon to improve results if the query has been submitted by many users in the past, this is not the case if a query is rare or even unique. One option to improve the search results for under-resourced queries is to preprocess the query before further han-

dling by the retrieval system. This encompasses techniques such as query understanding [55], query spelling correction [84], query expansion [86], query segmentation [79], and query categorization or classification. Due to its relevance to the contributions of this dissertation in general (and Chapter 4 in particular), the following exposition will focus on the latter technique as applied to question queries—a particular class of under-resourced queries.

Query categorization allows the retrieval system to account not only for the query's terms, but also its topic. The technique has benefited general search [14], query disambiguation and routing to vertical search [118], and search advertising [29]. The main difficulty in query classification is data sparseness: the short query strings. Search queries contain around three words on average [22, 151]; while question queries studied in Chapter 4 are somewhat longer—around six to seven words, according to different studies [123, 150]—they are still much shorter than web documents.

The data sparseness problem is usually addressed by enriching queries with additional information and performing the classification on these augmented representations. Queries are categorized based on the category labels of documents returned by a search engine [29] or are enriched by the search results containing document titles and snippets [173]. Bailey and coauthors [14] classify long queries with sparse user interaction data by matching them against shorter and more popular queries categorized based on past users' behavior. Li et al. [118] suggest to substantially expand the set of labeled queries using click-through information: user clicks on the same link returned for different queries are considered as a similarity indicator. Thus, iterative propagation of category labels from seed queries along click edges through co-clicked documents to unlabeled queries allows expanding the initial training set by several orders of magnitude.

Note that, to be practically applicable, all three approaches require the availability of search log information. In case of click-through information this is rather obvious. In case of using returned results or titles and snippets for categorization, the classification can be accounted for in a second retrieval run or performed off-line and then applied on-the-fly if the query appears again.

2.4.2 Result Set Postprocessing

Aside from preprocessing the query, context information can be incorporated, and retrieval performance enhanced, by modifying the result ranking. To this end, the retrieval system takes a top-k result set and reorders

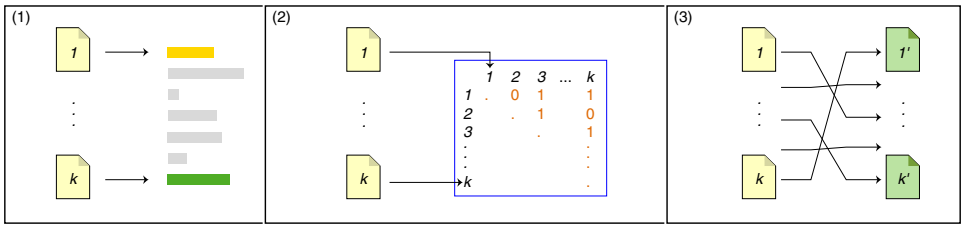


Figure 2.3: Pointwise, pairwise, and listwise learning-to-rank approaches: Given an initial top-k result set, (1) predict a relevance score independently for each individual document, (2) predict a ranking preference for each pair, or (3) directly optimize a permutation of the whole ranking.

or otherwise modifies it, so as to better reflect the anticipated usefulness based on the query, the characteristics of the result documents themselves, and any available context information. Postprocessing can be understood to encompass techniques like search result clustering [183] that affect primarily the way the search results are presented to the user, but in the what follows, we focus on those techniques that affect the ranking itself.

In this sense, the relevant result set postprocessing ideas were developed primarily in the learning-to-rank domain. There, the goal is to rank documents based on machine learning algorithms [124]. In general, three different approaches can be distinguished: pointwise, pairwise, and listwise ranking (Figure 2.3). In the pointwise approach, machine learning methods predict the rank for each document based on document-individual feature values. The pairwise approach instead uses pairs of documents to predict a ranking preferences for each pair [98]. The listwise approach does not learn a ranking function for individual documents or pairs but processes entire result lists. Independent of the employed learning approach, most learning-to-rank systems are built on top of a basis retrieval model: An initial document set, typically consisting of the basis model's top-k results, is retrieved and then re-ranked using the learned ranking method.

There are many directions for improving rankings in a learning-to-rank style. For example search engine logs provide a lot of implicit information that can be used to inform the learning process. Radlinski and Joachims [161] describe a learning-to-rank system that exploits click-through data in such a way. The technique described in Chapter 5, due to a lack of large click logs available for training, sticks to explicit feedback from the TREC relevance judgments for training. At first sight this may appear related to an approach of Veloso et al. [195] who use data mining techniques to learn association rules based on relevance judgments. However, instead of learning association rules, we take the set of axioms as given and learn only their

importance by inferring an aggregation function. Our idea of training different axiomatic rankers while optimizing the target performance measure of nDCG_{10} is inspired by the AdaRank framework [216] that also directly optimizes the performance measure instead of classification errors.

Due to the complexity of machine learned ranking models, learning to rank approaches typically follow a two-step approach in order to guarantee a fast system response: rather than ranking all documents in the index using the complex learned ranking function, a simple base retrieval model is used to retrieve an initial top-k result set, which is then re-ranked with the learned ranking model [39]. In this sense, the approach described in Chapter 5 of this dissertation is similar to a learning-to-rank approach, as well. However, rather than feedback signals from the query log, the focus is on incorporating theoretical insights on ranking functions into the retrieval process in a principled way. This approach is inspired by the ideas from axiomatic information retrieval outlined in the following section.

2.5 Axiomatic Ideas in Information Retrieval

As noted in Section 2.1, a wide variety of functions have been proposed to score the documents of a collection for ranking with respect to a query, with more or less solid theoretical foundations: some scoring functions are derived from a retrieval model prescribing how documents should be matched to queries, others are discovered via empirical experimentation, and often it's a mix of both. This section gives a brief outline of the subfield of axiomatic information retrieval, which approaches the search for a good scoring function from a different direction: axiomatic practitioners define first principles (i.e., axioms) that formally describe desirable properties of a good result ranking; from the axioms, new scoring functions (or improvements to existing ones) can be derived that guarantee these properties. Axioms for information retrieval play a rather "theoretical" role so far. Most of the respective studies focus on the question of whether the results of known retrieval models are in accordance with specific reasonable axioms that formalize ranking preferences. Using the theoretical ideas in this section as a starting point, Chapter 5 of this dissertation proposes a way of postprocessing result rankings at retrieval time—incorporating all of the known axioms in one retrieval system, with the possibility of adding new axioms in the future.

2.5.1 From Early to Modern Axiomatic IR

The earliest studies of axioms in the context of information retrieval date back more than 30 years now [32, 135, 136]. One of the first published ideas that can be considered "axiomatic" is a retrieval system based on production rules from artificial intelligence by McCune et al. [135], which led to some improvements over a simple Boolean model. Another approach using more formal rules (again, these could be viewed as axioms) is presented by Meghini et al. [136], who use terminological logics for building a retrieval model. The first real reference to a notion of axioms for information retrieval is contained in the aboutness study of Bruza and Huibers [32]. Actually, the authors do not propose a retrieval model but rather a way of expressing what should be expected from a good result ranking. Especially in the last decade, the interest in this direction of using axioms to describe what a good ranking looks like has increased substantially. Hui Fang's web page gives a good overview of the existing literature and axioms.[1]

This more modern branch of axiomatic retrieval research starts from the observation that those retrieval functions that are successful in practice tend to share similar sets of heuristics. Consider the standard formulation of the BM25 retrieval function (cf. Section 2.1): it includes term frequency, inverse document frequency, and document length normalization, and this is the case for the majority of contemporary, successful ranking functions [72]. This observation has inspired two key questions: *what is the full set of desirable heuristics*—and—*how should they be combined in a retrieval function?* Consequently, the goal of most modern axiomatic retrieval research is to propose reasonable axioms, and to evaluate how well existing retrieval models match the respective assumptions.

2.5.2 Overview of Known Axioms

The existing axiomatic literature can be divided by the goal of the axioms: term frequency, document length normalization and lower bounds, query aspects, semantic similarity, or term proximity. The following paragraphs briefly review important axiomatic ideas across these, as well as other axiomatic ideas that do not fit any of the aforementioned categories. An important distinction is the question whether a given axiom expresses a ranking preference—that is, formulates the conditions under which some document should be ranked higher or lower than another—or can be restated

[1] `http://www.eecis.udel.edu/~hfang/AX.html`

in such a way that it does. Only if that is the case, the corresponding axiom can be incorporated in the approach described in Chapter 5.

Term frequency Term frequency axioms follow the idea that documents containing query terms more often should be ranked higher. Fang et al. propose several such axioms (TFC1–TFC3 and TDC) [70, 73, 74] and experimentally show that satisfying them produces better rankings. The axiom TFC1 is a simple, but typical, example—it merely states that a document with more occurrences of a query term should be ranked higher. Formally, the TFC1 requirement is commonly stated as follows:

$$\begin{aligned} \textsc{Given} \quad & \text{a single term query } q = \{t\}, \\ & \text{and documents } d_1, d_2 \text{ with } |d_1| = |d_2|\,; \\ \textsc{If} \quad & tf(t, d_1) > tf(t, d_2) \\ \textsc{Then} \quad & \mathrm{SCORE}(q, d_2) > \mathrm{SCORE}(q, d_2) \end{aligned}$$

This formulation exemplifies some typical properties of the known axioms that are relevant to the approach described in Chapter 5: it states a set of requirements for the axioms to apply (in this case, a single-term query and equal document length), a specific condition that must hold with respect to features of the documents, and a ranking preference that results. We employ all of the above mentioned term frequency axioms in our approach. Na et al. [142] propose some specific axiomatic term frequency constraints tailored to language modeling (LM) retrieval approaches. Since their axioms cannot be easily rephrased to be generally applicable to non-LM retrieval, we decided not to include these axioms.

Document length Besides term frequency axioms, Fang et al. also propose document length axioms (LNC1, LNC2 and TF-LNC) [73] with the basic idea that in case of same term frequencies shorter documents should be ranked higher. We employ all of these axioms in our approach. A query-based document length constraint (QLNC) proposed by Cummins and O'Riordan [58] can not easily be reformulated to induce rank preferences so that we do not include it in our approach.

Lower bounds on term frequency normalization Combining frequency and document length, lower bound axioms state that long documents should not be penalized too much. Lv and Zhai propose two such axioms (LB1 and LB2) [126, 128]. For example, LB2 assumes two documents d_1 and

d_2 with identical ranking scores, and both containing some query term t; further, it assumes a document pair d'_1 and d'_2, where d'_1 is identical to d_1 up to an additional occurrence of t in d'_1 and d'_2 is identical to d_2 up to an additional occurrence of a new term t' that is contained in neither d_1 nor d_2. In this setting, d'_2 should be ranked higher than d'_1—simply stated, first occurrence of a term is more important than repeated occurrence.

Query aspects Zheng and Fang [224], and Wu and Fang [213] propose axioms (REG and AND) that aim at ranking documents higher that match more query terms or aspects. Gollapudi and Sharma [76] propose axioms (DIV) with a similar purpose, modeling the diversity of a result set as a whole. Interestingly, they show that no diversification function can satisfy all the axioms simultaneously. In their original formalizations, these query aspect related axioms do not induce rank preferences; we use adapted versions in our approach.

Semantic similarity It can often be important not to rely on exact term matching between query and results but to also take documents into account that contain semantically similar terms. Fang and Zhai [71] propose five axioms along these lines (STMC1–STMC3, TSSC1, TSSC2), which were later shown beneficial also in a query expansion setting [69]. We only use STMC1 and STMC2 in our approach since STMC3, TSSC1, and TSSC2 can not be restated to induce rank preferences.

Term proximity Term proximity axioms describe the importance of query terms appearing close to each other in results (e.g., phrases). Tao and Zhai [184] introduce several respective axioms (DIST1–DIST5)—but rather with the goal of improving a retrieval model's proximity feature than to induce rank preferences. Since their axioms do not induce rank preferences, Chapter 5 proposes new term proximity axioms instead.

Other axiom ideas There is a wide range of other axiomatic studies that do not fit the above groups. Many of these are not helpful in our setting since either the axioms have a completely unrelated purpose (e.g., axioms for evaluation [7, 35]) or the axioms do not induce rank preferences by nature. An exception is Altman and Tennenholtz' study of properties implied by graph-theoretic axioms for link graphs [6]. They show that their axioms are satisfied by the PageRank algorithm but the axioms do not induce any rank preferences. We include a modified PageRank-based axiom as one of our contributions.

Cummins and O'Riordan [56, 57] analyze axioms for learned ranking functions, but since none of the basis retrieval models we will use is machine-learning-based, the respective axioms would not help. Clinchant et al. [50, 51] describe axioms for pseudo-relevance feedback models (PRF) that are also not applicable in our setting since we do not employ PRF methods. Gerani et al. [75] propose axioms for combining scores in a multi-criteria relevance approach [75] that also do not fit the basis retrieval models we will employ. Zhang et al. [221] present an axiomatic framework for user-rating based ranking of items in Web 2.0 applications, but since our ad-hoc retrieval task is different, their axioms could not be applied. Karimzadehgan and Zhai [102] and Rahimi et al. [162] perform axiomatic analysis of translation language models in order to gain insights about how to optimize the estimation of translation probabilities; again, the purpose is different to our setting such that we do not include these axioms. Ding and Wang [62] show how axioms covering term dependency can be integrated into language-model-based retrieval approaches, but since their axioms do not induce preference lists and are not applicable to the non-LM approaches among our basis retrieval models, we do not include these axioms in our approach.

2.5.3 Benefits of Axiomatic Analysis: an Example

To complete the brief picture of axiomatic information retrieval presented here, we revisit the $\mathrm{BM25}$ retrieval function from Section 2.1.1 to show the potential benefits of axiomatic analysis as it is typically applied: Lv and Zhai [127] found $\mathrm{BM25}$ to violate the LB2 constraint given above: if a document is very long, the document length normalization term in the formula tends to drown out the scoring boost the document should receive for containing query terms not present in other documents in the ranking. Beyond that, Lv and Zhai also propose a simple correction, $\mathrm{BM25}^+$, that guarantees the LB2 constraint:

$$\mathrm{BM25}^+(q,d) = \sum_{i=1}^{n} idf(q_i) \cdot \left(\frac{tf(q_i,d) \cdot (k_1+1)}{tf(q_i,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{avgdl}\right)} + \delta \right)$$

Over the original formulation, only the additional free parameter δ is introduced; Lv and Zhai [126] show that whenever $\delta \geq \frac{k_1}{k_1+2}$ holds, the LB2 constraint is satisfied. As noted, Chapter 5 presents an attempt at incorporating axiomatic ideas into the retrieval process without the need to modify the scoring function.

2.6 Summary

This chapter has introduced the background and previous research that the contributions showcased in the following chapters build upon. Figure 2.2 has organized previous work, among other things, along a user-oriented or system-oriented continuum, and the chapter has roughly followed a trajectory from more system-oriented to more user-oriented. The following chapters repeat this same trajectory, with Chapter 3 making the most user-oriented, Chapter 5 the most system-oriented contribution, and Chapter 4 falling in between.

3

Understanding and Supporting Writing Tasks

The web has fundamentally changed how writers of non-fictional texts approach their task. In the past, research on a topic and writing about it typically happened separated in time and space (e.g., research in the library, writing at home). Nowadays, both can be done more or less simultaneously, since web search engines retrieve relevant information on almost any topic. Therefore, writers cab easily switch between search and writing whenever they perceive gaps of knowledge (i.e., information needs). This situation has accelerated the rate at which non-fictional texts are written as well as significantly decreased the costs of doing so, which is particularly true in cases where the resulting texts are not expected to be award-worthy, but merely publishable. Due to the frequency with which writers turn to a web search engine—for inspiration and ideas, to retrieve sources, or to check facts—-writing tasks are of interest for search engines to support better, but providing this support is not trivial: in contrast to less complex search tasks, the retrieval system needs to surface results that are not only relevant to the most recent query, but *useful* to furthering the writing task. Determining usefulness requires deep insight into the user's task progress.

With that overall end in mind, this chapter studies the writing process and search behavior of writers in action, and addresses the following sub-goals: (1) collecting a large dataset that captures the search and writing activity of essay authors who use a search engine to retrieve their sources [157]; (2) analyzing the search and writing activity to gain insights

into how writers search [156, 200]; (3) predicting writers' success at finding useful sources based on observed search and writing activity [194].

Accordingly, the present chapter lays out a series of results obtained along the way to making writing support by retrieval systems feasible. The primary roadblock to research on this problem has long been a lack of available datasets that cover simultaneous search and writing behavior. Section 3.1 reviews Webis-TRC-12 dataset; driven by research interest in text reuse analysis and plagiarism detection, the collection effort involved the creation of a petri-dish environment for search-supported essay writing: we hired 12 authors to write a total of 150 essays on that many topics, at least 5,000 words each, while recording a fine-grained log of text revisions, search queries, result clicks, and browsing. To attain reproducibility, we chose topics from the TREC web track and set up a static web search environment based on the ClueWeb09. Our search engine employs BM25F as the retrieval model, its user interface resembles those of commercial search engines, and its performance was optimized to allow for an average retrieval time of less than five seconds. Sections 3.2 and 3.3 elaborate the insights on search and writing behavior—far beyond the original goals—that the analysis of this data has enabled; in the process, these sections answer Research Question 1a (Writing Strategies) and Research Question 1b (Searching Strategies), respectively. Additionally, the latter section provides an answer to Research Question 2 (Measuring Usefulness) with the help of authors' text reuse behavior. Finally, Section 3.4 takes a tentative first step towards search result utility prediction: while investigating Research Question 3 (Predicting Retrieval Success), we automatically determine the degree to which search engine users are successful at finding useful sources for their tasks.

3.1 The Webis-TRC-12 Dataset

Humans frequently interact with search appliances in order to conduct the research deemed necessary to solve knowledge-intensive tasks, often via long-lasting interactions which may involve many search sessions spread out across several days. Modern web search engines, however, are optimized for the diametrically opposed task, namely to answer short-term, atomic information needs. Nevertheless, research has picked up this challenge: in recent years, a number of new solutions for task-based and exploratory search have been proposed and evaluated. However, most of them involve an overhauling of the entire search experience. But search engine users are already tackling complex search tasks in the real world, with

current web search interfaces, and that this fact has not been sufficiently investigated. Reasons for this shortcoming can be found in the lack of publicly available data to be studied. Ideally, for any given task that fits the aforementioned description, one would have a large set of search interaction logs from a diversity of humans solving it. Obtaining such data, even for a single task, has not been done at scale until now. Even search companies, which have access to substantial amounts of raw query log data, face difficulties in discerning individual complex tasks from their logs.

This section introduces the Webis text reuse corpus 2012 (Webis-TRC-12), a corpus of long, exploratory search missions and associated writing logs. The corpus was constructed via crowdsourcing by employing writers whose task was to write long essays on given TREC topics, using a ClueWeb09 search engine for research. Hence, our corpus forms a strong connection to existing evaluation resources that are used frequently in information retrieval. Further, it captures the way average users perform exploratory search today, using state-of-the-art search interfaces. The new corpus is intended to serve as a point of reference for modeling users and tasks as well as for comparison with new retrieval models and interfaces. Simultaneously with the writers' search activity, their revisions to the developing essay texts were recorded in fine-grained detail. Key figures of the search and writing logs are shown in Table 3.2.

Our dataset construction efforts have been guided by the previous approaches described in Section 2.2.2 (pages 23ff), addressing several shortcomings: *(a) Task diversity.* Qu and Furnas [160], Egusa et al. [66] as well as Liu and Belkin [123] employed only two different topics. Vakkari and Huuskonen [193] employ eleven topics, but all of them from the medical domain. We employ 150 topics, derived from the TREC web track, which are diverse and can be understood by laypeople. *(b) Connection of search and task outcome.* Qu and Furnas [160] and Egusa et al. [66] do not provide revisions of task outcomes. Our study aligns all search interactions with text revisions on a time line, which allows fine-grained analysis of the connection between search and task outcome, as proposed by Järvelin et al. [95]. *(c) Experimental setup and reproducibility.* Qu and Furnas [160], and Liu and Belkin [123], asked participants to use a search system in their lab—a maximally obtrusive setting [94]—whereas our participants could work from home. Unlike the other studies, we employ a well-known web corpus frequently used for evaluation purposes to create a static search scenario, that can be reproduced even after years. *(d) Incentives and motivation.* All four studies recruit undergraduate students as study subjects, which often introduces bias in diversity and motivation. Vakkari and Huuskonen ensure proper motiva-

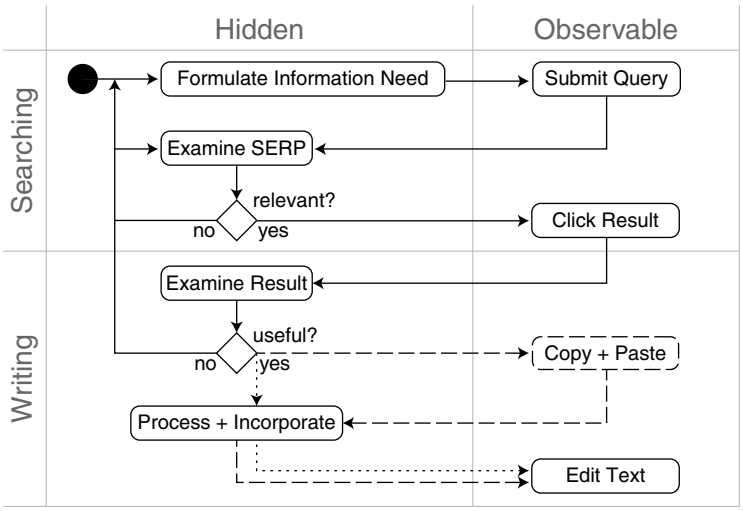


Figure 3.1: User actions in the search and writing process: our study design involving text reuse (dashed lines) allows for more direct observation of users' usefulness assessments of search results than would be possible without reuse (dotted lines).

tion, since their participants were graded and had the chance of earning credit points by completing the course; Liu and Belkin's participants received monetary compensation. In our case, we hired (semi-)professional writers from all over the world with a diversity of backgrounds, we had them sign a contract, and paid them on an hourly basis.

To justify our choice of an exploratory task, namely that of writing an essay about a given TREC topic, we refer to Kules and Capra [112], who manually identified exploratory tasks from raw query logs on a small scale, most of which turned out to involve writing on a given subject. Egusa et al. [66] describe a user study in which they asked participants to do research for a writing task, however, without actually writing something. This study is perhaps closest to ours, although the underlying data has not been published. The most notable distinction is that we asked our writers to actually write, thereby creating a much more realistic and demanding state of mind since their essays had to be delivered on time.

The Webis-TRC-12 writers were instructed to reuse (and, optionally, modify) text from web sources for their essays. This aspect of the study design allows the data to be used for the purpose of studying text reuse and plagiarism phenomena, and this was its original intent: The web has become one of the most common sources for text reuse. When reusing text from the web, humans may follow a three step approach: searching for appropriate sources on a given topic, copying of text from selected sources, modification and paraphrasing of the copied text [153]. A considerable

body of research deals with the detection of text reuse, and, in particular, with the detection of cases of plagiarism (i.e., the reuse of text with the intent of disguising the fact that text has been reused). Similarly, a large number of commercial software systems is being developed whose purpose is the detection of plagiarism. Both the developers of these systems as well as researchers working on the subject matter frequently claim their approaches to be searching the entire web or, at least, to be scalable to web size. However, there is hardly any evidence to substantiate this claim—rather the opposite can be observed: commercial plagiarism detectors have not been found to reliably identify plagiarism from the web [109], and the evaluation of research prototypes even under laboratory conditions shows that there is still a long way to go [154]. The disappointing state of the art can be explained at least in part by the lack of realistic, large-scale evaluation resources. However, as highlighted in Figure 3.1, the potential of the Webis-TRC-12 dataset goes beyond the study of text reuse for its own sake: the act of reuse also provides an instrument for observing the importance that users ascribe to individual search results in the process of completing their task, which will help answer Research Question 2 (Measuring Usefulness).

3.1.1 Corpus Construction

Two data sets form the basis for corpus construction, namely (1) a set of topics to write about and (2) a set of web pages to research about a given topic. For the former, we resort to topics used at TREC, specifically to those used at the Web Tracks 2009–2011, and for the latter, we employ the ClueWeb corpus from 2009[1] (and not the "real web in the wild"). The ClueWeb comprises more than one billion documents from ten languages and can be considered a representative cross-section of the real web. It is a widely accepted resource among researchers and has become one of the primary resources to evaluate the retrieval performance of search engines within several TREC tracks. The Webis-TRC-12's strong connection to TREC is deliberate—it will allow for unforeseen synergies. Based on these decisions, the corpus construction steps can be summarized as follows:

1. Rephrasing of the 150 topics used at the TREC Web Tracks 2009–2011 so that they explicitly invite people to write an essay.

[1] http://lemurproject.org/clueweb09

2. Indexing of the ClueWeb corpus category A (the entire English portion with about 0.5 billion documents) using the BM25F retrieval model plus additional features.

3. Development of a search interface that allows for answering queries within milliseconds and that is designed along the lines of commercial search interfaces.

4. Development of a browsing interface for the ClueWeb09, which serves ClueWeb09 pages on demand and which rewrites links on delivered pages so that they point to their corresponding ClueWeb09 pages on our servers.

5. Recruiting 12 professional writers at the crowdsourcing platform oDesk from a wide range of hourly rates for diversity.

6. Instructing the writers to write one essay at a time of at least 5000 words length (corresponding to an average student's homework assignment) about an open topic of their choice, using our search engine—hence browsing only ClueWeb pages.

7. Logging all writers' interactions with the search engine and the ClueWeb on a per-essay basis at our site.

8. Logging all writers' edits to their essays in a fine-grained edit log: a snapshot was taken whenever a writer stopped writing for more than 300ms.

9. Double-checking all of the essays for quality.

After having deployed the search engine and completed various usability tests, the actual corpus construction took nine months—from April 2012 through December 2012—with post-processing of the data taking another four months. Corpus construction proceeded in two batches: in the first run, 147 essays were written by a mix of volunteers and hired writers without use of the search engine; instead, each essay author was provided with a list of candidate source documents for the current topic, which were curated based on relevance judgements made by the assessors in past TREC runs (for three topics, a sufficient number of known-relevant sources was not available). While batch one served the dual purpose of field-testing the text editing infrastructure and collecting additional ground-truth data for text reuse research, the second batch—employing 12 professional writers who wrote 150 essays with access to ChatNoir and the full ClueWeb, but no

curated sources—became what is now known as the Webis-TRC-12 corpus, and is the primary focus of the remainder of this chapter. Before subsequent sections will delve into the research insights that these data have enabled thus far, the remainder of the present section will highlight different elements of the corpus construction set-up in greater detail.

Topic Preparation

Since the topics used at the TREC Web Tracks were not amenable for our purposes as-is, we rephrased them so that they ask for writing an essay instead of searching for facts. Consider for example topic 001 of the TREC Web Track 2009:

> *Query.* obama family tree
>
> *Description.* Find information on President Barack Obama's family history, including genealogy, national origins, places and dates of birth, etc.
>
> *Sub-topic 1.* Find the TIME magazine photo essay "Barack Obama's Family Tree."
>
> *Sub-topic 2.* Where did Barack Obama's parents and grandparents come from?
>
> *Sub-topic 3.* Find biographical information on Barack Obama's mother.

This topic is rephrased as follows:

> *Obama's family*. Write about President Barack Obama's family history, including genealogy, national origins, places and dates of birth, etc. Where did Barack Obama's parents and grandparents come from? Also include a brief biography of Obama's mother.

In the example, Sub-topic 1 is considered too specific for our purposes while the other sub-topics are retained. TREC Web Track topics divide into faceted and ambiguous topics. While topics of the first kind can be directly rephrased into essay topics, from topics of the second kind one of the available interpretations was chosen.

A Controlled Web Search Environment

To give the crowdsourcing workers a familiar search experience while maintaining reproducibility at the same time, they were instructed to use

the ChatNoir search engine [155], which indexes the ClueWeb09, while providing a user interface which follows industry standards, and an API that allows for user tracking. ChatNoir is based on the BM25F retrieval model [167], uses the anchor text list provided by [92], the PageRanks provided by the Carnegie Mellon University alongside the ClueWeb corpus, and the Spam rank list provided by [53]. ChatNoir comes with a proximity feature with variable-width buckets as described by [68]. ChatNoir's choice of retrieval model and ranking features is intended to provide a reasonable baseline performance. However, it is neither nearly as mature as those of commercial search engines nor does it compete with the best-performing models from TREC. Yet, it is among the most widely accepted models in information retrieval, which underlines the goal of reproducibility, and may be advantageous in other ways, since writers had to engage with the search engine to find sufficient material to write an essay of the aforementioned length.

When the user clicks on a search result, ChatNoir does not link into the real web but redirects into the ClueWeb. Though the ClueWeb provides the original URLs from which the web pages have been obtained, many of these pages have gone or been updated since. ChatNoir hence provides an API that serves web pages from the ClueWeb on demand: when accessing a web page, it is pre-processed before being shipped, removing automatic referrers and replacing all links to the real web with links to their counterpart inside the ClueWeb. This way, the ClueWeb can be browsed as if surfing the real web, while it becomes possible to track the user.

Crowdsourcing Writers

Our ideal writer has experience in writing, is capable of writing about a diversity of topics, can complete a text in a timely manner, possesses decent English writing skills, and is well-versed in using the aforementioned technologies. After bootstrapping our setup with 10 volunteers recruited at our university, it became clear that, because of the workload involved, accomplishing our goals would not be possible with volunteers only. Therefore, we resorted to hiring (semi-)professional writers and made use of the crowdsourcing platform oDesk.[2] Crowdsourcing has quickly become one of the cornerstones for constructing evaluation corpora, which is especially true for paid crowdsourcing. Compared to Amazon's Mechanical Turk [17], which is used more frequently than oDesk, there are virtually no workers at oDesk submitting fake results because of its advanced rating fea-

[2] `http://www.odesk.com`, nowadays `http://upwork.com`

Table 3.1: Demographics of the 12 Batch 2 writers.

Writer Demographics					
Age		*Gender*		*Native language(s)*	
Minimum	24	Female	67%	English	67%
Median	37	Male	33%	Filipino	25%
Maximum	65			Hindi	17%
Academic degree		*Country of origin*		*Second language(s)*	
Postgraduate	41%	UK	25%	English	33%
Undergraduate	25%	Philippines	25%	French	17%
None	17%	USA	17%	Afrikaans, Dutch,	
n/a	17%	India	17%	German, Spanish,	
		Australia	8%	Swedish each	8%
		South Africa	8%	None	8%
Years of writing		*Search engines used*		*Search frequency*	
Minimum	2	Google	92%	Daily	83%
Median	8	Bing	33%	Weekly	8%
Standard dev.	6	Yahoo	25%	n/a	8%
Maximum	20	Others	8%		

tures for workers and employers. Moreover, oDesk tracks their workers by randomly taking screenshots, which are provided to employers in order to check whether the hours logged correspond to work-related activity. This allowed us to check whether our writers used our environment instead of other search engines or text editors.

Table 3.1 gives an overview of the demographics of the twelve writers hired via crowdsourcing, based on a questionnaire and their resumes at oDesk. Most of them come from an English-speaking country, and almost all of them speak more than one language, which suggests a reasonably good education. Two thirds of the writers are female, and all of them have years of writing experience. Hourly wages were negotiated individually and range from 3 to 34 US dollars (dependent on skill and country of residence), with an average of about 12 US dollars. For ethical reasons, we payed at least the minimum wage of the respective countries involved. In total, we spent 20 468 US dollars to pay the writers—an amount that may be considered large compared to other scientific crowdsourcing efforts from the literature, but small in terms of the potential of crowdsourcing to make a difference in empirical science.

Table 3.2: Key figures of searching and writing in the Webis-TRC-12.

	Min	Q1	Mdn	Avg	Q3	Max	Sum
Queries							
– per essay	4.0	40.0	68.0	90.7	117.0	612.0	13,609
– per essay (unique)	1.0	12.0	20.0	23.6	31.5	121.0	3,538
– per physical session	0.0	0.0	0.0	4.9	4.0	231.0	13,609*
Clicks							
– per essay	12.0	55.0	87.0	111.3	144.5	431.0	16,698
– per essay (unique)	8.0	44.5	67.0	74.5	101.0	259.0	11,181
– per physical session	0.0	0.0	1.0	6.0	6.0	164.0	16,698*
– per query	0.0	0.0	0.0	2.3	0.0	76.0	8,779
Clicks per essay							
– on results	5.0	30.5	49.0	58.5	75.5	280.0	8,779*
– trail clicks	0.0	13.5	33.0	52.8	73.0	332.0	7,919
Writing sessions							
– per essay	11.0	28.0	42.0	46.3	59.5	178.0	6,943
– revisions (thousands)	0.2	1.8	2.9	2.9	3.8	6.8	–**
– words (thousands)	0.7	4.8	5.0	5.0	5.2	13.9	–**
– paste events	0.0	13.0	25.0	28.6	39.0	134.0	4,291
– references	3.0	11.0	16.0	18.4	21.0	69.0	2,761
Work time per essay							
– days passed	1.0	4.0	6.0	8.6	9.0	56.0	–**
– working days	1.0	4.0	5.0	5.5	7.0	17.0	–**
– working hours	1.8	5.2	7.5	7.9	9.8	23.0	1,191
– physical sessions	2.0	11.5	16.0	18.6	23.0	55.0	2,797
Minutes spent							
– reading per click	0.0	0.1	0.4	0.7	0.8	15.0	11,236
– writing per session	0.0	0.5	2.2	7.4	8.9	145.2	51,126

*Equal to some above value by definition.
**Sum not given to avoid misinterpretation.

3.1.2 Basic Corpus Statistics

Table 3.2 shows key statistics of the interaction logs, including absolute numbers for queries, clicks, working times, number of edits, words, and retrieved sources, as well as their ratios to essays, writers, and work time, where applicable. The average writer wrote 2 essays with a standard deviation of 15.9; one especially prolific writer managed to write 33 essays.

From a total of 13 609 queries submitted by the writers, each essay got an average of 91 queries. The average number of results clicked per query is 2.3. For comparison, we computed the average number of clicks per query

in the AOL query log [151], which is 2.0. In this regard, the behavior of our writers on individual queries does not differ much from that of the average AOL user in 2006. Most of the clicks that we recorded are search result clicks, whereas 7 919 of them are browsing clicks on web page links. Among the browsing clicks, 11.3% are clicks on links that point to the same web page (i.e., anchor links using the hash part of a URL). The longest click trail contains 51 unique web pages, but most trails are very short. This is a surprising result, since we expected a larger proportion of browsing clicks, but it also shows that our writers relied heavily on ChatNoir's ranking.

The query log of each writer divides into 931 search sessions with an average of 12.3 sessions per topic. Here, a session is defined as a sequence of queries recorded for a given topic which is not divided by a break longer than 30 minutes. Despite other claims in the literature [80, 100] we argue that, in our case, sessions can be reliably identified by timeouts because we have a priori knowledge about which query belongs to which essay. Typically, completing an essay took 6 days, which includes to a long-lasting exploration of the topic at hand.

The essays were written with a total of 424 017 edits. On average, writers needed 46 writing sessions to complete an essay, spread across 18 physical sessions. Here, we define a writing session as an uninterrupted span of time spent editing the text, whereas a physical session is an uninterrupted stretch of working, overall, and may comprise multiple writing sessions, interrupted e.g. by query submission and result examination.[3] Over the course of its inception, each essay was edited 2 826 times on average, and the standard deviation gives an idea about how diverse the modifications of the reused text were. Writers were not specifically instructed to modify the text as much as possible—rather they were encouraged to paraphrase in order to avoid detection by an automatic text reuse detector. This way, our corpus captures each writer's idea of the necessary modification effort to accomplish this goal. The average length of the essays is 5 006 words, but there are also some short essays if hardly any useful information could be found on the respective topics. About 18 sources have been reused in each essay, but some writers reused text from as many as 69 unique documents.

[3] In the visual language of Figure 3.5, each horizontal bar represents a physical session, and the beige boxes within the constituent writing sessions

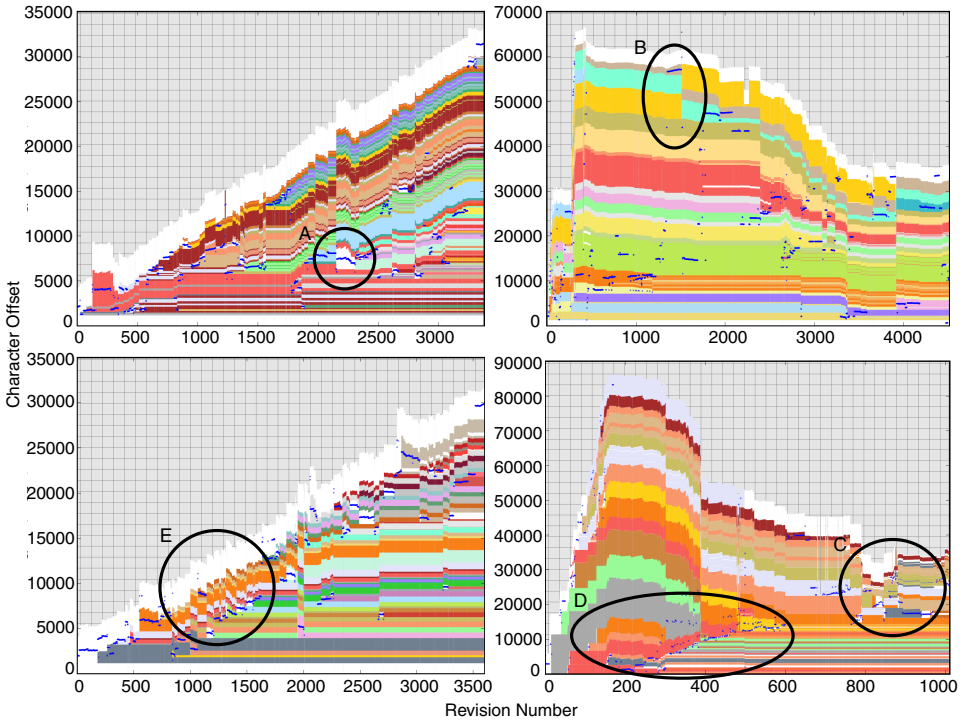


Figure 3.2: Types of text reuse: build-up reuse (left) versus boil-down reuse (right). Each plot shows the essay length in characters along the vertical axis, and the passage of time between first keystroke and essay completion along the horizontal; time is measured in revisions, wherein a new version was recorded whenever a writer stopped for more than 300ms, and longer breaks are collapsed. Colors encode different source documents. Original text is white; blue dots indicate the text position of the writer's most recent edit at that moment.

3.2 Insights into Writing Behavior

In pursuit of Research Question 1a (Writing Strategies) as stated in Chapter 1, we analyse the Webis-TRC-12 data with respect to the strategies that writers employ in the pursuit of complex, knowledge-intensive tasks like the one studied. To this end, the current section focuses specifically on the insights into writing behavior, and the specialized analysis tools developed to reveal them. The exploratory analysis of the writing log yields a series of other interesting and useful insights, which are presented opportunistically along the way.

3.2.1 Visualizing Edit Histories

To analyze the writers' writing style, that is to say, how writers reuse text and how the essay is completed, we recorded edit logs of their essays: whenever writing stopped for more than 300ms, a new edit was stored in a ver-

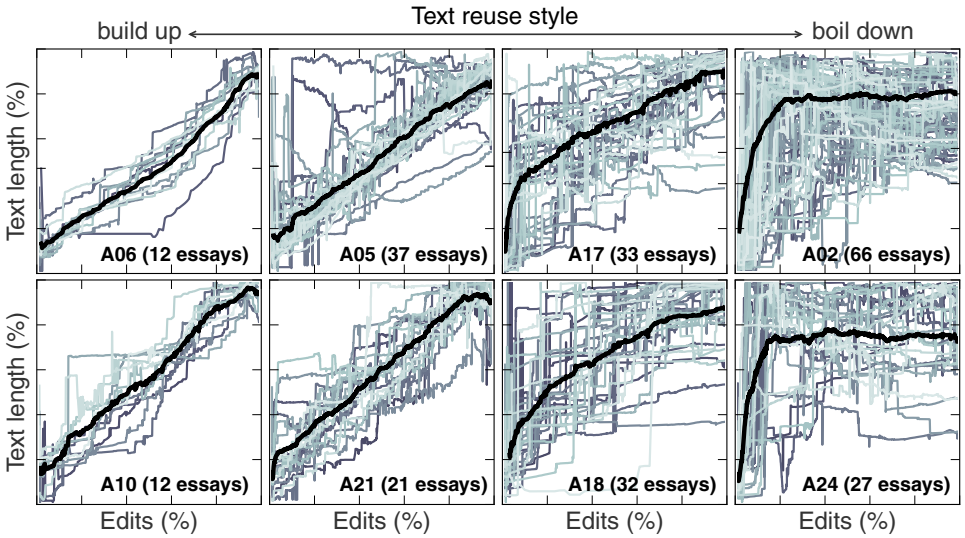


Figure 3.3: Text reuse styles ranging from build-up reuse (left) to boil-down reuse (right). A gray curve shows the normalized length of an essay over the edits that went into it during writing. Curves are grouped by writers. The black curve marks the average of all other curves in a plot.

sion control system at our site. The edit logs document the entire evolution of the text from first keystroke to essay completion. We have adapted the so-called history flow visualization to analyze the writing process [198]; Figure 3.2 shows four examples. Based on these visualizations, a number of observations can be made. In general, we identify two distinct writing-style types to perform text reuse, namely to *build up* an essay during writing, or, to first gather material and then to *boil down* a text until the essay is completed. Later in this section, we will analyze this observation in greater detail. Within the plots, a number of events can be spotted that occurred during writing: in the top left plot, encircled as area A, the insertion of a new piece of text can be observed. Though marked as original text at first, the writer worked on this passage and then revealed that it was reused from another source. At area B in the top right plot, one can observe the reorganization of two passages as they exchange places from one edit to another. Area C in the bottom right plot shows that the writer, shortly before completing this essay, reorganized substantial parts. Area D in the same plot shows how the writer went about boiling down the text by incorporating contents from different passages collected beforehand and, then, from one edit to another, discarded most of the rest. The saw-tooth shaped pattern in area E in the bottom left plot reveals that, even though the writer of this essay adopts a build-up style, she still pastes passages from her sources into the text one at a time, and then individually boils down each. Our visual-

Table 3.3: Contingency table: writers over reuse style.

Reuse	Writer ID										
Style	A02	A05	A06	A07	A10	A17	A18	A19	A20	A21	A24
build-up	4	27	11	4	9	13	12	4	9	18	2
mixed	10	3	0	1	1	7	6	0	0	3	1
boil-down	52	5	0	14	2	13	11	3	0	0	24

izations also show the text positions where writers modified the text as blue dots; in this regard, distinct writing patterns emerge where some writers go through a text linearly, and others do not.

3.2.2 Writing Strategies: Build-up Reuse versus Boil-down Reuse

Based on the edit history visualizations, we have manually classified the 297 essays of both batches into two categories, corresponding to the two styles build-up reuse and boil-down reuse. We found that 40% are instances of build-up reuse, 45% are instances of boil-down reuse, and 13% fall in between, excluding 2% of the essays as outliers due to errors or for being too short. The in-between cases show that a writer actually started one way and then switched to the respective other style of reuse so that the resulting essays could not be attributed to a single category. An important question that arises out of this observation is whether different writers habitually exert different reuse styles or whether they apply them at random. To obtain a better overview, we envision the applied reuse style of an essay by the skyline curve of its edit history visualization (i.e., by the curve that plots the length of an essay after each edit). Aggregating these curves on a per-writer basis reveals distinct patterns. For eight of our writers Figure 3.3 shows this characteristic. The plots are ordered by the shape of the averaged curve, starting from a linear increase (left) to a compound of steep increase to a certain length after which the curve levels out (right). The former shape corresponds to writers who typically apply build-up reuse, while the latter can be attributed to writers who typically apply boil-down reuse.

When comparing the plots we notice a very interesting effect: it appears that writers who conduct boil-down reuse vary more strongly in their behavior. The reuse style in some essays, however, falls in between the two extremes. Besides the visual analysis, Table 3.3 shows the distribution of reuse styles for the eleven writers who contributed at least five essays. Most writers use one style for about 80% of their essays, whereas two writers (A17, A18) are exactly on par between the two styles. Based on Pearson's chi-

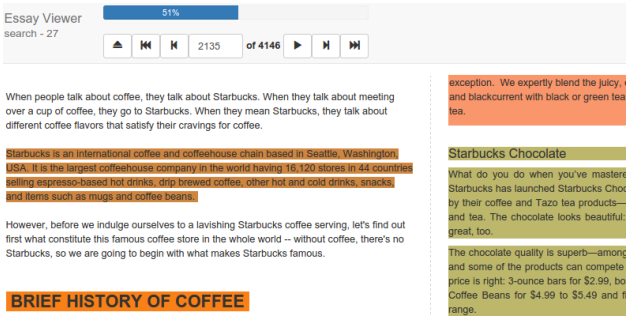


Figure 3.4: Screenshot of the essay viewer interface.

squared test, one can safely reject the null hypothesis that writers and text reuse styles are independent: $\chi^2 = 139.0$ with $p = 7.8 \cdot 10^{-20}$. Since our sample of authors and essays is sparse, Pearson's chi-squared test may not be perfectly suited which is why we have also applied Fisher's exact test, which yields a probability of $p = 0.0005$ under the independence hypothesis.

3.2.3 Interactively Inspecting Writing Behavior

In order to better understand the essay writing process, we implemented a web application that shows all revisions of a given essay in sequence. A screenshot of the interface is shown in Figure 3.4: The controls at the top of the page allow stepping forward and backward through the essay revisions, or jumping to a specific revision. The rest of the page shows the current state of the essay. Different colors indicate different ClueWeb09 sources for copied or paraphrased text. We envision extending this tool to include information from the query log, such that queries occurring at a given moment can be correlated with their contemporary writing interactions. The tool is available alongside the corpus.

3.2.4 Jointly Visualizing Writing and Searching

Consequently, we have developed a combined visualization to cover the entirety of the activity associated with a given essay, both in terms of text writing, and interactions with the search engine and source documents, to allow examining at a glance the full temporal course of actions that the authors took during their essay writing task. To this end, the physical working sessions are determined based on a 15 minutes inactivity gap. However, only the text-writing interactions have an exactly known end time; for query and click interactions, we estimate the durations. For queries, we apply a threshold of 60 seconds, because we assume that a writer would not stare on the

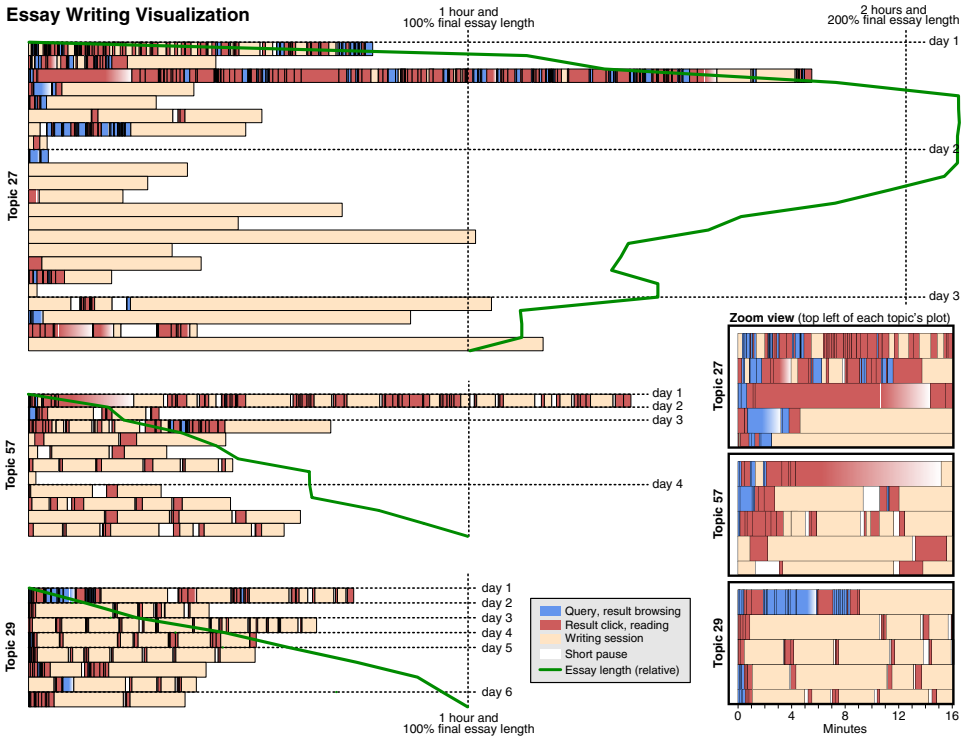


Figure 3.5: Visualization of the interactions for a selection of three of the 150 essays. Each stacked bar denotes an uninterrupted working session (15 minutes inactivity gap). Bar lengths indicate work time in minutes, blue boxes indicate querying and result browsing, red boxes indicate document views, and beige boxes indicate writing. White boxes and gradients in red or blue boxes indicate short pauses. The solid green line denotes the current essay length relative to the final essay.

result list for more than one minute without clicking any result. For clicked documents, we estimate the reading time based on the document length and an assumed reading speed of 250 words per minute [59]. A solid (green) line further shows the development of essay length in the sessions. Figure 3.5 shows examples of three topics. Each row depicts a physical session, and the horizontal dashed lines divide different working days (most sessions were about one hour long). The beige blocks represent text-writing interactions, and the blue and red ones depict queries and document views, respectively.

The author of the essay on topic 29 submitted few queries, but seems to have worked very purposefully: writing often directly follows document views; the author seems to deliberately learn and write about some particular aspect, visiting a few targeted documents in order to collect the needed information. The author of the essay on topic 27 has a very different working style, in that she starts with a couple of sessions foraging all possibly needed information; almost all sessions from the third working day onward deal

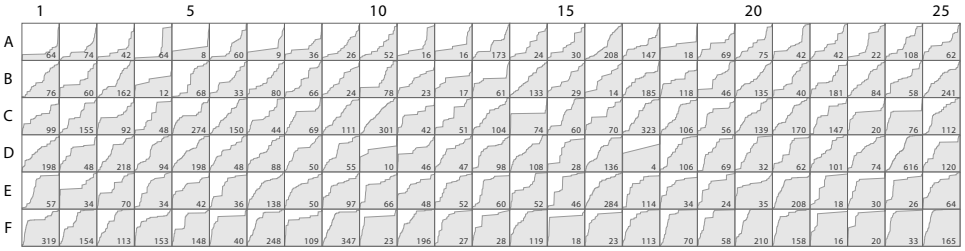


Figure 3.6: Spectrum of writer search behavior. Each grid cell corresponds to one of the 150 essays and shows a curve of the percentage of submitted queries (y-axis) at times between the first query until the essay was finished (x-axis). The numbers denote the amount of queries submitted. The cells are sorted by area under the curve, from the smallest area in cell A1 to the largest area in cell F25.

with rewriting and removing content from previously collected sources. Following the nomenclature from earlier in this section, the first author is a "build-up" writer, and the second a "boil-down" one. Section 3.3.2 will relate this to differences in searching behavior.

There is another interesting detail about the essay on topic 27: In the session before the last, a couple of document views are followed by very short writing interactions that influence the essay length only marginally. This behavior can be observed for many topics and different authors, and was also reported by Vakkari [192] previously. One possible explanation could be writers checking their essay for possible missing but important text passages from previously selected sources, or that they double-check their facts while finalizing their essays.

3.3 Insights into Search Behavior

Similarly to the preceding analysis into the writing behavior, and in pursuit of Research Question 1b (Searching Strategies), we are interested in identifying specific searching strategies that writers employed in the setting of the task they were given.

As outlined previously, we attempted to shift the attention of our writers toward searching for information, rather than spending time pondering over formulations, by allowing them to reuse in their essays the texts they found in the ClueWeb09. Nevertheless, the final essays were still required to be coherent and consistent—often resulting in reformulations of copy-pasted texts. To analyze the writers' search behavior during essay writing, we recorded detailed search logs of their queries while they used our search engine. Figure 3.6 shows for each of the 150 essays a curve of the percentage of queries at times between a writer's first query and an essay's completion.

Table 3.4: Origins of learned query terms.

Prior	Task	Search Results		
knowledge	**description**	Title	Snippet	Clicked doc.
312 (8.4%)	902 (24.3%)	291 (7.8%)	1,067 (28.7%)	1,147 (30.8%)

We have normalized the time axis and excluded working breaks of more than five minutes. The curves are organized so as to highlight the spectrum of different search behaviors we have observed: in row A, 70-90% of the queries are submitted toward the end of the writing task, whereas in row F almost all queries are submitted at the beginning. In between, however, sets of queries are often submitted in the form of "bursts," followed by extended periods of writing, which can be inferred from the steps in the curves (e.g., cell C12). Only in some cases (e.g., cell C10) a linear increase of queries over time can be observed for a non-trivial amount of queries, which indicates continuous switching between searching and writing. From these observations, it can be inferred that our writers sometimes conducted a "first fit" search and reused the first texts they found easily. However, as the essay progressed and the low hanging fruit in terms of search results were used up, they had to search more thoroughly in order to complete their essay. More generally, this data gives an idea of how humans perform exploratory search in order to learn about a given topic. Our current research on this aspect focuses on the prediction of search mission types, since we observe that the search mission type does not simply depend on the writer or the perceived topic difficulty.

3.3.1 Query Formulation

Over the course of exploratory tasks, searchers learn and extend or adapt their knowledge about a topic [66]. We expect the queries for an essay to also develop over time and examine when in the process specific terms occur and where they might stem from. For each query term entered for the first time, we assign it to one of the possible origins: the task description, a previously clicked document, the title or snippet of a previously shown search result, or the writer's initial knowledge. If a term has not occurred during any of the prior interactions, it is classified as *prior knowledge* only.

Following the scheme outlined above, Table 3.4 shows the origins of all 3,719 distinct query terms that appeared in the queries for the 150 topics: almost all terms could potentially have been learned while work on the topic.

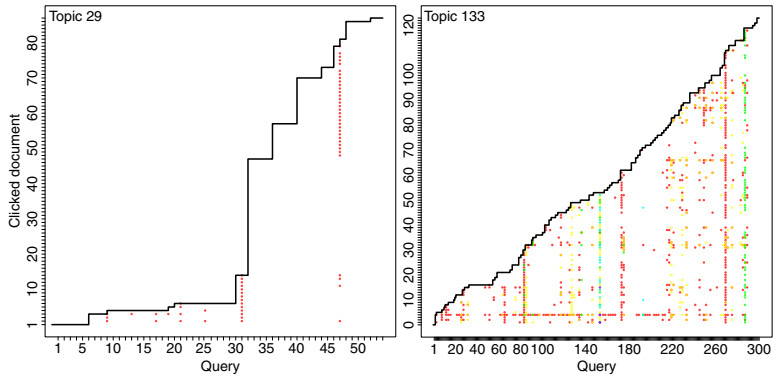


Figure 3.7: Potentially learned terms for topic 29 and 133: the line indicates which documents were visited as a result of which query; dots indicate which previously visited documents contain terms used in all queries.

Figure 3.7 shows when in the search process a writer introduced new terms and where they are likely to come from for topic 29 and 133. On the x-axis, one can see the current query number. The y-axis displays all clicked documents, and the staircase-shaped line depicts which click(s) happened as a result of which query. For topic 29, the first three clicks happened for the sixth query, another click followed after submitting the ninth query, and so on. The dots indicate a new term in the query and all previously clicked documents that contain this particular term. If two or even more terms were introduced in only one query, each of the terms is represented by another color. For instance, in topic 29 the queries 31 and 47 introduce the terms "games" and "reviews." These terms were contained in almost all of the clicked documents that were visited before the respective queries. Such a vertical line of dots can be interpreted as a change of subtopic because the writer ignored an often occurring term for quite a long time and then decided at some point to finally search for it. About 70 of the topics contain such clear subtopic changes based on recently visited documents. Topic 133 also shows another interesting pattern: a horizontal line (Figure 3.7). This indicates that document number 4 was influential for many queries (it is a detailed overview on the Declaration of Independence, the main theme of the topic).

Figure 3.8 visualizes what terms were used in which queries for two example essays, with the terms on the y-axis and the queries, in the sequence they were submitted, on the x-axis. From the two examples shown, it is obvious that many queries have numerous identical, immediate follow-up queries. Many instances of this can be explained through clicks on the search engine's "more"-button requesting 100 instead of 10 results; queries

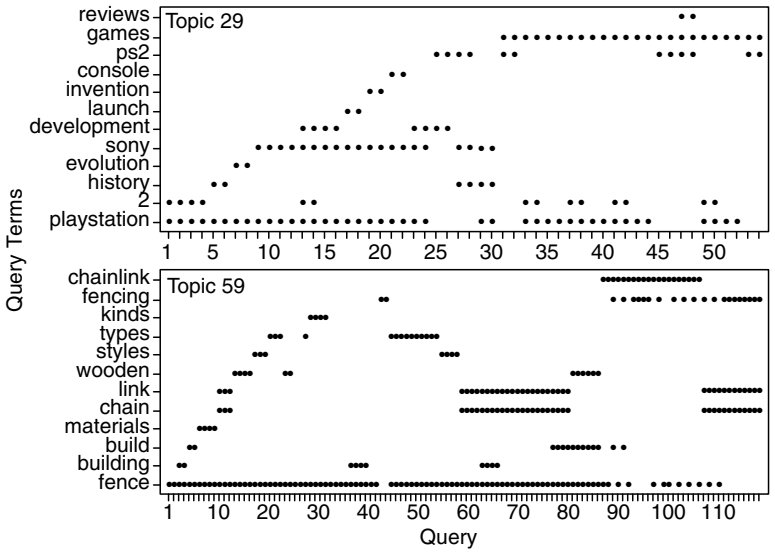


Figure 3.8: Query composition for topics 29 and 59: for all distinct query terms used by the writer, the dots indicate which terms occurred in which query.

are often resubmitted more than twice when there was a session break in between and the writer resumed work with the most recent query. However, there are also some odd cases like the query `[chain link fence]` that is submitted ten times in a row for topic 59 (queries 67 to 76 in topic 59). We have no satisfying explanation for this behavior; maybe the search engine was slow at this time such that the writer submitted the query again before having seen any result.

We consider identical queries that are re-submitted from time to time to be *anchor* queries, which we conjecture to support the author in her task in various ways: First, the results of such a query can point to many directions for further investigations and a writer might return to this query as soon as the work on one subtopic is finished. Second, anchor queries can serve to keep track of the main theme at any time and keep the writer on course. And third, writers might bring recently acquired knowledge into line with older knowledge structures and therefore want to return to previously seen documents. Typical anchor queries for many topics reflect the main theme of the task (i.e., the TREC topic itself).

3.3.2 Search Strategies: Clickers versus Queriers

We now focus on elementary differences in writers' searching strategies. Figure 3.9 shows the extreme cases of submitted queries over essay revisions for four authors (axes normalized to percentages). The curves are or-

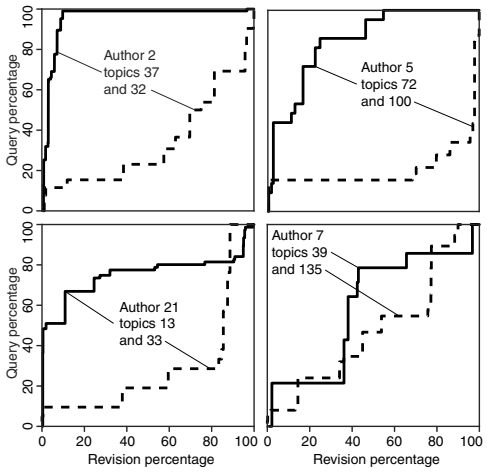


Figure 3.9: Examples for the spectrum of writer search behavior. Each curve shows the percentage of submitted queries (y-axis) per percentage of essay revisions (x-axis). For each author, we show the topics with the largest and smallest area under the curve (i.e., early queries vs. late queries).

ganized to highlight the spectrum of different search behavior for individual authors. Authors 2, 5, and 21, for instance, have topics for which they submit most of the queries rather early, but also topics with most queries at the end only (i.e., probably fact checking). Typically, sets of queries are submitted in short "bursts," followed by extended periods of writing, which can be inferred from the plateaus in the curves. For author 7, all the topics show a more linear increase of queries over the whole writing time for all topics, indicating continuous switching between searching and writing. From these observations, it can be inferred that query frequency alone is not a good indicator of task completion or the current stage of a task, even considering only a single author. Moreover, exploratory search systems have to deal with a broad behavior spectrum and be able to make the most of few queries, or be prepared that writers interact only a few times with them.

To further distinguish search behavior, we focus on the number of queries and clicks. As observed in Section 3.1.2, some authors submit only few queries but follow long click trails; others submit a variety of queries but rarely click on search results. We call the authors following the two strategies *clickers* and *queriers*. To distinguish between clickers and queriers, we count the number of queries and clicks that are performed until a document is clicked that is also used as a source in the essay. That is, we disregard how many queries and clicks occurred overall, but only consider how many of them occur between two clicks on such reference documents. The analysis for all essays reveals the two groups among our authors. Au-

Table 3.5: Behavioral differences of clickers and queriers. Shown are the respective median counts for each group, followed by the results of a Mann-Whitney U-test.

	Clickers	Queriers	Significance of difference
Queries	47.0	107.0	$U = 1058.5,\ z = -6.148,\ p < 0.01$
Clicks	102.5	79.5	$U = 2074.0,\ z = -2.136,\ p < 0.05$
Pastes	39.0	19.0	$U = 1361.5,\ z = -4.952,\ p < 0.01$
References	17.5	15.0	$U = 1876.0,\ z = -2.921,\ p < 0.01$

thors 5, 7, 20, 21 and 04 are clickers, and the authors 2, 6, 17 and 18 are queriers. Authors 1, 14 and 25 have worked on at most two topics only, yet the trend shows that they tend to be clickers.

Table 3.5 highlights the differences between clickers and queriers. Except for the number of clicks, which is also fairly high for queriers, all differences between both groups are highly significant as shown by a Mann-Whitney U-test (the data are not normally distributed). The fairly high number of clicks in the querier group simply seems to depend on the number of queries submitted. After all, the distributions of clicks for both groups differ not as much as the distributions of queries, pastes and references. This underpins the assumption that writers in exploratory search tasks consume some informative content before considering themselves to have learned enough. It is notable that clickers paste about twice as often as queriers do. It seems plausible that clickers pick up several possibly useful text passages during their information exploration phase, which they retain in their essays for later use. The number of reused references confirms this trend and it can be stated that queriers seem to be more selective with their reference documents than clickers.

3.3.3 Writer Dedication

Besides different search strategies, we also want to explore whether our data allow us to measure the degree of *writer dedication* to the exploratory search task. We try to reflect writer dedication by the effort a writer puts into the task, which can be valuable information for a search engine. For example, a truly dedicated writer might be interested in additional resources beyond the original query, whereas a writer who works only unwillingly her task might be only interested in overview pages without too many details. Recent studies investigating user engagement [144, 145] go beyond the simple features we can explore below, but we think that our search log-derivable measures can still be useful.

To distinguish "lazy" from more dedicated writers, we use the following nine features per essay: number of distinct queries, number of distinct clicks, number of copy-paste interactions, number of used references, total working hours, time spent reading documents, time spent writing, number of physical sessions, number of handled subtopics (determined by the number of session IDs a search mission detection algorithm returned [80]). In a next step, a ranking of all topics is produced for each feature individually, and each essay gets a score depending on its rank. For example, the essay on topic 133 contains the highest number of distinct queries and thus obtains 121 points (it is not 150 because 29 essays share the same number of distinct queries and obtain the same score). For the feature "distinct clicks," the essay on topic 133 is only on rank 18 and obtains 77 points. This is done for all features and the scores are summed up per essay; the resulting ranking is shown in Table 3.6. Remarkably, nine of the top-10 essays were written by author 2, who seems to have worked with high dedication on many essays, whereas authors 6 and 24 seem to have worked with little enthusiasm—even though authors always got to pick their preferred topic from those still available when starting a new essay.

To identify the most and least dedicated writers, we simply compute the average for each writer to work around the different numbers of treated topics per writer. It turns out that author 2 indeed belongs to the most dedicated writers with an average score of 403.5 but is slightly outperformed by author 21 with an average of 404.8. Note that author 2 wrote 33 different essays and the range of scores is varied, whereas author 21 worked only 12 topics, which all achieved quite high dedication scores. The least dedicated writers in our collection are author 6 and author 20 with an average score of 141.9 and 188.6, respectively. Note that the dedication ranking does not imply any conclusions on the quality of the essay itself but only about the relative effort that the authors invested. The quality of the essay has to be determined in a separate step—an idea could be to run the essays through text reuse detection software and assign higher quality scores to essays from which the ClueWeb09 sources cannot be readily detected anymore, similar to the source-based writing analyses of Sormunen et al. [177]. This could then also be used to confirm previous findings on how effort correlates with the task outcome [193].

3.3.4 Searching and Writing Styles

Section 3.2 identified two different writing styles: build-up and boil-down [157]. The first characterized by a rather continuous lengthening of

Table 3.6: Essays with topic (T) and author IDs (A) ranked (R) by the writer dedication score (S).

R	T	A	S	R	T	A	S	R	T	A	S
1	58	2	551	51	150	24	334	101	73	24	201
2	53	2	538	52	138	2	331	102	81	17	200
3	110	2	524	53	57	5	330	103	24	14	196
4	13	21	523	54	36	5	326	104	100	5	196
5	67	2	503	55	48	18	323	105	66	20	194
6	27	2	499	56	50	2	320	106	102	24	194
7	49	2	498	57	117	2	320	107	69	24	191
8	144	2	493	58	55	21	319	108	126	6	191
9	10	2	484	59	137	5	317	109	14	5	189
10	22	2	479	60	65	17	314	110	40	17	188
11	133	17	476	61	47	2	313	111	15	20	186
12	80	2	470	62	1	17	311	112	94	17	184
13	88	2	469	63	63	5	311	113	90	18	178
14	51	2	468	64	107	17	308	114	95	5	178
15	139	5	467	65	25	17	304	115	83	18	173
16	45	21	466	66	92	18	304	116	4	18	170
17	37	2	455	67	115	5	301	117	103	20	169
18	71	21	448	68	12	5	298	118	20	5	168
19	127	2	448	69	39	7	296	119	140	18	165
20	86	21	446	70	105	7	295	120	85	17	163
21	42	17	444	71	64	2	291	121	34	18	162
22	8	2	441	72	75	2	289	122	46	7	159
23	120	21	430	73	99	7	285	123	16	18	155
24	141	2	422	74	109	7	282	124	148	20	155
25	106	21	417	75	125	21	279	125	72	5	152
26	17	2	414	76	60	18	276	126	101	24	152
27	82	2	414	77	145	17	273	127	104	7	150
28	98	21	406	78	19	20	267	128	9	17	149
29	87	17	404	79	54	6	263	129	142	20	147
30	11	24	403	80	30	2	262	130	136	7	139
31	114	5	399	81	41	7	252	131	61	18	135
32	59	2	394	82	77	5	252	132	129	6	131
33	76	21	394	83	35	5	248	133	123	1	127
34	5	17	393	84	118	25	248	134	84	18	126
35	70	20	392	85	6	17	247	135	132	24	126
36	74	2	389	86	29	5	246	136	91	20	125
37	96	18	383	87	121	17	246	137	113	5	125
38	119	2	378	88	131	7	243	138	112	18	122
39	135	21	376	89	78	5	235	139	130	24	117
40	31	2	375	90	149	17	235	140	38	18	116
41	26	1	374	91	62	17	233	141	89	7	115
42	128	2	372	92	122	2	233	142	32	2	113
43	18	2	366	93	97	6	226	143	3	24	111
44	2	17	357	94	56	18	220	144	124	18	104
45	7	7	355	95	79	24	218	145	23	24	89
46	44	18	355	96	28	18	216	146	147	6	74
47	33	21	354	97	143	17	213	147	116	6	63
48	93	17	344	98	52	18	207	148	43	20	62
49	108	17	342	99	134	17	205	149	146	6	45
50	68	24	336	100	111	18	202	150	21	24	40

the essay over the whole period of writing, and the second style by a first quick length growth and subsequent reorganization and shortening. The essay on topic 27 reflects a typical boil-down writing while the essays on the topics 29 and 57 are build-up essays (cf. Figure 3.5). We now compare the writing style (as characterized by essay length growth) to the search and copy-pasting behavior. The hypothesis is that in build-up essays text passages are copy-pasted in rather regular intervals (and almost immediately adapted to fit the essay structure) while in boil-down essays most of the background research is hypothesized to happen at the beginning and thus most copy-paste interactions are to be expected at the beginning of working on a task.

As a simple measure to quantify the search behavior, we use the regularity of copy-paste events over the course of the writing process. One could argue that queries are a better search behavior measure but with the copy-paste events we focus on the search and web interactions that actually lead to some change in the essay. As for the regularity, we count the number of revisions between each consecutive pair of copy-paste events and compute the observed variance. For example, a 50-revisions essay with paste events in the revisions 10, 22 and 40, would result in the list $\langle 10, 12, 18, 10 \rangle$ (also containing the revisions prior to the first and after the last paste). A low variance in this list means that the paste events are rather equally distributed over the essay revisions, whereas a high variance indicates that a writer pasted very irregularly.

As a measure for the development of essay length, we count the number of revisions to the essay that increase its word count, and those that decrease it. Note that for simplicity, we do not consider the number of words added (or removed); only the trend matters (i.e., how many revisions lengthen the essay vs. how many shorten it). In an example 50-revision essay, this might result in 20 revisions in which content was removed and 30 in which content was added. The essay thus tends to grow, as 60% of the revisions lead to a longer essay. Yet, naturally each of the essays has to grow in total to reach an average of 5,000-words in length. Therefore, a low value like 60% rather is an indication of a boil-down writing style.

Figure 3.10 shows the plot resulting from the essay length development and the paste regularity for each topic. Different symbols (and colors) indicate different authors, thus revealing trends for each author's writing style. The x-axis ranges from about 50% to almost 100%; essays more to the right are from the less dedicated writers that hardly ever rephrased something they copy-pasted. The two authors 6 and 20 who are isolated from all other authors by reaching an essay growth of $\geq$85% also are the least dedicated

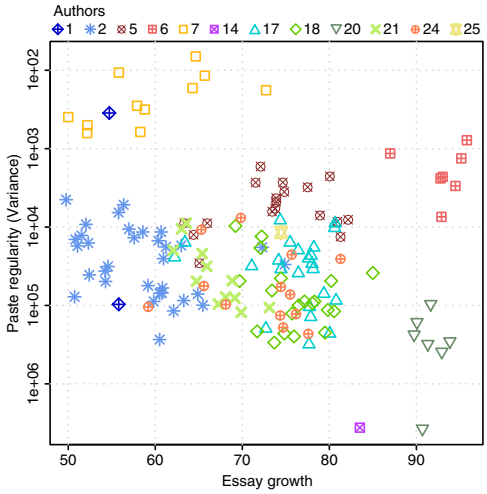


Figure 3.10: Authors' searching and writing styles in the form of essay growth (x-axis, percentage of revisions of an essay that lengthen it) vs. the regularity of copy-pasting content from search results (log-scaled y-axis, variance of the copy-paste revision number differences, low variance = high regularity). Each essay is a data point; essays from the same author typically have similar characteristics.

authors in Section 3.3.3. Many essays showing a build-up pattern in our earlier observations range from 70% to 85% in essay growth, while most essays with a growth below 70%, here especially those by author 2, are those that show the boil-down pattern. Yet, as can be seen on the y-axis, even a boil-down pattern might come with rather regular paste events (low variance with high regularity is on the top of the y-axis) meaning that some authors boiled down individual fragments rather than all useful passages at once. Interestingly, different authors' essays form clusters in our plot contrasting search behavior with the writing progress (copy-paste regularity vs. essay growth). Knowing to which category a writer belongs can help the search engine to better tailor its results. For instance, later follow-up queries are likely for build-up writers. The search engine could take some time while the author is writing to already prepare appropriate results in a "slow search" fashion (cf. Teevan et al. [188]).

3.3.5 Comparison of Working Phases

Finally, we investigate whether the authors work in distinct phases. Do they submit more queries early? Does writing form the major load at the end? Any patterns may inspire ideas to support writers in their respective working phases. In the beginning, a search engine could present not only results for the submitted query but also suggest shortcut queries [16] that helped

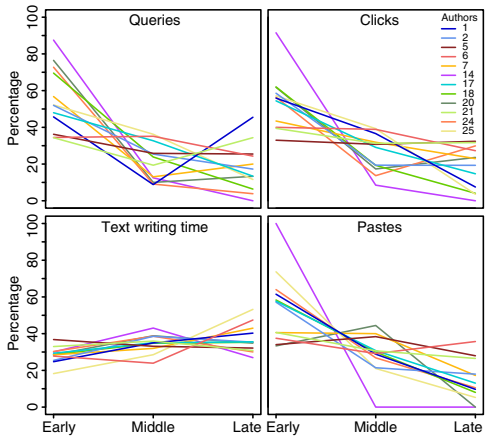


Figure 3.11: Work load in different working phases for all authors.

other users find relevant documents on the same topic. While this is helpful to quickly acquire an overview of different aspects of a topic, it might not be desirable in a later phase in which a writer is interested in specifics or checking facts.

For the sake of simplicity, we subdivide each topic into three working phases—early, middle and late—by splitting up the interactions in the actual working time into three parts of equal duration. For each phase, we measure the percentage of queries, clicks, writing and copy-paste interactions that happened in that phase. For example, if 25 queries out of 50 appeared in the very beginning, the query dimension score is 50% for the early phase etc. For each author and each phase, we take the median value over all their essays to "average" the scores. Figure 3.11 shows the plots for all authors. The general trend is that most queries, clicks and paste interactions happen in the early phases while writing in general seems to happen more in the later phases. This is not too surprising given the fact that most authors wrote essays on topics they were not familiar with and had to search for useful content to first explore the structure of the information space [207]. Still, some authors (and even more essays) show a V-pattern in their query or click load indicating that a large portion of queries also was submitted in the last phase (e.g., authors 1, 7, and 21). Interestingly, these authors did not have a high paste load in the last phase. This indicates that the authors might have checked the essay for possibly missing text passages from previously clicked documents or that they fact-checked some of their content before completion. Interestingly, for most authors the percentages of clicks and pastes over the different phases approximately correlate. At first glance, this might indicate that the authors did not improve their precision

Table 3.7: Confusion matrix of TREC judgments versus writer judgments.

TREC	Writer judgment			
judgment	useless	helpful	useful	unjudged
spam (-2)	3	0	1	2 446
spam (-1)	64	4	18	16 657
irrelevant (0)	219	13	73	33 567
relevant (1)	114	8	91	10 676
relevant (2)	44	5	56	3 711
key (3)	12	0	8	526
unjudged	5 506	221	1 690	–

(i.e., clicks vs. found relevant content in the form of copy-pasting) over the time spent on the topic. However, an in-depth analysis of this issue is left for future work.

3.3.6 Relevance and Usefulness of Sources

A key goal for a search system aiming to better support writers is to surface those search results that are most useful as sources in the writing process. As discussed in Section 2.2.5, usefulness and relevance have been found to not be perfectly correlated; hence, in pursuit of Research Question 2 (Measuring Usefulness), we investigate what we can learn about result document usefulness based on our writers' behavior, and how this relates to any known relevance judgments about the same documents.

Since our writers reused text from web pages they found during their search (and annotated the exact sites and passages that they used), we can directly make use of this information as a usefulness signal. We consider web pages from which text is directly reused as *useful documents* for the respective essay's topic, while web pages that are on a click trail leading to a useful document are termed *helpful*. The unusually high number of useful documents compared to helpful documents is explained by the fact that there are only few click trails of this kind, whereas most web pages have been retrieved directly. The remainder of web pages that were viewed but discarded by our writers are considered as irrelevant. Note that this procedure could be extended to obtain many additional—albeit less reliable—non-usefulness judgments by assuming a cascading user [49] who has considered every search result above the lowest-ranked clicked result.

Each year, the NIST assessors employed for the TREC conference manually review hundreds of web pages that have been retrieved by experimental retrieval systems that are submitted to the various TREC tracks. This was also the case for the TREC Web Tracks from which the topics of the

Webis-TRC-12 corpus are derived. We have compared the relevance judgments provided by TREC for these tracks with the implicit usefulness judgments from our writers. Table 3.7 contrasts the two judgment scales in the form of a confusion matrix. TREC uses a six-point Likert scale ranging from -2 (extreme Spam) to 3 (key document). For 733 of the documents visited by our writers, TREC relevance judgments can be found. From these, 456 documents (62%) have been considered useless by our writers, however, the TREC assessors disagree with this judgment in 170 cases. Regarding the documents considered as useful by our writers, the TREC assessors disagree on 92 of the 247 documents. As noted above, previous work comparing usefulness and relevance assessments has found similar discrepancies. In our particular case, a possible explanation for the disagreement can be found in the differences between the TREC ad hoc search task and our text reuse task: the information nuggets (small chunks of text) that satisfy specific factual information needs from the original TREC topics are not the same as the information "ingots" (big chunks of text) that satisfy our writers' needs.

3.4 Result Usefulness and Retrieval Success

Regardless of its reason, the discrepancy between relevance and usefulness assessments raises the suspicion that retrieval systems optimized against relevance judgments in Cranfield-style experiments will not perform optimally at surfacing useful results for users engaged in complex tasks. As a first step towards better supporting users during writing tasks in particular, the current chapter answers Research Question 3 (Predicting Retrieval Success) through regression models predicting the users' degree of success at retrieving useful sources. It should be noted that our models consider search result usefulness in aggregate—that is, they could be employed in practice to distinguish successful users from struggling ones. The next step—predicting the usefulness of individual documents for use as a ranking signal—remains future work.

As noted in Section 2.2.5, only few studies deal with search result usefulness so far, and they typically deal only with usefulness as perceived subjectively by searchers; simultaneously, usefulness is mostly measured by having searchers fill out questionnaires, rather than quantified implicitly from user behavior. The study described in this section employs the Webis-TRC-12 dataset to lift essay writing into the realm of fine-grained usefulness quantification: the "essay writing with text reuse" task makes the

usefulness of a search result directly observable, as a function of the writers' copy and paste activity (as Figure 3.1 at the beginning of this chapter already illustrates). Analyzing the large corpus of essays with text reuse, we identify two specific usefulness indicators based on text reuse behavior and build linear regression models that predict result usefulness based them. Keeping the limitations of our approach in mind, we believe that these results offer promising new directions for the development of search systems that support writing tasks at large.

3.4.1 Experimental Design

As noted, we base our investigation on the Webis-TRC-12 dataset described earlier in this chapter, deriving two notions of retrieval success, as expressed by the usefulness of the search results the 12 writers manage to retrieve while writing. From the same data, we also derive a set of quantitative measures from the writers' recorded searching and writing behavior while working on the 150 essays. The usefulness of search results forms the dependent variable, and the writer behavior the independent variables for the regression models discussed in Section 3.4.2.

Operationalizing the Usefulness of Search Results

We limit our conception of usefulness to cover only information usage that directly contributes to the task outcome in form of the essay text, and exclude more difficult to measure "indirect" information usage from our consideration, such as learning better query terms from seen search results.

Usefulness implies that information is obtained from a document to serve an underlying task. In the following, we quantify usefulness by focusing on cases where information is directly extracted from a document, not where it is first assimilated and transformed through the human mind to form an outcome. According to our definition, information is useful if it is extracted from a source and placed into an evolving information object to be modified. In the context of essay writing with text reuse, this means that information is copied from a search result and pasted in the essay to be written.

We measure the usefulness of documents for writing an essay in two dimensions, both based on the idea that a document is useful if information is extracted from it. First, we measure the number of words extracted from a document and pasted into the essay—this measure indicates the amount of text that has the potential to be transformed as a part of the essay. Second, we quantify usefulness as the number of times any text is pasted per clicked document.

Table 3.8: Means and standard deviations of study variables (n=130).

Query Variables	μ	σ	**Click Variables**	μ	σ
Queries	46.5	41.9	Clicks per query	4.0	4.6
Unique queries	24.9	18.1	Click trails per query	1.8	1.4
Anchor queries	5.2	6.1	% Useful clicks per query	30.8	12.9
Querying time per query	53.1	46.1	Result reading time per paste	262.0	357.7
% Unique queries of all queries	62.6	20.8	… per click per query	48.9	30.4
% Anchor queries of all queries	10.5	7.1	… per major revision	172.4	141.7
Query terms per query	5.4	1.5	*Text Editing Variables*		
Unique query terms (UQT) per query	0.8	0.4	Writing time per major rev.	867.3	666.0
UQT from documents per query	0.6	0.3	Revisions per paste	175.7	225.7
% UQT from snippets	78.6	8.7	Writing time per paste	1270.4	1100.5
% UQT from docs	67.1	20.8	Words in the essay	4988.1	388.8
% UQT per query	15.9	7.6	*Other Independent Variables*		
UQT per unique query	1.3	0.4	Search sessions	7.4	4.1
Dependent Variables					
Words per useful click per query				325.0	420.8
Pastes per useful click per query				1.2	1.8

The limitations of these measures include that they do not reflect the possible synthesis of pasted information or the importance of the obtained passage of text. It is evident that the amount and importance of information are not linearly related, although users were allowed to use the pasted text directly for the essay without originality requirements. Our idealization excludes the qualitative aspects of information use; the presupposition that an increasing amount of pasted text reflects usefulness directly resembles typical presuppositions in information retrieval research: for instance, Sakai and Dou [169] suppose that the value of a relevant information unit decays linearly with the amount of text the user has read. In general, a similar supposition holds for the DCG measure. These presuppositions are idealizations that we also apply in our analyses.

Independent Variables

For predicting the usefulness of search results, we focus on query, click, and text editing variables to build linear regression models. Temporally, querying and clicking precedes the selection of useful information, while the usage and manipulation of information succeeds it. Since we use aggregated data over all user sessions, we treat the search and writing process as a cross-sectional event, although querying, clicking, and text editing occur over several sessions. Since the editing of the essay text is connected with querying and clicking in a session, it is important to take into account also text editing variables in analyzing the usefulness of search results over all

Independent Variable	Group	β	R^2 Change
Clicks per Q	C	0.38***	0.400
Revisions per paste	T	−0.15*	0.144
Unique queries	Q	−0.34***	0.081
Search sessions	O	−0.18**	0.037
Words in the essay	T	0.12*	0.019
Result reading time per paste	C	−0.24***	0.010

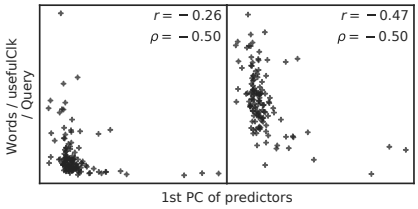


Figure 3.12: Left: Regression model for the number of words pasted per useful click per query (n=130), with predictors from the (Q)uery, (C)lick, (T)ext editing and (O)ther variables. Right: First principal component of this model's predictors and dependent variable; effect of the latter's log-transform on Pearson (r) and Spearman (ρ) correlation.

sessions. Therefore, we also select aggregated text editing variables for our models, although this solution is not ideal in every respect for representing the temporal order of the process.

Based on previous studies [81, 123, 133], we select 13 query variables, 6 click variables, 4 text editing variables, and 1 other variable, yielding the 24 variables in total depicted in Table 3.8. Here, anchor queries refer to those queries repeatedly revisited throughout a session, in order to keep track of the main theme of the task; time spent querying, reading, and writing is measured in seconds; a click trail begins on the search result page, potentially following further links in the result document. We build regression models for both dependent variables and apply a stepwise entering method of predictors [85].

Regression analysis requires linearity between independent and dependent variables but in our case, the associations of both measures of usefulness with the major independent variables turned out to be non-linear—as evidenced by a large discrepancy between the Pearson and Spearman correlation coefficients, shown in the right-hand plots in Figures 3.12 and 3.13. Therefore, we logarithmically transformed both words per useful click per query, and pastes per useful click per query (base of 10), enhancing linearity notably (Figure 3.12, right). The predictor result reading time per click per query still showed a non-linear association, and was log-transformed as well (Figure 3.13, right). While the writers were instructed to produce essays of about 5000 words, some essays were notably shorter or longer (cf. Section 3.1.2 and Hagen et al. [81]). We excluded essays shorter than 4000 and longer than 6000 words from the analysis, as well as four essays with missing variables, yielding 130 observations in total.

3.4.2 Predicting Retrieval Success

Based on the variables we derive from the dataset, we investigate two linear regression models of document usefulness—each predicting one of the dependent variables at the bottom of Table 3.8. The first model uses the number of words pasted per useful click per query as dependent variable, using the amount of text extracted as an indicator of a document's usefulness, whereas the second model quantifies usefulness as the number of times text was extracted, using the number of pastes per useful click per query as dependent variable.

The Number of Words Pasted per Document

The model is significant (R^2=.703; Adj R^2=.688; F=48.4; p<.000) consisting of six predictors. It explains 68.8% of the variation in the number of words pasted per useful click per query. The tolerance of all variables is greater than .60. The four strongest predictors—the number of clicks per query, the number of revisions per paste, the number of unique queries and the number of search sessions—cover 66.2 percentage points of the variation in the number of words pasted (Figure 3.12, left). The remaining two variables cover 2.9 percentage points of it. Limiting the model to the four major factors, it is possible to reach an accuracy of two thirds in predicting document usefulness.

As per the model coefficients shown in Figure 3.12, left, the more clicks users make per query, and the less time they spend reading result documents per paste, the more words are pasted per click per query. The number of revisions per paste reduces the number of words pasted. Increases in the number of search sessions and in the number of unique queries reduce the amount of text pasted, while an increase in the number of words in the essay increases the amount of text pasted. The number of unique queries and the number of search sessions are partially correlated, but contribute to the model in this case. Further, fewer clicks per query, more time reading documents, and a greater number of revisions per paste are associated with a smaller amount of text pasted. We hypothesize that difficulties in formulating pertinent queries lead to voluminous querying, and to a greater number of search sessions, which lead to fewer clicks, to longer dwell times per paste, and to a greater number of revisions per paste, all contributing to a smaller number of words pasted.

Independent Variable	Group	β	R^2 Change
Clicks per Q	C	0.33***	0.484
Writing time per paste (sec)	T	−0.42***	0.210
Queries	Q	−0.18**	0.162
Unique queries	Q	−0.24***	0.013
% Useful Clicks per Q	C	0.18***	0.008
Result reading time per C per Q	C	−0.15***	0.015
% Anchor Q of all Q	Q	0.12***	0.006
% Unique Q of all Q	Q	0.14**	0.007
% UQT from snippets of all UQT	Q	0.07*	0.004

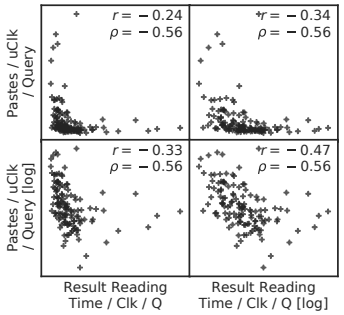


Figure 3.13: Left: Regression model for the number of pastes per useful click per query (n=129; groups as in Figure 3.12, left). Right: Scatter plots of "Result reading time per click per query" against the dependent variable, each before and after log-transform.

The Number of Pastes per Document

The regression analysis produces a model with nine variables significantly predicting the number of pastes per useful click per query. The model is significant (R^2=.908; Adj R^2=.902; F=131.2; p<.000) and covers 90.2% of the variance in the number of pastes per document (Figure 3.13, left). The three most important predictors—the number of clicks, the writing time per paste, and the number of queries—together explain 85.6% (R^2 Change) of the variation in the number of pastes; the remaining six predictors cover 4.6 percentage points of variation.

The direction of effect in click, query and text editing variables differs: Increasing values of click variables—except reading time—increase the chance that documents provide material for the essay. The query variables both increase and decrease the chance of finding useful documents, while an increase in writing time per paste decreases that chance. Compared to the previous model, click variables have a proportionally smaller contribution compared to query variables, while the relative contribution of text editing variables remains on about the same level. The direction of effect in predictors remains similar; the content of the model essentially resembles the previous one, although some predictors change: writing time per paste resembles revisions per paste, while result reading time per click per query resembles result reading time per paste. The proportions of anchor queries and unique queries are new predictors compared to the previous model.

The model indicates that the more clicks per query, the larger the proportion of useful clicks of all clicks and the shorter the dwell time in clicked documents per query, the more useful the retrieved documents are. Increases in the number of queries and unique queries decrease the usefulness of clicked

documents, while increases in the proportion of anchor queries and unique queries of all queries increase the chance that documents are useful.

Multicollinearity tolerance is the amount of variability of an independent variable (0-1) not explained by the other independent variables [85]. Five out of the nine predictors in the model were query variables. Tolerances of the number of queries (.240), the number of unique queries (.291) and the proportion of unique queries (.394) indicate that they depend quite heavily on other variables in the model. Therefore, leaving only the number of queries to represent querying would be reasonable and make the model more parsimonious.

We may conjecture that a smaller number of unique queries with good keywords from snippets produce a good result list. This contributes to a proportionally larger number of useful documents that require less dwell time for obtaining needed information for the essay. The information pasted is pertinent, not requiring much time to edit to match the evolving text. Naturally, the validity of this hypothetical process remains for later studies to test.

Comparing the Models

The explanatory power of the model predicting the number of words pasted is weaker, covering 68.8% of the variation in document usefulness, while the model for pastes covered about 90% of the variation.

The contributions of query, click and text editing variables vary between the models (Table 3.9). The relative effect ($\sum R^2$ Change) of click variables is notably greater in both models compared to query and text editing variables. Text editing variables have a somewhat greater role compared to query variables in predicting usefulness as indicated by the number of words pasted. Also the number of search sessions and the number of words in an essay have a minor impact on potential document usefulness. The models have only two predictors in common: the number of unique queries and the number of clicks per query.

In each model, three variables cover over 90% of the explained variation in document usefulness, one of them being a query, one a click and one a text editing variable. The most powerful variable is clicks per query in both models. Thus, one could predict each type of document usefulness by a very simple model.

In both models, the number of queries and the number of unique queries have a negative effect on document usefulness, while all proportional query variables have a positive effect. Clicks per query have a positive contribu-

Table 3.9: Summary of models: number of predictors, and relative importance ($\sum R^2$ change), per variable group for both models of search result usefulness.

Characteristics	Number of Words	Number of Pastes
Adj R^2	0.688	0.902
# Variables ($\sum R^2$ Change)	6	9
Query	1 (0.08)	5 (0.19)
Click	2 (0.41)	3 (0.51)
Text Editing	2 (0.16)	1 (0.21)
Other (sessions)	1 (0.04)	-

tion to usefulness, while dwell time has a negative contribution. Number of revisions and writing time per paste both have a negative effect on document usefulness. In the model for the number of pasted words, the number of sessions has a negative effect on usefulness, while the number of words in the essay has a positive effect.

Altogether, it seems that users find more useful result documents, if: the user issues fewer queries over fewer sessions, makes more clicks per query, but with shorter dwell time on individual documents, makes fewer revisions to the essay per pasted text snippet, and writes a longer essay. Although regression analysis does not indicate associations between independent variables, we conjecture that users who issue fewer queries have better result lists, click more per query, spend less time reading documents, all this producing more useful documents per click per query. This hypothetical process remains to be tested in future work.

3.4.3 Implications and Generalizability

Our regression models not only allow provide an answer to Research Question 3 (Predicting Retrieval Success), but also allow some informed speculation about the underlying processes: we observe that increased search result usefulness is associated with decreasing effort to edit the pastes for the essay. This is likely a result of the fact that writers were explicitly permitted to reuse text from the sources they found, without having to think about originality requirements. Hence, provided they found appropriate sources, writers could place passages from search results directly as a part of their essays. If the usefulness indicators reflect authors finding sources that require little editing, they should be correlated with less editing of pastes. To test this hypothesis, we measure the proportion of reused words out of all words in the essay (authors annotated the text they reused themselves, as

part of the original study); it can be reasonably assumed that the higher this proportion, the less the pasted text is edited. We find that Spearman correlations of the proportion of reused words with the number of pastes (ρ=.27**) and the number of words pasted (ρ=.18*) are significant. Thus, decreasing effort in editing pasted text reflects the usefulness of pastes in composing the essay.

Further, our models indicate that the fewer queries a user makes, the more clicks per query, and the less text editing takes place, the more useful the search results are. This matches well with previous findings: an increase in clicks has been shown to correlate with search satisfaction [88] and the perceived usefulness of documents [123]. However, our results also show that an increase in dwell time decreases search result usefulness. This contradicts many earlier findings that dwell time is positively associated with usefulness [121, 123, 133]. We believe that this difference is due to the study design underlying the dataset we used: First, previous studies have restricted task time considerably, while in the essay writing of the Webis-TRC-12 there was no time limit. Second, the required length of the essays is notably longer than in similar studies. Third, the writers of the essays in the Webis-TRC-12 were encouraged to reuse text from search results without originality requirements. These factors likely encouraged authors to copy-and-paste from search results, potentially editing the text later.

In a previous study, Liu and Belkin [121] observed that users kept their search result documents open while moving back and forth between reading documents and writing text. In their scenario, increased usefulness thus comes with increased dwell time. In the case of Webis-TRC-12 instead, many writers first selected the useful pieces from some search result, pasted them into their essay, and modified them later [157]. Thus, the actual dwell time on useful search results is lower in the Webis-TRC-12. Furthermore, the selection of useful text fragments likely resembles relevance assessments. It has been shown that it takes less time to identify a relevant document compared to a borderline case [78, 175]; essay writers likely needed less time to identify useful text passages in search results containing plenty of useful information, compared to documents with less such information. This can also further explain the negative association between dwell time and usefulness in the scenario of our study.

We believe that our results can be generalized to arbitrary writing tasks of long texts: In essay writing, it is likely that querying and result examination behavior is similar regardless of originality requirements, while text editing will vary by originality. An interesting future research question is how search and text editing contribute to document usefulness in the form

of information use, in the presence of stricter originality requirements. In the paragraphs above, we conjecture processes that could explain the associations between the predictors and the usefulness measures. While our regression models do not allow us to test these conjectures, such an analysis could form a promising future direction.

Our regression models cover about 90% of the variation in pastes from clicked search results and about 69% of the variation in the number of words pasted per clicked search result. We argue that the number of pastes and the number of pasted words reflect the actual usefulness of search results fairly validly: for the writers in our study, pasting precedes usage in the final essay. In both our regression models, three predictors cover 91–95% of the explained variation. In both cases, one of these is a query variable, one is a click variable, and one is a text editing variable. Thus, all three variable types are required for an accurate prediction of usefulness based on information usage. Click variables have the strongest effect on usefulness compared to query or text editing variables. However, it is essential to include also the latter ones in the models, as they cover a notable proportion of variation in usefulness. Consequently, personalization in real-world retrieval systems based on information use should include the major factors in these three variable groups, due to their strong effects.

3.5 Conclusion

This chapter has introduced the Webis-TRC-12 corpus, a crowdsourced dataset of 150 long essays written by 12 different writers with the support of a web search engine, while the writers' interactions with the search engine—as well as their changes to the essay text over time—were recorded in fine-grained detail. The initial exploratory analysis of the data has provided answers to Research Questions 1a and 1b, but the Webis-TRC-12 has been applied to other pursuits not covered here as well, including the evaluation of plagiarism detection systems [158] and source retrieval algorithms [83].

In pursuit of Research Question 1a (Writing Strategies), Section 3.2 identifies the *build-up* and *boil-down* writing strategies: the former is characterized by targeted and incremental material gathering over time, and reused text is adapted into the developing text on the fly. By contrast, adherents of the latter strategy collect big chunks of text in large batches, and distill a coherent essay in a later re-writing phase. Similarly, in answer to Research Question 1b (Searching Strategies), Section 3.3 finds evidence for two basic

searching strategies, which we label *clickers* and *queriers*: the former follow long click trails from a small number of queries, whereas the latter submit varied queries, but are more selective with clicks.

For web search tasks like the essay writing with text reuse studied in this chapter, we can propose also an answer to Research Question 2 (Measuring Usefulness): the act of reusing material from a source forms a strong usefulness signal—we term this usefulness by *actual use*, as opposed to measuring the perceived usefulness by asking the user (cf. Ahn et al. [3], He et al. [89]).

The study presented in Section 3.4 is one of the first attempts to analyze the actual usefulness of clicked search results based on information usage, and gives an initial answer to Research Question 3 (Predicting Retrieval Success): users who make fewer queries, click more, and edit the text less tend to be more successful at retrieving useful sources. Interestingly, long dwell times on result documents appear to be associated with lower retrieval success, while some previous studies found the opposite. We hypothesize that this is due to the greater complexity of our writers' task, and the lack of a time limit for task completion.

4

Analysing a Large Question-Query Log

This chapter tackles the problem of how search engine queries expressed in question form can be better supported by retrieval systems. To this end, we focus on the query preprocessing step in the task-based information retrieval process (cf. Figure 1.1 on page 3), and work towards answering Research Question 5 (Question Query Patterns) in search of patterns in the query log that can help e.g. in query disambiguation. In the process, we analyze a large query log of nearly 1 billion question queries in order to study the characteristics of question queries. Following this initial analysis, we address Research Question 4 (Question Query Classification), automatically categorizing the question-like queries by their topics. To this end, we mine a large training set of author-labeled questions from a Community Question Answering (CQA) platform, on which we train a classifier that is then transferred to the query classification setting. Our analysis with the help of the trained model helps us answer Research Question 5 (Question Query Patterns), gaining new insights on the characteristics of the questions that users type into search engines.

In the late 1990s, queries in question form comprised less than 1% of the query stream of a general-purpose search engine; the most common format was `[where can i find ...]` for general information on a topic [179]. Pang and Kumar report that question queries accounted for about 2% of the entire Yahoo query stream in 2010 [150]. Our analysis shows that question queries already constitute a 3–4% share of the query log we are using from 2012; questions are thus still on the rise even in keyword-based interfaces.

Why do users formulate search queries as questions? A possible explanation is the general tendency of smoother, more natural human-computer

interaction with information retrieval systems, as evidenced by touch, voice, and visual search interfaces. In particular, the increasing prevalence of voice search queries has been previously documented [172].

However, submitting queries in the form of natural language questions does not always yield better search results. As several studies show [11, 22, 150], web search engines perform worse at answering question queries compared to corresponding keyword queries. In view of the growing share of question queries, and the still lagging search quality for them, there is a strong need to improve the processing of such queries. For example, a recent major update to Google's search algorithm—codenamed Hummingbird—was targeted at answering long natural questions better.[1]

As elaborated in Chapter 2, query preprocessing with the aim of improving under-resourced queries often makes use of click-through data as an implicit relevance signal. In case of questions, however, the availability of click-through data is a big problem, as questions are typically rather unique, and have little associated log data. This rules out the above classification methods for our use case of question classification and we aim for another approach to analyze our large question query log.

The approach described in this chapter exploits Community Question Answering (CQA) data and its categorization scheme as a "bridge classification" for the question query classification problem. CQA services provide a vast amount of questions manually categorized by their users that can inform automatic query categorization. Similarly to [118], our primary goal is to expand the training set, using rather straightforward classification techniques. In our study we employ several million user-generated questions, along with top-level category labels, for building a question-query classifier. To the best of our knowledge, this approach is novel.

Topical classification of questions can be useful not only in a web search context. Robust topical classification can also boost the identification of users' information needs in contexts different from web search. Mobile voice-activated assistants like Apple's Siri—that suffer from a limited range of available classification domains [21]—may benefit just as the analysis of short interrogative posts on Twitter [223] or Facebook [140].

The contributions of this chapter are two-fold: First, we describe and analyze two large complementary datasets of Russian questions from 2012: (1) a year's worth of questions posted at a popular CQA service, and (2) question queries submitted to a large commercial search engine. To the best of our knowledge, this is the first study dealing with non-English question

[1] `http://onforb.es/1bfagwI`

datasets of this size. Second, we build a question classifier of high quality using CQA data and use it to analyze the information needs of web search question askers.

The remainder of the chapter is organized as follows: we introduce the datasets used in our analyses in Section 4.1 and explain our classification approach in Section 4.2. Besides experimental evaluation of the classification approaches, Section 4.3 also shows the application of our classification approach to one billion web search question queries to shed some light on what people ask their search engine. Finally, Section 4.4 summarizes the results and suggests interesting directions for future work.

4.1 Data Acquisition and Preparation

The basis for our question query classification are two datasets: a large amount of question-like queries collected from the query log of Yandex[2], a leading Russian search engine, and a year's worth of questions and answers from a popular Russian community question answering (CQA) platform Otvety@Mail.Ru[3]. Both datasets contain Russian queries only, although some of the queries contain words in other languages (mainly named entities such as movie or song titles, names of video games, etc.). Below, we outline the data acquisition process and provide further details on the datasets.

4.1.1 Web Search Questions

The initial dataset comprises all queries from Yandex' logs for the year 2012 containing one of 58 combinations of question word uni- or bigrams (e.g., *what, where, when, why, how, does, should, ..., in which, for what*, etc.). This is similar to previous processes of question extraction from query logs [22] except that the question word set was adapted to Russian. Each entry in the resulting question excerpt is annotated with the query string, time stamp, and user ID. The nearly 2 billion initially acquired questions form about 3–4% of the actual query log, indicating some further increase in the number of questions submitted to web search engines compared to the 2010 Yahoo figure of about 2% using similar extraction rules [150]. Under the agreement with the search engine, we have access only to the queries containing question words for research purposes; we have no access to other queries issued by the same users or to the search results. Since it was curated at the end

[2] `http://yandex.ru`
[3] `http://otvet.mail.ru`

Table 4.1: Cleaning the question queries extracted from the web query log.

Cleaning step	Unique users	Questions
Raw log	185,700,840	1,980,878,942
Spam & bots	184,630,648	1,903,716,272
Core questions	167,812,003	1,577,657,443
Repeats & prefixes	167,812,003	1,265,433,864
Unoriginal questions	145,688,746	923,482,955
Single-word questions	145,071,912	915,055,325

of 2012, the query log contains no entries for the second half of December. Hence, we omit all December entries from our analysis.

In an iterative process outlined below, we apply several data cleaning steps to retain only queries that represent actual question-asking information needs. Table 4.1 shows the individual steps of the data cleaning process and their impact.

We first target the removal of spam and bot queries from the log. After examining user activity statistics, we suggest to characterize a user as a bot when any of the following properties hold: (1) more than 2,000 total entries in our question-only excerpt of the query log over the entire year; (2) more than five questions within the most active one-minute window; (3) a median question length of more than 20 words; or (4) at least 50 questions in total, and the same leading 15 characters in at least 80% of them. The first two criteria are aimed at the number of questions per time slot, and the latter two at the type of questions submitted. Users submitting a very large number of questions in one year or in their peak activity minute behave rather "unhumanlike" and we view them as bots. Users submitting unusually long questions, or questions almost always starting with the same 2–3 words are also behaving rather unnaturally. Extensive spot-check inspections of users matching any of the above four criteria showed that all of them could manually be easily identified as bots. The specific numbers might be debatable, especially for the peak activity for some of the affected users, but we decided to rather aggressively remove users to base later examinations only on questions that were very likely submitted as a human information need.

Altogether, the first cleaning step removed about 1 million users and all their 77 million questions. Examples of removed bots include users submitting very many `[how to translate ...]` or `[how is the weather in ...]` questions that probably aim at scraping the search engine's translation or weather service, or for instance bots submitting thousands of long copy-pasted questions from exams. Interestingly, hardly any of the questions

containing an actual question mark remain after the first cleaning step; the ones that do remain almost always also seem to be copy-pasted from some exam. Having removed all entries for the suspicious users in this first step, we apply subsequent filtering steps to individual questions in the log.

In a second cleaning step, we retain only "core questions" with a question word in the first position, since extensive spot checks of the other queries showed a large number of queries with debatable question intent. Instead of devising sophisticated rules to decide for each such query whether it actually is a question intent or not, we again choose an aggressive removal to reduce the amount of non-question needs in the final dataset. This step removed about 326 million questions; about 17 million users had no remaining questions afterwards and were also removed.

The third cleaning step eliminates repeated questions and collapses prefixes. The goal is to remove bogus query submissions resulting from instant search, accidental submissions of unfinished question strings, or log entries of users paging through search engine results pages (SERPs) (always with the same question string but not really submitting new queries). If a user resubmits the same question within 90 minutes, without a different question in between, we only retain the first occurrence. To catch SERP paging behavior, we again choose to aggressively clean the query log using a long temporal window rather than a 2- or 5-minute gap. To remove "unfinished" questions stemming from instant searches or unintentionally submitted queries, we analyze pairs of questions submitted within 5 seconds. When the first query of such a pair is a prefix of the second, we retain only the latter (e.g., `[when was caesar bo]` is removed when immediately followed by `[when was caesar born]`).

In a fourth cleaning step, we remove unoriginal questions, by which we refer to questions not formulated by the user themselves, but probably stemming from some external source. For this, we first remove all questions that match one of 885 titles of Wikipedia articles (e.g., the Russian version of the movie title `[what women want]`). We then also identify questions that seek answers to crossword puzzles: if a query ends with the phrase `[`n `words]` for some value of n, we assume it came from a crossword puzzle. In addition, we also remove question queries that contain the phrase `[family feud]`, and variants of that TV show's name in its Russian incarnation.[4] For both the crossword and TV show questions, we include a "bootstrapping" step, in which we also remove all the questions that co-occur ten or more times with

[4] Family Feud is a popular TV show that prominently features questions like `[what is a problem most people have in their life]` for which the participants have to guess the most popular response of 100 people being asked that question.

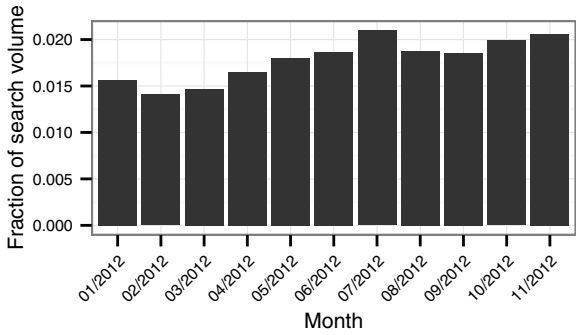


Figure 4.1: Question queries in the cleaned dataset as a monthly fraction of the total query traffic.

one of the characteristic phrases (about 7,600 question strings identified in the bootstrapping). Furthermore, we also remove questions matching a list of 1,764 questions published on fan websites of the Family Feud show. We believe that hardly any of the questions matching a Wikipedia article with the very same title, a crossword puzzle question, or a Family Feud question actually represent an original question intent of the user. Exceptions might be questions like `[when was family feud aired on abc]`, but again, we aggressively remove all of the about 342 million questions matching the above patterns, instead of a more detailed case-by-case decision.

In the fifth and final cleaning step, we filter out those entries containing only one word after stopword and question word removal. Although this also removes questions like `[when is christmas]` our spot checks found many of the single-word questions not to represent real question needs.

The cleaning steps altogether removed more than half of the originally sampled questions; the remaining dataset contains about 915 million question queries from about 145 million users. This represents about 1–2% of the search engine's query stream (cf. Figure 4.1 for the monthly fraction). Further characteristics and a comparison to our CQA dataset can be found in Section 4.1.4.

4.1.2 Community Question Answering Data

The CQA dataset we acquired comprises approximately 11 million questions submitted in Russian by over 2 million unique users to the Russian CQA platform Otvety@Mail.Ru[5] throughout the year 2012. Otvety@Mail.Ru (*otvety* means *answers*) is a Russian counterpart of Yahoo!

[5] `http://otvet.mail.ru/`

Table 4.2: Class distribution in the CQA dataset and the manually labeled question query test set.

Category	Number of instances	
	CQA	Test set
Society & Culture	1,267,700	95
Computers & Internet	965,834	131
Family & Relationships	950,180	33
Adult	526,465	13
Games & Recreation	524,533	61
Education	372,600	38
Home & Garden	355,906	117
Entertainment & Music	337,364	64
Cars & Transportation	335,659	89
Health	307,033	70
Consumer Electronics	193,685	43
Beauty & Style	173,825	23
Sports	165,959	16
Business & Finance	99,524	41
Σ	6,576,267	834

Answers with similar rules and incentives. Each question is manually categorized by the submitter into one of 28 top-level categories with altogether 189 leaf-level categories forming a two-level hierarchy. In the process of dataset acquisition, we omit several ambiguous categories, and merge closely related categories, leaving the 14 top-level categories shown in the first column of Table 4.2 as our classification targets.

When using query category labels as additional features for ranking along with hundreds of other features, coarse-grained flat categories usually suffice. This is an important difference to query classification for search advertising (advertising-to-query matching is based on category information only), automatic classification of web documents, or category suggestion for questions in the CQA scenario. In the latter cases, the amount of information items under leaf categories must be "digestible" by humans. Hence, hierarchical taxonomies with thousands of categories are used.

We paid special attention to noise in category labels, dissimilarity of the topic distributions in the two datasets, and the alignment of source (CQA) and target categories. Since the user posting a question on the CQA platform manually labels the question with a category—and this seems to be an error-prone task given the number of categories—we decided to further

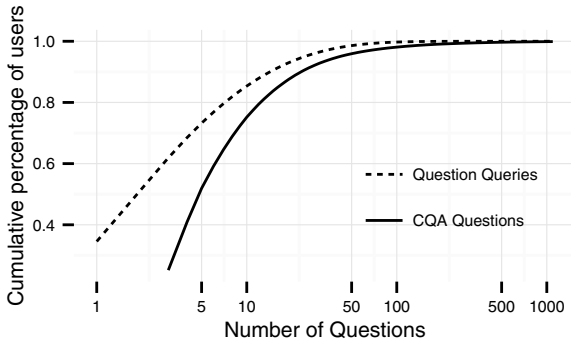


Figure 4.2: Number of questions per user.

clean the initial dataset. We only keep questions submitted by users that have posted at least three questions that got an answer. This criterion is meant to capture questions with better categorizations: users posting more than just one or two test questions can be viewed as more experienced with the category scheme and questions that got an answer form a further support of this hypothesis since other users found the query under its category.

The assigned categories in the remaining 6 million questions from the CQA platform are less noisy than the original 11 million questions, making the cleaned CQA data well-suited as a training set for our query classification task. The second column of Table 4.2 shows the number of instances in the CQA dataset per category. Further characteristics in comparison to our question queries dataset can be found in Section 4.1.4.

4.1.3 Web Search Question Test Data

In order to evaluate the performance of our classification pipeline on the question queries from the search engine log, we randomly sample 1,000 entries from the cleaned dataset. After labeling by three domain experts, no two annotators picked the same category for 166 of the questions. These more ambiguous questions were removed from the test set. The third column of Table 4.2 shows the class distribution in the remaining test set of 834 questions.

4.1.4 Descriptive Statistics

We have about 915 million questions from about 145 million users in the web search question query dataset and about 6 million questions from about

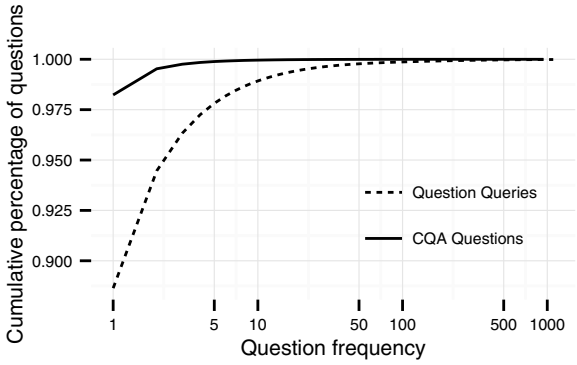


Figure 4.3: Question frequency in the CQA and question queries datasets.

0.5 million users in the CQA dataset. For both datasets, the distribution of the number of questions per user is shown in Figure 4.2.

Note that we do not have users with less than three questions in the CQA dataset due to our filtering rule. About one third of the CQA users in our cleaned dataset have posted three questions, another half have posted at most ten questions and the remaining 20% have submitted up to 5,000 questions in the year 2012. The average number of questions per user in the CQA data is about 16, with a maximum of 257 questions.

In the question queries data, the situation is slightly different, with the average user submitting about 6 questions; this is not surprising since we did not remove users with very few questions here. About 40% of the users only submitted a single question in the whole year. However, due to the user identification method on the server side, some questions from the same user might get logged with different user IDs. Similarly to the CQA data, another 40–50% of the users submit at most ten questions while only 10% of the users submit up to 2,000 questions in the whole year. Since 2,000 questions per year was a bot-removal threshold used in our cleaning process, there are no users with more than 2,000 question queries and only a few with more than 1,000 questions; the most interrogative user submitted about 1,500 question queries in the whole year.

The two datasets also differ in the most frequent question prefixes given in Table 4.3. Not surprisingly, the top prefixes of the question titles in the CQA data show that users often do not explicitly formulate a question but rather ask others for help (e.g., `[I need your help]` or `[can you help me]`). Due to the sampling strategy, the question queries have explicit question words as their initial n-grams. As was already observed in other studies,

Table 4.3: The five most frequent initial 2- and 3-grams per dataset.

N-gram	English translation	Frequency
CQA Dataset		
можно ли	is it possible	113,291
а вы	and you	89,193
у меня	I have	68,337
что делать	what to do	66,665
как вы	how you	64,235
что делать если	what to do if	36,990
где можно скачать	where can I download	21,665
как вы думаете	what do you think	20,057
а у вас	and you	16,631
как вы относитесь	what do you think	13,497
Question Queries Dataset		
как сделать	how to make	35,678,293
можно ли	is it possible	28,001,988
как правильно	how to correctly	23,014,202
сколько стоит	how much costs	19,533,978
где купить	where to buy	11,405,702
как избавиться от	how to get rid of	5,166,515
где можно купить	where to buy	2,804,874
как скачать музыку	how to download music	2,072,003
как доехать до	how to get to	2,028,746
какие документы нужны	what documents are needed	1,818,986

the most frequent questions are how-to questions that can be formulated using different bi- and trigrams in Russian.

Figure 4.3 shows the datasets' frequency distributions. In the CQA data, about 98% of queries are unique, as opposed to 88% for the question queries. Unsurprisingly, given that we count title and description as the query, the average question appears just once in the CQA data with the very short most frequent questions appearing five times. In the search engine question queries, the average question appears about two times while the most frequent question, `[how to download music from VK]`, has nearly a million occurrences.[6]

Figure 4.4 shows the question length distributions in both datasets. While almost all the question queries have at most ten words, only one third of the CQA questions are that short (but note that we combine the

[6]The query refers to the Russian social network site `vk.com`.

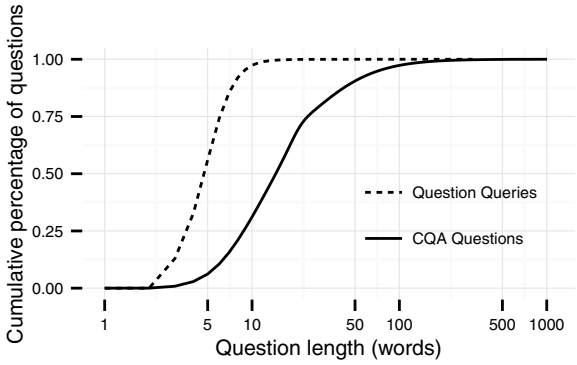


Figure 4.4: Question length in the CQA and question queries datasets.

question title and description fields). The average question query has a length of about six to seven words (about five to six not counting question words); the longest having 114 words was probably copy-pasted from an exam (not reprinted here). The average CQA question is much longer with about 24 words (28 including question words) and the longest CQA question is about 1,000 words including its description.

4.2 Question Query Classification

We employ machine learning to assign categories to the queries in the question query log: Using the CQA questions and their author-assigned categories as a training set, we train a classifier that predicts the categories of unlabeled search engine question queries with high accuracy. To this end, we first derive different feature representations from the question queries and CQA questions, using the representation strategies outlined below. We compare a bag-of-words representation, which is effective but unwieldy due to its amount of features and only applicable to a subset of the question query data, to a much more compact topic-model-based representation. We compare both of these machine learning based approaches to a simple retrieval-based baseline, which looks up the query to be classified in an inverted index of CQA questions, and assigns the majority category of the top ten results. Figure 4.5 gives a high-level overview of the three classification pipeline variants; additional details are given below.

Since we finally want to classify the question queries, the general process of transferring the trained classifier from CQA data to question queries is as follows: First, we extract features (bag-of-words or topic models) from the question queries, then train a classifier with these features on the CQA ques-

tions, and finally apply the classifier to the question queries. The (unsupervised) feature extraction from the target dataset ensures better transferability of the classifier.

4.2.1 CQA Retrieval Baseline

As a simple baseline for comparison, we implement a majority-vote classifier based on CQA retrieval. To this end, we index the CQA dataset using the Okapi BM25 retrieval model, which has served as a baseline in previous studies on CQA retrieval [217]. At classification time, we submit the unlabeled query to this index, and pick the most common category among the ten first search results. In case of ties, we pick the category with the higher aggregate retrieval score.

4.2.2 Bag-of-words Features

Our first actual machine learning model is based on a bag-of-words representation, where each question is represented as a term frequency vector of (case-folded and lemmatized) unigrams. A bag-of-words model can be very complex: across the entire question query dataset, there are more than 16 million distinct words. Since we are using the CQA questions as our training set, our classifier can only consider the 1.3 million words that also occur in CQA questions. Out of this number, we retain only those words that occur in at least ten question queries, resulting in 137,032 features for our question query representation.

Besides its complexity, the main drawback of the bag-of-words model is the divergence of the feature sets between the two datasets. Out of our nearly one billion question queries, only 85% contain vocabulary from the bag-of-words model. We employ probabilistic topic modeling in order to reduce the model complexity, as well as to improve the transferability of the classifier. We investigate two different probabilistic topic models: Latent Dirichlet Allocation [24] and the Biterm Topic Model (BTM) proposed by Cheng et al. [47].

4.2.3 Topic Model Features

Both LDA and BTM are generative Bayesian models that uncover latent topics in a given text corpus by modeling the formation of documents as the result of a probabilistic process. For the purposes of feature derivation for our classification task, they operate in two basic steps, which can be summarized as *inference* and *representation*. The inference step involves finding the

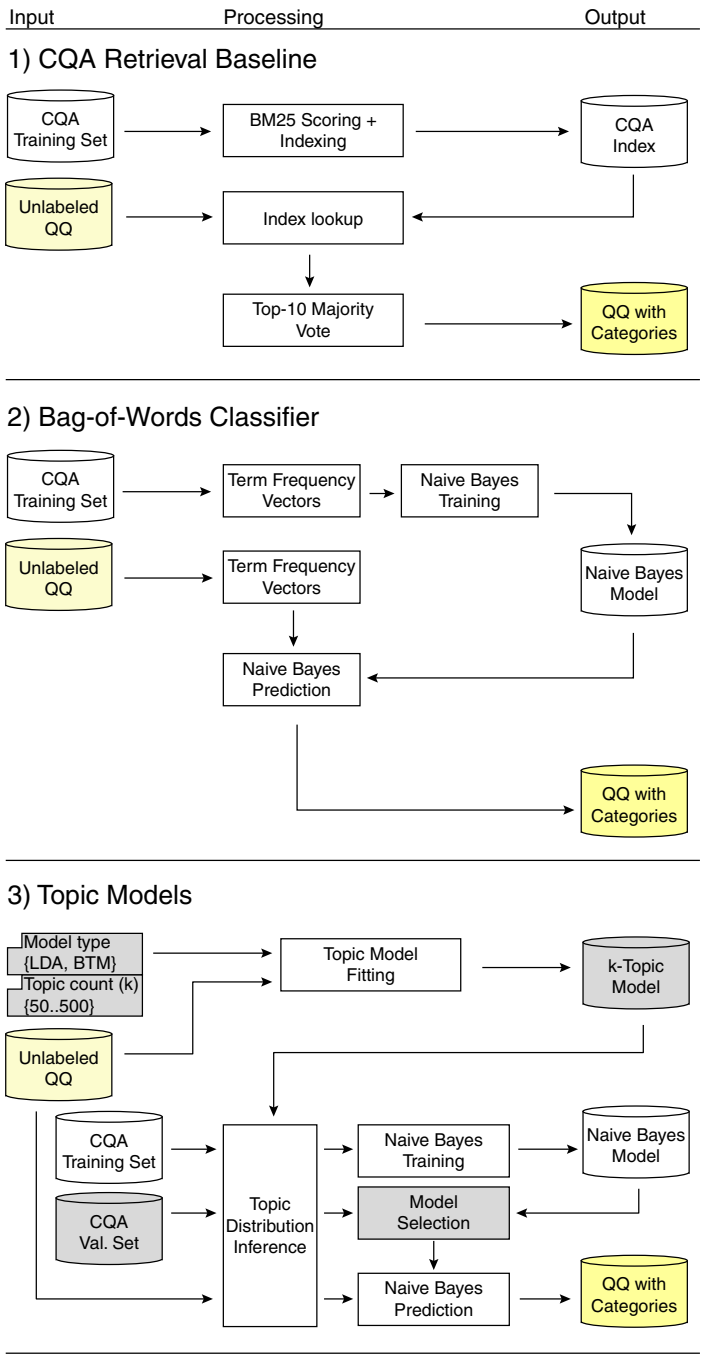


Figure 4.5: The three question query classification pipeline variants in our study: (1) Simple CQA retrieval baseline; (2) Naive Bayes classifier trained on Bag-of-words representation; (3) Naive Bayes Classifier trained on topic model representation.

model parameters that best fit the observed data (the questions/documents in the corpus) for a given topic number k. Given a topic model thus trained, documents can be represented as k-vectors of topic probabilities. The generative model that is assumed to have generated the observed documents differs significantly between LDA and BTM.

From the LDA perspective, each word in each document is generated by first drawing a topic from a document-specific topic distribution, and then drawing the word from the word distribution for that topic. In order to accurately infer the per-document topic distributions, LDA depends on document-level context, and tends to perform poorly on short texts where word co-occurrence information is sparse [47].

The Biterm Topic Model circumvents this sparseness problem by modeling term co-occurrence directly: BTM's generative model conceives of a document as a set of all word pairs (biterms) that co-occur in it. Each biterm in a given document is generated by drawing a topic from a single global topic distribution, and then drawing the biterm from that topic's biterm distribution.

The benefit of representing documents as vectors of latent topic probabilities is two-fold—first, the representation is much more compact than a bag-of-words model of similar performance, and second, it captures high-level semantic structure based on unigram occurrence alone, allowing a larger fraction of the question query log to be classified.

Our topic-model-based classification pipeline operates as follows: we apply stopword removal, case folding and lemmatization to all datasets. We then fit topic models to the question query dataset with the topic count k ranging from 10 to 500. For LDA, we employ the implementation available as part of the *gensim* Python software package.[7] To fit Biterm Topic Models, we use a C++ implementation maintained by one of the BTM authors.[8] We then represent the CQA questions using the topic models fitted on the question queries, and split the CQA questions into a training and validation set, comprising 70% and 30% of the questions, respectively. We use the validation set to select the best performing topic model, which we then evaluate on the web search question test set.

In the following section, we describe the results of our classification experiments and insights on questioning behavior in the search engine log.

[7] `https://github.com/piskvorky/gensim`

[8] `https://github.com/xiaohuiyan/OnlineBTM`

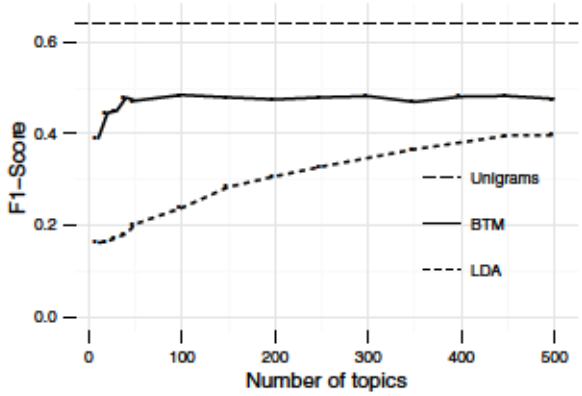


Figure 4.6: Classification performance of the topic model features on the CQA validation set.

4.3 Experimental Results

According to the above procedure, we train a multinomial naïve Bayes classifier on the CQA training set for each of our question query models, and compare their performance on the CQA validation set. Having selected the best performing models from this run, we train new classifiers on the entire CQA data and evaluate them using the web search question test set. In order to compare the performance of the different models, we compute the classification performance on each target class, and then average over the classes to arrive at the macro-average precision and recall, as defined by [176]. Finally, we classify all questions in the web search log in order to gain further insights into what users ask search engines. For the classification experiments described below, we employ the multinomial naïve Bayes implementation from the Apache Spark MLlib library.[9]

4.3.1 Performance on CQA Questions

In order to compare the performance of the different topic models on the CQA data, we first fit a topic model to the question query data for the different numbers of topics. Due to the large amount of input data, this is a time consuming process; fitting the 500-topic BTM model requires approximately 80 hours of wall-clock time on a machine with sixteen 1.6 GHz CPU cores, while the largest LDA model requires about 24 hours. For both topic models, we use an incremental variant of the inference algorithm. Our

[9] `https://spark.apache.org/mllib/`

Table 4.4: Performance of the Bag-of-words and BTM models on the web search query test data. The final column shows the change in F_1-score relative to the validation set.

Features	Precision	Recall	F_1-Score	Gain
CQA Retrieval Baseline				
—	0.67	0.66	0.66	—
Bag-of-words				
137,032	0.61	0.7	0.65	+2%
Biterm Topics				
100	0.47	0.53	0.50	+1%
200	0.46	0.49	0.47	±0%
300	0.46	0.50	0.48	±0%
400	0.46	0.50	0.48	±0%
450	0.49	0.53	0.51	+4%

observations confirm those of [47]—while the processing time for BTM is higher than for LDA, the memory requirements are lower.

Figure 4.6 shows the classification performance of the topic model-based features on the validation set, with the number of latent topics ranging from ten to 500. The biterm topic model outperforms LDA by a large margin for all topic counts. Considering the sparse word co-occurrence information found in web search queries, this result confirms our expectations. Both topic models' performance increases with growing number of topics, but the effect is more pronounced for LDA. More fine-grained latent topics make more informative features for query categorization in both cases. While the bag-of-words model outperforms both BTM and LDA, it is at the cost of greater model complexity: the number of dimensions in the feature vector for the bag-of-words model is four orders of magnitude larger.

4.3.2 Performance on Web Search Test Data

Based on the above results, we conclude that the biterm topic model is better suited than LDA to our application domain. Hence, we compare the performance of the BTM features to the bag-of-words features on the web search query test set.

The results of our comparison are summarized in Table 4.4, where we show the test set performance of CQA retrieval, the bag-of-words model, and the BTM models which perform best on the validation set. The rightmost column of the table shows the relative performance gain (or loss) in-

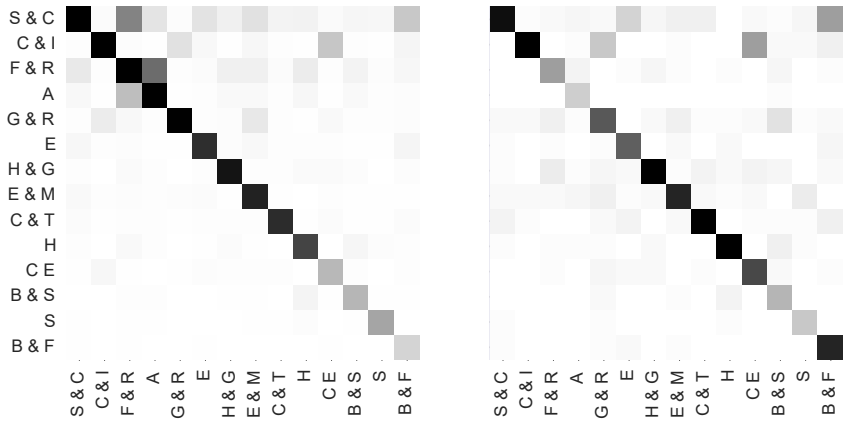


Figure 4.7: Confusion matrices for the bag-of-words classifier on the CQA validation set (left) and the question queriest test set (right). The rows are the true classes, the columns are the predictions; the ordering of the classes is the same as in Table 4.2.

curred in the transfer from the CQA to the web search data: it is notable that no performance loss occurs here, which suggests that the word distributions in both datasets are sufficiently similar. While bag-of-words features still outperform the topic model on the test set, the difference in F1-score between bag-of-words and the best-performing BTM model is smaller compared to the validation set.

Being the best-performing of our machine learning models, we select the bag-of-words classifier to investigate the topic distribution in the web search question dataset; for the purpose of our post-hoc analysis, classification speed is not a major concern. However, in a live retrieval setting, we argue that one may prefer the BTM classifier despite its lower performance: due to the more compact feature vector, classification with BTM is much faster; on a 100-node Hadoop cluster running many classifications in parallel, the bag-of-words classifier requires on average three milliseconds of CPU time to classify a single question, compared to 1.3 milliseconds for the BTM classifier.

As shown in Figure 4.7, the classifier succeeds at distinguishing most of the categories rather well. Two exceptions are the "Family & Relationships" and "Adult" categories, which are frequently confused, as well as the "Computers & Internet" and "Consumer Electronics" categories. In both cases, a likely explanation is the natural overlap in vocabulary between these pairs of categories.

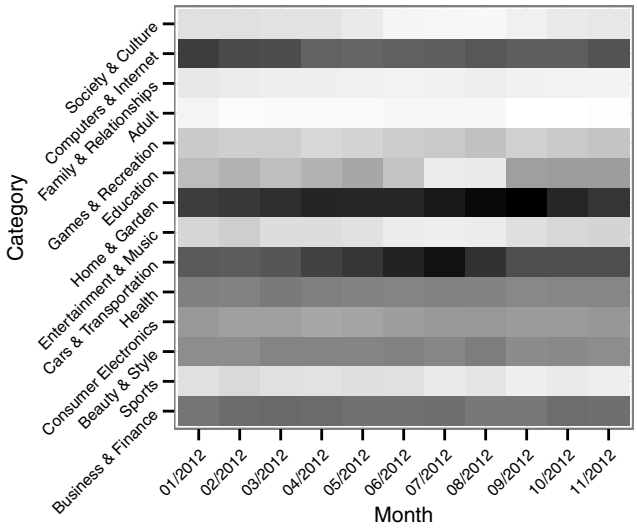


Figure 4.8: Distribution of monthly question query volume over categories. For each month, the shadings of the grid cells represent the categories' relative contributions to that month's total number of queries.

While the CQA retrieval classifier achieves a slightly higher F1-score than bag-of-words on the web search question test set, it incurs a much larger computational overhead—an average of 407 milliseconds per query, with the index stored on a solid-state disk. More advanced retrieval models have been shown to outperform BM25 in terms of CQA retrieval performance [217]. However, the overhead of an index lookup for each classification may prove prohibitive in a live retrieval setting.

4.3.3 Categorizing Web Search Questions

Below, we showcase some of the insights gained from the category distribution of the question queries in our query log. Since even our three human annotators were unable to reach a majority agreement regarding the category assignment in 17% of cases, and our classifier agrees with the annotators only two thirds of the time, the category assigned to any individual query should be taken with a grain of salt. However, we do consider our model good enough to study general trends in the data.

Figure 4.8 shows the distribution of question query categories by month over the entire dataset. The shading in the cells shows the contribution of each category to the total query volume for the corresponding month. The category axis is ordered by descending frequency in CQA questions, for easy comparison with Table 4.2. The category distribution that our classifier

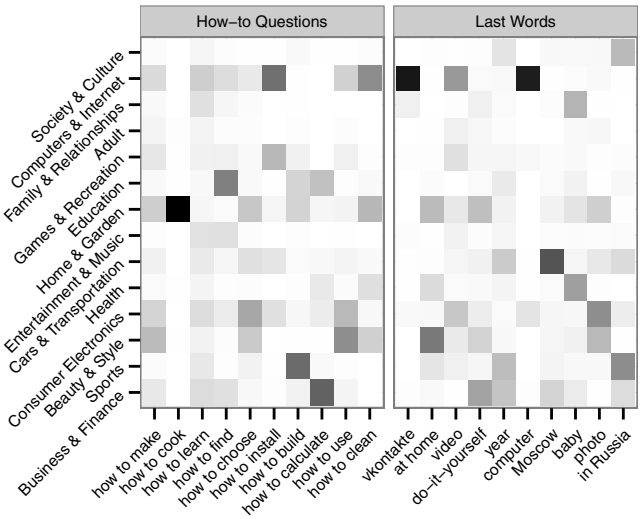


Figure 4.9: Distribution of common question prefixes (left) and suffixes (right) over categories. The items on the x-axis are ordered by total number of occurrences in the query log, highest at the left. For each item, the shadings of the grid cells represent the categories' relative contributions to that item's occurrences.

infers for the web search questions is quite different from the distribution of category sizes among the CQA data. For instance, "Home & Garden" is the largest web search question category, covering over 13% of the web search queries, as opposed to 5% of CQA questions. Only 4% of web search queries are assigned to the "Society & Culture" category, compared to 18% of CQA questions.

Beyond this, the development of categories' query volume over time is of interest. While the query volume for some categories, such as "Health" or "Beauty & Style," remains more or less constant throughout the year, others show a pronounced seasonal variation. Most notably, the "Education" category reaches its low point during the months of July and August, while "Cars & Transportation" peaks around the same time. This may reflect askers embarking on their summer vacations, and abandoning education-related inquiries for travel-related ones.

Figure 4.9 shows the category distribution for some prominent question prefixes and suffixes. For the prefixes, we select a set of *how-to* question prefixes that are the most frequent in the query log, and compute the category proportions for the questions starting with each prefix. Some prefixes are strongly correlated with a single category, such as the `[how to cook ...]` questions with "Home & Garden." Other question prefixes, like `[how to make ...]` or `[how to learn ...]` are more evenly split among cate-

gories and occur to some extent in each one. As a side benefit, this analysis serves as a sanity check for our classification model: expressions with several plausible contexts are distributed across the appropriate categories. For instance, the `[how to clean ...]` questions, with their corresponding housekeeping-, computer-maintenance-, and personal-hygiene-related contexts, appear most frequently in the "Home & Garden," the "Computers & Internet," and the "Beauty & Style" categories, respectively.

In the right half of Figure 4.9, we show the distribution of the ten most common question suffixes over the categories, which reveals similar patterns to the questions' initial parts.

In an additional avenue of inquiry, we investigate the prevalence of advanced search operators—such as quoting, boolean expressions, or restricting the search to certain domains or file types—among question queries. Studies of general query logs have found a single-digit percentage of queries to use operators. For instance, [206] report 1.12% of queries recorded over a 13-week period containing operators, and 8.7% of users employing operators at least once during that time. We conjecture that among queries formulated as natural-language questions, operator use will be even rarer. Indeed, out of the nearly one billion question queries in our dataset, only 0.2% contain any search operators; only 1% of the 145 million unique users use operators at least once.

In our query log, the quotation operator for phrasal search is by far the most prevalent, accounting for about 96% of all operator occurrences. Well-known operators like quotation and exact word match are equally prevalent across all categories, while the use of more advanced functionality often appears concentrated to a single category. For instance, the word distance operators (for retrieving only documents where all query terms occur within a user-specified number of words of each other) occur most often in the "Education" category.

4.4 Conclusion

We have conducted the first large-scale analysis of non-English question querying behavior on a web search engine. Our main goal was to answer Research Question 5 (Question Query Patterns) by analyzing the topics that searchers are interested in over the time of one year. To this end we have based our study on the about 1 billion questions submitted to a large commercial search engine in 2012. An initial superficial analysis of the query log, along with over 6 million question posted to a Community Question

Answering (CQA) site, allowed us to gain insights about the characteristics of question queries: in short, question queries are longer, rarer, and more likely to be unique than typical search engine queries, but shorter, and less frequently unique, than CQA questions. Deeper analysis of the query log's topic distribution required answering Research Question 4 (Question Query Classification) in the process.

In solving the problem of classifying question queries, we could not follow the same practices used for classifying general web queries. There, established technologies use the search results to enrich the short query strings and to classify a query based on the results or the documents clicked by a user; however, in the case of questions that are rarely submitted by more than one user, click-through is much sparser. Since we also had in mind to develop a classifier that can be used in an online search engine, the fact that result information is not available for most of the questions ruled out the use of the standard procedure. Contrary to query classification for ad-matching, or classification of questions at question answering platforms (that often classify into huge hierarchies with many classes) we aim at a flat set of only a few categories that can be easily integrated as additional features in the retrieval process (e.g., to select appropriate verticals).

Our suggested approach to question query classification is to use features extracted from the question queries to train a classifier on labeled CQA questions (where the asker assigns categories to posted questions) and then transfer this classifier back to the web search question queries. Our experiments show this approach to work very well in the setting with 14 target classes. Hence, even though studies have shown that users tend to submit different questions to search engines than to CQA services, a fact also visible in our analyses, the classification transferability is not harmed. Training the classifiers on all the questions posted to a CQA service in the same year as the search engine questions, an F-measure of about 0.5 shows a decent performance given the 14 classes. Interestingly, the accuracy of the very efficient biterm topic model-based classifier is not much worse than the less efficient bag-of-words-based classifiers that had been proposed in previous studies for question classification.

Our experimental study of the year-long question query log shows some interesting first insights on categorized question asking behavior on a non-English search engine. Not too surprisingly, education questions are hardly observed in the months of summer vacation, while travel questions have their peak appearance in this time. The ratio of questions related to home and garden or health is rather stable over the year, while not too surprisingly "adult" topics are much less present in questions than in general web search

queries. Further analyses on how-to questions, the questions' last words, and search operator use, also revealed some interesting insights.

Still, our first analyses should be seen as a starting point to use the question query classification for future work that can help improve retrieval performance on questions by better tailoring the results to the users' needs. This is especially important for questions that cannot directly be answered by showing related CQA questions. The amount of questions not directly answerable from CQA data is still an important direction for future research. Complementing our results with a similar study of question categories on English questions could shed some light on cultural differences in asking behavior and might help search engines to better address the different markets. Since the sheer amount of questions in the total query stream still is increasing, such topics will only get more important in the future. Potential applications abound—for instance in mobile voice search—to enable users to more naturally interact with retrieval systems via questions.

5

Enhancing Result Rankings With Axioms

Axioms have featured in the information retrieval literature as a way of formally expressing the properties that a good result ranking should have for more than 30 years (cf. Section 2.5). So far, axiomatic ideas have been limited to the realm of ranking theory, serving the analysis of existing retrieval functions, and the derivation of new ones. While axioms have certainly advanced the state of information retrieval research, the purely theoretical approach has certain drawbacks: any change to the consensus on the set of desirable axioms would require re-evaluating the retrieval functions used—and perhaps re-indexing the collections kept—by search engines. Past axiomatic studies have often suggested subtle changes to the original retrieval models to better conform with specific axioms, and then demonstrated retrieval performance improvements based on these changes, although typically only a handful of specific axioms are considered. Up to now, no "operationalized" axiomatic retrieval model has been proposed that *by construction* conforms with as many axioms as possible and that hence could lead to substantial retrieval performance gains.

This observation leads to Research Question 6 (Axiomatic Result Reranking): Is it possible—and how—to seamlessly integrate axioms for ranking preferences into the retrieval process, in order to improve the results of a basis retrieval model? Our proposed solution is inspired by the learning-to-rank framework: Given some basis retrieval model, a carefully weighted axiom combination re-ranks the top k results and produces an axiom-

compliant output. In this regard, we consider as many of the published axioms as possible and also suggest new term proximity axioms.

This chapter focuses on the result set postprocessing step in the task-based IR process (cf. Figure 1.1 on page 3). While postprocessing can be accomplished in a variety of ways—as discussed previously in Section 2.4.2—the axiomatic reranking approach discussed in this chapter closely follows the learning-to-rank paradigm. In the approach presented here, we employ a mixture of the pairwise and the listwise learning-to-rank approach: our axioms yield pairwise ranking preferences, but our optimization criteria measure performance over a range of result lists of different queries used for training—an approach inspired by a study of Cao et al. [42].

Most axioms in the information retrieval literature have a similar basic structure: for a pair or triple of documents, ranking preferences are deduced from standard features such as document length, term frequency, or semantic similarity (cf. the review in Section 2.5, pages 35ff). When such an axiom is applied to all pairs or triples of documents in a retrieval model's result list, the matrix of the inferred preferences may induce a result re-ranking. For example, consider a situation with an axiom A and three initially retrieved documents d_1, d_2, and d_3. After applying axiom A to all document pairs, one might end up with the preferences $d_2 >_A d_1$, $d_2 >_A d_3$, and $d_1 >_A d_3$, where $d_i >_A d_j$ means that document d_i should be ranked above d_j according to axiom A. Only the ranking $[d_2, d_1, d_3]$ matches these preferences and will thus become the re-ranked document list. However, in the general case there are many axioms (typically of different importance) and contradictory rank preferences will become likely. As a solution and a way of combining the weighted axioms' matrices of rank preferences, we apply fusion algorithms that were developed in the field of computational social choice.

In the following, Section 5.1 presents our axiomatic reranking framework and its components, including our operationalization of existing axioms in a practical retrieval setting, a set of new term proximity based axioms, and our approach to aggregating multiple axioms' preferences while resolving contradictions. Section 5.2 studies the effectiveness of our axiom-based retrieval system in a large-scale evaluation with 17 basis retrieval models in the setting of the TREC Web tracks 2009–2014. As a result, the performance of almost all basis retrieval models is improved via axiomatic result re-ranking. It is thus possible to improve existing retrieval models in an "ex-post manner," considering the latest insights from the research on retrieval axioms. Finally, Section 5.3 points out possible avenues for future work.

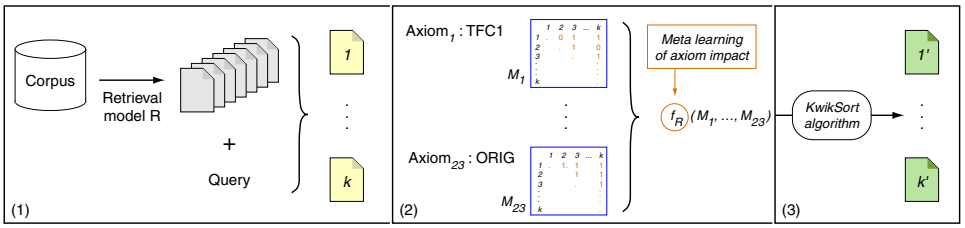


Figure 5.1: Illustration of our axiomatic result re-ranking approach. (1) Initial result set construction of size k using a basis retrieval model. (2) Deduction of axiom-specific partial orderings (matrices) for the result set documents, which are combined into a single matrix based on a previously learned axiom aggregation function. (3) Re-ranking of the original result list by solving the Kemeny rank aggregation problem with the KwikSort algorithm.

5.1 The Axiomatic Re-Ranking Approach

We put axiomatic re-ranking to work within three steps. First, an initial search is done with some basis retrieval model; the returned top k results are used as re-ranking candidates (in our experiments we set $k = 50$). Recall that our approach is not restricted to a certain retrieval model—a fact which is later demonstrated in the experimental evaluation. Second, each axiom is evaluated regarding the retrieved documents, and the resulting pairwise rank preferences are stored as a matrix. Using a machine learning algorithm on a training set of document pairs with known relevance judgments, we infer an aggregation function to combine multiple axiom preferences into a joint preference matrix. Third, on the resulting matrix a rank aggregation is applied that utilizes ideas from the field of computational social choice. In particular, we derive the final re-ranked results by employing the KwikSort algorithm [4] to solve the Kemeny rank aggregation problem [105] on the sum matrix. We argue that the training should yield different axiom aggregation functions for different basis retrieval models. Hence, when applying axiomatic re-ranking given some basis retrieval model's results, we consult the corresponding learned aggregation function. The general setup of our approach is illustrated in Figure 5.1.

In the remainder of this section we explain which axioms from the axiomatic IR literature we use and the sometimes necessary modifications. We also present our newly developed term proximity axioms, and detail the employed rank aggregation method and the axiom aggregation scheme.

5.1.1 Requirements on Axioms

We analyzed the literature on published retrieval model axioms and carefully selected those that can be restated to induce rank preferences for result lists. In this regard we decided to restrict to axioms that formalize rank preferences on pairs of documents—reflecting the pairwise approach to learning to rank. From its syntax, an axiom A in our framework is formulated as a triple:

$$A = (\mathit{precondition}, \mathit{filter}, \mathit{conclusion}),$$

where $\mathit{precondition}$ is any evaluable condition, *filter* is a more specific filter condition, and $\mathit{conclusion}$ is a rank preference $d_i >_A d_j$ (semantics: document d_i should be ranked above d_j according to A). For each axiom A and for all pairs of documents these rank preferences are stored in a matrix M_A:

$$M_A[i,j] = \begin{cases} 1 & \text{if } d_i >_A d_j, \\ 0 & \text{otherwise.} \end{cases}$$

Note that, at least theoretically, the application of an axiom requires the iteration over all pairs of candidate documents to check $\mathit{precondition}$ and *filter* to infer the rank preferences. In this initial investigation, however, we do not focus on the practical efficiency of axiomatic re-ranking but demonstrate its effectiveness. Further tuning the efficiency of the axiomatic re-ranking approach will be an interesting task for future research given the promising experimental improvements of retrieval quality we achieve (cf. Section 5.2).

5.1.2 Existing Axioms And Modifications

We start with remarks on modifications that pertain to most axioms and then present the analyzed axioms with potential individual modifications.

General Modifications Some axioms from the literature rely on preconditions or filters that require two documents to have exactly the same length (e.g., TFC1 and TFC2) in order to be admissible. However, while this condition is theoretically sound, real-world top k search result sets are unlikely to contain many documents that fulfill this (or some other) value constraint exactly. We hence relax such conditions, and require only a fuzzy match, allowing a difference of at most 10%—a length condition $\mathit{length}(d_i) = \mathit{length}(d_j)$ requiring exactly the same length of d_i and d_j is

thus adapted to $length(d_i) \approx_{10\%} length(d_j)$ with the following semantics:

$$\frac{|length(d_i) - length(d_j)|}{\max\{length(d_i), length(d_j)\}} \leq 0.10.$$

Conversely, if an axiom stipulates a length *difference*, this corresponds to a difference of more than 10%, denoted $>_{10\%}$, in our framework. Besides document length, axioms with equality constraints on term frequency are also adapted with a 10% relaxation. Some axioms' conditions require identical term discrimination values for two terms. We use idf in such cases, but do not apply the 10% rule since this would result in too many terms with the "same" idf values. Instead, we round values to two decimal places, and then consider two terms to have the "same" term discrimination value if the rounded idf values are the same. In some axioms, a semantic similarity measure $\mathrm{s}(w_1, w_2)$ for two terms is employed. We use WordNet[1] in such cases. Also note that while some axioms conclude properties of some abstract query-document scoring function $score(d, q)$, we simply map these properties to the induced rank preferences.

Term Frequency Axioms The basic idea of the term frequency axioms TFC1–TFC3 and TDC is to formulate reasonable assumptions on the correlation between term frequency and document ranks. As a first example, axiom TFC1 is given with the original description and our restated version in full, while subsequent axioms will be described in less detail. Axiom TFC1 assigns higher scores to documents that contain a query term more often based on the following original definition [73]:

TFC1: Let $q = \{t\}$ be a query with only one term t. Assume $|d_1| = |d_2|$. If $tf(t, d_1) > tf(t, d_2)$, then $score(d_1, q) > score(d_2, q)$.

We transform TFC1 to our triple notation as follows:

$$\begin{aligned} precondition &:= length(d_1) \approx_{10\%} length(d_2), \\ filter &:= tf(t, d_1) >_{10\%} tf(t, d_2), \text{ and} \\ conclusion &:= d_1 >_{\mathrm{TFC1}} d_2. \end{aligned}$$

For queries with more than one term, we use the sum of the individual term frequencies in the *filter* condition as a generalized version of TFC1. We generalize and transform the axioms TFC2, TFC3, and TDC. Axiom TFC2 requires special treatment, as it compares three documents and checks the

[1] `http://wordnet.princeton.edu/`

term frequency gaps between these documents, but concludes $score$ differences $score(d_2, q) - score(d_1, q) > score(d_3, q) - score(d_2, q)$ for the three documents that cannot directly be modeled in our framework. According to the precondition of TFC2, d_3 has the highest term frequency and d_1 the lowest; hence, we change the $conclusion$ to $d_3 >_{\text{TFC2}} d_2$ and $d_2 >_{\text{TFC2}} d_1$. This way, we treat TFC2 as a transitive version of TFC1; as such, it probably does not add much to an axiomatic re-ranking that also includes TFC1.

The axioms TFC3 and TDC conclude scoring properties for two-keyword queries based on term discrimination values (rounded idf values in our setting). The document containing the terms more often—or containing terms with higher idf values—is favored. To be applicable also to longer queries, we generalize TFC3 and TDC by applying them to every query term pair.

Document Length Axioms The axioms LNC1, LNC2, and TF-LNC all target document length normalization [73]: LNC1 compares two documents that have the same term frequency for all query terms (up to the 10% relaxation in our setting), and prefers the shorter document.

Axiom LNC2 checks whether one document is an m-times copy of another document, and prefers the shorter one. Since we consider it quite unlikely that a real top k ranking will contain a document that is an m-times copy of another, we modify this condition as follows: We first calculate the Jaccard coefficient of the documents' vocabularies (i.e., measuring the degree of term overlap). If this comes out to at least 80%, we compute a surrogate value for m as the ratio of the minimum and maximum term frequencies of the shared terms.

Axiom TF-LNC combines term frequency and document length for single term queries $\{t\}$: it prefers the document with the higher term frequency for t if the documents have the same length (up to 10% relaxation) without t. Similar to the LNC1 generalization above, we further generalize TF-LNC to multi-term queries using the sum of the term frequencies.

Lower Bound Axioms The axioms LB1 and LB2 capture a heuristic of lower bounding term frequency such that long documents are not overly penalized [126]: LB1 examines document pairs that have the same retrieval score $score(q, d)$ (with 10% relaxation in our case). If there is a query term t with $tf(t, d_1) = 0$ and $tf(t, d_2) > 0$, then $d_1 <_{\text{LB1}} d_2$. Since in its original formulation, LB2 concludes rank preferences for artificially generated documents not contained in the original result list, we modify it to work on pairs of actual documents: If the tf values of query terms t and t' differ by at most

10% in two documents d_1 and d_2, the document with the higher frequency of the query term that appears earlier in the query is preferred.

Semantic Similarity Axioms Matching semantically similar terms instead of exact matches of the query terms might be helpful in vocabulary mismatch situations but also for enhancing small result sets. We explore the use of WordNet to determine semantically similar terms in the context of the axioms STMC1–3 and TSSC1–2 [69, 71]. The conditions of STMC3, TSSC1, and TSSC2 are highly specific and cannot be "softened" such that we do not include these axioms in our framework. The original formulation of STMC1 assumes that the query and both documents under consideration each consist of only one term. We generalize this setting as follows: given a document d and query q, we calculate the semantic similarity of each word from d with each query term $t \in q$ and denote the average of these similarities as $\sigma(d, q)$. Given two documents d_1 and d_2, the one having the larger σ value is preferred. Along similar lines, we generalize the formulation of STMC2: Given a pair of documents d_1 and d_2, we identify the non-query term t from any of both documents that is maximally similar to any query term t'. We conclude $d_1 >_{\text{STMC2}} d_2$ iff $|d_2|/|d_1| =_{approx_{20\%}} tf(t, d_2)/tf(t', d_1)$.

Query Aspect Axioms The axioms REG and AND [213, 224] focus on the individual query terms. We modify REG as follows: let t be the query term most similar to all other query terms; if both d_1 and d_2 contain all the other query terms, the document with the higher tf value for t is preferred. Our implementation of the AND axiom compares document pairs (d_1, d_2) where only d_1 contains all query terms, and prefers d_1. We adapt the original formulation of the diversity-inducing axiom DIV [76] as follows: let $J(d, q)$ be the Jaccard coefficient between the set of terms in document d and the set of query terms. If $J(d_1, q) < J(d_2, q)$, we conclude $d_1 >_{\text{DIV}} d_2$. To suppress duplicates among the top k results, we further propose a new axiom RSIM that computes the simhash-based similarities of all document pairs; from any similarity cluster thus formed, it only favors one particular document over all the others, while not having preferences for documents from different clusters.

Other Axioms We include a rather straightforward new axiom P-RANK that simply prefers the document with higher PageRank. All the other published axioms from the literature (as introduced in Section 2.5) are either too specific, could not be formulated in a triple formulation, are already

Table 5.1: Axioms included in our re-ranking scheme.

Purpose	Acronyms	Source	Incl.
Term frequency	TFC1–TFC3	[73]	Yes
	TDC	[73]	Yes
Document length	LNC1 + LNC2	[73]	Yes
	TF-LNC	[73]	Yes
	QLNC	[58]	No
Lower bound	LB1 + LB2	[126]	Yes
Query aspects	REG	[213, 224]	Yes
	AND	[213, 224]	Yes
	DIV	[76]	Yes
	RSIM	new	Yes
Semantic similarity	STMC1 + STMC2	[71]	Yes
	STMC3	[71]	No
	TSSC1 + TSSC2	[71]	No
Term proximity	QPHRA	new	Yes
	PROX1–5	new	Yes
	PHC + CCC	[184]	No
Other	ORIG	new	Yes
	P-RANK	[6]	Yes
	CPRF	[50]	No
	CTM	[102]	No
	CMR	[75]	No
	CEM	[7]	No

covered by other used axioms, or do not target retrieval (e.g., they axiomatize evaluation instead) such that we do not implement them here. Besides the axioms described above, and our new proximity axioms explained in the next section, we also include a simple axiom ORIG that represents the original top k ranking as returned by the base retrieval model (and does not modify any rank decisions). Its purpose is to give some "voting power" in the subsequent rank aggregation to the reasonable ideas underlying the base retrieval model.

Summary Table 5.1 lists the known axioms and whether we include them in our axiomatic re-ranking framework, including our newly developed term proximity axioms explained in the next section, and the aforementioned ORIG axiom as a fallback option if no other axiom is effective.

5.1.3 Our New Term Proximity Axioms

As a first "proximity"-style axiom, we propose the new QPHRA axiom, aimed at queries that contain phrase search operators (i.e., double quotes in typical search engine syntax): A document that contains all the query phrases is favored over a document that does not. Beyond this, the existing literature is not completely devoid of suggestions regarding term proximity—Tao and Zhai [184] propose two axiomatic constraints on proximity importance for retrieval. However, those axioms formalize desirable properties a distance measure should have, and are thus not meaningful in our rank preference framework. Hence, we propose the new term proximity axioms PROX1–PROX5, inspired by Tao and Zhai's ideas, below. As a common setting to all of them, let $q = \{t_1, \ldots, t_n\}$ be a multi-term query and d_1, d_2 be two different documents. We further assume that all query terms appear in both of the documents: $\forall t_j \in q : tf(t_j, d_i) > 0$, for $i \in \{1, 2\}$.

Our first proximity axiom captures proximity via the average position difference of all query term pairs.

PROX1: Let $\pi(q, d)$ be the average difference of query term pair positions calculated as follows:

$$\pi(q, d) = \frac{1}{|P|} \sum_{(i,j) \in P} \delta(d, i, j)$$

where $P = \{(i, j) \mid i, j \in q, i \neq j\}$ is the set of all query term pairs and $\delta(d, i, j)$ calculates the average number of words between the terms t_i and t_j in the document d based on all positions in d where t_i and t_j occur.

If $\pi(q, d_1) < \pi(q, d_2)$, we conclude $d_1 >_{\text{PROX1}} d_2$ assuming that a document with a lower π value—which corresponds to pairs of query term being closer together on average—should get a better rank.

We note two caveats regarding PROX1: First, it uses all pairs of term occurrences in a document, while naturally many of these will be far apart even if some pairs are very close together. Second, it might also be desirable for the first close co-occurrence of query terms to be near the document's beginning, such that the searcher will encounter them early while reading. The following axioms PROX2 and PROX3 address these issues.

PROX2: Let $\mathit{first}(t_i, d)$ be the position (i.e., word index) of the first occurrence of query term t_i in document d and let $\mu(d, q)$ be the sum of first positions over all query terms. If $\mu(d_1, q) < \mu(d_2, q)$, we conclude $d_1 >_{\text{PROX2}} d_2$.

Axiom PROX2 considers each query term separately, disregarding whether documents contain phrases from the query.

PROX3: Let $\tau(d, q)$ be position of the first occurrence of the entire query q as one phrase in the document d; if q is not a phrase in d, we set $\tau(d, q) = \infty$. If $\tau(d_1, q) < \tau(d_2, q)$, we conclude $d_1 >_{\text{PROX3}} d_2$.

A problem of PROX3 is that important documents may not contain the whole query as one phrase, but many subsets of the query terms as shorter phrases. The following axiom measures proximity using the closest tuples of query terms.

PROX4: Let $\omega(d, q)$ be a pair $(a, -b)$, where a is the number of non-query words within the closest grouping of all terms from query q in document d, and b is the number of times such a minimal grouping occurs in d. If $\omega(d_1, q) < \omega(d_2, q)$, we conclude $d_1 >_{\text{PROX4}} d_2$.

We assume that the document with the lower ω value better matches the query, since all query terms are closer together in the document. Further improving the proximity notion, we propose an axiom PROX5 based on the width of the smallest word window that contains all query terms.

PROX5: Given a query term $t_i \in q$ and a document d, we determine the size of the smallest text span containing all query terms around each occurrence of t_i. Let $\bar{s}(d, q)$ be the average smallest text span across all occurrences of all query terms in d. If $\bar{s}(d_1, q) < \bar{s}(d_2, q)$, we conclude $d_1 >_{\text{PROX5}} d_2$.

Having established the set of axioms that are part of our framework, we now consider how to handle situations where multiple axioms express (possibly conflicting) preferences.

5.1.4 Rank-aggregation

As stated previously, each axiom's ranking preferences for a given top k result set are expressed as a matrix M_A, whose elements (i, j) determine whether or not document d_i should be ranked before document d_j according to axiom A. In order to re-rank a top k result set based on these preferences, we derive an aggregation function that yields a single, combined preference matrix M using a machine learning model described in detail in Section 5.1.5.

However, after axiom preference aggregation, the resulting matrix M is likely to contain conflicts: if for instance $M[i, j] > M[j, i]$ and $M[j, k] >$

$M[k,j]$ but $M[k,i] > M[i,k]$, it is not clear what document to rank the highest. Rank aggregation problems of this kind are typical to the domain of computational social choice, and a variety of possible rank aggregation schemes exist to address them [48]. In what follows, we choose the Kemeny rank aggregation scheme, since it has been shown beneficial in meta-search engines [65]. Kemeny rank aggregation merges m rankings into one global ranking while minimizing a distance function to the original m rankings (e.g., the number of pairs that are ranked in a different ordering) [105].

Solving Kemeny rank aggregation is a well known NP-complete problem [90], but variety of viable approximation schemes have been proposed. Out of these, we choose the KwikSort algorithm presented by Ailon et al. [4]. As originally formulated, KwikSort solves the minimum feedback arc set problem in weighted tournaments, but can be easily transferred to our setting, since the matrix M can be viewed as the incidence matrix of a directed weighted tournament graph with the vertex set $V = \{d_1, \ldots, d_n\}$.

5.1.5 Learning Axiom Preference Aggregation

We use the 23 axioms shown in Table 5.1 based on various ideas regarding the quality of a result ranking (term frequency, proximity, etc.). Given a query q and a pair of documents (d_i, d_j) from the result set for q, each axiom A may express a preference for ranking d_i higher ($M_A[i,j] > M_A[j,i]$), lower ($M_A[i,j] < M_A[j,i]$) or the same ($M_A[i,j] = M_A[j,i]$) as d_j. In a set of documents with known relevance judgments, the optimal ordering for each document pair is known. Hence, we view the problem of axiom preference aggregation as a supervised classification problem at the level of document pairs, seeking to infer the aggregation function that best approximates the partial ordering induced by the relevance judgments.

We train a Random Forest classifier to predict the documents' relative ordering in an optimal ranking, using the individual axiom preferences as predictors, and relevance judgments as ground truth. For each document pair, we assign a class attribute from the set $\{\text{lower}, \text{higher}, \text{same}\}$. Since the relative ordering of documents with the same relevance has no influence on the measured quality of the final ranking, we employ an instance weighting scheme that halves the impact of the "same" class. And since not all axioms might be equally important for different retrieval models (e.g., *tf-idf* already has a term frequency component), we train separate preference aggregation functions for each retrieval model.

Due to a lack of large click logs or other large-scale implicit user feedback on our side, we use the nDCG_{10} over relevance judgments for TREC

queries as the performance measure in our experiments (i.e., the normalized discounted cumulative gain over the top ten ranks of the result lists, as described in Section 2.1.2). We randomly split the queries into a training set to learn the retrieval-model-specific aggregation functions, and a test set to evaluate their retrieval performance before and after applying our axiomatic re-ranking scheme.

5.2 Evaluation on TREC Queries

Our experimental evaluation of the axiomatic re-ranking scheme is conducted as a large-scale study on the TREC Web tracks of 2009–2014 with a variety of basis retrieval models serving the initial top k results. For the experiments on the 200 queries from the Web tracks 2009–2012, we employ 16 different basis retrieval models included in the Terrier framework [147], which we use to index the ClueWeb09 Category B. For the 100 queries from the Web tracks 2013 and 2014, we use the TREC-provided Indri[2] and Terrier baselines for the ClueWeb12 as our two basis retrieval models.

To speed up the experimental process, we perform the training and testing of the axiom aggregation schemes, each axiom's individual ranking, and the KwikSort Kemeny rank aggregation on a 135 node Hadoop cluster that also hosts the ClueWeb09 and ClueWeb12 documents and corpus statistics (e.g., idf values) needed for some of the axioms.

5.2.1 Axiomatic Web Track Performance

We evaluate axiomatic re-rankings of the queries from the TREC Web tracks of 2009–2014. From the ClueWeb09-based Web tracks of 2009–2012, there are a total of 198 queries with available relevance judgments. After discarding 18 queries for which none of the basis retrieval models find any relevant results, we randomly select 120 of the remaining 180 queries as the training set, and use the other 60 as the test set. The 16 basis retrieval models shown in Table 5.2 are employed; a few basics on the functioning of retrieval models are discussed in Section 2.1, and further details on these models in particular can be found in the extensive Terrier documentation.[3] We have set up Terrier to index the Category B part of the ClueWeb09 and train the axiom aggregation functions for each model separately on the training set topics as described in Section 5.1.

[2] http://lemurproject.org/clueweb12/services.php
[3] http://terrier.org/docs/v4.0/configure_retrieval.html

Table 5.2: Average retrieval performance (nDCG_{10}) of the different retrieval models over the 60 test set queries. Each row shows the basis model's performance (Basis), with axiomatic re-ranking (+AX), and with MRF term dependence. Significant differences between Basis/+AX, Basis/MRF and MRF/MRF+AX (paired two-sided t-test, $p \leq 0.05$) are marked with a dagger†; the effect size (Cohen's d) is given in brackets below each value. The final column shows the nDCG_{10} of the best possible re-ranking.

Model	Basis	+AX	MRF	MRF+AX	max
DPH	0.273	**0.291**	**0.307**†	**0.314**	0.642
		(0.062)	*(0.112)*	*(0.025)*	
DFRee	0.205	**0.236**	**0.230**	**0.245**	0.599
		(0.121)	*(0.091)*	*(0.057)*	
In_expC2	0.205	**0.214**	**0.229**	**0.238**	0.591
		(0.038)	*(0.091)*	*(0.031)*	
TF_IDF	0.202	**0.228**	**0.239**	0.200	0.589
		(0.098)	*(0.134)*	*(-0.155)*	
In_expB2	0.201	**0.202**	**0.234**	**0.237**	0.592
		(0.006)	*(0.124)*	*(0.011)*	
DFReeKLIM	0.199	**0.213**	**0.224**	0.224	0.591
		(0.057)	*(0.095)*	*(-0.001)*	
BM25	0.198	0.188	**0.229**	0.216	0.587
		(-0.044)	*(0.116)*	*(-0.049)*	
InL2	0.197	0.197	**0.235**	0.212	0.593
		(-0.001)	*(0.139)*	*(-0.091)*	
BB2	0.195	**0.197**	**0.236**†	0.234	0.587
		(0.005)	*(0.151)*	*(-0.006)*	
DFR_BM25	0.194	**0.206**	**0.236**	0.220	0.591
		(0.049)	*(0.156)*	*(-0.062)*	
LemurTF_IDF	0.187	**0.224**†	**0.221**†	**0.237**†	0.576
		(0.151)	*(0.132)*	*(0.060)*	
DLH13	0.164	**0.187**	**0.184**	**0.201**	0.499
		(0.100)	*(0.080)*	*(0.067)*	
PL2	0.16	**0.213**†	**0.190**†	**0.211**	0.550
		(0.221)	*(0.125)*	*(0.084)*	
DLH	0.153	**0.187**	**0.181**	**0.197**	0.470
		(0.144)	*(0.113)*	*(0.064)*	
DirichletLM	0.139	**0.242**†	**0.192**†	**0.253**†	0.564
		(0.456)	*(0.276)*	*(0.249)*	
Hiemstra_LM	0.107	**0.167**†	**0.161**†	**0.163**	0.397
		(0.277)	*(0.245)*	*(0.005)*	

The evaluation results on the test set topics are depicted in Table 5.2. The models in the table are ordered according to their "Base" performance without axiomatic re-ranking. For the sake of fairness, it should be pointed out that the poor performance of the Hiemstra_LM retrieval model, in particular, reveals no fundamental flaw with that model per se, but an error in its implementation in the Terrier IR platform, which has since been corrected.[4] The experiments described in this section employ the faulty Hiemstra_LM implementation—but even with that caveat, the numbers indicate that axiomatic reranking can repair even badly broken result rankings to some extent. After the average basis performance over all test set topics, shown in the second column of the table, the two subsequent columns show the nDCG_{10} after applying axiomatic re-ranking ("+AX"), and Terrier's Markov Random Field term dependency score modifier ("MRF") to the basis result set, respectively. We note that while MRF term dependency improves upon the average basis performance in all cases, the magnitude of the improvement is larger for axiomatic re-ranking for nearly half of the studied retrieval models. The fifth column shows the average nDCG_{10} when applying axiomatic re-ranking after MRF term dependency; the effect sizes reported in this column are computed with respect to the "MRF" values. The final column of Table 5.2 shows the maximum nDCG_{10} achievable on the basis model's top 50 result set—i.e., when ranking these documents in an "oracle"-style directly by their TREC relevance judgments—this gives an indication of the quality of the whole result set that the reranking model had available, and indicates an upper bound of what could be possible with a hypothetical ideal axiom combination.

Except for two retrieval models of middling performance, our axiomatic re-ranking consistently improves the average basis retrieval performance. This improvement is statistically significant (paired two-sided t-test, $p \leq 0.05$) for only four retrieval models at the lower end of the performance spectrum; however, we note that MRF term dependency achieves a significant improvement in only two further cases, while the magnitude of the effect tends to be smaller. Even on our fairly small test set, axiomatic retrieval yields a mid-sized effect on the performance of poorly performing basis retrieval models. It should be noted that the performance improvements seen especially for the models with weaker basis performance only come from the axiomatic re-ranking of the top 50 results of the weak model, not by incorporating knowledge from the better performing models, or any documents ranked at positions beyond 50 by the weak model.

[4] http://web.archive.org/web/20171229220200/http://terrier.org/issues/browse/TR-183

Table 5.3: Retrieval performance (nDCG_{10}) on the Web track 2014 topics before and after applying the axiomatic re-ranking approach. The axiom aggregation functions are trained on the topics of the Web track 2013. Significant differences between before and after (paired two-sided t-test, $p \leq 0.05$) are marked with a dagger (†) and effect size according to Cohen's d is given.

Model	Before	After	Effect size
Terrier DPH	0.471	0.446	-
Indri LM	0.346	0.502†	0.69

There are several interesting observations from these initial experiments. First, the retrieval model with the second worst basis performance (DirichletLM) achieves the second best performance after axiomatic re-ranking, both with and without MRF term dependency scoring. Second, the differences between retrieval models after re-ranking are smaller than before. However, this leveling effect is not due to the re-ranked results being almost optimally ranked. As the final column of Table 5.2 shows, none of the studied re-ranking approaches achieve more than half of the nDCG_{10} of the optimal re-ranking, on average; there is a considerable potential for improvement in moving the retrieval performance closer to the optimum with stronger axioms. Future re-ranking ideas probably would need to include axioms capturing more sophisticated relevance signals than the rather simplistic assumptions of the axioms used here. We will shed some more light on the influence of the different axioms in another experiment below.

Before analyzing the individual axioms' impact, we conduct an experiment similar to the above for the TREC Web track baselines of the years 2013 and 2014. We did not index the ClueWeb12 ourselves for this experiment but relied on the rankings provided by the Web track organizers as the baselines. In this run, the preference aggregation schemes are trained on the topics of the Web track 2013 and tested on the topics of 2014, yielding 50 topics each for training and testing. The results are depicted in Table 5.3.

As can be seen, the performance of the Indri baseline is significantly improved with a medium effect size while the Terrier baseline's performance is decreased—although not significantly. One possible explanation for the decreased Terrier DPH performance is that for this ClueWeb12 experiment, we only used 50 topics for training, while for the ClueWeb09 experiments we used 120 topics. Applying the aggregation function trained for DPH on the Web track 2009–2012 topics to the Web track 2014 test set yields a slight, albeit non-significant, performance improvement to an nDCG_{10} of 0.48 for DPH on the Web track 2014 topics. This indicates that the fifty

Web track 2013 topics might not suffice to train a good aggregation function for axiomatic re-ranking of DPH results.

Similarly to the ClueWeb09 setting, this experiment again indicates that axiomatic re-ranking can even out performance discrepancies between different basis retrieval models, such that the specific model used for the initial top k retrieval has less of an impact. Still, there is room for further improvement, as can be seen from the possible performance given an optimally re-ranked top k result set—for the Web track 2013/14 experiment, the optimal nDCG_{10} is close to 1.0 on average.

5.2.2 Impact of the Different Axioms

To gain further insights into the influence of the different axioms, we analyze the ClueWeb09 experiment in more detail. In particular, we investigate the performance of different axiom subsets, and how often they are applied and actually change the ranking decisions compared to the ORIG axiom that reproduces the basis model's ranking. Further, we analyze the overlap of the different retrieval models' top k results to account for the more homogeneous performance of the different retrieval models after axiomatic re-ranking.

Axiom subsets We study the influence of different axioms in the setting of ClueWeb09 experiment described earlier. We run the same experimental process (learning the aggregation function on the 120 training set topics, testing on the 60 other topics) for individual subsets of axioms and for the set of all axioms without each subset (the ORIG axiom is always included). A summary of the results is shown in Table 5.4; refer to either Table 5.1 or 5.5 for the individual axioms included in each of the subsets.

As the top half of Table 5.4 shows, of the six axiom subsets—document length, lower bound, query aspects, semantic similarity, term frequency, and proximity—query aspects and semantic similarity don't improve any of the retrieval models by themselves. The other four groups improve at least one model on their own, with the term proximity axioms improving the largest number of basis retrieval models, albeit by a small percentage.

A further observation can be made about subsets containing all axioms except one of the groups, as shown in the lower half of the table: Without the lower bound axioms, without the query aspects axioms, and without the proximity axioms, the fewest improvements are possible. This hints at the relative importance of these axioms. Without document length, 10 improvements are still possible. For all of the axiom subsets, the relative im-

Table 5.4: Improvements in nDCG_{10} on the test set for different axiom subsets. The second column shows the number of retrieval models (out of 16) whose performance is improved on average across the test set topics. The last column shows the average difference in nDCG_{10} across all models.

Axiom Subset	Improved	Avg. Diff.
Term frequency axioms only	5	-0.80%
Document length axioms only	1	-0.02%
Lower bound axioms only	2	-7.79%
Query aspects axioms only	0	-15.62%
Semantic similarity axioms only	0	-14.70%
Term proximity axioms only	6	+1.37%
All without term frequency	5	-1.78%
All without document length	10	+4.54%
All without lower bound	1	-6.55%
All without query aspects	1	-11.57%
All without semantic similarity	6	-1.24%
All without term proximity	2	-5.98%

provements in nDCG_{10} are much smaller than for the full set. This hints at a rather complex interplay between the different axioms in achieving a better top 10 ranking.

The subset experiments show large differences in the importance of individual axioms that we further examine by analyzing the impact of the different axioms in the preference aggregation function, and to how many document pairs the different axioms could be applied.

Axiom importance, usage and rank differences In order to examine the different axioms' importance, we study how much they contribute to the performance of the learned preference aggregation functions. Table 5.5 shows the mean decrease in model accuracy for the axiom preference aggregation functions of the best and worst performing basis retrieval model from the experiment shown in Table 5.2. For each axiom, the corresponding value in the table shows by what percentage the aggregation model's accuracy (at predicting the correct ordering of a given document pair) would decrease without that variable. The contributions of the different axioms tend to be fairly similar across retrieval models, but there are some key differences. The contribution of the ORIG axiom appears to decrease with the performance of the basis retrieval model. While certain axioms never have a large impact on the aggregation functions (TDC, TF-LNC and LB2), there

Table 5.5: Feature importance for a selection of axioms (by mean decrease in accuracy without that axiom) in the axiom preference aggregation function for the best- and worst-performing basis model.

Purpose	Axiom	Decrease Accuracy DPH	Hiemstra_LM
Term frequency	TFC1	19.21	10.53
	TFC2	9.70	1.97
	TDC	0.15	1.99
Document length	LNC1	3.18	3.93
	TF-LNC	1.44	0.00
Lower bound	LB1	33.22	26.04
	LB2	5.54	3.73
Query aspects	REG	21.33	20.31
	DIV	27.71	24.85
Semantic similarity	STMC1	31.18	27.32
	STMC2	15.25	15.16
Term proximity	PROX1	19.41	18.64
	PROX2	16.61	12.96
	PROX3	25.08	18.59
	PROX4	17.76	17.84
	PROX5	17.60	15.66
Other	P-RANK	18.77	12.79
	ORIG	23.54	14.82

is at least one high-impact axiom in almost all axiom groups. An exception are the document length axioms that never contribute more than five percent to the aggregation accuracy.

In a related avenue of enquiry, we examine how many ranking preferences of $d_i >_A d_j$ each individual axiom A specifies in the ClueWeb09 experiment—i.e., how often its *preconditions* are met. The distributions are quite similar for the different retrieval models. Interestingly, STMC1 (semantic similarity) is applied most frequently by far, but as can be seen from the axiom subsets experiment, it often probably draws non-useful conclusions. The axioms PROX2 and LB1 are the second most commonly applied, followed by the other proximity axioms, then TFC1, TFC2 and LNC1. By contrast, the axioms LNC2, TFC3 and TF-LNC are used very rarely.

To underpin this investigation, we study the difference in the top 10 result rankings caused by individual axioms. Again, STMC1 alone yield the

highest difference in the rankings, but it does not have a high impact on any of the learned aggregation functions. The term proximity axioms, as well as TFC1 and LB1, change about 50% of the top 10 result sets. Given the high impact of especially PROX3 and LB1, along with the axiom subset results, this indicates that their share of the improved re-ranked performance is the highest. The axioms LNC1, TF-LNC, TDC, and LB2 hardly ever change a top-10 ranking by themselves. Along with their lower impact, this indicates that they are the least important among our selection.

Result overlap of the basis models As mentioned previously, a particular outcome of our reranking experiments is that the performance measurements of different basis retrieval models become more similar after re-ranking than they were before. A large overlap between the top k result sets of the different retrieval models would explain not just this effect, but also the rather similar axiom impact across models.

To analyze the overlap, we measure the Jaccard coefficient between any two basis models' top 50 results. The average Jaccard coefficient of the 7,200 possible pairs is 0.6, confirming our suspicion of a large degree of overlap. Furthermore, limiting the analysis to the documents with a TREC judgment of 2 or higher (i.e., those judged at least highly relevant by TREC's assessors), the average overlap increases to 0.8. When these documents are moved to the top of a result list by the re-ranking process, the increase in nDCG_{10} is especially pronounced (see also Section 2.1.2). Since these highly relevant documents are treated similarly for individual basis retrieval models by the similarly aggregated axiom combinations, this explains the leveling effect in retrieval performance that we have observed.

5.3 Conclusion

In an attempt to answer Research Question 6 (Axiomatic Result Reranking), we have introduced an axiom-based framework to re-rank a basis retrieval model's top k results. This way, we exploit all the findings from the last decade on axiomatic information retrieval in a unified setup. For the first time, we demonstrate how a variety of axioms can be used in a practical setting. Our experimental analyses show the axiom-based re-ranking to improve retrieval performance for almost all of the basis retrieval models studied—often with a medium effect size. That said, our experiments also show that there is much room for further improvement, since hypothetical optimal re-rankings of the top k result sets would perform much better still.

Since our framework can be easily extended to include further preference-inducing axioms, investigating new or differently formulated axioms is a promising direction besides improving the actual aggregation scheme. Worthwhile candidates include axioms that we have not yet been able to adapt to our scheme, or in a highly modified form (e.g., semantics or query aspects), axioms capturing not-yet-axiomatized retrieval aspects (e.g., search sessions or missions [80], readability, freshness, usefulness), or improved formalizations of already included axiomatic ideas (e.g., better proximity preferences based on query segmentation information [79]). The inclusion of further axioms should further increase retrieval performance, since many facets of relevance (or other notions of result set quality) are unlikely to be fully covered by the axioms already included.

Aside from further improving retrieval with axioms, another important topic for future research is the efficiency of the axiomatic re-ranking process: while we have demonstrated the potential effectiveness (i.e., improvements of retrieval performance), the current implementation exhibits a run time of several seconds for re-ranking a single top 50 result set, on average. This is not tolerable for use in a live system. However, since our current experimental setup has not yet been optimized for speed, the necessary efficiency gains to reach practical applicability may well be achievable.

6 Conclusion

This chapter concludes the dissertation; it reviews our main findings with respect to the research questions from Chapter 1, and possible implications and applications thereof, in Section 6.1. Afterwards, Section 6.2 discusses aspects of our research questions that remain unanswered, as well as open problems and follow-up questions for future work.

6.1 Main Findings and Implications

As discussed back in Chapter 1, and illustrated in Figures 1.1 and 1.2 on pages 3 and 4, Chapters 3 through 5 study different aspects of the task-based web-search process, and make contributions from different perspectives: Chapter 3 assumes a user-focused perspective, and investigates the collection and exploitation of context information during complex writing tasks. Chapter 4 takes a partially user-focused and system-focused view, studying both query preprocessing and log analysis. Chapter 5 makes the most system-oriented contribution and focuses on result set postprocessing considerations primarily related to properties of the retrieval model

Chapter 3 introduces the Webis-TRC-12 dataset, a corpus of combined search engine interaction and writing logs of 150 long essays composed by 12 writers during an extensive crowdsourcing study. The dataset is made available to the research community at large[1] to foster future insights into search and writing behavior. Since the dataset builds upon standardized text retrieval resources—namely, the ClueWeb09 web crawl and TREC

[1] It can be accessed via `https://webis.de/data/webis-trc-12.html`.

queries—we expect it will be conducive to a variety of research tasks, both those discussed in later sections of Chapter 3, as well as new tasks not yet conceived. With respect to our own research, the dataset has enabled insights into the essay writers' behavior, into quantifying the usefulness of search results for writing tasks with text reuse, and into predicting how successful writers are at their search tasks based on the observed behavior.

First, Chapter 3 provides answers to Research Question 1a (Writing Strategies) and Research Question 1b (Searching Strategies): we discover that the writers in our essay-writing-with-text-reuse scenario follow one of two distinct writing strategies, which we call *build-up* and *boil-down* writing. The former is characterized by incremental, targeted material gathering while writing: build-up writers switch back and forth between material gathering and writing continuously, immediately integrating each newly retrieved source into the developing essay right after discovering it. By contrast, boil-down writers exhibit a stricter separation between a material-gathering and a writing phase: they first collect many sources up-front, copy-pasting large amounts of text into the essay first, and then reorganizing the collected passages into a coherent essay in a second pass. The study participants did not adhere to the same strategy in all essays they wrote, but the vast majority exhibit a strong preference for one strategy or the other. Similarly, we find evidence of two contrasting searching strategies: some authors submit few queries, but click on many search results and follow long click trails; others submit a variety of queries, but are much more selective with the search results they actually visit. We call the former type of searchers *clickers*, and the latter *queriers*; aside from the query and click frequencies, we find another key distinction in the fact that clickers paste significantly more passages, from significantly more source documents, than queriers. A subsequent joint analysis of searching and writing strategies indicates that both may be independent: writing and searching strategies can be observed in all combinations, with no clear and obvious correlation.

Our insights into searching and writing strategies have practical implications on retrieval personalization: for instance, a web search engine that is able to detect whether the user follows the querier or clicker searching strategy could adapt its user interface accordingly. Queriers, who appear to explore the information space mainly through varied searches, are likely to benefit more from high-quality query suggestions, as well as the integration of varied information modalities into the search result page (as exemplified by Google's OneBox UI elements). Clickers, by contrast rely more on the documents linked from the result page, and would benefit more from highly optimized or specialized ranking functionality; for instance, the ranking

may trade off between diversity and relevance of the result lists depending on the query and available context information about the user's task. A hypothetical retrieval-and-writing system that detects that the user is a build-up writer would want to promote especially those results that are closely related to the current editing location in the text, whereas for a boil-down user, an accurate detection of the user's current working phase would be much more important: in the initial material gathering phase might benefit from highly diverse result sets, whereas during the later re-writing phase, search results should look more like those for a build-up user.

While investigating Research Question 2 (Measuring Usefulness), we identify text reuse as an instrument for measuring search result usefulness: the experimental setup of the Webis-TRC-12 dataset allows for precise observation of which documents our writers have reused by copy-pasting into their essay text—and consequently, which documents they considered useful. In a comparison of our writers' implicit usefulness judgments and pre-existing TREC relevance judgments for the essay topics, we find only a small overlap in the set of documents judged by our writers, and the TREC assessors, respectively. Those documents judged by both groups show a similar discrepancy between relevance and usefulness judgments as have been found in previous studies (some of which are discussed in Section 2.2.5). In order to answer Research Question 3 (Predicting Retrieval Success), we operationalize retrieval success in terms of reuse-based search result usefulness with two derived measures, one counting the number of times users extracted useful content from search results, and one counting the number of words thus extracted. We study two regression models predicting these retrieval success measures from user behavior while searching and writing. Our models differ in explanatory power, covering 90% and 68% of the variance of the respective retrieval success measures, but both share the number of clicks per query as the predictor with the strongest positive correlation with retrieval success, while measures of writing effort, as well as querying effort, are negatively correlated with success in both models.

These results seem to indicate that, of the previously identified searching strategies, the clicker group is more likely to be successful at retrieval. Along the same lines as the previously discussed ideas regarding retrieval personalization, this information is potentially useful to a retrieval system aiming to improve the support of struggling users—for instance, through specially tailored query suggestions (as has previously been suggested by Odijk et al. [146], for example). However, the exact nature of the causal relationship between retrieval success, and the associated behavioral signals, remains to be investigated.

Chapter 4 studies a large Russian-language search engine log containing queries in question form, with the goal of improving retrieval support for such question queries by way of query preprocessing techniques. Question queries are of growing importance to search engines—e.g., due to the increasing prevalence of voice search—and are more challenging to support due to their greater rarity. A first contribution from our investigation is a reliable data preprocessing pipeline which removes noise from the query log. As a result of this preprocessing, we have a clean dataset of about one billion question queries, down from initially two billion entries that included spam submissions by bots, and repeated or incomplete entries resulting from paging or instant search. Our end goal is to improve our understanding of question patterns in the query log, answering Research Question 5 (Question Query Patterns), and to ultimately exploit the gained insights for the purpose of improving retrieval performance. With this end in mind, we first address Research Question 4 (Question Query Classification), and design a flexible classification pipeline—which predicts the main topic for a given question query—with the help of a secondary collection comprising six and a half million questions posted to a Russian Community Question Answering (CQA) site.

Our approach is based on the working hypothesis that CQA data will be useful as a training dataset for classification in this setting, since CQA users themselves assign topics to the questions they submit, and that the characteristics of the two settings are sufficiently similar that a classifier trained on CQA can be successfully transferred to question queries. Section 2.2.3 mentions several past studies on topical categorization of CQA questions. However, in contrast to our approach, all these CQA methods do not extend beyond CQA (i.e., they use CQA data for learning and consequently perform classification on the data of the same origin). One of our contributions instead is to show how a classifier trained on CQA questions can be used to classify web search questions as well, affording the opportunity of including on-the-fly class information in the retrieval process for questions submitted to search engines. Based on these ideas, we design three variants for our classification pipeline—one based on retrieval of CQA questions from an inverted index and class assignment by majority vote, one based on a simple naïve Bayes classifier trained on a unigram representation, and one trained on a more compressed topic model based representation; all variants distinguish the same set of 14 target categories. While the retrieval-based variant achieves the highest classification accuracy, it is also the slowest by far. The topic-model based variants are especially efficient at prediction time, at the cost of a higher preprocessing and training overhead, and a loss in classifi-

cation accuracy by about 15%. The unigram-based approach is of middling efficiency, and nearly as accurate as the retrieval-based one.

To address Research Question 5 (Question Query Patterns), we hence select the unigram-based approach to classify all of the nearly one billion question queries in the log, due to its high accuracy and sufficient speed. Based on our findings, we explore patterns in the query stream that can help improve retrieval performance for question queries. Of these, the changes in prevalence over time of the different topics show the most promise: while some of the trends are unsurprising—such as travel-related questions becoming most prevalent, and education-related ones least prevalent, during the summer months—they can serve as useful context information to a query disambiguation system. As an example, consider a query about some geographic landmark—based on the previous observation, a travel-related interpretation should be assigned a higher probability during summer months, and an education-related one at other times; in practice this will of course be offset against the rest of the available context information, such as the user's immediately preceding queries.

Finally, Chapter 5 tackles the problem of result set postprocessing by way of axiomatic reranking. As discussed in Section 2.5, retrieval axioms aim to formally capture desirable properties of a result ranking, and have previously been applied primarily to the analysis and design of ranking functions. While developments like the $\mathrm{BM25}^{+}$ ranking function [126] as highlighted on page 39 have clearly benefited the field, past axiomatic research has generally only considered few axioms at a time. The implementation of new ranking functions into an existing retrieval system can be cumbersome, and may require re-indexing the entire collection. Hence, Chapter 5 poses Research Question 6 (Axiomatic Result Reranking), which targets an algorithmic axiomatic reranking framework that can integrate (appropriately formalized) axioms directly into the retrieval process, by using them to rerank the top k result set returned by an initial basis retrieval model.

To this end, we present a multi-stage approach that first retrieves an initial top k result set using the basis model, and for each axiom under consideration, computes a preference matrix that describes the optimal ranking according to that axiom. The individual preference matrices are then aggregated with the help of a machine learning model optimized with respect to the particular basis retrieval model used. Any contradictions are resolved with the help of a rank aggregation algorithm. We evaluate our approach with a large selection of basis retrieval models and axioms; our results show that significant improvements in retrieval performance are possible in many

cases, and that the variance in performance across different basis retrieval models decreases. This latter fact is notable, since it vindicates our assumption that some more or less reasonable choice of basis retrieval model is sufficient—the exact model chosen becomes less important after reranking.

As exemplified by our newly proposed axioms for integrating term proximity requirements into the retrieval process, axiomatic reranking has the potential to greatly speed up the turnaround in this research area. Using our framework, newly discovered axioms can be quickly evaluated with respect to their real-world impact on a variety of retrieval models. Conversely, axiomatic reranking can give a quick first indication of which axioms a new retrieval model may or may not satisfy, by looking at how much reranking with different axioms can improve that models' performance.

6.2 Open Problems and Future Work

To conclude this chapter, and the dissertation as a whole, we highlight some aspects of our research questions that remain unanswered, or that have interesting follow-up questions. In the process, we discuss also limitations of our research and promising avenues for future work.

The level of detail captured by the Webis-TRC-12 dataset presented in Chapter 3 is unprecedented, and it constitutes—to the best of our knowledge—the largest publicly available corpus of task-based searching and writing behavior. That said, a key limitation of the dataset presented lies in the number of participants in that study: while the twelve writers were sufficient to identify distinct writing and searching strategies, and to achieve statistically significant models of retrieval success, the question remains how well our findings will generalize to the general population; a replication study with a larger scope is desirable. Our experiment setup—which encouraged authors to reuse text wherever possible—may present a similar obstacle to the replicability of our findings; any follow-up studies should consider including a control group that cannot reuse, or not as much, and verify whether the same patterns can be observed then. It is conceivable that additional searching or writing strategies can be observed with a larger sample size, or more advanced analysis techniques.

Another interesting avenue for follow-up work to explore is a deeper look into the interplay between the searching and writing behaviors we have identified. Since the querier vs. clicker searching strategies appear to be largely independent of the build-up vs. boil-down writing strategies, there should, in theory, be some real-world users that occupy all four cells of the

resulting contingency table; a comparative analysis of the task outcomes of these four groups may be insightful. In terms of concrete measures of task outcome, we can go beyond the simple retrieval success notions we explored in Chapter 3. For instance, with some suitable quantification, the quality of the resulting essays is a target measure of interest, where an analysis of the causal relationship between search (and writing) behavior and outcome would likely be insightful.

Finally, the implications of our research in Chapter 3 on retrieval personalization warrant further inquiry. We consider retrieval personalization based on actual information use a promising proposition: the query and click variables we measure are already logged in standard search logs. Beyond that, modern web search engines tend to be operated by companies that also offer writing support tools, where they could measure text editing variables, as well. By showing that writers' aggregate retrieval success can be predicted by a simple model consisting of three variables, we have taken a first tentative step in this direction. Directly predicting the utility of individual candidate documents for a particular writing task will be important future work, in order to apply this idea in practice.

With respect to answering question queries, commercial web search engines have made immense progress in the few years since the study described in Chapter 4 was first published. As an example, at the time of this writing (early 2019), you can type `[how much does a tesla flamethrower cost]` into Google and receive, as the top result, a OneBox with an article covering the merchandise alluded to in the query, with the $500 price tag highlighted, the mis-appropriated entity notwithstanding.[2] For simple factoid queries—and a large number of `[how to]`-like instruction seeking inquiries—the web search question answering problem can be considered solved. That said, there are still questions that stump Google search: ask `[why is the word knowledge hyphenated "knowl-edge"]`, and you do not get an answer on the first result page—the query spelling correction unhelpfully suggests to get rid of the hyphen in the quoted segment.[3] This latter example points the way to what is still a frontier of question query research: questions requiring some kind of argumentative answer, like justifications or weighing of alternatives. While pioneering efforts at argument search are well underway [203], the technology currently isn't at a level of maturity where it could be integrated into a commercial search engine. Hence,

[2] In a January 2018 publicity stunt, "The Boring Company" sold flamethrower-shaped blowtorches as mechandise. They share a founder with Tesla, Inc., but are otherwise distinct.

[3] In American English, the rule is to hyphenate by pronunciation; the "l" in the first syllable is necessary to distinguish its pronunciation from the word "know."

a worthwhile follow-up study would focus on the development of question query mining techniques for specialized question types—such as argumentative or comparative questions—and deriving new datasets dedicated to the processing of these types of queries.

Our reranking experiments in Chapter 5 show that there is still room for improvement towards optimal top 50 rankings; more advanced axioms may be able to capture more fine-grained notions of relevance, and further increase ranking performance. Even more interesting is the possibility of retrieval axioms that go beyond the simple single-query relevance notions that we have considered so far. It should be possible to formulate axioms that govern the introduction of contextual information into the retrieval process, including previously submitted queries, clicked search results, or on-task behavior such as text writing. Axioms for specialized retrieval situations, such as question answering, are conceivable, as well.

Given the highly complex scoring functions used by commercial search engines nowadays, axioms can promise to make the processes behind web search ranking more transparent to users. While it is not possible to directly explain how a neural ranking model with millions of parameters arrives at a given result ranking, the retrieval system could fit an axiomatic ranking to reproduce the results of the neural ranker as closely as possible, and then use the parameters of the axiomatic model to synthesize an explanation of which factors were especially important to arrive at the initial ranking. For production deployment, however, the efficiency of the axiomatic rank aggregation still needs to be improved.

While the ranking preference aggregation function described in Chapter 5 uses TREC relevance judgments as training data, a different optimization target is possible here, as well. For instance, preference aggregation could be optimized towards task-based usefulness judgments (such as those derived from the writer behavior in Chapter 3), rather than single-query relevance. Appropriate training data may even be derived from the Webis-TRC-12 dataset and related future endeavors.

Teaching retrieval systems to rank based on usefulness judgments in such a way would tie the various contributions of this dissertation together. Given the ever-rising information flood we are subjected to, the demand for retrieval systems to better anticipate which information items will be useful to us seems likely to increase. There is little doubt that task-based web search, and the question how the search engines of tomorrow can better support it, will remain a fertile field of inquiry for the foreseeable future.

Bibliography

[1] Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. Learning User Interaction Models for Predicting Web Search Result Preferences. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 3–10, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148175.

[2] Eugene Agichtein, Ryen W. White, Susan T. Dumais, and Paul N. Bennet. Search, Interrupted: Understanding and Predicting Search Task Continuation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 315–324, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5. doi: 10.1145/2348283.2348328.

[3] Jae-wook Ahn, Peter Brusilovsky, Daqing He, Jonathan Grady, and Qi Li. Personalized web exploration with task models. In *Proc. WWW'08*, pages 1–10, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-085-2. doi: 10.1145/1367497.1367499. URL `http://doi.acm.org/10.1145/1367497.1367499`.

[4] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), 2008. doi: 10.1145/1411509.1411513. URL `http://doi.acm.org/10.1145/1411509.1411513`.

[5] Milad Alshomary, Michael Völske, Tristan Licht, Henning Wachsmuth, Benno Stein, Matthias Hagen, and Martin Potthast. Wikipedia Text Reuse: Within and Without. In Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra, editors, *Advances in Information Retrieval. 41st European Conference on IR Research (ECIR 2019)*, volume 11437 of *Lecture Notes in Computer Science*, pages 747–754, Berlin Heidelberg New York, April 2019. Springer. doi: 10.1007/978-3-030-15712-8_49.

[6] Alon Altman and Moshe Tennenholtz. Ranking systems: the pagerank axioms. In John Riedl, Michael J. Kearns, and Michael K. Reiter,

editors, *Proceedings 6th ACM Conference on Electronic Commerce (EC-2005), Vancouver, BC, Canada, June 5-8, 2005*, pages 1–8. ACM, 2005. doi: 10.1145/1064009.1064010. URL http://doi.acm.org/10.1145/1064009.1064010.

[7] Enrique Amigó, Julio Gonzalo, and Felisa Verdejo. A general evaluation measure for document organization tasks. In Gareth J. F. Jones, Paraic Sheridan, Diane Kelly, Maarten de Rijke, and Tetsuya Sakai, editors, *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*, pages 643–652. ACM, 2013. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484081. URL http://doi.acm.org/10.1145/2484028.2484081.

[8] Jaime Arguello. Predicting Search Task Difficulty. In Maarten Rijke, Tom Kenter, Arjen P. de Vries, Cheng Xiang Zhai, Franciska Jong, Kira Radinsky, and Katja Hofmann, editors, *Advances in Information Retrieval*, volume 8416 of *Lecture Notes in Computer Science*, pages 88–99. Springer International Publishing, 2014. ISBN 978-3-319-06027-9. doi: 10.1007/978-3-319-06028-6_8.

[9] Robert Armstrong, Dayne Freitag, Thorsten Joachims, and Tom M. Mitchell. WebWatcher: A Learning Apprentice for the World Wide Web. Technical report, AAAI, 1995.

[10] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. Improvements that don't add up: Ad-hoc Retrieval Results since 1998. In David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy J. Lin, editors, *Proceedings of the Eighteenth ACM Conference on Information and Knowledge Management*, CIKM '09, pages 601–610, Hong Kong, China, November 2009. ACM. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1646031.

[11] Anne Aula, Rehan M. Khan, and Zhiwei Guan. How Does Search Behavior Change As Search Becomes More Difficult? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 35–44, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753333. URL http://doi.acm.org/10.1145/1753326.1753333.

[12] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology behind Search*. Addison Wesley, New York, second edition edition, 2011. ISBN 978-0-321-41691-9.

[13] Ricardo A. Baeza-Yates, Arjen P. de Vries, Hugo Zaragoza, Berkant Barla Cambazoglu, Vanessa Murdock, Ronny Lempel, and Fabrizio Silvestri, editors. *Advances in Information Retrieval - 34th European Conference on IR Research, ECIR 2012, Barcelona, Spain, April 1-5, 2012. Proceedings*, volume 7224 of *Lecture Notes in Computer Science*, 2012. Springer. ISBN 978-3-642-28996-5. doi: 10.1007/978-3-642-28997-2. URL `http://dx.doi.org/10.1007/978-3-642-28997-2`.

[14] Peter Bailey, Ryen W White, Han Liu, and Giridhar Kumaran. Mining historic query trails to label long and rare search engine queries. *ACM Transactions on the Web (TWEB)*, 4(4):15, 2010.

[15] Judit Bar-Ilan, Zheng Zhu, and Mark Levene. Topic-specific analysis of search queries. In *Proceedings of the 2009 workshop on Web Search Click Data*, pages 35–42. ACM, 2009.

[16] Ranieri Baraglia, Fidel Cacheda, Victor Carneiro, Diego Fernández, Vreixo Formoso, Raffaele Perego, and Fabrizio Silvestri. Search shortcuts: a new approach to the recommendation of queries. In Lawrence D. Bergman, Alexander Tuzhilin, Robin D. Burke, Alexander Felfernig, and Lars Schmidt-Thieme, editors, *Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009*, pages 77–84. ACM, 2009. ISBN 978-1-60558-435-5.

[17] Jeff Barr and Luis Felipe Cabrera. AI gets a Brain. *Queue*, 4(4):24–29, May 2006.

[18] Steven M Beitzel, Eric C Jensen, Abdur Chowdhury, Ophir Frieder, and David Grossman. Temporal Analysis of a Very Large Topically Categorized Web Query Log. *Journal of the American Society for Information Science and Technology*, 58(2):166–178, 2007.

[19] Steven M Beitzel, Eric C Jensen, David D Lewis, Abdur Chowdhury, and Ophir Frieder. Automatic Classification of Web Queries Using Very Large Unlabeled Query Logs. *ACM Transactions on Information Systems (TOIS)*, 25(2):9, 2007.

[20] N.J. Belkin, M. Cole, and J. Liu. A model for evaluating interactive information retrieval. In *SIGIR Workshop on the Future of IR Evaluation, July 23, 2009, Boston*, 2009.

[21] Jerome R Bellegarda. Spoken Language Understanding for Natural Interaction: The Siri Experience. In *Natural Interaction with Robots, Knowbots and Smartphones*, pages 3–14. Springer, 2014.

[22] Micheal Bendersky and W. Bruce Croft. Analysis of Long Queries in a Large Scale Search Log. In *Proceedings of the Workshop on Web Search Click Data, WSCD '09, Barcelona, Spain, February 09, 2009*, pages 8–14, 2009. ISBN 978-1-60558-434-8.

[23] Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. Modeling the Impact of Short- and Long-term Behavior on Search Personalization. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 185–194, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5. doi: 10.1145/2348283.2348312.

[24] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003. ISSN 1532-4435.

[25] Cristiana Bolchini, Carlo A. Curino, Elisa Quintarelli, Fabio A. Schreiber, and Letizia Tanca. A Data-oriented Survey of Context Models. *SIGMOD Rec.*, 36(4):19–26, December 2007. ISSN 0163-5808. doi: 10/bfdcgv.

[26] K. D. Bollacker, S. Lawrence, and C. L. Giles. Discovering relevant scientific literature on the Web. *IEEE Intelligent Systems and their Applications*, 15(2):42–47, March 2000. ISSN 1094-7167. doi: 10/dmshmf.

[27] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, and Piero Fraternali. Liquid Query: Multi-domain Exploratory Search on the Web. In *Proceedings of the 19th international conference on World wide web*, WWW'10, pages 161–170, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772708.

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

[29] Andrei Z Broder, Marcus Fontoura, Evgeniy Gabrilovich, Amruta Joshi, Vanja Josifovski, and Tong Zhang. Robust Classification of Rare Queries Using Web Knowledge. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 231–238. ACM, 2007.

[30] Marc Bron, Jasmijn van Gorp, Frank Nack, Maarten de Rijke, Andrei Vishneuski, and Sonja de Leeuw. A Subjunctive Exploratory Search Interface to Support Media Studies Researchers. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR'12, pages 425–434, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5. doi: 10.1145/2348283.2348342.

[31] Jonathan Brown. Jonathon Fletcher: How the Forgotten Father of the Search Engine Was Trumped by Google. *The Independent*, September 2013.

[32] Peter Bruza and T. W. C. Huibers. Investigating aboutness axioms using information fields. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pages 112–121. ACM/Springer, 1994. ISBN 3-540-19889-X. URL `http://dl.acm.org/citation.cfm?id=188521`.

[33] Jay Budzik and Kristian J. Hammond. Watson : Anticipating and Contextualizing Information Needs. In *Proceedings of the ASIS Annual Meeting*, volume 36, pages 727–740, December 1999.

[34] Vannevar Bush. As We May Think. *The Atlantic*, July 1945.

[35] Luca Busin and Stefano Mizzaro. Axiometrics: An axiomatic approach to information retrieval effectiveness metrics. In Kurland et al. [113], page 8. ISBN 978-1-4503-2107-5. doi: 10.1145/2499178.2499182. URL `http://doi.acm.org/10.1145/2499178.2499182`.

[36] Katriina Byström and Preben Hansen. Conceptual framework for tasks in information studies. *Journal of the American Society for Information Science and Technology*, 56(10):1050–1061, August 2005. ISSN 1532-2890. doi: 10/dwt49b.

[37] Fei Cai, Shuaiqiang Wang, and Maarten de Rijke. Behavior-based personalization in web search. *Journal of the Association for Information Science and Technology*, 68(4):855–868, April 2017. ISSN 23301635. doi: 10/gdg4kh.

[38] Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. Large-Scale Question Classification in CQA by Leveraging Wikipedia Semantic Knowledge. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1321–1330. ACM, 2011.

[39] B. Barla Cambazoglu, Hugo Zaragoza, Olivier Chapelle, Jiang Chen, Ciya Liao, Zhaohui Zheng, and Jon Degenhardt. Early Exit Optimizations for Additive Machine Learned Ranking Systems. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 411–420, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-889-6. doi: 10.1145/1718487.1718538.

[40] Huanhuan Cao, Daxin Jiang, Jian Pei, Enhong Chen, and Hang Li. Towards Context-aware Search by Learning a Very Large Variable Length Hidden Markov Model from Search Logs. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 191–200, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526736.

[41] Xin Cao, Gao Cong, Bin Cui, and Christian S Jensen. A generalized framework of exploring category information for question retrieval in community question answer archives. In *Proceedings of the 19th international conference on World wide web*, pages 201–210. ACM, 2010.

[42] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In Zoubin Ghahramani, editor, *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 129–136. ACM, 2007. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273513. URL `http://doi.acm.org/10.1145/1273496.1273513`.

[43] Marc-Allen Cartright, Ryen W. White, and Eric Horvitz. Intentions and Attention in Exploratory Health Search. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR'11, pages 65–74, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2009929.

[44] Wen Chan, Weidong Yang, Jinhui Tang, Jintao Du, Xiangdong Zhou, and Wei Wang. Community Question Topic Categorization via Hierarchical Kernelized Classification. In *Proceedings of the 22nd ACM International Conference On Information & Knowledge Management*, pages 959–968. ACM, 2013.

[45] Wei-Fan Chen, Matthias Hagen, Benno Stein, and Martin Potthast. A User Study on Snippet Generation: Text Reuse vs. Paraphrases. In

41st International ACM Conference on Research and Development in Information Retrieval (SIGIR 2018), pages 1033–1036. ACM, July 2018. doi: 10.1145/3209978.3210149. URL http://doi.acm.org/10.1145/3209978.3210149.

[46] Wei-Fan Chen, Henning Wachsmuth, Khalid Al-Khatib, and Benno Stein. Learning to Flip the Bias of News Headlines. In *11th International Natural Language Generation Conference (INLG 2018)*, pages 79–88. Association for Computational Linguistics, November 2018. URL http://aclweb.org/anthology/W18-6509.

[47] Xueqi Cheng, Yanyan Lan, Jiafeng Guo, and Xiaohui Yan. BTM: Topic Modeling over Short Texts. *IEEE Transactions on Knowledge and Data Engineering*, page 1, 2014.

[48] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. A short introduction to computational social choice. In Jan van Leeuwen, Giuseppe F. Italiano, Wiebe van der Hoek, Christoph Meinel, Harald Sack, and Frantisek Plasil, editors, *SOFSEM 2007: Theory and Practice of Computer Science, 33rd Conference on Current Trends in Theory and Practice of Computer Science, Harrachov, Czech Republic, January 20-26, 2007, Proceedings*, volume 4362 of *Lecture Notes in Computer Science*, pages 51–69. Springer, 2007. ISBN 978-3-540-69506-6. doi: 10.1007/978-3-540-69507-3_4. URL http://dx.doi.org/10.1007/978-3-540-69507-3_4.

[49] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. Click Models for Web Search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(3):1–115, July 2015. ISSN 1947-945X. doi: 10/gdj32m.

[50] Stéphane Clinchant and Éric Gaussier. A document frequency constraint for pseudo-relevance feedback models. In Gabriella Pasi and Patrice Bellot, editors, *COnférence en Recherche d'Infomations et Applications - CORIA 2011, 8th French Information Retrieval Conference, Avignon, France, March 16-18, 2011. Proceedings*, pages 73–88. Éditions Universitaires d'Avignon, 2011. ISBN 978-2-35768-024-1. URL http://asso-aria.org/coria/2011/73.pdf.

[51] Stéphane Clinchant and Éric Gaussier. A theoretical analysis of pseudo-relevance feedback models. In Kurland et al. [113], page 6. ISBN 978-1-4503-2107-5. doi: 10.1145/2499178.2499179. URL http://doi.acm.org/10.1145/2499178.2499179.

[52] W. S Cooper. On selecting a measure of retrieval effectiveness. *JASIST*, 24(2):87–100, 1973.

[53] Gordon V. Cormack, Mark D. Smucker, and Charles L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, 14(5):441–465, 2011.

[54] W. Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley, Boston, 2010. ISBN 978-0-13-607224-9. OCLC: ocn268788295.

[55] W. Bruce Croft, Michael Bendersky, Hang Li, and Gu Xu. Query Representation and Understanding Workshop. *SIGIR Forum*, 44(2):48–53, January 2011. ISSN 0163-5840. doi: 10/dxs5dd.

[56] Ronan Cummins and Colm O'Riordan. An axiomatic comparison of learned term-weighting schemes in information retrieval: clarifications and extensions. *Artif. Intell. Rev.*, 28(1):51–68, 2007. doi: 10.1007/s10462-008-9074-5. URL `http://dx.doi.org/10.1007/s10462-008-9074-5`.

[57] Ronan Cummins and Colm O'Riordan. Analysing ranking functions in information retrieval using constraints. *Information Extraction from the Internet, CreateSpace Independent Publishing Platform (August 2009)*, 2009.

[58] Ronan Cummins and Colm O'Riordan. A constraint to automatically regulate document-length normalisation. In Xue-wen Chen, Guy Lebanon, Haixun Wang, and Mohammed J. Zaki, editors, *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 2443–2446. ACM, 2012. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2398662. URL `http://doi.acm.org/10.1145/2396761.2398662`.

[59] M. De Leeuw and E. De Leeuw. *Read Better, Read Faster: A New Approach to Efficient Reading*. Pelican Original. Penguin Books, 1965. URL `http://books.google.de/books?id=jW1EAAAAYAAJ`.

[60] Brenda Dervin. From the Mind's Eye of the User: The Sense-making Qualitative-quantitative Methodology. In Jack D. Glazier and Ronald R. Powell, editors, *Qualitative Research in Information Management*, volume 9, pages 61–84. Libraries Unlimited, Englewood, CO, 1992.

[61] Brenda Dervin. Sense-making theory and practice: An overview of user interests in knowledge seeking and use. *Journal of Knowledge Management*, 2(2):36–46, December 1998. ISSN 1367-3270. doi: 10/cfg9jm.

[62] Fan Ding and Bin Wang. An axiomatic approach to exploit term dependencies in language model. In Hang Li, Ting Liu, Wei-Ying Ma, Tetsuya Sakai, Kam-Fai Wong, and Guodong Zhou, editors, *Information Retrieval Technology, 4th Asia Infomation Retrieval Symposium, AIRS 2008, Harbin, China, January 15-18, 2008, Revised Selected Papers*, volume 4993 of *Lecture Notes in Computer Science*, pages 586–591. Springer, 2008. ISBN 978-3-540-68633-0. doi: 10.1007/978-3-540-68636-1_68. URL http://dx.doi.org/10.1007/978-3-540-68636-1_68.

[63] Gideon Dror, Yoelle Maarek, Avihai Mejer, and Idan Szpektor. From query to question in one click: suggesting synthetic questions to searchers. In *Proceedings of the 22nd international conference on World Wide Web*, pages 391–402. International World Wide Web Conferences Steering Committee, 2013.

[64] Huizhong Duan, Yunbo Cao, Chin-Yew Lin, and Yong Yu. Searching Questions by Identifying Question Topic and Question Focus. In *Proceedings of ACL-08: HLT*, pages 156–164, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[65] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In Vincent Y. Shen, Nobuo Saito, Michael R. Lyu, and Mary Ellen Zurko, editors, *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 613–622. ACM, 2001. ISBN 1-58113-348-0. doi: 10.1145/371920.372165. URL http://doi.acm.org/10.1145/371920.372165.

[66] Yuka Egusa, Hitomi Saito, Masao Takaku, Hitoshi Terai, Makiko Miwa, and Noriko Kando. Using a concept map to evaluate exploratory search. In *Proceedings of the third symposium on Information interaction in context*, IIiX'10, pages 175–184, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0247-0. doi: 10.1145/1840784.1840810.

[67] David Ellis. A Behavioural Approach to Information Retrieval System Design. *Journal of Documentation*, 45:171–212, 1989. doi: 10/fwd77b.

[68] Tamer Elsayed, Jimmy J. Lin, and Donald Metzler. When close enough is good enough: approximate positional indexes for efficient ranked retrieval. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 1993–1996, 2011.

[69] Hui Fang. A re-examination of query expansion using lexical resources. In Kathleen McKeown, Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, editors, *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 139–147. The Association for Computer Linguistics, 2008. ISBN 978-1-932432-04-6. URL http://www.aclweb.org/anthology/P08-1017.

[70] Hui Fang and ChengXiang Zhai. An exploration of axiomatic approaches to information retrieval. In Ricardo A. Baeza-Yates, Nivio Ziviani, Gary Marchionini, Alistair Moffat, and John Tait, editors, *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19, 2005*, pages 480–487. ACM, 2005. ISBN 1-59593-034-5. doi: 10.1145/1076034.1076116. URL http://doi.acm.org/10.1145/1076034.1076116.

[71] Hui Fang and ChengXiang Zhai. Semantic term matching in axiomatic approaches to information retrieval. In Efthimis N. Efthimiadis, Susan T. Dumais, David Hawking, and Kalervo Järvelin, editors, *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 115–122. ACM, 2006. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148193. URL http://doi.acm.org/10.1145/1148170.1148193.

[72] Hui Fang and ChengXiang Zhai. Axiomatic Analysis and Optimization of Information Retrieval Models. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 1288–1288, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2257-7. doi: 10.1145/2600428.2602294.

[73] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In Mark Sanderson, Kalervo Järvelin, James Allan, and Peter Bruza, editors, *SIGIR 2004: Proceedings of the 27th*

Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004, pages 49–56. ACM, 2004. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009004. URL http://doi.acm.org/10.1145/1008992.1009004.

[74] Hui Fang, Tao Tao, and ChengXiang Zhai. Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.*, 29(2):7, 2011. doi: 10.1145/1961209.1961210. URL http://doi.acm.org/10.1145/1961209.1961210.

[75] Shima Gerani, ChengXiang Zhai, and Fabio Crestani. Score transformation in linear combination for multi-criteria relevance ranking. In Baeza-Yates et al. [13], pages 256–267. ISBN 978-3-642-28996-5. doi: 10.1007/978-3-642-28997-2_22. URL http://dx.doi.org/10.1007/978-3-642-28997-2_22.

[76] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl, editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 381–390. ACM, 2009. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526761. URL http://doi.acm.org/10.1145/1526709.1526761.

[77] Tim Gollub, Michael Völske, Matthias Hagen, and Benno Stein. Dynamic Taxonomy Composition via Keyqueries. In *14th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2014)*, pages 39–48. ACM/IEEE, September 2014. ISBN 978-1-4799-5569-5. URL http://dl.acm.org/citation.cfm?id=2740769.2740777.

[78] J. Gwizdka. Characterizing relevance with eye-tracking measures. In *Proceedings of the 5th Information Interaction in Context Symposium*, pages 58–67. ACM, 2014.

[79] Matthias Hagen, Martin Potthast, Anna Beyer, and Benno Stein. Towards Optimum Query Segmentation: In Doubt Without. In Xuewen Chen, Guy Lebanon, Haixun Wang, and Mohammed J. Zaki, editors, *21st ACM International Conference on Information and Knowledge Management (CIKM 2012)*, pages 1015–1024. ACM, October 2012. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2398398.

[80] Matthias Hagen, Jakob Gomoll, Anna Beyer, and Benno Stein. From Search Session Detection to Search Mission Detection. In *10th Inter-*

national Conference Open Research Areas in Information Retrieval (OAIR 2013), pages 85–92. ACM, May 2013. URL `http://dl.acm.org/citation.cfm?id=2491769`.

[81] Matthias Hagen, Martin Potthast, Michael Völske, Jakob Gomoll, and Benno Stein. How Writers Search: Analyzing the Search and Writing Logs of Non-fictional Essays. In Diane Kelly, Rob Capra, Nick Belkin, Jaime Teevan, and Pertti Vakkari, editors, *1st ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR 2016)*, pages 193–202. ACM, March 2016. doi: 10.1145/2854946.2854969.

[82] Matthias Hagen, Michael Völske, Steve Göring, and Benno Stein. Axiomatic Result Re-Ranking. In *25th ACM International Conference on Information and Knowledge Management (CIKM 2016)*, pages 721–730. ACM, October 2016.

[83] Matthias Hagen, Martin Potthast, Payam Adineh, Ehsan Fatehifar, and Benno Stein. Source Retrieval for Web-Scale Text Reuse Detection. In Ee-Peng Lim, Marianne Winslett, Mark Sanderson, Ada Wai-Chee Fu, Jimeng Sun, J. Shane Culpepper, Eric Lo, Joyce C. Ho, Debora Donato, Rakesh Agrawal, Yu Zheng, Carlos Castillo, Aixin Sun, Vincent S. Tseng, and Chenliang Li, editors, *26th ACM International Conference on Information and Knowledge Management (CIKM 2017)*, pages 2091–2094. ACM, November 2017. doi: 10.1145/3132847.3133097.

[84] Matthias Hagen, Martin Potthast, Marcel Gohsen, Anja Rathgeber, and Benno Stein. A Large-Scale Query Spelling Correction Corpus. In Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White, editors, *40th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2017)*, pages 1261–1264. ACM, August 2017. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3080749.

[85] J. F. Hair, W. C. Black, B. J. Babin, and R.E Anderson. *Multivariate data analysis*. Prentice-Hall, New Jersey, 2010.

[86] Donna Harman. Towards Interactive Query Expansion. *ACM SIGIR Forum*, 51(2):79–89, August 1988. ISSN 01635840. doi: 10/gfj2vb.

[87] Morgan Harvey, Fabio Crestani, and Mark J. Carman. Building user profiles from topic models for personalised search. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 2309–2314, New York, NY,

USA, 2013. ACM. ISBN 978-1-4503-2263-8. doi: 10.1145/2505515.2505642.

[88] A. Hassan, R. Jones, , and K. Klinkner. Beyond DCG: user behavior as a predictor of a successful search. In *Proc. WSDM'10*, pages 221–230. ACM, 2010.

[89] Daqing He, Peter Brusilovsky, Jaewook Ahn, Jonathan Grady, Rosta Farzan, Yefei Peng, Yiming Yang, and Monica Rogati. An evaluation of adaptive filtering in the context of realistic task-based information exploration. *IP & M*, 44(2):511–533, 2008. ISSN 0306-4573. doi: 10/c9kggv.

[90] Edith Hemaspaandra, Holger Spakowski, and Jörg Vogel. The complexity of kemeny elections. *Theor. Comput. Sci.*, 349(3):382–391, 2005. doi: 10.1016/j.tcs.2005.08.031. URL `http://dx.doi.org/10.1016/j.tcs.2005.08.031`.

[91] William Hersh, Chris Buckley, T. J. Leone, and David Hickam. OHSUMED: an interactive retrieval evaluation and new large test collection for research. In *SIGIR'94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–201. Springer-Verlag New York, Inc., 1994. ISBN 0-387-19889-X.

[92] Djoerd Hiemstra and Claudia Hauff. MIREX: MapReduce information retrieval experiments. Technical Report TR-CTIT-10-15, University of Twente, 2010.

[93] Hugo C. Huurdeman and Jaap Kamps. From Multistage Information-seeking Models to Multistage Search Systems. In *Proceedings of the 5th Information Interaction in Context Symposium*, IIiX '14, pages 145–154, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2976-7. doi: 10.1145/2637002.2637020.

[94] Bernard Jansen, Amanda Spink, and Isak Taksa. Research and Methodological Foundations of Transaction Log Analysis. *Handbook of Research on Web Log Analysis*, pages 1–16, 2008.

[95] Kalervo Järvelin and Peter Ingwersen. Information Seeking Research Needs Extension Towards Tasks and Technology. *Information Research*, 10(1), 2004.

[96] Kalervo Järvelin, Pertti Vakkari, Paavo Arvola, Feza Baskaya, Anni Järvelin, Jaana Kekäläinen, Heikki Keskustalo, Sanna Kumpulainen, Miamaria Saastamoinen, Reijo Savolainen, and Eero Sormunen. Task-based information interaction evaluation: The viewpoint of program theory. *ACM Trans. Inf. Syst.*, 33(1):3:1–3:30, March 2015. ISSN 1046-8188. doi: 10.1145/2699660. URL `http://doi.acm.org/10.1145/2699660`.

[97] J. Jiang, D. He, and J. Allan. Comparing In Situ and Multidimensional Relevance Judgments. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 405–414, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5022-8.

[98] Thorsten Joachims. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM. ISBN 978-1-58113-567-1. doi: 10.1145/775047.775067.

[99] Thorsten Joachims, Dayne Freitag, and Tom M. Mitchell. Web Watcher: A Tour Guide for the World Wide Web. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, IJCAI 97, pages 770–777, Nagoya, Japan, 1997. Morgan Kaufmann.

[100] Rosie Jones and Kristina Lisa Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In James G. Shanahan, Sihem Amer-Yahia, Ioana Manolescu, Yi Zhang, David A. Evans, Aleksander Kolcz, Key-Sun Choi, and Abdur Chowdhury, editors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 699–708. ACM, 2008. ISBN 978-1-59593-991-3.

[101] Nattiya Kanhabua and Kjetil Nørv\a ag. Learning to Rank Search Results for Time-sensitive Queries. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2463–2466, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1156-4. doi: 10/gfs32g.

[102] Maryam Karimzadehgan and ChengXiang Zhai. Axiomatic analysis of translation language model for information retrieval. In

Baeza-Yates et al. [13], pages 268–280. ISBN 978-3-642-28996-5. doi: 10.1007/978-3-642-28997-2_23. URL http://dx.doi.org/10.1007/978-3-642-28997-2_23.

[103] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *Proc. SIGIR'04*, pages 377–384. ACM, 2004.

[104] Diane Kelly and Jaime Teevan. Implicit Feedback for Inferring User Preference: A Bibliography. *SIGIR Forum*, 37(2):18–28, September 2003. ISSN 0163-5840. doi: 10/d47tkq.

[105] John G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):pp. 577–591, 1959. ISSN 00115266. URL http://www.jstor.org/stable/20026529.

[106] J. Kim, J. Teevan, and N. Crasswell. Explicit in situ user feedback for web search results. In *Proc. SIGIR'16*, pages 829–832. ACM, 2016.

[107] 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*, WSDM '14, pages 193–202, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2351-2. doi: 10.1145/2556195.2556220.

[108] Jürgen Koenemann and Nicholas J. Belkin. A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '96, pages 205–212, New York, NY, USA, 1996. ACM. ISBN 978-0-89791-777-3. doi: 10/fpgn25.

[109] Katrin Köhler and Debora Weber-Wulff. Plagiarism detection test 2010. http://plagiat.htw-berlin.de/wp-content/uploads/PlagiarismDetectionTest2010-final.pdf, 2010.

[110] Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando, editors. *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, 2007. ACM. ISBN 978-1-59593-597-7.

[111] Carol C. Kuhlthau. Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science*, 42(5):361–371, June 1991. ISSN 1097-4571. doi: 10/cc9pnq.

[112] Bill Kules and Robert Capra. Creating Exploratory Tasks for a Faceted Search Interface. In *Proceedings of the Second Workshop on Human-Computer Interaction and Information Retrieval*, HCIR'08, pages 18–21, 2008.

[113] Oren Kurland, Donald Metzler, Christina Lioma, Birger Larsen, and Peter Ingwersen, editors. *International Conference on the Theory of Information Retrieval, ICTIR '13, Copenhagen, Denmark, September 29 - October 02, 2013*, 2013. ACM. ISBN 978-1-4503-2107-5. URL `http://dl.acm.org/citation.cfm?id=2499178`.

[114] Martin Kurth. The Limits and Limitations of Transaction Log Analysis. *Library Hi Tech*, 11(2):98–104, 1993.

[115] A. Lally, J. M. Prager, M. C. McCord, B. K. Boguraev, S. Patwardhan, J. Fan, P. Fodor, and J. Chu-Carroll. Question analysis: How Watson reads a clue. *IBM Journal of Research and Development*, 56(3.4):2:1–2:14, May 2012. ISSN 0018-8646, 0018-8646. doi: 10.1147/JRD.2012.2184637.

[116] David B. Leake, Travis Bauer, Ana Maguitman, and David C. Wilson. Capture, Storage and Reuse of Lessons about Information Resources: Supporting Task-Based Information Search. In *In Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems*, pages 33–37. AAAI Press, 2000.

[117] Baichuan Li, Irwin King, and Michael R Lyu. Question Routing In Community Question Answering: Putting Category In Its Place. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2041–2044. ACM, 2011.

[118] Xiao Li, Ye-Yi Wang, and Alex Acero. Learning Query Intent From Regularized Click Graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346. ACM, 2008.

[119] Xin Li and Dan Roth. Learning question classifiers. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*, 2002. URL `http://aclweb.org/anthology/C02-1150`.

[120] Jian Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. How Do Users Grow Up Along With Search Engines?: A Study Of Long-Term Users'

Behavior. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1795–1800. ACM, 2013.

[121] Jingjing Liu and Nicholas J. Belkin. Personalizing Information Retrieval for Multi-session Tasks: The Roles of Task Stage and Task Type. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 26–33, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835457.

[122] Qiaoling Liu, Eugene Agichtein, Gideon Dror, Evgeniy Gabrilovich, Yoelle Maarek, Dan Pelleg, and Idan Szpektor. Predicting Web Searcher Satisfaction With Existing Community-Based Answers. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 415–424. ACM, 2011.

[123] Qiaoling Liu, Eugene Agichtein, Gideon Dror, Yoelle Maarek, and Idan Szpektor. When Web Search Fails, Searchers Become Askers: Understanding The Transition. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 801–810. ACM, 2012.

[124] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[125] Claudio Lucchese, Salvatore Orlando, Raffaele Perego, Fabrizio Silvestri, and Gabriele Tolomei. Identifying task-based sessions in search engine query logs. In Irwin King, Wolfgang Nejdl, and Hang Li, editors, *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 277–286. ACM, 2011. ISBN 978-1-4503-0493-1.

[126] Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. In Craig Macdonald, Iadh Ounis, and Ian Ruthven, editors, *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 7–16. ACM, 2011. ISBN 978-1-4503-0717-8. doi: 10.1145/2063576.2063584. URL `http://doi.acm.org/10.1145/2063576.2063584`.

[127] Yuanhua Lv and ChengXiang Zhai. When Documents Are Very Long, BM25 Fails! In *Proceedings of the 34th International ACM SIGIR Con-*

ference on Research and Development in Information Retrieval, SIGIR '11, pages 1103–1104, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2010070.

[128] Yuanhua Lv and ChengXiang Zhai. A log-logistic model-based interpretation of tf normalization of bm25. In *Advances in Information Retrieval*, pages 244–255. Springer, 2012.

[129] Paul P. Maglio, Rob Barrett, Christopher S. Campbell, and Ted Selker. SUITOR: An Attentive Information System. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, IUI '00, pages 169–176, New York, NY, USA, 2000. ACM. ISBN 978-1-58113-134-5. doi: 10.1145/325737.325821.

[130] Ana Maguitman. Searching in the Context of a Task: A Review of Methods and Tools. *CLEI Electronic Journal*, 21(1):1, April 2018. ISSN 0717-5000. doi: 10/gfjzf9.

[131] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

[132] J. Mao, Y. Liu, K. Zhou, J. Nie, M. Zhang, and S. Ma. When does relevance mean usefulness and user satisfaction in web search. In *Proc. SIGIR'16*, pages 463–472. ACM, 2016.

[133] Jiaxin Mao, Yiqun Liu, Huanbo Luan, Min Zhang, Shaoping Ma, Hengliang Luo, and Yuntao Zhang. Understanding and predicting usefulness judgment in web search. In *Proc. SIGIR'17*, pages 1169–1172, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5022-8. doi: 10.1145/3077136.3080750. URL `http://doi.acm.org/10.1145/3077136.3080750`.

[134] Gary Marchionini. Exploratory Search: From Finding to Understanding. *Commun. ACM*, 49(4):41–46, April 2006. ISSN 0001-0782. doi: 10/cg3s62.

[135] Brian P. McCune, Richard M. Tong, Jeffrey S. Dean, and Daniel G. Shapiro. RUBRIC: A system for rule-based information retrieval. *IEEE Trans. Software Eng.*, 11(9):939–945, 1985. doi: 10.1109/TSE.1985.232827. URL `http://doi.ieeecomputersociety.org/10.1109/TSE.1985.232827`.

[136] Carlo Meghini, Fabrizio Sebastiani, Umberto Straccia, and Costantino Thanos. A model of information retrieval based on a terminological logic. In Robert Korfhage, Edie M. Rasmussen, and Peter Willett, editors, *Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993*, pages 298–307. ACM, 1993. ISBN 0-89791-605-0. doi: 10.1145/160688.160753. URL `http://doi.acm.org/10.1145/160688.160753`.

[137] Alessandro Micarelli and Filippo Sciarrone. Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. *User Modeling and User-Adapted Interaction*, 14(2-3):159–200, June 2004. ISSN 0924-1868, 1573-1391. doi: 10/d9g6dh.

[138] Masahiro Morita and Yoichi Shinoda. Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval. In Bruce W. Croft and C. J. van Rijsbergen, editors, *SIGIR '94*, pages 272–281. Springer London, 1994. ISBN 978-1-4471-2099-5.

[139] Dan Morris, Meredith Ringel Morris, and Gina Venolia. SearchBar: A Search-centric Web History for Task Resumption and Information Re-finding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'08, pages 1207–1216, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: 10.1145/1357054.1357242.

[140] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. What Do People Ask Their Social Networks, And Why?: A Survey Study Of Status Message Q&A Behavior. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1739–1748. ACM, 2010.

[141] Yashar Moshfeghi and Frank E. Pollick. Search Process As Transitions Between Neural States. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pages 1683–1692, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-5639-8. doi: 10.1145/3178876.3186080.

[142] Seung-Hoon Na, In-Su Kang, and Jong-Hyeok Lee. Improving term frequency normalization for multi-topical documents and application to language modeling approaches. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White, editors, *Advances in Information Retrieval , 30th European Conference on IR Research,*

ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings, volume 4956 of *Lecture Notes in Computer Science*, pages 382–393. Springer, 2008. ISBN 978-3-540-78645-0. doi: 10.1007/978-3-540-78646-7_35. URL http://dx.doi.org/10.1007/978-3-540-78646-7_35.

[143] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, and Dinh Q. Phung. A Capsule Network-based Embedding Model for Search Personalization. *CoRR*, abs/1804.04266, 2018.

[144] Heather L O'Brien and Elaine G Toms. What is user engagement? a conceptual framework for defining user engagement with technology. *Journal of the American Society for Information Science and Technology*, 59 (6):938–955, 2008.

[145] Heather L O'Brien and Elaine G Toms. The development and evaluation of a survey to measure user engagement. *Journal of the American Society for Information Science and Technology*, 61(1):50–69, 2010.

[146] Daan Odijk, Ryen W. White, Ahmed Hassan Awadallah, and Susan T. Dumais. Struggling and Success in Web Search. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1551–1560, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3794-6. doi: 10/gcsgws.

[147] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Christina Lioma. Terrier: A high performance and scalable information retrieval platform. In *Proceedings of the OSIR Workshop*, pages 18–25. Citeseer, 2006.

[148] Umut Ozertem, Olivier Chapelle, Pinar Donmez, and Emre Velipasaoglu. Learning to Suggest: A Machine Learning Framework for Ranking Query Suggestions. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 25–34, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1472-5. doi: 10.1145/2348283.2348290.

[149] Weike Pan and Li Chen. GBPR: Group Preference Based Bayesian Personalized Ranking for One-class Collaborative Filtering. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2691–2697, Beijing, China, 2013. AAAI Press. ISBN 978-1-57735-633-2.

[150] Bo Pang and Ravi Kumar. Search in the lost sense of query: Question formulation in web search queries and its temporal changes. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 135–140. Association for Computational Linguistics, 2011.

[151] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A Picture of Search. In Xiaohua Jia, editor, *Proceedings of the 1st International Conference on Scalable Information Systems, Infoscale 2006, Hong Kong, May 30-June 1, 2006*, volume 152 of *ACM International Conference Proceeding Series*, page 1. ACM, 2006. ISBN 1-59593-428-6.

[152] Michael Pazzani, Jack Muramatsu, and Daniel Billsus. Syskill & Webert: Identifying Interesting Web Sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, pages 54–61, Portland, Oregon, 1996. AAAI Press. ISBN 978-0-262-51091-2.

[153] Martin Potthast. *Technologies for Reusing Text from the Web*. Dissertation, Bauhaus-Universität Weimar, December 2011. URL `http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:gbv:wim2-20120217-15663`.

[154] Martin Potthast, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. An Evaluation Framework for Plagiarism Detection. In Chu-Ren Huang and Dan Jurafsky, editors, *23rd International Conference on Computational Linguistics (COLING 10)*, pages 997–1005, Stroudsburg, Pennsylvania, August 2010. Association for Computational Linguistics.

[155] Martin Potthast, Matthias Hagen, Benno Stein, Jan Graßegger, Maximilian Michel, Martin Tippmann, and Clement Welsch. ChatNoir: A Search Engine for the ClueWeb09 Corpus. In Bill Hersh, Jamie Callan, Yoelle Maarek, and Mark Sanderson, editors, *35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2012)*, page 1004. ACM, August 2012. ISBN 978-1-4503-1472-5. doi: 10.1145/2348283.2348429.

[156] Martin Potthast, Matthias Hagen, Michael Völske, and Benno Stein. Exploratory Search Missions for TREC Topics. In Max L. Wilson, Tony Russell-Rose, Birger Larsen, Preben Hansen, and Kristian Norling, editors, *3rd European Workshop on Human-Computer Interaction and Information Retrieval (EuroHCIR 2013)*, pages 11–14. CEUR-

WS.org, August 2013. URL http://www.cs.nott.ac.uk/~pszmw/euroHCIR2013/proceedings/paper3.pdf.

[157] Martin Potthast, Matthias Hagen, Michael Völske, and Benno Stein. Crowdsourcing Interaction Logs to Understand Text Reuse from the Web. In Pascale Fung and Massimo Poesio, editors, *51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1212–1221. Association for Computational Linguistics, August 2013. URL http://www.aclweb.org/anthology/P13-1119.

[158] Martin Potthast, Tim Gollub, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In Evangelos Kanoulas, Mihai Lupu, Paul Clough, Mark Sanderson, Mark Hall, Allan Hanbury, and Elaine Toms, editors, *Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 2014)*, pages 268–299, Berlin Heidelberg New York, September 2014. Springer. ISBN 978-3-319-11381-4. doi: 10.1007/978-3-319-11382-1_22.

[159] Bo Qu, Gao Cong, Cuiping Li, Aixin Sun, and Hong Chen. An Evaluation Of Classification Models For Question Topic Categorization. *Journal of the American Society for Information Science and Technology*, 63 (5):889–903, 2012.

[160] Yan Qu and George W. Furnas. Model-driven Formative Evaluation of Exploratory Search: A Study Under a Sensemaking Framework. *Information Processing and Management*, 44(2):534–555, 2008.

[161] Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback. In Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, editors, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, pages 239–248. ACM, 2005. ISBN 1-59593-135-X. doi: 10.1145/1081870.1081899. URL http://doi.acm.org/10.1145/1081870.1081899.

[162] Razieh Rahimi, Azadeh Shakery, and Irwin King. Axiomatic analysis of cross-language information retrieval. In Jianzhong Li, Xiaoyang Sean Wang, Minos N. Garofalakis, Ian Soboroff, Torsten Suel, and Min Wang, editors, *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM*

2014, Shanghai, China, November 3-7, 2014, pages 1875–1878. ACM, 2014. ISBN 978-1-4503-2598-1. doi: 10.1145/2661829.2661915. URL http://doi.acm.org/10.1145/2661829.2661915.

[163] Matthew Richardson. Learning About The World Through Long-Term Query Logs. *ACM Transactions on the Web (TWEB)*, 2(4):21, 2008.

[164] Phyllis A. Richmond. Review of the Cranfield Project. *American Documentation*, 14(4):307–311, October 1963. ISSN 1936-6108. doi: 10/cv672q.

[165] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X.

[166] Stephen E Robertson. The probability ranking principle in ir. *Journal of documentation*, 33(4):294–304, 1977.

[167] Stephen E. Robertson, Hugo Zaragoza, and Michael J. Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 42–49, New York, NY, USA, 2004. ACM. ISBN 1-58113-874-1. doi: 10.1145/1031171.1031181. URL http://doi.acm.org/10.1145/1031171.1031181.

[168] Daniel M. Russell, Mark J. Stefik, Peter Pirolli, and Stuart K. Card. The Cost Structure of Sensemaking. In *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, pages 269–276. ACM, 1993.

[169] T. Sakai and Z. Dou. Summaries, ranked retrieval and sessions: A unified framework for information access evaluation. In *Proc. SIGIR'13*, pages 473–482. ACM, 2013.

[170] Daniel Salber, Anind K. Dey, and Gregory D. Abowd. The Context Toolkit: Aiding the Development of Context-enabled Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 434–441, New York, NY, USA, 1999. ACM. ISBN 978-0-201-48559-2. doi: 10.1145/302979.303126.

[171] Gerard Salton and Michael E. Lesk. The SMART automatic document retrieval systems - an illustration. *Commun. ACM*, 8(6):391–398, 1965. doi: 10.1145/364955.364990. URL `https://doi.org/10.1145/364955.364990`.

[172] Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope. "Your Word is my Command": Google Search by Voice: A Case Study. In *Advances in Speech Recognition*, pages 61–90. Springer, 2010.

[173] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Building Bridges for Web Query Classification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 131–138. ACM, 2006.

[174] Amit Singhal. Modern Information Retrieval: A Brief Overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[175] M.D. Smucker and C. Jethani. Time to judge relevance as an indicator of assessor error. In *Proc. SIGIR'12*, pages 1153–1154. ACM, 2012.

[176] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[177] Eero Sormunen, Jannica Heinstrom, Leena Romu, and Risto Turunen. A method for the analysis of information use in source-based writing. *Information Research: An International Electronic Journal*, 17(4):n4, 2012.

[178] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[179] Amanda Spink and H Cenk Ozmultu. Characteristics of Question Format Web Queries: An Exploratory Study. *Information processing & management*, 38(4):453–471, 2002.

[180] Amanda Spink, Bernard J Jansen, Dietmar Wolfram, and Tefko Saracevic. From e-sex to e-commerce: Web Search Changes. *Computer*, 35(3):107–109, 2002.

[181] Amanda Spink, Huseyin Cenk Özmutlu, and Seda Özmutlu. Multitasking information seeking and searching processes. *Journal of the American Society for Information Science and Technology*, 53(8):639–652, 2002.

[182] Benno Stein, Tim Gollub, and Dennis Hoppe. Beyond Precision@10: Clustering the Long Tail of Web Search Results. In Bettina Berendt, Arjen de Vries, Wenfei Fan, Craig Macdonald, Iadh Ounis, and Ian Ruthven, editors, *20th ACM International Conference on Information and Knowledge Management (CIKM 2011)*, pages 2141–2144. ACM, October 2011. ISBN 978-1-4503-0717-8. doi: 10.1145/2063576.2063910.

[183] Benno Stein, Tim Gollub, and Dennis Hoppe. Search Result Presentation Based on Faceted Clustering. In Xuewen Chen, Guy Lebanon, Haixun Wang, and Mohammed J. Zaki, editors, *21st ACM International Conference on Information and Knowledge Management (CIKM 2012)*, pages 1940–1944. ACM, October 2012. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2398548.

[184] Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information retrieval. In Kraaij et al. [110], pages 295–302. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277794. URL `http://doi.acm.org/10.1145/1277741.1277794`.

[185] Robert S. Taylor. The Process of Asking Questions. *American Documentation*, 13(4):391–396, October 1962. ISSN 1936-6108. doi: 10/cfwqv6.

[186] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. To Personalize or Not to Personalize: Modeling Queries with Variation in User Intent. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 163–170, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390364.

[187] Jaime Teevan, Meredith Ringel Morris, and Steve Bush. Discovering and Using Groups to Improve Personalized Search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 15–24, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-390-7. doi: 10.1145/1498759.1498786.

[188] Jaime Teevan, Kevyn Collins-Thompson, Ryen W. White, and Susan Dumais. Slow Search. *Commun. ACM*, 57(8):36–38, August 2014. ISSN 0001-0782. doi: 10/gd8363.

[189] Elaine G. Toms. Task-based information searching and retrieval. In Ian Ruthven and Diane Kelly, editors, *Interactive Information Seeking, Behaviour and Retrieval*, pages 43–59. Facet Pub, London, 2011. ISBN 978-1-85604-707-4.

[190] Pertti Vakkari. A Theory of the Task-based Information Retrieval Process: A Summary and Generalisation of a Longitudinal Study. *Journal of Documentation*, 57(1):44–60, 2001. ISSN 0022-0418. doi: 10/d4j2ft.

[191] Pertti Vakkari. Task-Based Information Searching. *Annual Review of Information Science and Technology*, 37(1):413–464, January 2005. ISSN 00664200. doi: 10/c2r576.

[192] Pertti Vakkari. Exploratory Searching as Conceptual Exploration. *Proceedings of the 4th International Workshop on Human-Computer Interaction and Information Retrieval*, pages 24–27, 2010.

[193] Pertti Vakkari and Saila Huuskonen. Search Effort Degrades Search Output but Improves Task Outcome. *Journal of the American Society for Information Science and Technology*, 63(4):657–670, 2012.

[194] Pertti Vakkari, Michael Völske, Matthias Hagen, Martin Potthast, and Benno Stein. Predicting Retrieval Success Based on Information Use for Writing Tasks. In *22nd International Conference on Theory and Practice of Digital Libraries (TPDL 2018)*, pages 161–173, Porto, September 2018. Springer. doi: 10.1007/978-3-030-00066-0_14.

[195] Adriano Veloso, Humberto Mossri de Almeida, Marcos André Gonçalves, and Wagner Meira Jr. Learning to rank at query-time using association rules. In Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong, editors, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 267–274. ACM, 2008. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390381. URL `http://doi.acm.org/10.1145/1390334.1390381`.

[196] Suzan Verberne, Lou Boves, and Wessel Kraaij. Bringing why-qa to web search. In *Advances in Information Retrieval*, pages 491–496. Springer, 2011.

[197] Kim J. Vicente. *Cognitive Work Analysis: Towards Safe, Productive, and Healthy Computer-Based Work*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1999. ISBN 978-0-8058-2396-7.

[198] Fernanda B. Viégas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. In *CHI '04: Proceedings of the SIGCHI conference on Human factors*

in computing systems, pages 575–582, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-702-8. doi: http://doi.acm.org/10.1145/985692.985765.

[199] Michael Völske, Tim Gollub, Matthias Hagen, and Benno Stein. A Keyquery-Based Classification System for CORE. In Laurence Lannom, editor, *3rd International Workshop on Mining Scientific Publications (WOSP 2014)*, volume 20. Corporation for National Research Initiatives (CNRI), September 2014. doi: 10.1045/november14-voelske. URL `http://www.dlib.org/dlib/november14/voelske/11voelske.html`.

[200] Michael Völske, Pavel Braslavski, Matthias Hagen, Galina Lezina, and Benno Stein. What users ask a search engine: Analyzing one billion russian question queries. In *24th ACM International Conference on Information and Knowledge Management (CIKM 2015)*, pages 1571–1580. ACM, October 2015. ISBN 978-1-4503-3794-6. doi: 10.1145/2806416.2806457. URL `http://doi.acm.org/10.1145/2806416.2806457`.

[201] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to Learn Automatic Summarization. In *EMNLP 2017 Workshop on New Frontiers in Summarization*, pages 59–63. Association for Computational Linguistics, September 2017. doi: 10.18653/v1/W17-4508. URL `http://aclweb.org/anthology/W17-4508`.

[202] Thanh Tien Vu, Alistair Willis, and Dawei Song. Modelling Time-aware Search Tasks for Search Personalisation. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 131–132, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3473-0. doi: 10.1145/2740908.2742714.

[203] Henning Wachsmuth, Martin Potthast, Khalid Al-Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. Building an Argument Search Engine for the Web. In *Fourth Workshop on Argument Mining (ArgMining 2017)*, pages 49–59. Association for Computational Linguistics, September 2017.

[204] Ingmar Weber, Antti Ukkonen, and Aris Gionis. Answers, Not Links: Extracting Tips From Yahoo! Answers to Address How-To Web Queries. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 613–622. ACM, 2012.

[205] Ryen W. White and Gary Marchionini. Examining the Effectiveness of Real-time Query Expansion. *Information Processing and Management*, 43(3):685–704, 2007. ISSN 0306-4573. doi: 10.1016/j.ipm.2006.06.005.

[206] Ryen W. White and Dan Morris. Investigating the querying and browsing behavior of advanced search engine users. In Wessel Kraaij, Arjen P. de Vries, Charles L. A. Clarke, Norbert Fuhr, and Noriko Kando, editors, *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 255–262. ACM, 2007. ISBN 978-1-59593-597-7.

[207] Ryen W. White and Resa A. Roth. *Exploratory Search: Beyond the Query-Response Paradigm*. Synthesis Lectures on Information Concepts, Retrieval, and Services Series. Morgan & Claypool, 2009. ISBN 9781598297836.

[208] Ryen W. White, Gheorge Muresan, and Gary Marchionini, editors. *Proceedings of the ACM SIGIR 2006 Workshop on Evaluating Exploratory Search Systems*, 2006.

[209] Ryen W. White, Gary Marchionini, and Gheorghe Muresan. Editorial: Evaluating Exploratory Search Systems. *Information Processing and Management*, 44(2):433–436, 2008. ISSN 0306-4573. doi: 10.1016/j.ipm.2007.09.011.

[210] Ryen W. White, Peter Bailey, and Liwei Chen. Predicting User Interests from Contextual Information. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 363–370, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-483-6. doi: 10.1145/1571941.1572005.

[211] Ryen W. White, Paul N. Bennett, and Susan T. Dumais. Predicting Short-term Interests Using Activity-based Search Context. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1009–1018, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. doi: 10.1145/1871437.1871565.

[212] Ryen W. White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang. Enhancing Personalized Search by Mining and Modeling Task Behavior. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1411–1420, New York,

NY, USA, 2013. ACM. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488511.

[213] Hao Wu and Hui Fang. Relation based term weighting regularization. In Baeza-Yates et al. [13], pages 109–120. ISBN 978-3-642-28996-5. doi: 10.1007/978-3-642-28997-2_10. URL http://dx.doi.org/10.1007/978-3-642-28997-2_10.

[214] Biao Xiang, Daxin Jiang, Jian Pei, Xiaohui Sun, Enhong Chen, and Hang Li. Context-aware Ranking in Web Search. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 451–458, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835525.

[215] Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trancactions on Information Systems*, 18(1):79–112, 2000. ISSN 1046-8188. doi: http://doi.acm.org/10.1145/333135.333138.

[216] Jun Xu and Hang Li. Adarank: a boosting algorithm for information retrieval. In Kraaij et al. [110], pages 391–398. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277809. URL http://doi.acm.org/10.1145/1277741.1277809.

[217] Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. Retrieval models for question and answer archives. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 475–482, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390416. URL http://doi.acm.org/10.1145/1390334.1390416.

[218] Xiaobing Xue, Yu Tao, Daxin Jiang, and Hang Li. Automatically mining question reformulation patterns from search log data. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 187–192. Association for Computational Linguistics, 2012.

[219] Liu Yang, Qi Guo, Yang Song, Sha Meng, Milad Shokouhi, Kieran McDonald, and W. Bruce Croft. Modeling User Interests for Zero-Query Ranking. In *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 171–184. Springer, Cham, March 2016. ISBN 978-3-319-30670-4 978-3-319-30671-1. doi: 10.1007/978-3-319-30671-1_13.

[220] E. Yilmaz, M. Verma, N. Craswell, F. Radlinski, and P. Bailey. Relevance and effort: an analysis of document utility. In *Proc. CIKM'14*, pages 91–100. ACM, 2014.

[221] Dell Zhang, Robert Mao, Haitao Li, and Joanne Mao. How to count thumb-ups and thumb-downs: User-rating based ranking of items from an axiomatic perspective. In Giambattista Amati and Fabio Crestani, editors, *Advances in Information Retrieval Theory - Third International Conference, ICTIR 2011, Bertinoro, Italy, September 12-14, 2011. Proceedings*, volume 6931 of *Lecture Notes in Computer Science*, pages 238–249. Springer, 2011. ISBN 978-3-642-23317-3. doi: 10.1007/978-3-642-23318-0_22. URL `http://dx.doi.org/10.1007/978-3-642-23318-0_22`.

[222] Shiqi Zhao, Haifeng Wang, Chao Li, Ting Liu, and Yi Guan. Automatically generating questions from queries for community-based question answering. In *IJCNLP*, pages 929–937, 2011.

[223] Zhe Zhao and Qiaozhu Mei. Questions About Questions: An Empirical Analysis of Information Needs on Twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1545–1556. International World Wide Web Conferences Steering Committee, 2013.

[224] Wei Zheng and Hui Fang. Query aspect based term weighting regularization in information retrieval. In Cathal Gurrin, Yulan He, Gabriella Kazai, Udo Kruschwitz, Suzanne Little, Thomas Roelleke, Stefan M. Rüger, and Keith van Rijsbergen, editors, *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings*, volume 5993 of *Lecture Notes in Computer Science*, pages 344–356. Springer, 2010. ISBN 978-3-642-12274-3. doi: 10.1007/978-3-642-12275-0_31. URL `http://dx.doi.org/10.1007/978-3-642-12275-0_31`.