A Framework for Evaluation and Optimization of Relevance and Novelty-based Retrieval

Abhimanyu Lad

CMU-LTI-11-003

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
www.lti.cs.cmu.edu

Thesis Committee:

Yiming Yang (*Chair*)
Jaime Carbonell
Jamie Callan
Jan Pedersen (*Microsoft, Inc.*)

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in Language and Information Technologies*

To my wife, my family, my teachers, and my friends.

Abstract

There has been growing interest in building and optimizing retrieval systems with respect to relevance and novelty of information, which together more realistically reflect the usefulness of a system as perceived by the user. How to combine these criteria into a single metric that can be used to measure as well as optimize retrieval systems is an open challenge that has only received partial solutions so far. Unlike relevance, which can be measured independently for each document, the novelty of a document depends on other documents seen by the user during his or her past interaction with the system. This is especially problematic for assessing the retrieval performance across multiple ranked lists, as well as for learning from user's feedback, which must be interpreted with respect to other documents seen by the user. Moreover, users often have different tolerances towards redundancy depending on the nature of their information needs and available time, but this factor is not explicitly modeled by existing approaches for novelty-based retrieval.

In this thesis, we develop a new framework for evaluating as well as optimizing retrieval systems with respect to their utility, which is measured in terms of relevance and novelty of information. We combine a nugget-based model of utility with a probabilistic model of user behavior; this leads to a flexible metric that generalizes existing evaluation measures. We demonstrate that our framework naturally extends to the evaluation of session-based retrieval while maintaining a consistent definition of novelty across multiple ranked lists.

Next, we address the complementary problem of optimization, *i.e.*, how to maximize retrieval performance for one or more ranked lists with respect to the proposed measure. Since the system does not have knowledge of the nuggets that are relevant to each query, we propose a ranking approach based on the use of observable query and document features (*e.g.*, words and named entities) as surrogates for the unknown nuggets, whose weights are automatically learned from user feedback. However, finding the ranked list that maximizes the coverage of a given set of nuggets leads to an NP-hard problem. We take advantage of the sub-modularity of the proposed measure to derive lower bounds on the performance of approximate algorithms, and also conduct experiments to assess the empirical performance of a greedy algorithm under various conditions.

Our framework provides a strong foundation for modeling retrieval performance in terms of non-independent utility of documents across multiple ranked lists. Moreover, it allows accurate evaluation and optimization of retrieval systems under realistic conditions, and hence, enable rapid development and tuning of new algorithms for novelty-based retrieval without the need for user-centric evaluations involving human subjects, which, although more realistic, are expensive, time-consuming, and risky in a live environment.

Acknowledgments

This thesis would not have been possible without the support of my friends, family, faculty, and colleagues. First and foremost, I am deeply grateful to my advisor, Yiming Yang, who patiently steered and supported me over the course of this PhD. She was always ready with fresh ideas and directions that tremendously helped me in shaping up my thesis. I am thankful to her for taking such personal interest in my research, which kept me motivated through these years. I would also like to express my gratitude to my thesis committee members, Jaime Carbonell, Jamie Callan, and Jan Pedersen, for all their time and effort. Their unique perspectives and thoughtful feedback played a crucial role in refining the ideas in this thesis.

I am also thankful to my professors at my undergraduate college, Indian Institute of Information Technology, Allahabad, where I got my first taste of research in information retrieval. Dr. Sudeep Sanyal and Dr. G.C. Nandi highly encouraged me to work on challenging problems and provided me with all resources to pursue my research interests.

To all my friends in Pittsburgh: thank you for your support, and for keeping me sane and happy. Special thanks to Alok Parlikar and Satanjeev Banerjee for all the laughter! I am also grateful to all past and present members of my research group: Abhay Harpale, Fan Li, Jian Zhang, Konstantin Salomatin, Monica Rogati, Siddharth Gopal, and Shinjae Yoo. Thank you for all your help and for providing early feedback on my research. Outside of CMU, I would like to call out Manish Saggar, University of Texas, Austin. We studied together at undergraduate college, and applied to graduate schools together. His tenacity and incredibly positive attitude towards research and life in general have been a constant source of inspiration for me.

I am forever indebted to my parents who have always encouraged me to settle for nothing but the best. My father has been a great mentor and advisor to me throughout my life. His support and encouragement played a tremendous role in my decision to pursue a PhD.

Last and most of all, thank you, Sudipti, for all your sacrifice and support through this journey. It is a testament to your strength that we have made it through this PhD together. This dissertation is as much yours as it is mine.

Contents

List of Tables

List of Figures

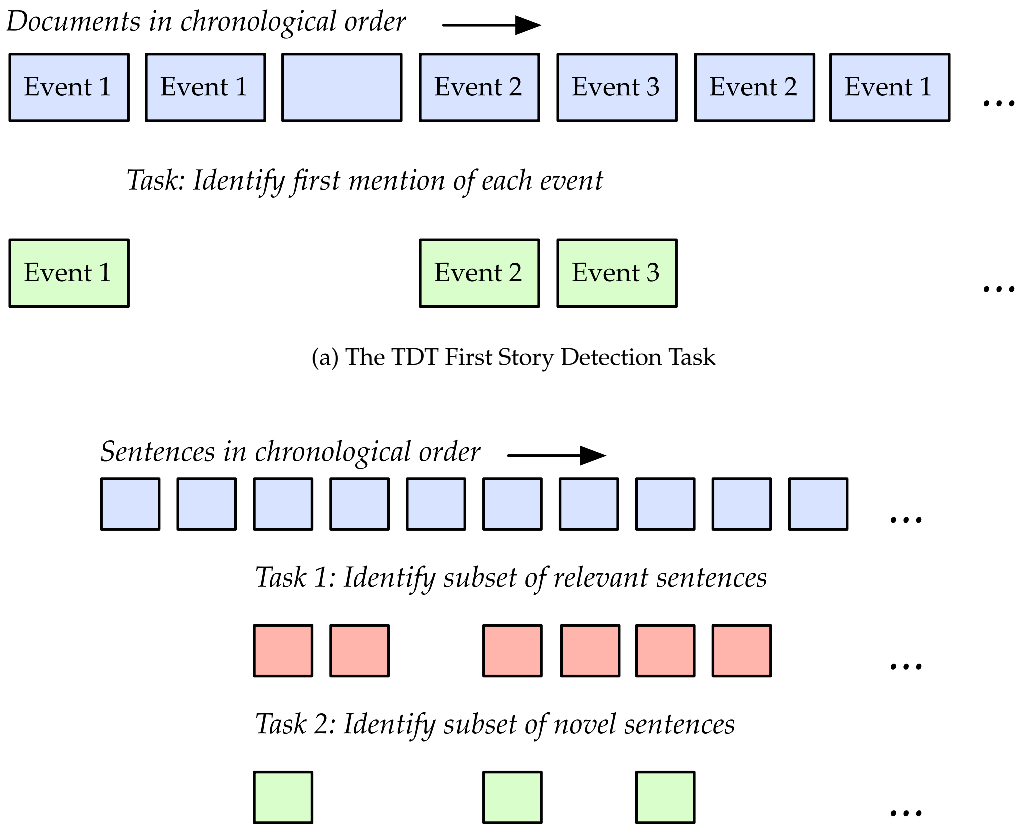
Chapter 1

Introduction

Today's search engines index numerous sources of information including news websites, blogs, and more recently, user-generated content on micro-blogging services like Twitter, discussion forums, and news aggregation services like Digg and Reddit. Naturally, a popular news story or topic is reported and discussed on multiple websites; such overlapping content leads to redundant information in search results. Moreover, users often engage in search sessions where they iteratively find information on a topic by using a sequence of interrelated queries [Bates, 1989; Spink et al., 2002]. Current search engines treat each query independently, which often leads to the presence of same or similar documents in subsequent ranked lists. Such overlapping and redundant content puts an undue burden on the user to locate *novel* information that he or she has not seen previously.

Hence, there has been growing interest in building and optimizing retrieval systems with respect to relevance as well as novelty of information, which together more realistically reflect the usefulness of a retrieval system as perceived by the user. Therefore, a framework is required for measuring and optimizing retrieval performance in terms of these two criteria in a principled manner so as to guide the development and tuning of new algorithms for novelty-based retrieval. Significant research has been devoted to the evaluation and optimization of purely relevance-based retrieval in the form of various test collections, evaluation metrics, and learning algorithms. However, these cannot be directly applied to novelty-based retrieval; combining novelty with relevance for evaluating and optimizing ranked retrieval poses several challenges that have not been fully addressed so far.

Novelty detection has been studied in the information retrieval community in the past. For instance, the *Topic Detection and Tracking* (TDT) forum [Yang et al., 1997; Allan et al., 1998] studied the problem of *first story detection*, where the goal was to monitor a stream


Documents in chronological order
Event 1
Event 1
Event 2
Event 3
Event 2
Event 1
...
Task: Identify first mention of each event
Event 1
Event 2
Event 3
...
Sentences in chronological order
...
Task 1: Identify subset of relevant sentences
...
Task 2: Identify subset of novel sentences
...

(a) The TDT First Story Detection Task

(b) The TREC Novelty Detection Task

Figure 1.1: Past Research on Novelty Detection.

of articles and identify the first mention of a news event (see Figure 1.1a). Similarly, the *Text REtrieval Conferences* (TREC) [Soboroff, 2004] studied the problem of *sentence-level novelty detection*, where the goal was to monitor a stream of documents and identify sentences that are relevant to a given information need, and further identify a subset of these sentences that contain novel information, *i.e.*, information not seen in previously selected sentences (see Figure 1.1b). In both these research studies, the presentation order of documents was assumed to be fixed so as to enable an operational definition of novelty in terms of previous documents, *i.e.*, previous with respect to time. In such a retrieval setting, the system merely filters a temporal stream of documents without reordering them. The user is assumed to read each document as and when it is selected by the system. On the other hand, a search engine produces a ranked list of documents in response to a user query, where documents are ordered according to their estimated utility to the user. This is known as *ranked retrieval*, which is a far more common mode of interaction with today's retrieval systems (search engines) and is well-suited to finding information in large collections of text. Moreover, ranked retrieval makes a realistic assumption that users will not (and cannot) read all documents selected by the retrieval system. The TDT and TREC forums did not address the problem of how to combine relevance and novelty in the context of ranked retrieval.

The problem of minimizing the redundancy in single ranked lists has been addressed to some extent in the field of *diversity-based retrieval*. For instance, *Maximum Marginal Relevance* (MMR) [Carbonell and Goldstein, 1998; Goldstein et al., 2000a] proposed a linear combination of relevance and novelty scores to rank documents, where novelty is measured in terms of cosine similarities between pairs of documents. Similarly, other approaches like *sub-topic retrieval* [Zhai et al., 2003] and *intent-aware retrieval* [Agrawal et al., 2009] have been proposed for diversifying search results, which mainly differ in the way similarity between documents is measured. However, all these methods focus on "implicit" novelty without taking the user into account. In other words, they do not take into account what information the user has already seen in his or her past interaction with the retrieval system. This is especially important for session-based retrieval that consists of multiple rounds of interaction with the system, and hence, is more likely to provide redundant content to the user. In such a setting, it is not clear which documents from previous ranked lists should be deemed as read by the user for the purpose of modeling the redundancy in the current ranked list. A naive solution is to assume that all documents in previously displayed ranked lists are read by the user. However, users generally do not read all documents presented to them, especially in long ranked lists generated by search engines. In fact, users are more likely to read the top-ranked documents, and stop at some position based on their patience or satisfaction [Joachims et al., 2005]. An alternative solution is to assume that users read a fixed number of documents (say, top ten) in each ranked list [Järvelin et al., 2008]. However, any such arbitrarily chosen stopping position would lead to a crude approximation that is not representative of the behavior of all users. Moreover, documents below the chosen rank will be completely ignored for the purposes of evaluation as well as optimization of the system.

Clearly, the flexibility a user has in interacting with one or more ranked lists presented by the system means that a model of the user's browsing behavior is required for accurately estimating the system's performance as perceived by a typical user. In relevance-based retrieval, where a document's usefulness is independent of others, a commonly used substitute for an explicit user browsing model is the placement of more weight on higher ranks, *e.g.*, implicit top-heaviness of *Average Precision* (`AvgP`) [Buckley and Voorhees, 2005] or explicit discount factors in *Discounted Cumulated Gain* (`DCG`) [Järvelin and Kekäläinen, 2002]. On the other hand, for novelty-based evaluation, the usefulness of each document must be conditioned on the user's browsing history. However, user behavior is non-deterministic: Users can decide to stop at any position

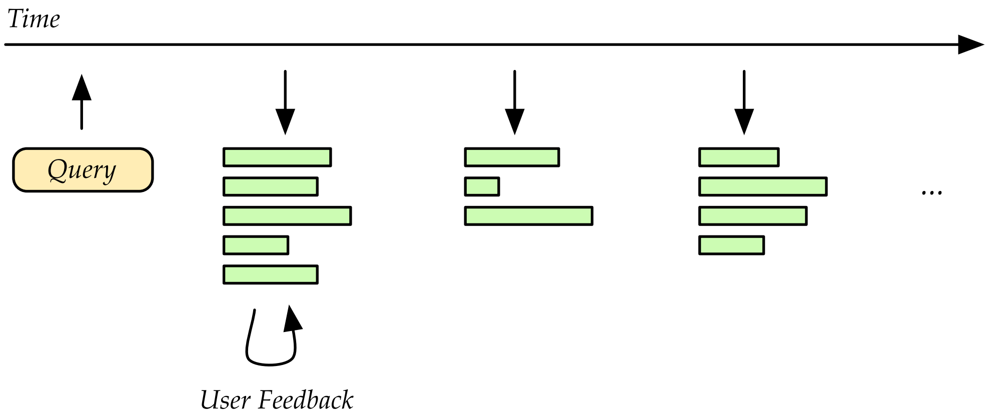


Figure 1.2: Information Distillation: Given a query, track a stream of documents and rank specific relevant spans of text at regular intervals of time.

in a ranked list for various reasons like satisfaction or frustration. Therefore, we can only talk about the *likelihood* of the user reading a particular document, which in turn would determine the redundancy of subsequent documents. In other words, we require a probabilistic model of user behavior that can be used to derive the *expected* utility of the system for a targeted population of users.

Moreover, users often have different tolerances towards redundancy based on their information need or available time. While some users might only want to see previously unseen documents, other users might desire a certain level of redundancy in the ranked list for various reasons like corroboration of information, or assessing the consensus or opinions on a single topic or product based on different news sources, reviewers, or blogs. However, existing approaches for novelty-based ranking do not take this factor into account.

Finally, there has been growing interest in modeling more complex information retrieval scenarios like *information distillation* [Hakkani-Tur et al., 2007; Yang et al., 2007; White et al., 2008], which combines aspects of ranked retrieval and information filtering. Information distillation is targeted towards intelligence analysts who need to gather relevant information on news events by monitoring various sources of information. Given an information need in the form of a query, the distillation system returns relevant content in a ranked list at regular intervals of time (see Figure 1.2). A unique aspect of the distillation setting is that the retrieval system is allowed to present any span of text that is relevant to the information need. Moreover, the retrieval system is penalized for producing long ranked lists containing redundant information, which would place undue burden on the intelligence analyst who must go through all returned documents. This is unlike the web search setting where search engines produce a very long ranked list, assuming that the user will stop when he or she finds enough information on a topic. Therefore, accurately modeling the utility in such a distillation setting requires, in turn, the accurate modeling of the relevance, novelty, and reading costs of arbitrary spans of text like documents, passages, or sentences. The problem of how to model the utility of such arbitrary spans of text has been addressed to some extent in the fields of summarization and question answering through the use of "nuggets" (*i.e.*, pieces of information) as units of retrieval [Voorhees, 2003; Lin and Demner-Fushman, 2005; Marton and Radul, 2006; Dang et al., 2006]. However, how to use such a nugget-based approach in the context of ranked retrieval while taking into account the users' browsing behavior as well as different tolerances towards redundancy has remained an open challenge.

In sum, to build ranked retrieval systems that are more likely to satisfy users' information needs, we must answer the following questions:

1. How can we model the utility of a ranked list in terms of relevance and novelty of information?
2. How can we model users' browsing behaviors across one or more ranked lists?
3. How can we model users' different tolerances towards redundancy of information?
4. How can we model the utility of arbitrary spans of text while taking their different reading costs into account?

Our goal is this thesis is to answer these questions by developing a new framework for accurately modeling the utility of retrieval systems with respect to relevance as well as novelty of information. We model a document's utility (*i.e.*, relevance and novelty) in terms of its nuggets (pieces of information). The credit received for subsequent presentation of the same nugget is subject to a diminishing returns property, which reflects how the value of repeated information is perceived by users. The non-deterministic nature of users' browsing behaviors is captured using a probabilistic model, which allows us to calculate the *expected utility* over all possible ways of interacting with one or more ranked lists of documents. The flexible modeling of the user's tolerance of redundancy as well as his browsing behavior leads to a measure called *Expected Global Utility* (EGU) that generalizes existing measures used in information retrieval research, and also naturally extends to session-based retrieval while maintaining a consistent definition of novelty across multiple ranked lists.

After proposing a new measure of the utility of a retrieval system, we focus on the problem of *optimization*, *i.e.*, how to produce ranked lists that maximize the proposed utility measure. There are two main challenges associated with directly optimizing a nugget-based measure: First, the system does not have a-priori knowledge of the nuggets that are relevant to a given query and the documents in which they are present. We propose a new approach based on using various document features as surrogates for the unknown nuggets. We also address the problem of learning from user feedback while taking into account the non-independent nature of the utility of documents. We compare the proposed techniques against existing approaches for novelty-based retrieval on two test collections that reflect two different information-seeking behaviors.

Second, even with perfect knowledge of the nuggets, producing the best ranked list leads to a hard combinatorial problem: Maximizing the coverage of discrete elements (nuggets in our case) can be reduced to the SET-COVER problem [Zhai et al., 2003; Carterette, 2009], which is known to be NP-hard. We show that our proposed measure has a special structure called *sub-modularity* that allows approximate algorithms to achieve good performance. We derive a new lower-bound on the performance of a greedy algorithm for optimizing the proposed metric. We also conduct experiments on synthetic data to assess the empirical performance of the greedy algorithm under various conditions.

Roadmap for the Thesis

- In Chapter 2, we present background information and definitions that are essential to understanding the rest of the thesis. We define the concept of relevance and novelty, highlighting their underlying assumptions as well as inherent differences. We survey some of the representative work on novelty detection and also evaluation techniques that have been used in the past.

- In Chapter 3, we address the problem of evaluating ranked retrieval performance in terms of relevance and novelty of information. We present a new evaluation framework that uses a nugget-based model of utility with a probabilistic model of user behavior to estimate the expected utility of a retrieval system. Specifically, we propose a new evaluation measure call Expected Global Utility (EGU).

- In Chapter 4, we analyze the behavior of EGU on real retrieval systems and datasets. We conduct experiments on the task of diversity-based retrieval using existing TREC runs. We also explore the task of information distillation, for which we create a new benchmark dataset by extending the TDT4 dataset. We create nuggets and corresponding "nugget-matching rules" that allow evaluation of retrieval systems that output arbitrary spans of text, as opposed to traditional evaluations that assume a fixed unit of retrieval, *e.g.*, a document.

- In Chapter 5, we describe our analysis of the computational challenges associated with optimizing ranked lists with respect to the proposed evaluation measure. We also conduct experiments on synthetic data to assess the empirical performance of approximate algorithms for optimizing novelty-based performance measures under different conditions.

- In Chapter 6, we address the problem of system optimization, *i.e.*, how a retrieval system should generate the optimal ranked list with respect to the user's query. We propose a new approach based on using observable document features as surrogates for nuggets, and compare this approach with various baseline approaches for novelty and diversity-based ranking.

- In Chapter 7, we summarize the contributions and conclusions of this thesis, and describe some of the future work that follows from this thesis.

Chapter 2

Background

In this chapter, we provide background information and definitions that are essential to understanding the rest of the thesis. Instead of presenting an exhaustive survey of all related work here, we describe applicable related work at the end of each of the subsequent chapters, where it is easier to put into context and compare against our work.

In Section 2.1, we define the concept of relevance and novelty and then motivate the need for novelty detection in various information retrieval scenarios in Section 2.2. In Section 2.3, we describe the underlying assumptions in relevance and novelty-based retrieval, followed by a survey of some of the representative past work on novelty detection in Section 2.4. We describe some of the common evaluation measures used for relevance-based retrieval as well as novelty and diversity-based retrieval in Section 2.5. We also note how novelty-based ranking is related to diversity-based ranking and multi-document summarization in Sections 2.6 and 2.7, respectively.

2.1 Relevance and Novelty

In the context of information retrieval, *relevance* refers to how well a document addresses a given information need, which is generally expressed by the user in the form of a query. There are many aspects of relevance; for instance, *topical relevance* refers to the inherent relatedness of a document to the information need, *i.e.*, whether the document is "on topic". In a broader sense, relevance is defined with respect to the user's context, which might include various factors like reading difficulty, cost, and novelty of the information with respect what the user already knows [Mizzaro, 1998; Bateman, 1998].

As a matter of convention in this thesis, we will use the term “relevance” to refer to the topical relevance or topicality of a document, while treating novelty as a separate factor that, together with (topical) relevance, influences the overall utility of a document with respect to addressing the user’s information need. In this sense, our goal is to take into account both relevance and novelty while ranking documents in response to a query.

Novelty refers to the presence of new information with respect to what the user already knows, which might include the user’s prior knowledge as well as the information he gains as he interacts with a retrieval system. The task of identifying documents, passages, or sentences that contain new information is known as *novelty detection*. The goal of novelty detection is obvious: Avoid the presentation of redundant information to the user. Redundancy in search results arises in various retrieval scenarios and novelty detection can be particularly useful in these settings. Let us look at some of these scenarios in the next section.

2.2 Information Retrieval Scenarios

The nature of the information retrieval task depends heavily on the goals and intentions of the target user: The information-seeking behavior of a lawyer looking for all pertinent case files is very different from a typical web surfer looking for general information on a topic, whose behavior in turn differs from that of an intelligence analyst tracking an evolving news event.

The differences in these retrieval scenarios determine the design of the retrieval system as well as the the evaluation methodology. Let us examine the following retrieval scenarios: ad hoc search, session-based search, information filtering, and information distillation. Our goal is to identify the kinds of information-seeking behaviors where novelty or diversity-based retrieval is especially crucial in improving the overall experience of the user.

2.2.1 Ad hoc search

Ad hoc retrieval is the most commonly studied retrieval task. The goal is to return documents that are relevant to an immediate or short-lived information need, which is expressed in the form of a query. Ad hoc queries on the web are mainly categorized as *navigational* and *informational* queries [Broder, 2002; Rose and Levinson, 2004]. Navigational queries indicate the user’s intent to reach a particular website. Ideally, the result page of such queries includes the target page at the first rank, in which case, the user ignores the rest of the search results. Informational queries indicate the user’s intent to gather information on a broad topic. Users generally click and read multiple web pages appearing on the results page to satisfy their information need.

Informational queries can benefit from novelty detection since the user is interested in multiple documents, in which case, it is best to avoid redundant documents in the ranked list. Both navigational and informational queries can be ambiguous, in which case, it is prudent for the retrieval system to diversify its results with the hope of including at least one document that satisfies the different intents of all users, thus, reducing the likelihood of abandonment by any user.

2.2.1.1 Session-based Retrieval

Users often engage in search sessions, where they sequentially issue multiple interrelated queries to gather information on a particular topic [Bates, 1989; Spink et al., 2002, 2006]. Like single queries, it is prudent for the retrieval system to avoid documents that are redundant with respect to documents that have already been seen by the user in previous ranked lists. This mode of novelty detection is user-centric, *i.e.*, documents are deemed novel or redundant with respect to a particular user's interaction with the system.

2.2.2 Information Filtering

Information filtering refers to the task of selecting documents that are relevant to a long-lasting information need of the user. Such an information need is generally represented by a persistent user profile. The information need of the user might change or evolve over time, and this is usually captured by seeking feedback from the user on the documents selected by the system, which is then used to update the user profile. Such a learning-based setup is known as adaptive filtering and has been widely studied in literature [Allan, 1996; Callan, 1998; Yang et al., 2007; Robertson and Soboroff, 2002; Fiscus and Wheatley, 2004].

Zhang et al. [2002] extended this task to the problem of returning documents that are relevant as well as novel with respect to previously selected documents. This is an interesting retrieval setting for novelty detection and we describe it in more detail in Section 2.4.1.

2.2.3 Information Distillation

Information distillation is a new mode of information retrieval that was the focus of DARPA's Global Autonomous Language Exploitation (GALE) project [Hakkani-Tur et al., 2007; White et al., 2008]. It is targeted towards intelligence analysts who need to gather relevant information on events around the world by monitoring various sources of information. A distillation system is allowed to produce any readable response text that is consistent with the sources. This may include paraphrasing or summarizing the original text. Naturally, such a setting with no fixed retrieval unit requires the development of new retrieval algorithms as well as new strategies for evaluation. A general solution to the evaluation problem is to use nuggets as evaluation units, which is also the basis of the evaluation framework developed in this thesis (Chapter 3).

The overarching goals of the GALE project require a tight integration of speech transcription, language translation, question answering, information filtering and ranked retrieval systems; In our previous work [Yang et al., 2007], we focused on the retrieval aspect of the problem and designed a system that combines adaptive filtering, passage retrieval, and novelty detection to efficiently present relevant and novel snippets of information on evolving news events. The retrieval setup involves the following components:

"Chunked" Adaptive Filtering. Similar to the standard adaptive filtering setup, the retrieval system monitors a stream of documents and selects those that are relevant to the user's information need. However, in a realistic setting, a user cannot be expected to examine and provide feedback on each document as and when it is presented by the system. That is, the system must hold the selected documents till the user is ready to examine them. Therefore, we produce output only at regular intervals or "chunks" of time, *e.g.*, once per week, which corresponds to requesting the system at the end of each week to present the highlights or updates to the analyst on a particular topic of interest.

Passage Retrieval. Presenting a (potentially large) set of documents at the end of each time chunk puts an undue burden on the user to find the useful pieces of information. Therefore, the system splits the documents into snippets of text (*i.e.*, passages), and then presents them to the user in a ranked list for easy browsing.

Novelty Detection. This distillation setup is designed for long-lasting information needs that are addressed through multiple rounds of filtering, passage ranking, and feedback over an extended period of time. Moreover, there are multiple sources of information (newswire sources) that provide similar or same information on a news event. Therefore, novelty detection plays an important role in this setup. The retrieval system ranks passages based on relevance and novelty with respect to other passages already presented to the user in the current or previous ranked lists.

Our evaluation framework (Chapter 3) as well the dataset developed (Chapter 4) in this thesis are particularly suited for such a retrieval setting, apart from being generally applicable to any type of ranked retrieval that requires novelty or diversity-based retrieval. We perform detailed experiments on the distillation setup in Chapter 4.

Next, let us look at the fundamental differences between relevance and novelty-based retrieval, and how these differences affect the strategies applicable to both types of retrieval.

2.3 Relevance vs. Novelty-based Retrieval

2.3.1 Relevance-based Retrieval

Most research in relevance-based ranked retrieval is implicitly based on two assumptions:

Assumption 2.1 *The usefulness of each document to the user is independent of other documents in the corpus.*

Assumption 2.2 *The user browses the ranked list in a top-down manner, and therefore, is more likely to see a document that appears higher in the ranked list.*

Although unrealistic, the first assumption simplifies the design of the retrieval system and allows it to score each document in its corpus independent of other documents in response to a user's query. Under the second assumption, documents that are more likely to satisfy the user's information need should be presented first. This idea is formalized by the *Probability Ranking Principle* [Robertson, 1977; Robertson et al., 1982; Robertson and Belkin, 1993], which states that overall effectiveness of the system is maximized if it orders the documents by decreasing probabilities of relevance to the user, where the probabilities are estimated as accurately as possible on the basis of whatever information is made available to the system.

These assumptions have important implications for evaluation as well as optimization of retrieval systems. Most evaluation measures for relevance-based retrieval follow a common theme: They favor the presence of more relevant documents at the top ranks. The contribution of a document is not affected by the presence of other documents. To wit, these evaluation measures do not distinguish between two relevant documents that contain different information about the user's query, and two relevant documents that are duplicates. We will see evaluation measures in more detail in Section 2.5.

On the optimization side, these assumptions imply the following general strategy for relevance-based retrieval systems: Given a query, Step 1: Score each document in terms of its likelihood of satisfying the user, and Step 2: Rank the documents by their decreasing scores. Obviously, the second step (*i.e.*, sorting) is trivial. The focus is entirely on the first step, *i.e.*, how to score documents to accurately reflect their likelihood of relevance to the user's query, which is the subject of much research in information retrieval; some of the representative approaches are as follows. The vector space model [Salton, 1971; Salton et al., 1975] represents queries and documents as vectors with elements corresponding to words, weighted by a term weighting scheme, *e.g.*, TF–IDF [Salton and McGill, 1986; Salton and Buckley, 1988]. Then, the score of each document is simply the cosine similarity between the query and the document vector. More sophisticated measures like BM25 also take the document length into account [Robertson and Walker, 1994]. Language modeling based approaches to information retrieval are based on a probabilistic model of query and document generation [Lafferty and Zhai, 2003; Zhai and Lafferty, 2006]. Recently, learning to rank approaches have been proposed to automatically learn the scoring function as an optimal combination of several query and document features [Burges et al., 2005; Agarwal et al., 2005]. Nevertheless, all these methods ultimately lead to a scoring function that estimates the usefulness of a document to a query independent of other documents.

2.3.2 Novelty-based Retrieval

By definition, the novelty of a document depends on other documents seen by the user. Therefore, Assumption 2.1 is violated in case of novelty-based retrieval. In terms of retrieval strategies, this means that each document can no longer be scored independently of other documents that will be shown to the user. The particular form of the ranking algorithm depends on the nature of the information retrieval task (*e.g.*, information filtering vs. ranked retrieval). We describe the research conducted in novelty-based retrieval in the next section.

2.4 Past Research on Novelty Detection

The task of novelty detection has been investigated in *Topic Detection and Tracking* (TDT) studies as well as *Text REtrieval Conferences* (TREC). TDT investigated the task of *First Story Detection*: Monitor a stream of chronologically ordered documents and identify the first mention of a previously unknown news event [Yang et al., 1997; Allan et al., 1998; Carbonell et al., 1999]. This corresponds to inter-topic or inter-event novelty detection. The *TREC Novelty Track* investigated novelty detection within topics: From a stream of chronologically ordered sentences that are relevant to a given topic, identify sentences that do not repeat information that is covered by previous sentences [Allan et al., 2003; Soboroff, 2004].

As discussed in Chapter 1, both these retrieval settings assume a fixed (chronological) order of presentation of the documents. The retrieval system in this case merely performs *information filtering*, *i.e.*, filter a temporal stream of documents without reordering them. The user is assumed to read each document as and when it is selected by the system, which is clearly unrealistic. On the other hand, search engines produce a ranked list of documents in response to a user query, where documents are ordered according to their estimated utility to the user. This setting, known as *ranked retrieval* is a far more common mode of interaction with today's retrieval systems (search engines), and is well-suited to finding information in large collections of text. Moreover, ranked retrieval makes a realistic assumption that users will not (and cannot) read all documents selected by the retrieval system.

2.4.1 Combining Relevance and Novelty for Adaptive Filtering

Adaptive filtering is the task of monitoring a temporal stream of documents and finding documents that are relevant to the user's long-lasting information need [Robertson and Soboroff, 2002; Fiscus and Wheatley, 2004]. Zhang et al. [2002] extended this task to the problem of detecting documents that are relevant as well as novel with respect to what the user has already seen. The authors proposed a two-step process for filtering documents. In the first step, documents that are not relevant to the user's information need are filtered out. In the second step, documents deemed relevant are further tested for novelty with respect to the user's profile. Documents below a certain threshold are dropped. Thus, the decision to deliver a document depends on its relevance *and* novelty.

Zhang et al. [2002] proposed and compared various similarity measures for estimating novelty, including cosine similarity between term vectors, word overlap, and KL divergence between the language models of documents.

More importantly, unlike the TREC novelty detection task, which treated novelty (or redundancy) as an inherent property of a sentence independent of the user, Zhang et al. [2002] treat novelty as a user-centric decision: documents are novel or redundant for a specific user based on what he or she has been presented in the past. Our model of novelty in this thesis encompasses both these views: (i) novelty as a user-centric property, which is useful when addressing long-lasting information needs, and (ii) novelty as an inherent property of documents or sentences that is independent of the user, which is useful when diversifying search results (also see Section 2.6 for discussion on novelty and diversity). In this thesis, we study an extended version of the filtering task that combines adaptive filtering, novelty detection, and passage retrieval (see Chapter 4).

The two step filtering process proposed by Zhang et al. [2002] is appropriate for retrieval scenarios where the order of documents is fixed—in this case, chronological order. In such a setting, the system merely makes a binary decision, *i.e.*, whether to present a document to the user or not. However, in ranked retrieval, where the documents are freely ordered by the system, relevance and novelty must be somehow combined to produce a ranked list for the user. With the proliferation of ranked retrieval systems (*e.g.*, web search engines), this problem has received increased attention, as we will see next.

2.4.2 Combining Relevance and Novelty for Ranking Documents

Maximum Marginal Relevance. One of the early works on novelty and diversity based ranking, known as *Maximum Marginal Relevance* (MMR) [Carbonell and Goldstein, 1998; Goldstein et al., 2000a], proposed a linear combination of the relevance and novelty scores for the purpose of scoring and ranking documents. MMR incrementally builds the ranked list by choosing the next document with the highest *marginal* relevance, *i.e.*, high relevance to the query, and low similarity to already selected documents in the ranked list:

$$f(d_i|q, d_{1:(i-1)}) = \lambda \cdot \mathrm{sim}(d_i, q) - (1 - \lambda) \cdot \max_{d_j \in d_{1:(i-1)}} \mathrm{sim}(d_i, d_j) \tag{2.1}$$

where λ controls the relative importance of choosing relevant versus novel documents.

MMR assumes that relevance and novelty are measured independently and then combined to score each document. That is, relevance and novelty are treated as compensatory: high novelty can compensate for low relevance. Hence, it is possible for such a method to favor a document that is highly novel but irrelevant to the given query. However, users generally treat relevance as a pre-condition for usefulness of information [Mizzaro, 1998; Greisdorf, 2003]. Therefore, it is more appropriate to directly target the retrieval of relevant *and* novel information. We explore such an approach in this thesis (see Chapter 6).

Sub-topic retrieval. The basic idea of MMR, *i.e.*, incrementally building the ranked list based on marginal relevance, has been extended and applied to the task of diversity-based retrieval. For instance, Zhai et al. [2003] studied the task of subtopic retrieval and proposed the following scoring function for creating the ranked list:

$$f(d_i|q, d_{1:(i-1)}) = \mathrm{Pr}_{Rel}(d_i|q)\Big(\mathrm{Pr}_{Nov}(d_i|d_{1:(i-1)}) - \rho - 1\Big) \tag{2.2}$$

where ρ is a trade-off parameter similar to λ in MMR. The probabilities $\mathrm{Pr}_{Rel}(d_i|q)$ and $\mathrm{Pr}_{Nov}(d_i|d_{1:(i-1)})$ represent the likelihoods of relevance and novelty of document d_i, estimated using language modeling techniques. Again, the relevance and novelty of a document are estimated separately, which Zhai et al. [2003] also observe as a shortcoming to be addressed in future work.

Intent-aware retrieval. Agrawal et al. [2009] proposed an approach for increasing the likelihood of user satisfaction in case of ambiguous queries by diversifying search results with respect to the topical categories of documents, which capture the ambiguous intents of different users. They use a MMR-like greedy algorithm based on the following definition of marginal utility, which can be interpreted as the probability that the candidate document will satisfy the user, given that none of the previous documents did so:

$$g(d|q,c,S) = \sum_{c} U(c|q,S)V(d|q,c) \tag{2.3}$$

where $U(c|q,S)$ denotes the conditional probability that the query belongs to category c, given that all previously selected documents (set S) failed to satisfy the user. It is equal to:

$$U(c|q,S) = \Pr(c|q) \prod_{d' \in S} (1 - V(d'|q,c)) \tag{2.4}$$

where $V(d|q,c)$ denotes the value or relevance of the document with respect to the query q when the intended category was c, and $\Pr(c|q)$ denotes the likelihood of query q belonging to category c.

In words, this method incrementally builds the ranked list based on marginal relevance, which in this case is defined in terms the categories of a document, discounted by categories that have already been covered by previously-selected documents in the ranked list. Such an approach directly models relevant and novel information, which is a desirable property. However, while Agrawal et al. [2009] only use categories as the criteria for diversification, we explore various features like words, named entities, latent topics, as well the source of information (see Chapter 6).

2.5 Evaluation in Information Retrieval

An important part of research in information retrieval is the design of evaluation methods that allow systematic and objective comparison between different retrieval systems. A good evaluation strategy can provide valuable insights into various aspects of the performance of a system, and hence, guide the development of new retrieval approaches. The nature of the information retrieval task affects the the evaluation strategy in terms of the test collection as well as performance measures used for evaluation.

A standard approach for evaluation involves a collection of documents, a set of information needs expressed as queries, and relevance judgments, *i.e.*, a set of assessments that determine the relevance of a document with respect to each query. Once such a test collection is available, a retrieval system is run on the set of queries, and its output (*i.e.*, documents returned for each query) is evaluated against the known relevance judgments, in terms of various performance measures, which we describe below.

2.5.1 Evaluating Relevance-based Retrieval

2.5.1.1 Set-based Measures

When the order of the retrieved documents is not important, retrieval performance is most commonly measured in terms of the following measures.

Recall. Recall refers to the fraction of relevant documents that are retrieved by the system.

$$\text{Recall} = \frac{\#(\text{relevant documents retrieved})}{\#(\text{relevant documents})} \tag{2.5}$$

Precision. Precision refers to the fraction of retrieved documents that are relevant.

$$\text{Precision} = \frac{\#(\text{relevant documents retrieved})}{\#(\text{retrieved documents})} \tag{2.6}$$

The concepts of recall and precision were first introduced by Kent et al. [1954], and have been commonly used as well as analyzed in the information retrieval community [Salton, 1971; Raghavan et al., 1989; Cleverdon, 1993].

F measure. Neither recall nor precision alone provide a complete picture of a retrieval system's performance. It is trivial to maximize one at the expense of the other. Therefore, an ideal system achieves a balance between recall and precision. Moreover, the importance of recall vs. precision varies based on the nature of the information retrieval task. The F measure [Van Rijsbergen, 1979] addresses these issues by trading off recall against precision through a parameter $\beta \in [0, \infty]$.

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{2.7}$$

A commonly used value for β is 1. This leads to the F_1 measure, which is equal to the harmonic mean of recall and precision.

2.5.1.2 Ranked Retrieval Measures

In many scenarios including web search, the retrieval system produces a ranked list of documents in response to a query, so that more relevant documents appear higher in the ranked list. This assumes that users browse the ranked list in a top-down manner, and therefore, are more likely to examine the top-ranked documents. Correspondingly, ranked retrieval is evaluated using performance measures that are top-heavy, *i.e.*, pay more attention to documents at the higher ranks. We describe some of the commonly used measures below.

Precision at k (Prec@k). This refers to the fraction of documents that are relevant in the first k documents ranked by the system:

$$\text{Prec@}k = \frac{\#(\text{relevant documents in top } k \text{ ranks})}{k} \tag{2.8}$$

Such a measure is a good indication of the perceived quality of the first page of results produced by a search engine, and therefore, is commonly used for web search evaluation [Agichtein et al., 2006; Manning et al., 2008; Carterette et al., 2009].

Mean Average Precision (AP). Average precision [Buckley and Voorhees, 2005] refers to the average of the precision values at each relevant document in the ranked list.

$$\text{AP} = \frac{1}{|R|} \sum_{j:d_j \in R} \text{Prec@}j \tag{2.9}$$

where R refers to the set of relevant documents for the given query, and d_j refers to the document placed at rank j by the system. Average precision values are further averaged over queries to obtain Mean Average Precision (MAP).

Discounted Cumulative Gain (DCG). DCG is a recently proposed measure that is seeing increasing adoption in evaluating and comparing web retrieval systems, *i.e.*, search engines [Järvelin and Kekäläinen, 2002]. It is based on the notion of cumulative gain, *i.e.*, the gain accrued by a user who browses a ranked list in a top-down manner. Moreover, users are less likely to examine lower-ranked documents; this fact is incorporated by discounting the gain based on the rank of the document. The DCG at rank k as defined as follows:

$$DCG@k = \sum_{i=1}^{k} \frac{rel(d_i)}{(1 + \log_b(i))} \tag{2.10}$$

where $rel(d_i)$ is the relevance level of the i^{th} document in the ranked list, and the logarithmic base b controls the strength of the discount factor. NDCG is obtained by normalizing this score by the DCG score of the ideal ranked list.

DCG has the advantage of naturally incorporating graded relevance judgments, which is not directly possible with MAP, although alternatives have been proposed in literature, *e.g.*, the Q-measure [Sakai, 2005].

2.5.2 Evaluating Novelty-based Retrieval

2.5.2.1 Set-based Measures

The TDT and TREC studies treated novelty detection as a filtering task. Therefore, they evaluated retrieval systems using set-based measures. The TDT first story detection task used a cost function [Yang et al., 2002] based on number of misses and false alarms:

$$C_{fsd} = C_{miss} \cdot P_{miss} \cdot P_{target} + C_{fa} \cdot P_{fa} \cdot P_{non-target} \tag{2.11}$$

where C_{miss} and C_{fa} are the costs of a miss and false alarm, P_{miss} and P_{fa} are the retrieval system's conditional probabilities of a miss and false alarm, and P_{target} and $P_{non-target}$ are corpus-dependent prior probabilities of serving an on-target and off-target document, respectively.

The TREC novelty detection task used recall, precision, and F_1 (the harmonic mean of recall and precision), defined in terms of retrieval of relevant and novel sentences [Soboroff, 2004].

Since ranked retrieval was not the focus of these studies, no evaluation methodologies were developed that could assess the quality of a ranked list of documents in terms of their relevance and novelty.

2.5.2.2 Ranked Retrieval Measures

Sub-topic Recall. Certain metrics developed for evaluating diversity-based retrieval are also applicable to the evaluation of novelty-based retrieval. For instance, Zhai et al. [2003] proposed to measure diversity in terms of the coverage of sub-topics of the given query. Specifically, *$\mathcal{S}$-recall* was defined as the fraction of sub-topics covered by documents up to a given rank j:

$$\mathcal{S}\text{-}recall@j = \frac{|\bigcup_{i=1}^{j} \textsc{SubTopics}(d_i)|}{|\mathcal{S}|} \tag{2.12}$$

where $\textsc{SubTopics}(d_i)$ denotes the sub-topics covered by the i^{th} document, and $\mathcal{S}$ denotes the set of all sub-topics relevant to the given query. Such a measure can be applied to novelty-based retrieval if the coverage of (appropriately defined) new sub-topics is used as an indicator of the novelty of a document.

NDCU and $\alpha-$NDCG. Recently, the problem of developing novelty-sensitive metrics for ranked retrieval was attacked more directly by Yang et al. [2007] and Clarke et al. [2008], who proposed nugget-based variations of a popular metric called *Normalized Cumulated Discounted Gain* (NDCG) [Järvelin and Kekäläinen, 2002]. Yang et al. [2007] proposed

Normalized Cumulated Discounted Utility (NDCU) and Clarke et al. [2008] proposed α-NDCG. The main idea in both of these metrics is to compute the gain of each document in terms of the nuggets it contains, and each subsequent presentation of the same nugget leads to a diminishing return to reflect the decreased value provided by redundant information. Specifically, $\alpha-$NDCG defines the gain of the k^{th} document as follows:

$$\mathrm{G}[k] = \sum_{i=1}^{m} I(d_k, i)(1-\alpha)^{r_{i,k-1}} \tag{2.13}$$

where i iterates over all m nuggets, $I(d_k, i)$ indicates whether document d_k contains nugget i, and $r_{i,k-1}$ is the number of times nugget i appeared in documents up to rank $k-1$. The free parameter α is interpreted as the probability of assessor error, *i.e.*, the likelihood that the assessor incorrectly determined that a document contains a nugget. The total gain up to rank k is then computed as follows:

$$\alpha\text{-DCG}[k] = \sum_{j=1}^{k} \mathrm{G}[j]/\log_2(1+j) \tag{2.14}$$

However, none of these metrics are based on a model of user's browsing behavior, which limits their interpretability and prevents their extension to evaluation of multiple ranked lists in a consistent manner. These shortcomings also raise questions about the objective functions that are targeted by current approaches for optimizing novelty or diversity-based retrieval performance.

2.6 Relationship of Novelty-based Retrieval with Diversity-based Retrieval

Novelty detection is related to diversity-based retrieval, which refers to approaches that attempt to maximize the coverage of specific items of interest, *e.g.*, sub-topics of a query, topical categories of documents, and so on. The *SIGIR Workshop on Redundancy, Diversity, and Interdependent Document Relevance* [Radlinski et al., 2009] identified two types of diversity:

- **Extrinsic diversity**, or diversity as uncertainty about the information need. This type of diversity in search results aims to reduce the risk of misinterpreting an ambiguous query. For instance, in response to the query “apple”, the system could produce a ranked list of documents that are diverse with respect to their topical categories (*e.g.*, “food and health” and “corporate affairs”).

- **Intrinsic diversity**, or diversity as part of the information need. Even with an unambiguous query, the user might expect the search results to cover different aspects of the information need. For instance, in response to a broad query like "machine learning", the system could produce documents that cover multiple aspects like "inductive learning", "transductive learning", "active learning", etc.

Thus, the goals of novelty and diversity-based ranking are similar, especially intrinsic diversity as defined above: Maximizing the coverage of information in a finite set of documents indirectly corresponds to avoiding redundant information. This leads to considerable overlap in the techniques used for evaluation as well as optimization of novelty and diversity-based retrieval. We explore this connection in more detail in Section 3.4.

2.7 Relationship with Multi-document Summarization

Novelty-based ranking shares two goals with multi-document summarization: *maximize coverage*, *i.e.*, include all representative points from the documents, and *minimize redundancy*, *i.e.*, minimize the repetition of information between the passages included in the summary from different documents. These goals lead to similarities in the evaluation methodologies for the two retrieval tasks. In both scenarios, the system is rewarded for coverage of more information, which is measured in terms of distinct key facts in summarization [Mani et al., 1998], and in terms of distinct sub-topics or aspects of the query in case of ranked retrieval [Zhai et al., 2003].

This overlap of goals also leads to the use of similar techniques for novelty-based ranking and summarization. For instance, event-focused summarization techniques strive to summarize news events in terms of key relevant facts; Many approaches are based on using features like words and named entities [Ge et al., 2003; Toutanova et al., 2007] or latent topics [Hennig and Labor, 2009] as surrogates for the keys facts. Redundancy in the summary is minimized by using variants of the Maximum Marginal Relevance approach, *e.g.*, see Goldstein et al. [2000b]. Our proposed techniques for optimizing novelty-based ranking (Chapter 6) are therefore similar in spirit and use the same features for selecting the most informative documents or passages for inclusion in the ranked list: *i.e.*, words, named entities, and latent topics.

However, unlike summarization, we are interested in ranked retrieval, where the system must present the documents or passages in an optimal order, which takes into account the typical browsing behavior of users with respect to one or more ranked lists in a search session. This has important implications for the nature of the performance measures that we propose (Chapter 3) as well the ranking algorithm and approach for learning from user feedback (Chapter 6).

Chapter 3

A New Framework for Retrieval Evaluation

Our goal is to create a realistic measure of system utility as perceived by the user, which depends on the relevance as well as novelty of the information produced by the system in a single ranked list or in multiple ranked lists that are part of a single search session. The main problem associated with evaluating novelty-based retrieval is the non-independent nature of the utility of each document: The novelty of each document depends on other documents in the ranked list that will be seen by the user. Therefore, creating a static collection with novelty judgments in not possible. Instead, we model the utility of a document in terms of "nuggets", *i.e.*, pieces of information that are relevant to the user's information need (Section 3.1). The credit received for subsequent presentation of the same nugget is subject to a diminishing returns property, which reflects how the value of repeated information is perceived by real users (Section 3.1.1).

The dependence of novelty on the documents seen by the user means that the utility of a ranked list must be conditioned on the user's browsing behavior. However, user behavior is non-deterministic: Not every document in the system-produced ranked lists is necessarily read by the user. Therefore, our evaluation metric is based on a probabilistic model of user behavior (Section 3.2.1), which is used to calculate the *expected utility* of the system by summing over all possible ways of interacting with one or more ranked lists (Section 3.2.2).

This modeling power comes at the cost of increased computational complexity, which we address using efficient approximation techniques that lead to further insight and connections with existing evaluation methods (Section 3.3). The flexible modeling of the user's tolerance of redundancy as well as his browsing behavior leads to a metric that generalizes existing measures used in information retrieval research (Section 3.6.1) and also naturally extends to session-based retrieval while maintaining a consistent definition of novelty across multiple ranked lists.

3.1 Nugget-based Utility

In traditional information retrieval setups, the ground truth consists of relevance judgments for each query-document pair. These judgments can either be binary ("relevant" vs "non-relevant"), or graded. However, creating novelty judgements is not straightforward: One cannot assign a novelty level to each document independently. Instead, a document is novel or redundant only with respect to one or more other documents. Creating novelty judgments for every such combination of documents is clearly unfeasible. In order to create a concrete and operational definition of novelty, we borrow the idea of "nuggets" from the field of question answering (QA). In QA evaluations, a nugget is defined as any piece of information that an assessor can mark as either present or absent from a document [Voorhees, 2003; Lin and Demner-Fushman, 2005; Marton and Radul, 2006; Dang et al., 2006]. Obviously, a single document can contain zero or more nuggets, and each nugget can be present in zero or more documents.

As an example, given a query, *"What is a golden parachute?"*, two of the correct nuggets that answer this question are:[1]

```
Nugget 1.  Provides remuneration to executives who lose jobs.
Nugget 2.  Remuneration is usually very generous.
```

Note that the system's response may not contain these exact wordings to receive credit for these nuggets. For instance, the following response is marked as containing both these nuggets:

...The arrangement, which includes lucrative stock options, a hefty salary, and a golden parachute if Gifford is fired....

Naturally, this raises the question of how to determine whether a nugget is present in an arbitrary span of text produced by the retrieval system. If the retrieval unit is fixed, *e.g.*, documents, then one approach is to manually annotate the presence of nuggets in documents. This was the approach adopted by the TREC Interactive Track [Hersh and Over, 2000] as well as the diversity task in the TREC Web Track [Clarke et al., 2009]. To automatically determine the presence of nuggets in arbitrary spans of text, we propose a new approach based on "nugget-matching rules" in Section 4.3.1.

[1] This example appears in Voorhees [2003].

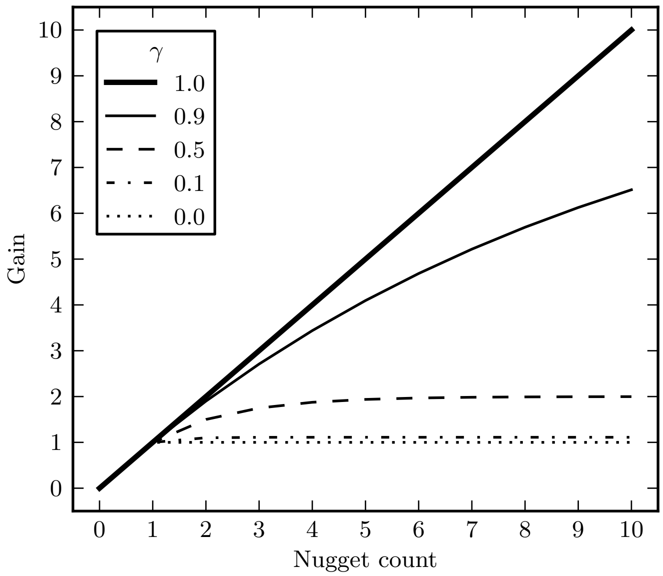


Figure 3.1: Gain as a function of nugget counts for different values of γ.

Once we represent documents as bags of nuggets, we can define the graded relevance of a document d with respect to a given query q as follows:

$$rel(d|q) = \sum_{\delta \in \Delta_q} I(\delta, d)\mathbf{w}(\delta, q) \tag{3.1}$$

where δ represents a nugget, Δ_q is the set of all nuggets relevant to query q, $I(\delta, d)$ indicates whether δ is present in document d, and $\mathbf{w}(\delta, q)$ represents the weight or importance of nugget δ with respect to query q.

3.1.1 Diminishing Returns

Eq. (3.1) does not account for novelty of information, *i.e.*, it does not differentiate between relevant-and-novel versus relevant-and-redundant nuggets. This is because the presentation of the same nugget multiple times leads to a linear increase in the gain to the user. In reality, each successive presentation of the same nugget decreases its value to the user. We capture this phenomenon of diminishing returns by introducing a parameter $\gamma \in \{0, 1\}$, which represents the factor by which the value of each subsequent presentation of a nugget is reduced. That is, the first occurrence of a nugget receives a unit gain of 1, the second occurrence receives γ, the third occurrence receives a gain of γ^2, and so on. Thus, the marginal gain of the i^{th} document, when each nugget δ has already occurred $\eta_\delta(d_{1:(i-1)})$ number of times in documents up to rank $i-1$ is given by:

$$G(d_i|q) = \sum_{\delta \in \Delta_q} I(\delta, d_i)\mathbf{w}(\delta, q)\gamma^{\eta_\delta(d_{1:(i-1)})} \tag{3.2}$$

The parameter γ represents the user's tolerance towards redundancy: Small values denote less tolerance, *e.g.*, $\gamma = 0.1$ reduces the value of a nugget by a factor of 10 for each subsequent presentation. Similarly, higher values of γ denote a high tolerance towards redundancy. In the extreme case, when $\gamma = 0$, the user is interested in seeing each nugget only once, and any subsequent appearance of the same nugget produces no additional gain. At the other extreme, $\gamma = 1$ represents the traditional assumption used in information retrieval, *i.e.*, each presentation of a relevant nugget produces the same increase in gain, irrespective of how many times it has already been seen by the user. Figure 3.1 shows the "diminishing returns" property of the gain function for different values of γ.

3.1.2 Cost of Reading

The cost of reading accounts for the total time and effort spent by the user in going through the system's output. This is a realistic factor in many scenarios, *e.g.*, adaptive filtering [Robertson and Soboroff, 2002; Fiscus and Wheatley, 2004], where the user is interested in tracking a particular topic. Obviously, the user does not have infinite time to read everything presented by the system, and is dependent on the system for filtering out irrelevant information. Therefore, the system must be penalized for producing outputs that require more time to read or browse. Another example is passage retrieval, where the system must choose between passages that differ not only in their relevance to the query but also in their lengths. In such a case, it would be natural to favor a retrieval system that can balance relevance as well as the total size of the system's output.

Let us look at a few ways in which we can define the cost of reading.

Unit cost per document. The simplest approach is to assign a constant cost to each document.

$$C(d_i|q) = c \tag{3.3}$$

Such a definition of cost penalizes the system in terms of the number of documents retrieved, and therefore, favors systems with high precision. The balance between gain and cost correspondingly controls the tradeoff between recall and precision of the system's output. Note that this definition does not differentiate between documents of different lengths, which might be an important consideration in certain retrieval setups.

Length-based cost. This approach uses length of the document as a proxy for the time and effort required for reading the document:

$$C(d_i|q) = \text{len}(d_i) \tag{3.4}$$

The length can be defined in terms of the number of words in the document. Such a definition of cost penalizes systems that retrieve long documents. The balance between gain and cost, then, controls the "information density" of information in the ranked list, *i.e.*, the amount of information per word in the ranked list.

Asymmetric costs. The above definitions of reading costs do not account for the fact that a user can immediately reject irrelevant documents, possibly based on the title or a snippet of the document. Instead, the user will concentrate his or her reading effort on the relevant documents. Therefore, relevant documents should be subject to a higher reading cost compared to irrelevant documents to accurately reflect the browsing effort of users.

$$C(d_i|q) = \begin{cases} c_1 & d_i \in R_q \\ c_2 & \text{else} \end{cases} \tag{3.5}$$

where c_1 is the reading cost of a relevant document, and is higher than c_2, which is the reading cost of an irrelevant document. R_q denotes the set of documents that are relevant to query q. The parameters c_1 and c_2 can be determined through user studies, especially eye-tracking experiments to determine the amount of time spent on each document in the ranked list.

3.1.3 Utility of a Ranked List

Gain. The total gain of a ranked list L with respect to a query q can be defined in terms of the per-document gain from Eq. (3.2):

$$\mathbb{G}(L|q) = \sum_{i=1}^{|L|} G(d_i|q)$$

(Using $\eta_\delta(L_{i-1})$ to denote number of times δ appears in documents up to rank $i-1$)

$$= \sum_{i=1}^{|L|} \sum_{\delta \in \Delta_q} I(\delta, d_i) \mathbf{w}(\delta, q) \gamma^{\eta_\delta(L_{i-1})}$$

$$= \sum_{\delta \in \Delta_q} \mathbf{w}(\delta, q) \left(\sum_{i=1}^{|L|} I(\delta, d_i) \gamma^{\eta_\delta(L_{i-1})} \right)$$

(Using $\eta_\delta(L)$ to denote number of times δ appears in the entire ranked list)

$$= \sum_{\delta \in \Delta_q} \mathbf{w}(\delta, q) \left(1 + \gamma + \gamma^2 + \cdots + \gamma^{\eta_\delta(L)-1}\right)$$

$$= \sum_{\delta \in \Delta_q} \mathbf{w}(\delta, q) \frac{1 - \gamma^{\eta_\delta(L)}}{1 - \gamma} \tag{3.6}$$

Cost. Similarly, the total cost can be defined in terms of per-document costs:

$$\mathbb{C}(L|q) = \sum_{i=1}^{|L|} C(d_i|q) \tag{3.7}$$

Utility. Now, we can define the total utility of a ranked list:

$$\mathbb{U}(L|q) = \mathbb{G}(L|q) - \mathbb{C}(L|q) \tag{3.8}$$

However, Eq. (3.8) assumes that the user reads each document returned by the system. Obviously, this is an unrealistic assumption. Users tend to read a ranked list in a top-down manner, and can abandon the ranked list at any rank due to various reasons, like satisfaction, frustration, and so on. To account for such non-deterministic behavior, we define a probability distribution over all possible user behaviors, which is then used to calculate the Expected Utility of the system, as described in the next section.

3.2 Expected Utility

We define Ω as the space of all possible user browsing patterns: each element $\omega \in \Omega$ denotes a possible way for a user to browse the ranked list, *i.e.*, to read a specific subset of the documents that appear in the ranked lists. Let us assume a probability distribution over the space Ω, such that $\Pr(\omega)$ corresponds to how likely it is for a user to read this set of documents. Intuitively, $\Pr$ should assign higher probability to subsets that include documents at top ranks, reflecting common user behavior. We leave the specific details of modeling user behavior to Section 3.2.1.

Once we have a way of representing different user interaction patterns ω, we can define the utility as a function of ω, *i.e.*, $\mathbb{U}(\omega)$. Note that $\mathbb{U}(\omega)$ is a random quantity, since ω is a random variable. Therefore, the obvious next step is to calculate the expected value of $\mathbb{U}$ with respect to the probability distribution defined over Ω. We call this quantity as Expected Global Utility:

$$\text{EGU} = \sum_{\omega \in \Omega} \Pr(\omega) \mathbb{U}(\omega) \tag{3.9}$$

3.2.1 User Browsing Patterns

As mentioned earlier, a user can read any subset of the documents presented by the system. We use Ω to denote the set of all subsets that the user can read. Naturally, the most general definition of Ω would be the power set of all documents in the ranked list, and the size of such a state space would be $2^{\sum_{i=1}^{K}|l_i|}$. This is a very large state space that would lead to difficulties in estimating a probability distribution as well as computing an expectation over the entire space. Another alternative is to restrict the space of possible browsing patterns by assuming that the user browses each ranked list in a *top down* manner without skipping any document, until he or she decides to stop. Thus, each possible user interaction is now denoted by a K-dimensional vector $\omega = \{s_1, s_2, ..., s_K\}$, such that $s_k \in \{1..|l_k|\}$ denotes the stopping position in the k^{th} ranked list. This leads to a state space of size $\prod_{i=1}^{K}|l_k|$, which is much smaller than the earlier *all-possible-subsets* alternative. We further make a reasonable assumption that the stopping positions in different ranked lists are independent of each other, i.e., $\Pr(\omega) = \Pr(s_1, s_2, ..., s_K) = \Pr(s_1)\Pr(s_2)\cdots\Pr(s_K)$.

The particular form of $\Pr(s)$, i.e., the probability distribution of stopping positions in a ranked list, can be chosen appropriately based on the given domain, user interface, and user behavior. Let us consider a few alternatives:

3.2.1.1 Persistence Model

One alternative is to assume that the user's decision to stop is only dependent on the rank, which indicates the persistence of the user. That is, more persistent users stop at a lower rank, whereas less persistent users stop at one of the top ranks in the search results. To model such user behavior, let us use a geometric distribution with an adjustable parameter p, which is the approach taken by Moffat and Zobel [2008] and also has empirical justification as we will see below. Note that the standard geometric distribution has an infinite domain, but each ranked list returned by the system will have a finite length. Therefore, we use a truncated geometric distribution, defined as follows.

Truncated Geometric Distribution. For a ranked list of length ℓ, the left-over probability mass beyond rank ℓ is assigned to the stopping position ℓ, to reflect the intuition that users who intended to stop before rank ℓ will be oblivious to the limited length of the ranked list, but all users who intended to stop at a rank lower than ℓ will be forced to stop at rank ℓ due to the limited length of the ranked list.

Hence, the stopping probability distribution for the k^{th} ranked list can be expressed by the following recursive formula:

$$\Pr(S_k = s) = \begin{cases} (1-p)^{s-1}p & 1 \leq s < |l_k| \\ 1 - \Pr(S_k < |l_k|) & s = |l_k| \\ 0 & \text{else} \end{cases} \tag{3.10}$$

Empirical Support for the Persistence Model. The above-mentioned persistence model that uses a geometric distribution to model stopping positions is further supported by empirical observations: [Joachims et al., 2005] conducted a user study to understand how users interact with the list of ranked results by tracking their eye movements. In Figure 3.2, we have plotted the probability of examining a document at each rank as per our persistence model (*i.e.*, $\Pr(S \geq s)$ in the geometric distribution), and compare it against the observed distribution of fixations (prolonged gaze) by the subjects of their user study. Such fixations are known to represent instances of information acquisition and processing [Just and Carpenter, 2002]. Evidently, the observed behavior of real users can be closely modeled using the geometric distribution for appropriate choice of parameters [2].

For the sake of completeness, in the appendix (Section A.1), we discuss another option for defining a stopping distribution on finite ranked lists, *i.e.*, a normalized geometric distribution, which seems more obvious at first, but leads to non-intuitive behavior of the utility metric. For this reason, we do not use the normalized geometric distribution, and Section A.1 can be skipped by the reader without loss of continuity.

3.2.1.2 Satisfaction-based Model

Till now we have considered stopping probability distributions that are only rank-based, but not satisfaction-based. In other words, the stopping probability in Eqs. (3.10) and (A.1) only depends on the rank, not on how much the user is satisfied by the document at that rank. However, recent studies have shown that the user's decision to stop depends on his satisfaction level while browsing the ranked list [Craswell et al., 2008; Chapelle and Zhang, 2009]. To reflect such user behavior, we can use more sophisticated probability distributions that take user satisfaction into account. While the "cascade model" [Craswell et al., 2008] makes a strong assumption that the user is interested in only one document and stops as soon as he finds it, we can go one step ahead and model the user's likelihood of stopping as proportional to the gain he has received up to the current rank, which is a more natural proxy for user satisfaction in

[2] We used $p = 0.2$ in a zero-based geometric distribution to allow for the fact that users can read nothing, *i.e.*, stop at the 0^{th} rank.

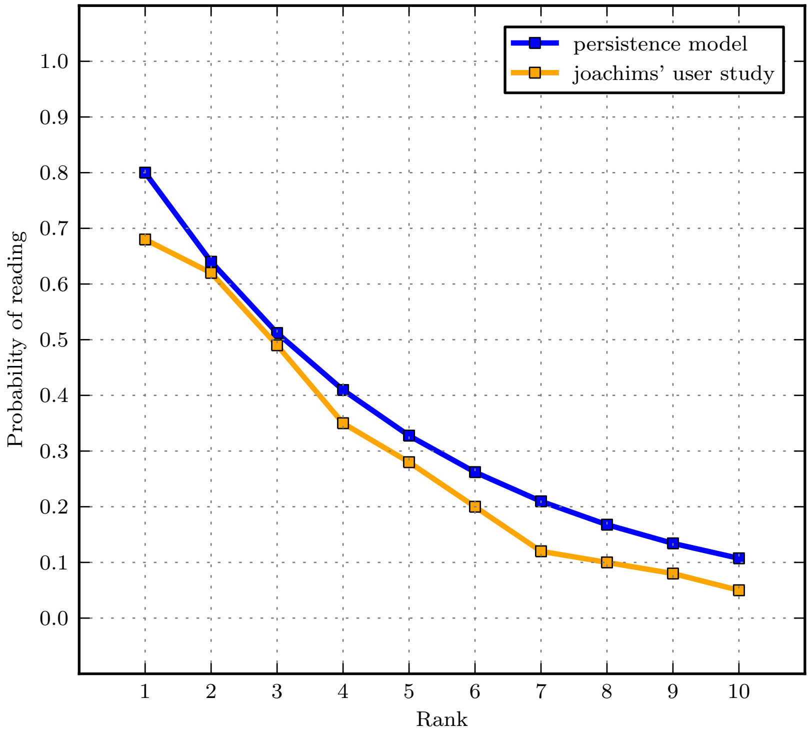


Figure 3.2: Comparison of the persistence model vs. the empirical distribution of eye fixations in the user study performed by Joachims et al. [2005].

our framework. Intuitively, we can use the following form for the stopping probability distribution:

$$\Pr(s) \propto \mathbb{G}(L_{1:s}) \tag{3.11}$$

Let us formalize this model by specifying the stopping distribution in terms of persistence as well as satisfaction:

Persistence-Satisfaction Model. Under this model, users decide to stop at a particular position either due to satisfaction (in terms of information collected) or lack of persistence (also known as abandonment). That is, a user's goal is to collect information on a particular topic; if the user is not satisfied at a particular rank, she will move onto the document at the next rank, but may also abandon her search due to frustration or lack of persistence. In accordance with our previous persistence model based on geometric distribution over stopping positions, let the probability of abandonment be p. Also, let the probability of satisfaction at rank s, denoted by $P_{sat}(s)$, be defined in terms of the current gain accumulated by the user, as follows:

$$P_{sat}(s) = \frac{\mathbb{G}(L_{1:s})}{\mathcal{G}} \tag{3.12}$$

where $\mathbb{G}(L_{1:s})$ denotes the gain accumulated up to rank s, as defined in Eq. 3.6, and $\mathcal{G}$ is the normalization factor that corresponds to the gain at which the user is completely satisfied.

Thus, the probability of stopping at position s is:

$$P(s) = \underbrace{(1-p)^{s-1}}_{A} \cdot \underbrace{\prod_{i=1}^{s-1} \Big(1 - P_{sat}(i)\Big)}_{B} \cdot \underbrace{\Big(P_{sat}(s) + \big(1 - P_{sat}(s)\big) \cdot p\Big)}_{C} \tag{3.13}$$

In words, probability of stopping at position s is equal to the product of the probability of not being satisfied by the first $s-1$ documents (term B), the probability of deciding to continue looking past the first $s-1$ documents (term A), and the probability of either being satisfied by the document at rank s or giving up at this point (term C). Note that if probability of satisfaction $P_{sat}(\cdot)$ is set to zero, Eq. 3.13 reduces to our original persistence model, *i.e.*, satisfaction plays no role in determining the stopping position of the user.

To understand the behavior of this user browsing model, in Figure 3.3, we plot this “persistence-satisfaction” based probability distribution and compare it with the simpler persistence-based probability distribution over stopping positions in the ranked list. The persistence-based probability distribution is simply a truncated geometric distribution as described above.

The persistence-satisfaction model depends on the positions of relevant information in the ranked list, unlike the persistence model, which only depends on the ranks. We used the following hypothetical ranked list (lower-case letters denote the nuggets present in each document):

d_1: `[]`
d_2: `[a,b]`
d_3: `[]`
d_4: `[]`
d_5: `[c,d,e]`
d_6: `[]`
d_7: `[]`
d_8: `[f]`
d_9: `[]`
d_{10}: `[g,h]`

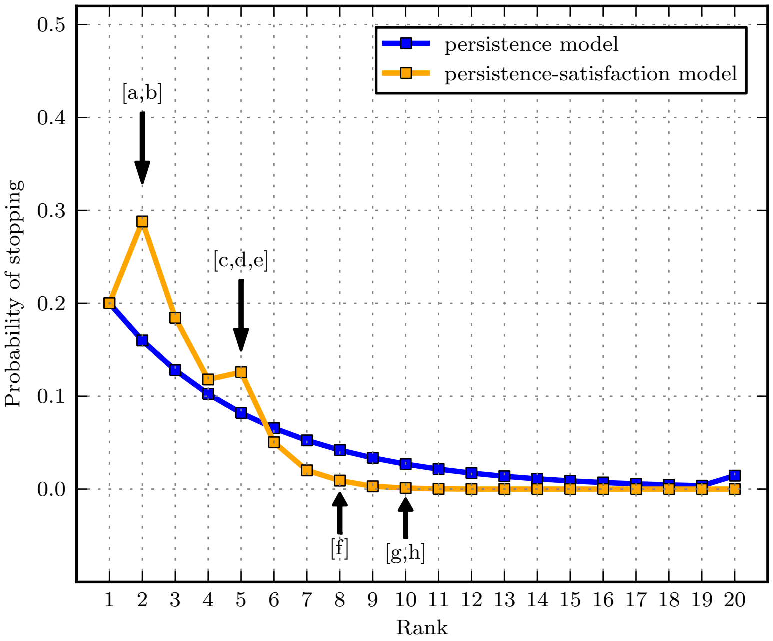


Figure 3.3: Comparison of the persistence and persistence-satisfaction user browsing models. The documents containing relevant nuggets have been indicated at their respective ranks.

Note how the peristence-satisfaction model spikes at ranks containing relevant documents. According to this model, the user is relatively more likely to stop at one of the relevant documents, which is intuitive and was also posited by other researchers as a reasonable assumption [Robertson, 2008]. Overall, this model shifts the probability mass towards the top ranks to account for the fact that if many relevant documents are present at the top ranks, then the user is more likely to be satisfied by reading fewer documents.

3.2.2 Utility Conditioned on User Browsing Patterns

The utility of multi-session ranked lists $l_1, l_2, ..., l_K$ depends on how a user interacts with them. We now define $\mathbb{U}(\omega)$ as the utility of multiple ranked lists conditioned on a user interaction pattern. Recall that $\omega = (s_1, s_2, ..., s_K)$ specifies the stopping positions in each of the ranked lists, allowing us to construct the list of documents actually read by the user for any given ω. We denote this list as $\mathcal{L}(\omega) = \mathcal{L}(s_1, s_2, ..., s_K)$, obtained by concatenating the top $s_1, s_2, ..., s_K$ documents from ranked lists $l_1, l_2, ..., l_K$, respectively. The conditional utility $\mathbb{U}(\omega)$ is defined as:

$$\mathbb{U}(\omega) = \mathbb{G}(\mathcal{L}(\omega)) - a \cdot \mathbb{C}(\mathcal{L}(\omega)) \tag{3.14}$$

Using $\eta_\delta(\mathcal{L}(\omega))$ to denote nugget counts in the synthetic ranked list based on stopping positions ω, we can write utility as:

$$\mathbb{U}(\omega) = \frac{1}{1-\gamma} \sum_{\delta \in \Delta_q} \mathbf{w}(\delta, q) \left(1 - \gamma^{\eta_\delta(\mathcal{L}(\omega))}\right) - a\,\mathrm{len}(\mathcal{L}(\omega)) \tag{3.15}$$

Expected Global Utility. Given the utility of multi-session ranked lists conditioned on each specific user browsing pattern, calculation of the expectation over all patterns is straightforward:

$$\begin{aligned} \mathbb{E}\left[\mathbb{U}(\omega)\right] &= \sum_{\omega \in \Omega} \Pr(\omega)\mathbb{U}(\omega) \\ &= \sum_{s_1=1}^{|l_1|} \cdots \sum_{s_K=1}^{|l_K|} \left(\prod_{k=1}^{K} \Pr(s_k) \right) \mathbb{U}(\underbrace{s_1, ..., s_K}_{\omega}) \end{aligned} \tag{3.16}$$

Note that we are making the reasonable assumption that the user's stopping positions in each of the ranked lists are independent of each other.

3.3 Tractable Computation over Multiple Ranked Lists

Unfortunately, the exact utility calculation quickly becomes computationally intractable as the number and lengths of ranked lists grow. Therefore, we must find a way to approximate the expected utility of multiple ranked lists.

First, we note that gain depends on the ranked lists and the stopping positions only through the nugget counts $\eta(\mathcal{L}(\omega))$, which act as "sufficient statistics" for calculating the total gain. To make this explicit, we can rewrite the gain function as follows:

$$\mathbb{G}(\mathcal{L}(\omega)) \equiv \mathbb{G}(\eta(\mathcal{L}(\omega))) \tag{3.17}$$

With this definition of gain, we can rewrite `EGU` in terms of expected gain and expected cost:

$$\mathbb{E}\left[\mathbb{U}(\omega)\right] = \mathbb{E}\left[\mathbb{G}(\eta(\mathcal{L}(\omega)))\right] - \mathbb{E}\left[\mathrm{len}(\mathcal{L}(\omega))\right] \tag{3.18}$$

Our approximations are based on the observation that we can move the expectation operator inside to obtain an approximate but more efficiently computable function.

First approximation. By moving the expectation operator inside the gain function, we can approximate the total gain (the first term above) as:

$$\mathbb{E}\left[\mathbb{G}(\eta(\mathcal{L}(\omega)))\right] \approx \mathbb{G}(\mathbb{E}\left[\eta(\mathcal{L}(\omega))\right]) \tag{3.19}$$

Thus, instead of calculating the *expected gain* with respect to different browsing patterns, we compute the gain obtained by the expected number of times each nugget will be read from all the ranked lists, *i.e.*, $\mathbb{E}\left[(\eta(\mathcal{L}(\omega)))\right]$.

Since the expected number of times each nugget is seen is calculated with respect to the probability distribution over stopping positions, $\mathbb{E}\left[(\eta(\mathcal{L}(\omega)))\right]$ is not restricted to integer values. Specifically, the approximated gain is equal to:

$$\mathbb{G}(\mathbb{E}\left[\eta(\mathcal{L}(\omega))\right]) = \sum_{\delta \in \Delta_q} \frac{1 - \gamma^{\mathbb{E}[(\eta_\delta(\mathcal{L}(\omega)))]}}{1 - \gamma} \tag{3.20}$$

Now, let us see why $\mathbb{G}(\mathbb{E}\left[\eta(\mathcal{L}(\omega))\right])$ can be calculated more efficiently than the original definition of gain. Since the number of times each nugget will be read in a single ranked list only depends on the possible stopping positions in that list and is independent of the stopping positions in other ranked lists, the computation can be decomposed into K terms as follows:

$$\begin{aligned}\mathbb{E}\left[\eta_\delta(\mathcal{L}(\omega))\right] &= \sum_{k=1}^{K} \mathbb{E}\left[\eta_\delta(l_k(s_k))\right] \\ &= \sum_{k=1}^{K} \sum_{s_k=1}^{|l_k|} \Pr(s_k)\eta_\delta(l_k(s_k))\end{aligned} \tag{3.21}$$

where $\eta_\delta(l_k(s_k))$ denotes the number of times nugget δ would be read in the k^{th} ranked list when the stopping position is s_k. Thus, the approximate computation requires a sum over $O(|l_1| + |l_2| + ... + |l_K|)$ terms, instead of the $O(|l_1| \times |l_2| \times ... \times |l_K|)$ terms in the original calculation, which must consider all combinations of stopping positions in the K ranked lists.

Jensen's inequality. Note that gain, as a function of nugget counts, is a concave function (see Figure 3.1). Therefore, by Jensen's inequality, our approximation overestimates the actual gain:

$$\mathbb{E}\left[\mathbb{G}(\eta(\mathcal{L}(\omega)))\right] \leq \mathbb{G}(\mathbb{E}\left[\eta(\mathcal{L}(\omega))\right]) \tag{3.22}$$

The equality is achieved for $\gamma = 1$, when $\mathbb{G}(.)$ becomes a linear function of the nugget counts. When $\gamma < 1$, the curvature of the gain function increases with decreasing γ, and the approximate calculation becomes more prone to overestimating the true value of gain.

To assess the quality of the approximation, we compared the approximate calculation against the exact calculation on randomly generated ranked lists for different values of γ; see Figure 3.4 on Page 45. The first row in Figure 3.4a shows the first approximation for unnormalized gain values. As expected, the quality of the approximation improves with increasing values of γ. The first row in Figure 3.4b shows the corresponding first approximation for the normalized gain values, *i.e.*, divided by the score of the ideal ranked list. In this case, the scores of the system's ranked list as well as the ideal ranked list are overestimated by the approximation procedure, which negates the error to some extent. Therefore, the first approximation proposed here is a good surrogate for the exact calculation for all values of γ, as long as we are interested in the normalized `EGU` scores.

Second approximation. We can further approximate gain by moving the expectation operator further inside:

$$\mathbb{G}(\mathbb{E}\left[\eta(\mathcal{L}(\omega))\right]) \approx \mathbb{G}(\eta(\mathcal{L}(\mathbb{E}\left[\omega\right]))) \tag{3.23}$$

which corresponds to calculating the gain for the nugget counts that will be obtained if the user stopped at the expected stopping position in each ranked list. In our case, this is $1/p$, the expected value of the geometric distribution with parameter p. This level of approximation is equivalent to calculating the gain at a fixed "cut-off" position, as is commonly done with traditional metrics, *e.g.*, recall@k, NDCG@k.

However, unlike the first approximation, where we used the concavity of the gain function to derive the upper-bound relationship of the approximation against the exact calculation, we cannot derive such a relationship here, because the nugget counting function $\eta(.)$ is neither convex nor concave. In fact, it is a non-decreasing step function of the stopping position: When the stopping position increases by one, $\eta_\delta(.)$ increases by one if the next document contains the nugget δ, otherwise, it remains the same as its previous value. This leads to a worse approximation as compared to the first approximation described above, which is also evident in the second rows of Figures 3.4a as well as 3.4b.

Efficient computation of cost. The cost of reading, as defined in Section 3.1.2, is independent for each document, and hence, can is also independent for each ranked list, only dependent on the stopping position in the current ranked list. Therefore, it can be efficiently calculated without requiring any approximation:

$$\begin{aligned}\mathbb{E}\left[\mathrm{len}(\mathcal{L}(\omega))\right] &= \sum_{s_1}\cdots\sum_{s_K}\Pr(s_1)\cdots\Pr(s_K)\,\mathrm{len}(\mathcal{L}(s_1,\cdots,s_K)) \\ &= \sum_{s_1}\cdots\sum_{s_K}\Pr(s_1)\cdots\Pr(s_K)\left[\mathrm{len}(l_1(s_1))+\cdots+\mathrm{len}(l_K(s_K))\right] \\ &= \left[\sum_{s_1}\Pr(s_1)\,\mathrm{len}(l_1(s_1))\right]\times\cdots\times\left[\sum_{s_K}\Pr(s_K)\,\mathrm{len}(l_K(s_K))\right]\end{aligned}$$

which can be calculated in $O(|l_1|+|l_2|+...+|l_K|)$ or $O(l_{avg}\cdot K)$ time.

3.4 Applicability to Diversity-based Retrieval

In Chapter 2, we argued that the goals of diversity-based retrieval often overlap with novelty-based retrieval. Fortunately, the use of nuggets as units of retrieval enables a unified treatment of novelty and diversity-based retrieval in our framework. Specifically, we can use a flexible definition of nuggets that includes sub-topics of the query, pieces of factual information (factoids), topical categories, or other properties of documents like author, geographical location, and so on, that are of interest to the user. The use of sub-topics as nuggets corresponds to the task of "aspect retrieval", which aims to cover multiple distinct aspects of the information need (*e.g.*, different applications of robotics) [Zhai et al., 2003]. The use of categories as nuggets corresponds to the task of "intent-aware retrieval", which aims to diversify search results in response to an ambiguous query (*e.g.*, food and business-related documents in response to "apple") [Agrawal et al., 2009].

Moreover, these definitions of nuggets are not exclusive. In fact, a user is likely to desire diversity across multiple dimensions, albeit, with different tolerances for redundancy for each dimension. For instance, the user might want to see different aspects of a product review, preferably from different websites, but definitely from different users. Such a model of diversity can be easily incorporated in our evaluation framework by allowing the answer key for each query to comprise multiple types of nuggets (in this case—product aspects, website URLs, and authors of reviews). Moreover, each such type would have its own dampening factor γ to model different tolerances towards redundancy in each nugget type.

Thus, a document's gain (Eq. 3.2) can be re-written as:

$$G(d_i|q) = \sum_{\delta\in\Delta_q} I(\delta,d_i)\mathbf{w}(\delta,q)\gamma_{T(\delta)}^{\eta_\delta(d_{1:(i-1)})} \tag{3.24}$$

where $T(\delta)$ represents the type of the nugget δ, *e.g.*, author, geographical location, etc. The weights $\mathbf{w}(\cdot)$ can also be used to assign more importance to certain types of nuggets, *e.g.*, to capture the assumption that the authorship of an article is more important than the geographical location of the article, and so on.

3.5 Normalized Expected Global Utility

To make EGU scores interpretable across queries, we normalize the EGU score with respect to the best score that can be obtained for the given query. Note that EGU scores can be negative due to the notion of cost. Therefore, we use the following "shifted" form for normalizing EGU scores:

$$\text{Normalized } \mathtt{EGU} = \frac{\mathtt{EGU} \;\; - \mathtt{EGU}_M}{\mathtt{EGU}_I - \mathtt{EGU}_M} \tag{3.25}$$

where $\mathtt{EGU}_M$ refers to the minimum possible value for EGU, and $\mathtt{EGU}_I$ refers to the best possible value of EGU as obtained on the ideal ranked list. In the subsequent sections, we show how to compute the ideal (highest) as well the minimum EGU scores.

3.5.1 Computing the Ideal Ranked List

The ideal ranked list is by definition the ranked list with the highest possible EGU score that can be obtained for the given query. For purely relevance-based metrics, which assume that the utility of each document is independent of others, computing the ideal ranked list is straight-forward: simply rank the documents by decreasing grades of relevance, breaking ties arbitrarily.

For measures like EGU where the score is based on coverage of discrete elements (nuggets in our case), computing the ideal ranked list turns out to be an NP-hard problem. In Chapter 5, we prove this for EGU, and also study the performance of approximate algorithms.

3.5.2 Computing the Minimum Possible Value of EGU

Similar to computing the ideal ranked list, finding the ranked list with the smallest EGU is non-trivial. However, a trivial lower bound for EGU can be obtained as follows: Rank documents by decreasing value of their costs, and compute the expected value of reading these documents without taking their gains into account. A similar approach has been used for computing the lower bound for combinatorial objective functions that can take negative values, *e.g.*, see the uncapacitated facility location problem

[Cornuejols et al., 1977b]. Due to the geometric distribution over stopping positions, the minimum $\texttt{EGU}$ will be a finite value[3] even for an infinite list of documents.

For retrieval scenarios where cost is not involved, $\texttt{EGU}_M$ is 0, which leads to a simpler definition of normalized $\texttt{EGU}$

$$\text{Normalized } \texttt{EGU} = \frac{\texttt{EGU}}{\texttt{EGU}_I} \tag{3.26}$$

We use this form in retrieval scenarios where reading cost is not of any concern, *e.g.*, standard adhoc search. The "shifted" form (Eq. 3.25) will be used to evaluate filtering scenarios where cost is taken into account.

3.6 Related Work

User Models for Evaluation. Recently, there has been increasing interest in evaluation metrics that are based on a model of user behavior. For example, Moffat and Zobel [2008] proposed *Rank-Biased Precision* (RBP), which corresponds to the expected rate of gain (in terms of graded relevance) obtained by a user who reads a ranked list *top down*, and whose stopping point in a ranked list is assumed to follow a geometric distribution. Similarly, Robertson [2008] re-interpreted *average precision* as the expected precision observed by a user who stops with uniform probability at one of the relevant documents in the ranked list returned by the system. Sakai and Robertson [2008] proposed a more general metric called *Normalized Cumulative Utility*, which combines graded relevance with a probabilistic model of user's stopping behavior. However, all these metrics are designed for measuring utility purely in terms of *relevance* – binary or graded. Recently, Chapelle et al. [2009] proposed a new metric called *Expected Reciprocal Rank* (ERR) that is based on a model of how users interact with ranked lists. ERR penalizes documents that are ranked below relevant ones, thus, simulating the non-independent nature of the utility of documents as perceived by the user. However, ERR does not explicitly model novelty. Instead, any relevant document decreases the value of other documents, even if they contain novel and relevant information.

Clickthrough-based Evaluation. There has been a recent trend in evaluating retrieval systems directly based on clickthrough data, which is abundantly available in a deployed system, *e.g.*, a web search engine. Such approaches have the advantage of not requiring manual relevance assessments. Moreover, they reflect the preferences of real users of the system. On the downside, such approaches necessarily expose the the retrieval system to real users, which might not be prudent when testing a new and potentially risky retrieval algorithm. Some of the simplest click metrics include

[3] For example, if the probability of stopping is p and the cost of reading each document is c, then the lower bound on $\texttt{EGU}$ is $-c/p$.

"number of clicks received in a retrieval session", *"whether or not a click was observed at all in a session"*, and other variants like *"max, min, or mean reciprocal ranks of clicks"* [Radlinski et al., 2008b; Chapelle et al., 2009]. Joachims [2002] proposed an approach for comparing two search engines by observing clickthrough behavior on a combined ranking of documents from the two search engines, while avoiding presentation bias. This is achieved through a special experimental design that involves sending the query to both search engines and mixing their results. However, such an approach assumes that users click on documents independent of the presence of other documents. That is, the interaction between documents is ignored, which makes this approach unsuitable for novelty or diversity-based evaluation. Dupret and Liao [2010] proposed a model of user browsing behavior to estimate the intrinsic relevance of a document based on clickthrough data. Their approach uses observed clicks to learn the underlying utilities of documents through a logistic model. While such an approach allows modeling of user behavior in a search session, it is based on a linear utility function, *i.e.*, the total utility of a session is the sum of utilities of clicked documents, with no regard to repetition of information in the documents. This makes the model inapplicable to novelty-based evaluation, unlike our proposed model that correctly models the diminishing returns of redundant information.

Simulation-based Evaluation. [Keskustalo et al., 2008] proposed an approach for realistic evaluation of a retrieval system by simulating a user's interaction with the system, which involves reading documents and providing relevance feedback. However, their user model provides a coarse approximation of reality: Although the authors acknowledge that users may have different patience levels and hence stop at different rank positions, they make deterministic assumptions about stopping positions, and experiment with only a few values: 1, 5, 10, and 30. We address the problem of non-deterministic user behavior by defining a probability distribution over all possible user interactions, and using it as a basis for calculating the expected performance of the system. In other words, we can directly estimate the average performance that would be obtained if the simulation were repeated many times, except that we do not have to explicitly run the simulations. Instead, we can calculate the statistical expectation in closed form.

Novelty-detection in TDT and TREC. Most novelty detection approaches and benchmark evaluations conducted in TDT and TREC have shared a convention of producing novelty judgments in an *offline* manner: All the documents (or sentences) which are relevant to a query are listed in a pre-specified (temporal) order, and a binary judgment about the novelty of each document is made, based on how its content differs from previous documents in the list [Allan et al., 2003]. Such novelty judgments would be correct from a user's perspective only if *all* these documents were presented by the system to the user, and in the exact same order as they were laid out during the ground truth assignment. These conditions may not hold in realistic use of a retrieval system, which could show both relevant and non-relevant documents to the user, ranked according to its own notion of "good" documents.

Evaluation of Session-based Retrieval. To evaluate retrieval systems in multi-session search scenarios, Järvelin et al. [2008] proposed an extension to the Discounted Cumulated Gain (DCG) metric, known as session-based DCG (sDCG) that discounts relevant results from later retrieval sessions to favor early retrieval of relevant information in multi-session search scenarios. This is based on the assumption that examining retrieved results and reformulating the query involves an effort on the part of the user. However, sDCG lacks a model of user behavior, *i.e.*, a model of stopping positions in each ranked list. Therefore, when detecting duplicate documents in the current ranked list, there is no natural way for modeling the documents that were read in the previously seen ranked lists. The authors are forced to make deterministic assumptions, *e.g.*, *"All users read 10 documents in each ranked list"*.

User's Reading Effort. There has been limited work in combining relevance with reading cost to perform document retrieval. In the context of XML retrieval, Shimizu and Yoshikawa [2007] proposed a ranking method based on benefit and reading effort of XML elements. The reading effort allows taking into account the size and nesting level of the retrieved XML elements. Arvola et al. [2010] proposed a new approach for passage and XML retrieval based on the concept of reading effort measured in terms of number of text characters that the user is expected to browse through. Expected Search Length (ESL) [Cooper, 1968] defines retrieval performance as the expected user effort in terms of the number of irrelevant documents that the user has to browse in order to retrieve a give number of relevance documents. The reading effort in ESL corresponds to using unit cost per document, while our framework allows more general definitions of cost. Moreover, we allow the tradeoff between gain and cost to be controlled using a parameter so as to flexibly model the constraints of various retrieval scenarios.

3.6.1 Comparison and Relationship with Existing Metrics

NDCG. Järvelin and Kekäläinen [2002] proposed *Normalized Discounted Cumulative Gain* to measure the performance of retrieval systems in terms multiple grades of document relevance, while explicitly discounting the contribution of documents at lower ranks. NDCG has become a very popular as an intuitive measure of retrieval performance. Multiple definitions exist; Järvelin et al. [2008] define the discounted cumulative gain at rank k as:

$$DCG@k = \sum_{i=1}^{k} \frac{rel(d_i)}{(1 + \log_b(i))} \tag{3.27}$$

where $rel(d_i)$ is the relevance level of the i^{th} document in the ranked list, and the logarithmic base b controls the strength of the discount factor. NDCG is obtained by normalizing this score by the DCG score of the ideal ranked list.

Our metric generalizes NDCG by replacing $rel(i)$ by a nugget-based definition of gain in terms of relevance and novelty, and including a notion of reading cost. In fact, EGU becomes equivalent to NDCG when:

- Choosing $\gamma = 1$, *i.e.*, which corresponds to a purely relevance-based definition of gain,
- Choosing $a = 0$ (see Eq. 3.8), *i.e.*, which corresponds to no reading cost and making the metric purely recall-based, and
- Choosing a user browsing model such that probability of reading a document at rank i is proportional to $1/(1 + \log(i))$.

Since NDCG is a purely recall-based metric, it is generally evaluated up to a "cut-off" rank, referred to as NDCG@k. In Section 3.3, we described how such a rank cut-off corresponds to a second-level approximation of EGU.

***$\mathcal{S}$*-recall.** Zhai et al. [2003] proposed *$\mathcal{S}$-recall* for measuring the coverage of sub-topics in a system-produced ranked list (cf. Chapter 2). Our proposed metric overcomes two shortcomings of *$\mathcal{S}$-recall*:

1. *$\mathcal{S}$-recall* is purely set-based: a sub-topic or nugget is either covered or not covered by the ranked list. It cannot model a user who desires some level of redundancy in the ranked list. Our proposed framework provides a flexible approach for accommodating different tolerances towards redundancy through the parameter γ, and *$\mathcal{S}$-recall* simply corresponds to the special case of $\gamma = 0$.
2. *$\mathcal{S}$-recall* does not account for the browsing behavior of real users, whose likelihood of reading a document decreases with its rank. No distinction is made between the documents up to rank j at which *$\mathcal{S}$-recall* is computed. In our framework, the probability distribution over stopping positions provides a flexible approach to emphasizing the importance of top ranks to different extents, and *$\mathcal{S}$-recall@j* corresponds to putting all probability mass at rank j in the probability distribution over stopping positions in the ranked list.

α-NDCG. Clarke et al. [2008] proposed α-NDCG as a variation of NDCG to model relevance and novelty in terms of nuggets. However, there are two important differences between α-NDCG and our framework:

1. α-NDCG does not explicitly model the user's tolerance towards redundancy. Instead, they use "probability of assessor error" as an indirect way to model the novelty of the ranked list. In their framework, a parameter α denotes the probability that a document contains a nugget when the assessor marks it as such. α acts as an indirect way of controlling the tolerance for redundancy: To model a user who does not care about novelty (*i.e.*, the traditional relevance-based retrieval setup), the authors use $\alpha = 0$, which—indirectly—corresponds to the case that the assessor is always wrong when he or she detects a nugget in a document.

   In our framework, we take a more direct route: We assume that nugget judgments are accurate, and model user's tolerance towards redundancy using a parameter γ, which corresponds to the factor by which the gain of a nugget is reduced on each subsequent presentation to the user.

2. α-NDCG is not based on a model of user browsing behavior, *i.e.*, the likelihood of user stopping at various ranks. Therefore, it does not naturally extend to multiple ranked lists since it is not clear which nuggets in documents from previous ranked lists would be deemed as read by the user for the purpose of evaluating the current ranked list.

   Our framework is based on a probabilistic model of user behavior, which allows us to extend the notion of novelty to multiple ranked lists in a principled manner.

3.7 Summary

In this chapter, we proposed a new evaluation measure called Expected Global Utility (EGU) that realistically models the utility derived by a user going through one or more ranked lists in a search session. EGU is based on three components: (i) a notion of gain, which is defined in terms of relevant and novel nuggets, (ii) a notion of cost, which reflects the effort required to examine the search results, and (iii) a probabilistic model of user browsing behavior, which enables the formulation of the *expected* utility of a ranked list for an average user.

Our definition of gain follows a diminishing returns property, which allows modeling different tolerances towards redundancy. We proposed three different notions of reading cost, defined in terms of the number of documents, the word-length of the documents, and also in terms of the relevance of the documents (*i.e.*, different costs for relevant and irrelevant documents). Our evaluation measure, EGU, admits the various task-dependent parameters as shown in Table 3.1.

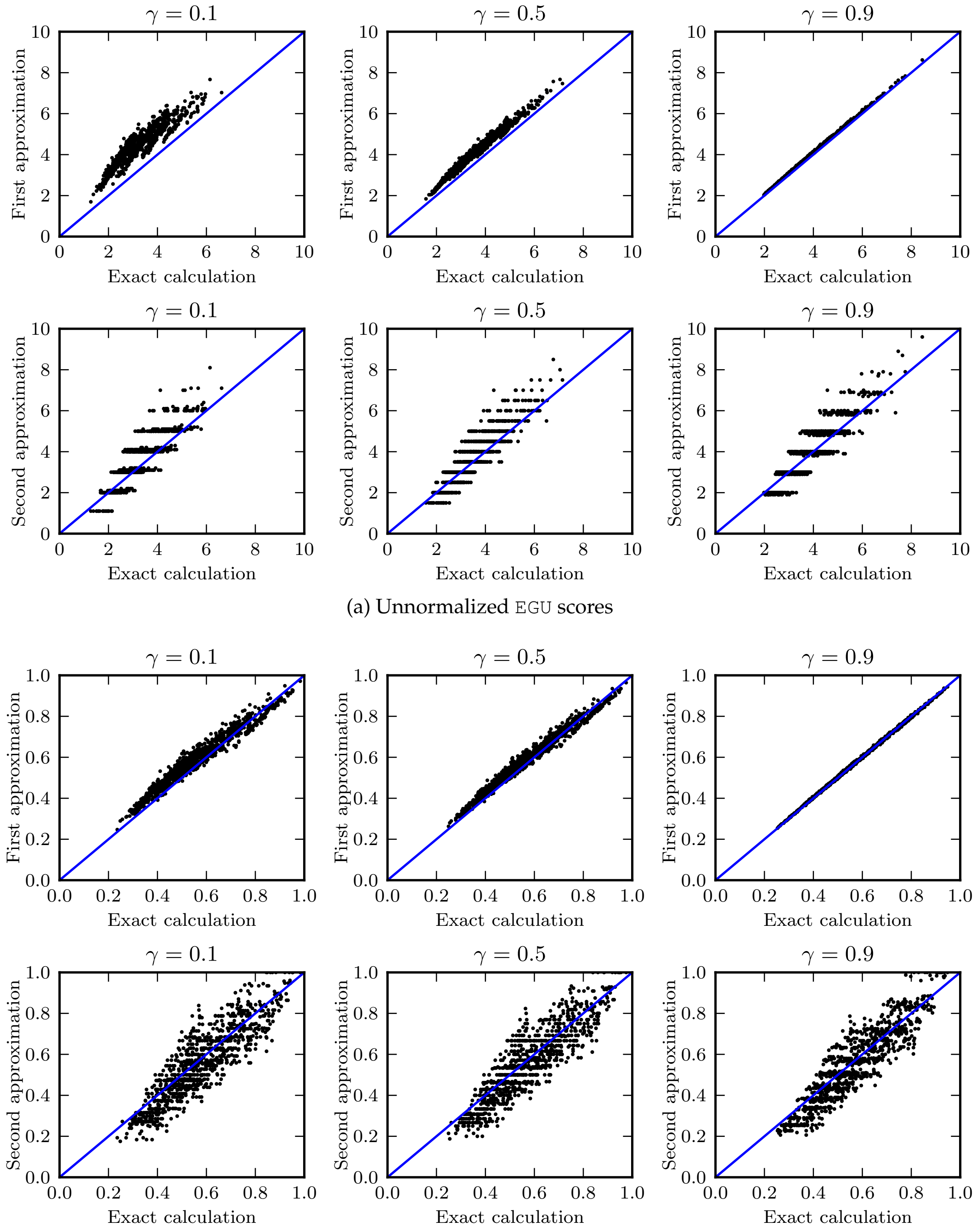
Table 3.1: Task-dependent parameters in EGU.

Parameter	Description
γ	Tolerance towards redundancy (Higher values signify more tolerance for redundancy)
p	Browsing persistence (Higher values signify impatient users)
c	Reading effort (Higher values signify higher cost of reading each retrieved document)

We explored two alternatives for the modeling of user browsing behavior: (i) The persistence model posits that the user's likelihood of stopping in the ranked list depends solely on the rank, and (ii) The persistence-satisfaction model more realistically posits that the user's likelihood of stopping depends on the rank as well as his or her level of satisfaction at the given rank. However, it is not known how the use of these models affects the evaluation of a retrieval system. We answer this question by conducting experiments with these two models on existing TREC runs in Chapter 4.

We also showed how the measure can be computed efficiently over multiple ranked lists without summing over a very large space of user interactions. We remarked that computing the ranked list that will maximize EGU is an NP-hard problem. We explore the effectiveness of approximate algorithms for this problem in Chapter 5.

The applicability of the proposed nugget-based evaluation measure is limited without an appropriate testbed that includes ground truth in the form of nuggets. We have created such a testbed that can be used for evaluating novelty-based retrieval, and also more sophisticated retrieval scenarios like information distillation, which allow systems to output arbitrary spans of text in response to queries. We describe this dataset next in Chapter 4, where we also study the behavior of EGU on another task: diversity-based retrieval that was part of the TREC 2009 Web Track.



(a) Unnormalized EGU scores

(b) Normalized EGU scores

Figure 3.4: Comparison of exact and approximate calculations of EGU for different values of γ.

Chapter 4

Behavior of EGU on Actual Retrieval Systems and Datasets

We would like to understand how the proposed evaluation measure behaves on actual retrieval systems, and how it compares with other well-understood measures like precision and $\alpha-$NDCG [Clarke et al., 2008]. Note that EGU is not a single evaluation metric, but a family of metrics parameterized by user persistence p, *i.e.*, the stopping probability of users, and redundancy tolerance γ, and reading cost c (see Table 3.1). Therefore, it is instructive to understand how different settings of these parameters affect the evaluation of actual retrieval systems.

To enable automatic evaluation using the proposed framework, we need a testbed comprising a set of documents, information needs, and associated ground truth in the form of nuggets. For certain retrieval scenarios, existing test collections can be reused for nugget-based evaluation. For instance, the TREC interactive track [Hersh and Over, 2000] investigated the task of aspect retrieval, where the goal is to retrieve multiple distinct aspects of a given topic. More recently, the TREC Web Track [Clarke et al., 2009] investigated the task of diversity-based retrieval on the web. In both cases, the answer keys were defined in terms of the aspects or sub-topics of the information need that must be covered by the documents returned by the system. These aspects or sub-topics can be treated as nuggets for the purposes of the evaluation framework proposed in this thesis. We analyze the behavior of the proposed evaluation measure on the TREC diversity-based retrieval task in Section 4.1. In this chapter, we only focus on the relative performance of *existing* retrieval approaches that were submitted to the TREC Web Track. We conduct more detailed comparisons of existing approaches with *our* proposed novelty-based ranking approach in Chapter 6.

Another retrieval scenario that we are interested in is information distillation, which truly showcases the full power of the proposed evaluation framework, as we argue in

Section 4.2. As explained in Section 2.2.3, information distillation is targeted towards intelligence analysts, who need to gather information on events around the world by monitoring various sources. A unique aspect of this task is that distillation systems are allowed to produce any readable response text that is consistent with the sources, optionally paraphrasing or summarizing the source text [White et al., 2008]. Therefore, we can no longer use document-level relevance judgments to evaluate such systems. Instead, we need the ability to automatically determine the relevance and novelty of arbitrary spans of text. Nuggets are a natural choice as the evaluation unit in this case. However, we cannot create a test collection by manually assigning nuggets to every possible span of text. Instead, we require an automatic approach for determining the presence of a nugget in the system's output.

To solve these problems associated with evaluation of distillation systems, we have created a new test collection by extending a publicly available dataset used in Topic Detection and Tracking studies with new queries that correspond to the information needs of intelligence analysts. We also created corresponding answer keys as well as "nugget-matching rules" that automatically match these nuggets with arbitrary system output (Section 4.3).

First, let us begin by describing our experiments on the diversity task in the TREC 2009 Web Track.

4.1 TREC 2009 Web Track: Diversity-based Retrieval

In diversity-based retrieval, the goal is to retrieve multiple distinct aspects or facets of the user's information need. We focus on the diversity task that was part of the TREC 2009 Web Track [Clarke et al., 2009]. This task used 50 information needs, each further divided into multiple (4.9 on average) facets. Annotators manually determined the presence of each facet in (a subset of) the documents of the test collection. These facets can be treated as nuggets in our evaluation framework. Table 4.1 shows a sample information need that was part of the test suite in the diversity task in TREC 2009. Evaluation was based on the ClueWeb09 dataset, which consists of over 1 billion webpages crawled from the Internet in January and February, 2009 by researchers at Carnegie Mellon University[1]. The diversity task used a subset of the ClueWeb09 dataset that consists of 500 million English webpages.

We use the outputs of retrieval systems submitted to TREC, and measure their performance using various settings of the proposed metric. Each such output is known as a "run" in TREC terminology, and consists of the document rankings produced by a retrieval system, or a particular configuration of a retrieval system, on the provided

[1] http://boston.lti.cs.cmu.edu/Data/clueweb09/

Table 4.1: A sample information need used in the diversity task in TREC 2009.

Information need:	Find information on air travel, airports, and airline companies.
Query:	`air travel information`
Subtopic 1:	What restrictions are there for checked baggage during air travel?
Subtopic 2:	What are the rules for liquids in carry-on luggage?
Subtopic 3:	Find sites that collect statistics and reports about airports, such as flight delays, weather conditions, etc.
Subtopic 4:	Find the AAA's website with air travel tips.
Subtopic 5:	Find the website at the Transportation Security Administration (TSA) that offers air travel tips.

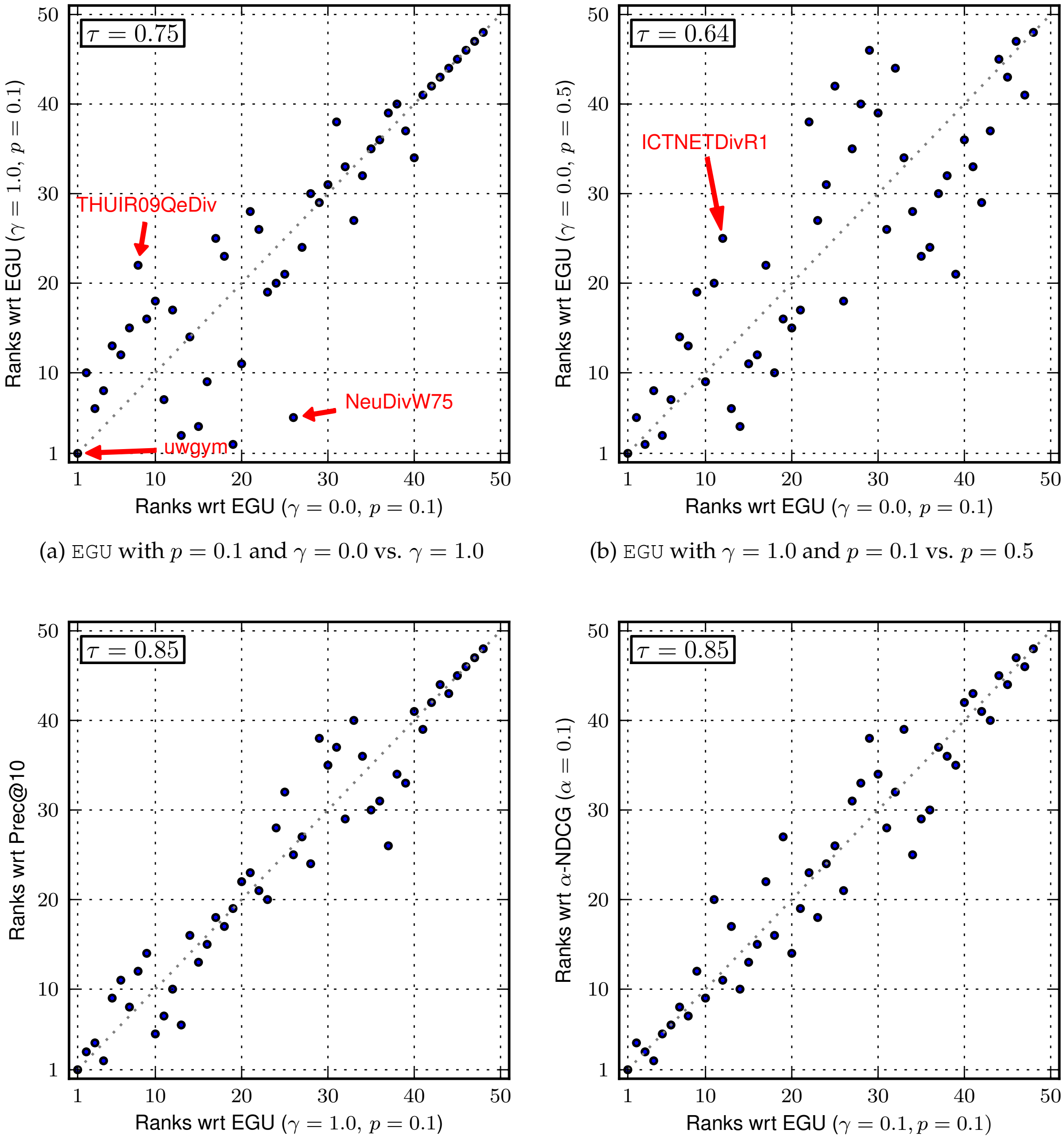
set of 50 queries. We have access to 48 such runs that were submitted by various universities and research labs for this task.

Our goal is to evaluate these runs under various parameterizations of EGU and observe the changes in their relative performance, as described below:

Effect of redundancy. First, we would like to understand how system rankings change when redundancy is taken into account. In Figure 4.1a, we compare the rankings of systems for EGU with $\gamma = 0.0$, *i.e.*, no redundancy tolerance, on the X-axis, against EGU with $\gamma = 1.0$, *i.e.*, full redundancy tolerance, on the Y-axis, which corresponds to the traditional relevance-based retrieval setting. The former case corresponds to diversity-based retrieval, while the latter case corresponds to traditional ad-hoc retrieval, which does not penalize repetition of information. In both cases, the user stopping probability was set to $p = 0.1$, which indicates an average reading length of ten documents for each user.

The general trend of the scatterplot along the diagonal line in Figure 4.1a indicates that runs that perform well on ad-hoc retrieval also perform well on diversity-retrieval. For instance, uwgym (in the lower left corner) is the top-ranked for ad-hoc as well as diversity-based retrieval[2]. However, there are several well-performing runs (*i.e.*, highly-ranked runs, which corresponds to the lower left region of the graph) that deviate from the diagonal, which indicates that their rankings change substantially when evaluated using EGU with $\gamma = 0.0$ vs. $\gamma = 1.0$. Two extreme examples have been marked in the graph: NeuDivW75 is a run that performs very well on the ad-hoc

[2] Note that this run was specially crafted using a combination of three popular commercial search engines and does not represent any participating retrieval system.



(a) EGU with $p = 0.1$ and $\gamma = 0.0$ vs. $\gamma = 1.0$

(b) EGU with $\gamma = 1.0$ and $p = 0.1$ vs. $p = 0.5$

(c) EGU with $p = 0.1$ and $\gamma = 1.0$ vs. Prec@10

(d) EGU with $p = 0.1$ and $\gamma = 1.0$ vs. α-NDCG

Figure 4.1: TREC retrieval system rankings with respect to different variations of evaluation metrics. The Kendall's rank correlation coefficients (τ) have also been shown in each graph.

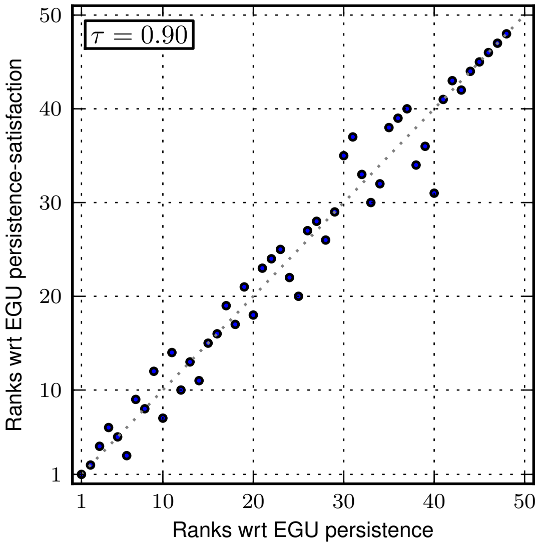


Figure 4.2: TREC retrieval system rankings with respect to the persistence-model vs. the persistence-satisfaction model in EGU.

task and is ranked 5^{th}, but when evaluated using the diversity-based retrieval setting, it ranks 26^{th}. This indicates that NeuDivW75 has substantial repetition in its results, which is penalized by EGU with the $\gamma = 0.0$ setting. On the other hand, THUIR09QeDiv ranks 22^{nd} for the ad-hoc retrieval task, but ranks 8^{th} on the diversity-based retrieval task. This indicates that THUIR09QeDiv is more effective in minimizing repetition, and hence, is particularly suited for diversity-based retrieval.

Effect of user persistence. Next, we observe the effect of user persistence on the system rankings. In Figure 4.1b, we compare the rankings when runs are evaluated using $p = 0.1$ and $p = 0.5$ as the parameter of the geometric distribution over stopping positions (see the persistence model in Section 3.2.1.1). The former case corresponds to persistent users who read the top ten documents on average, whereas the latter case corresponds to impatient users who only read the top two documents on average. The scatterplot indicates that there are substantial changes in rankings for the two scenarios. For instance, ICTNETDivR1 performs relatively well (ranked 12^{th}) when users are persistent (*i.e.*, $p = 0.1$) but slips to rank 25^{th} when evaluated against impatient users (*i.e.*, $p = 0.5$). In other words, ICTNETDivR1 is slightly less likely to place the most useful documents at the first two ranks, as compared to other top-ranked runs.

Correlation of EGU and precision. Next, we would like to see how well EGU correlates with traditional measures like Prec@10. Figure 4.1c shows the scatterplot of rankings with respect to EGU with $\gamma = 1.0$ and Prec@10. Note that neither metric in this case

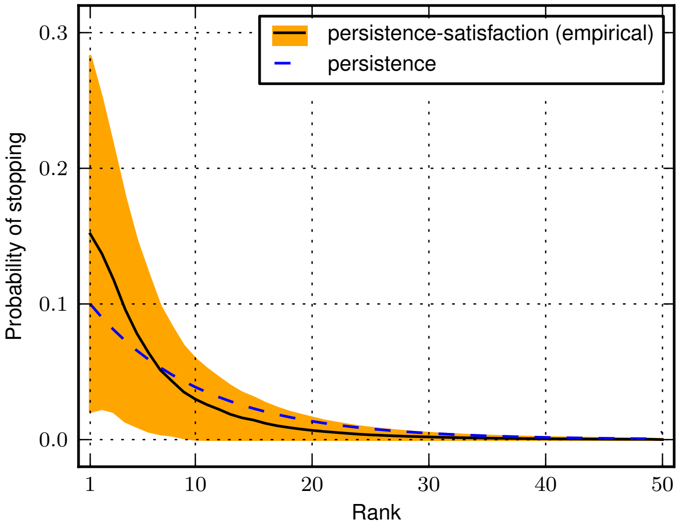


Figure 4.3: The empirical stopping distribution based on the persistence-satisfaction model, generated using the TREC 2009 diversity runs.

penalizes redundancy. Therefore, we see a strong correlation between the rankings of runs with respect to these two metrics, apart from some lower-ranked runs (ranks between 30 and 40).

Correlation of EGU and $\alpha-$NDCG In Figure 4.1d, we have plotted the rankings with respect to EGU with $\gamma = 0.1$, and $\alpha-$NDCG with the corresponding parameter $\alpha = 0.9$. $\alpha-$NDCG [Clarke et al., 2008] is a recently-proposed evaluation measure that has also been used in the diversity retrieval task of the TREC 2009 and 2010 Web Tracks [Clarke et al., 2009]. There is a high correlation in the rankings based on the two measures, which is expected since $\alpha-$NDCG has a mathematical form very similar to EGU at least for the purposes of evaluating a single ranked list. However, this similarity breaks down when evaluating multiple ranked lists in a search session; we analyze this scenario in Chapter 6. Also see Section 3.6.1 for more details on the similarities and differences between the two evaluation measures.

Effect of user browsing model. In Section 3.2.1, we described two main alternatives for the user browsing model: the persistence-based model, and the persistence-satisfaction model. The former model posits that users' likelihood of stopping can be characterized purely in terms of the rank, while the latter model posits that users' likelihood of stopping depends on the rank as well as their level of satisfaction up to the current rank. To understand how the usage of these two models affects the measured performance of systems, in Figure 4.2, we plot the rankings of TREC runs with respect to these two models. Interestingly, the rankings of the systems do not vary significantly under the two user browsing models. Nevertheless, to understand the characteristics of the two models, in Figure 4.3, we have plotted the empirical distribution over stopping positions (according to the persistence-satisfaction model) using the TREC runs. That is, based on the ranked lists in the TREC runs, we used the persistence-satisfaction model to calculate the probability of stopping at each rank, averaged over all (48) runs and all (50) queries. We have plotted the mean probability, flanked on both sides by bands representing one standard deviation. It is evident that under the persistence-satisfaction model, users are likely to stop early in the ranked list, compared to our geometric distribution-based persistence model.

Nevertheless, when only the relative performances of retrieval algorithms is of concern, the simpler persistence-based model can be regarded as a practical approximation for the more complicated persistence-satisfaction model, since both lead to similar rankings of runs (Kendall's $\tau = 0.9$) as seen in Figure 4.2.

4.2 Information Distillation

The second retrieval scenario that is of interest to us is information distillation, which was the focus of DARPA's Global Autonomous Language Exploitation (GALE) project [Hakkani-Tur et al., 2007; White et al., 2008]. It is targeted towards information analysts who need to gather relevant information on events around the world by monitoring various sources of information. One instantiation of information distillation is due to Yang et al. [2007], which involves a pipeline of adaptive filtering, passage retrieval, and novelty detection (see Section 2.2.3 for more details) to support long-lasting information needs over a search session involving multiple rounds of retrieval and user feedback. The adaptive filtering component helps in tracking evolving news events: the "adaptive" element learns from user feedback to adjust the model according to the user's evolving information needs, and the "filtering" element intelligently limits the the number of items shown to the user to reduce the amount of time required to examine the updates—and hence—the overall effort in tracking the news event. To further reduce the burden on the user to find the most pertinent information, the passage retrieval extracts and ranks passages from the selected documents. The novelty component ensures that the ranked list of passages is not redundant with respect to previous passages seen by the user.

There are several aspects of this setup that make it particularly well-suited for evaluation using the proposed framework in this thesis:

1. A retrieval system is allowed to produce any readable response text (in this case—passages) that is consistent with the sources, including paraphrasing and

summarizing source text. Therefore, nuggets are well-suited for evaluating the relevance and novelty of arbitrary spans of text.

2. To evaluate the filtering aspect of the task, the evaluation measure requires a notion of cost so as to reward systems that can intelligently limit the number of documents selected for the user's attention in each time chunk. In other words, if there are no updates in a particular time chunk, the system should produce no output, instead of presenting a ranked list of irrelevant items.

3. This scenario involves long-lasting queries in a search session with multiple rounds of retrieval. Therefore, the performance of the system must be measured over the entire session, instead of unrealistically treating each ranked list independently. This involves accurately measuring novelty across ranked lists, which is possible with EGU, but not well-defined for other measures like $\alpha-$NDCG.

To evaluate this retrieval setting, we first develop a new test collection by extending an existing benchmark dataset in Section 4.3. We then describe our experiments on this dataset using the CMU Adaptive Filtering Engine (CAFÉ) Yang et al. [2007], which is a state of the art information distillation system that combines adaptive filtering, passage retrieval, and novelty detection.

4.3 Extending the TDT4 Dataset

TDT4 was a benchmark corpus used in the *Topic Detection and Tracking* (TDT) studies in 2002 and 2003 [Fiscus and Doddington, 2002]. The corpus consists of about 28,000 news articles in English from various newswire services like AP, NYT, CNN, ABC, NBC, MSNBC, Voice of America, published between October 2000 and January 2001. Due to the presence of multiple news sources and the fact that many news events evolve over time, the TDT4 data collection exhibits considerable redundancy across documents, which makes it an ideal testbed for novelty-based retrieval evaluation.

The *Linguistic Data Consortium* (LDC) annotated the corpus with 100 topics and corresponding relevance judgments at the document level. Based on 12 news events that appear in the time frame of this collection, we created 120 queries that capture various information needs and sub-tasks that would be of interest to an intelligence analyst. For each query, we identified the pieces of information (*i.e.*, nuggets) that satisfy the information need, and then created rules for automatically determining whether a given span of text contains the nugget (Section 4.3.1)[3].

[3] The queries were created with the assistance of Profs. Yiming Yang and Jaime Carbonell. The bootstrapping approach for identifying nuggets and deriving the corresponding "nugget-matching rules", as well as the software necessary for creating and validating these nugget-matching rules were developed

Table 4.2: Sample information needs in the extended TDT dataset.

(Queries related to the Singapore Airlines crash)	
Information Need:	Track developments related to role of explosion or fire, indicating a bomb.
Query:	`singapore airlines sq006 crash explosion fire bomb`
Information Need:	Role of weather.
Query:	`singapore airlines sq006 crash weather`
Information Need:	Track reports of casualties.
Query:	`singapore airlines sq006 crash casualties survivors`
(Queries related to the Texas prison break)	
Information Need:	Track sightings of escapees.
Query:	`texas prison escape confirmed sighting false alarms`
Information Need:	Find known associates of escaped inmates.
Query:	`texas prison escape link associate aide conspire`
Information Need:	Track the reward offered for their capture.
Query:	`texas prison escape reward`

Table 4.3: Some statistics about the dataset and the associated ground truth.

Documents	28,390
Queries	120
Total relevant docs	654
Average nuggets per query	7
Total nuggets	890

Table 4.2 shows two sample news events that appear in the dataset, and the corresponding queries that we created. Some statistics about the dataset and the associated ground truth are shown in Table 4.3.

by the author. The annotation effort was coordinated by the author, and accomplished with the assistance of fellow graduate students Abhay Harpale, Ni Lao, and research programmer Bryan Kisiel.

4.3.1 Nugget-Matching Rules

Since the ground truth is defined in terms of nuggets instead of directly marking documents as relevant or irrelevant to each query, a method is required for determining which nuggets are present in an arbitrary span of text returned by the system. The approach used in sub-topic retrieval or diversity-based retrieval is to pre-determine the nuggets present in each document (or a reasonable subset) of the corpus. However, such a strategy is tedious, and also prevents flexible evaluation at finer granularities, *e.g.*, passages, or other arbitrary spans of text that the system may choose to return, as is the case in information distillation. In other words, an automatic method is required for reliably determining whether a snippet of text contains a given nugget. We propose such a method, known as *nugget-matching rules*, which are generated using a semi-automatic procedure explained below. These rules are essentially Boolean queries that will only match against snippets that contain the nugget. For instance, a candidate rule for matching answers to "How many prisoners escaped?" is `(texas AND seven AND escape AND (convicts OR prisoners))`, possibly with other synonyms and variants in the rule. For a corpus of news articles, which usually follow a typical formal prose, it is fairly easy to write such simple rules to match expected answers using a "bootstrap" approach, as described below.

We propose a two-stage approach that combines the strength of humans in identifying semantically equivalent expressions and the strength of the system in gathering statistical evidence from a human-annotated corpus of documents. In the first stage, human subjects annotated (using a highlighting tool) portions of on-topic documents that contained answers to each nugget. In the second stage, subjects used our rule generation tool to create rules that would match the annotations for each nugget. The tool allows users to enter a Boolean rule as a disjunction of conjunctions (e.g. `((a AND b) OR (a AND c AND d) OR (e))`). Given a candidate rule, our tool uses it as a Boolean query over the entire set of on-topic documents and calculates its recall and precision with respect to the annotations that it is expected to match. Hence, the subjects can start with a simple rule and iteratively refine it until they are satisfied with its recall and precision. We observed that it was very easy for humans to improve the precision of a rule by tweaking its existing conjunctions (adding more `AND`s), and improving the recall by adding more conjunctions to the disjunction (adding more `OR`s).

As an example, let's try to create a rule for the nugget which says that seven prisoners escaped from the Texas prison. We start with a simple rule – `(seven)`. When we input this into the rule generation tool, we realize that this rule matches many spurious occurrences of `seven` (e.g. '...seven states...') and thus gets a low precision score. We can further qualify our rule – `texas AND seven AND convicts`. Next, by looking at the 'missed annotations', we realize that some news articles mentioned "...seven prisoners escaped...". We then replace `convicts` with the disjunction `(convicts`

`OR prisoners)`. We continue tweaking the rule in this manner until we achieve a sufficiently high recall and precision, *i.e.*, the (small number of) misses and false alarms can be safely ignored.

Thus we can create nugget-matching rules that succinctly capture various ways of expressing a nugget, while avoiding matching incorrect (or out of context) responses. Human involvement in the rule creation process ensures high quality generic rules which can then be used to evaluate arbitrary system responses reliably. Section A.2 in the appendix lists some sample nuggets and their corresponding nugget-matching rules.

The full set of information needs, nuggets, and nugget-matching rules are available for download at `http://nyc.lti.cs.cmu.edu/downloads/etdt4.tar.gz`.

4.4 Experiments on the Information Distillation Task

Our goal is to demonstrate the effectiveness of the proposed framework for accurately evaluating and comparing the performance of retrieval systems in terms of multiple factors: relevance, novelty, and cost of reading across multiple ranked lists in a search session. We are also interested in observing the behavior of the proposed measure, `EGU`, as the relevance and novelty parameters of the retrieval systems are varied.

4.4.1 Experimental Setup

We simulate the information distillation task as described in Section 2.2.3: Consider a hypothetical intelligence analyst who needs to track evolving news events around the world by monitoring multiple information sources. To start her search, she uses long-lasting query, which is used by the adaptive filtering component of the system to detect relevant articles. However, instead of presenting them to the analyst as and when they appear and unrealistically expecting her to provide immediate feedback, the retrieval system holds the articles and presents them at configurable time intervals or "chunks". Moreover, the articles are not presented as a set, which is difficult to examine, but in the form of a list of passages ranked according to their relevance and novelty. The analyst examines some of these passages based of her interest, which is recorded as implicit feedback and used by the retrieval system to adjust its user profile.

To simulate such a user-in-the-loop retrieval setting in an offline setup, we divide the 4-month span of the TDT dataset into chunks of 12 consecutive days each. At the end of each such time chunk, the retrieval system is expected to produce a ranked list of documents using the past 12 days of documents as the corpus. The system receives feedback from the user, and then produces a new ranked list for the next chunk, and so on. This setup simulates a user who is following an evolving news event over an extended period of time—expecting the retrieval system to return a personalized ranked list of relevant and novel documents after every 12 days.

Also, for consistency with Yang and Lad [2009], we split the queries into a validation set and a test set comprising 59 and 45 queries, respectively. We use the validation set to tune the parameters of the retrieval system, as described next.

4.4.2 Retrieval Systems

Unlike our experiments on the TREC Web Track where we had access to 48 runs, for this task, we have access to a single system: the CMU Adaptive Filtering Engine (CAFÉ) [Yang et al., 2007], which combines adaptive filtering, passage retrieval, and novelty detection. CAFÉ has two main parameters:

1. Relevance threshold ($t_{rel} \in \{0, 1\}$), which controls the number of passages shown to the user. CAFÉ uses logistic regression to estimate the relevance of each passage. Passages with scores less than t_{rel} are filtered out, *i.e.*, not shown to the user. Hence, higher values of t_{rel} represent stronger filtering, and therefore, fewer passages shown to the user.
2. Novelty threshold ($t_{nov} \in \{0, 1\}$), which controls the level of redundancy in the ranked list. CAFÉ uses cosine similarity between passages to estimate their novelty. Passages that have a cosine similarity greater than $1 - t_{nov}$ with one of previously-selected passages are removed from the ranked list. Hence, higher values of t_{nov} represent stricter novelty detection, *i.e.*, only the most novel passages are retained.

Varying these two parameters and tuning them with respect to the different evaluation measures of interest gives us insight into the behavior of EGU with respect to other commonly-used measures.

4.4.3 Evaluation Measures

We report three evaluation measures, EGU, $\alpha-$NDCG, and MAP. In EGU, we use $\gamma = 0.1$, which corresponds to low tolerance for redundancy. We use $p = 0.1$, *i.e.*, users read the top ten documents in each ranked list, on average. Since we are interested in filtering performance, we use a non-zero cost in EGU, *i.e.*, a per-passage cost of $c = 0.01$ (Section 3.1.2). This penalizes systems that present long ranked lists containing irrelevant information. To penalize redundancy, we use $\gamma = 0.1$ in EGU, and $\alpha = 0.9$, which is the equivalent redundancy parameter setting for $\alpha-$NDCG.

Table 4.4: Performance of CAFÉ using three evaluation measures. CAFÉ [EGU] means CAFÉ's parameters were tuned for target measure EGU, and so on.

System [Target]	Evaluation measures			Optimal Parameters	
	EGU	$\alpha-$NDCG	MAP	t_{rel}	t_{nov}
CAFÉ [EGU]	**0.4701**	0.2163	0.3087	0.65	0.25
CAFÉ [$\alpha-$NDCG]	0.1946	**0.3847**	0.4367	0.00	0.40
CAFÉ [MAP]	0.1632	0.3292	**0.5019**	0.00	0.00

Note that $\alpha-$NDCG is not designed for evaluating search sessions comprising multiple ranked lists. Specifically, how to determine the novelty or redundancy of documents across ranked lists is not well-defined for $\alpha-$NDCG. Therefore, we make the reasonable approximation that the top ten documents in each ranked list will be read by all users, and hence, will be used to update the nugget counts for evaluation of subsequent ranked lists. In EGU, no such assumption is required, since reading behavior is assumed to be a probabilistic process that seamlessly extends to multiple ranked lists.

4.4.4 Results and Analysis

We conducted the following experiment. We have three evaluation measures: EGU, $\alpha-$NDCG, and MAP, and one retrieval system, CAFÉ, which admits two configurable parameters, relevance threshold (t_{rel}) and novelty threshold (t_{nov}). We used the validation set of queries to tune these parameters (through parameter sweep) in turn for each of the three evaluation measures, and then observed the performance in each case using all three evaluation measures. This leads to a 3×3 grid as shown in Table 4.4. Each row represents one configuration of CAFÉ tuned for a particular target metric, *e.g.*, CAFÉ [EGU] denotes CAFÉ tuned for EGU, and so on. In each row, the optimal values of the two system parameters are also shown in the two right-most columns. For each of the three evaluation measures, the highest value in the column appears in bold.

The bold entries on the diagonal indicate that the highest value for each evaluation measure is obtained when the same measure is used to optimize the system. As a corollary, optimizing for one measure does not at all optimize for other measures, which shows that each of the three measures have very different characteristics.

Now let us look at how optimizing for each performance measures affects the behavior of the retrieval system in terms of its relevance and novelty-based filtering. The optimal parameter values when targeting EGU (first row in the table) are: $t_{rel} = 0.65$ and $t_{nov} = 0.25$, *i.e.*, a relatively strict relevance-based filtering and moderate novelty-based filtering. On the other hand, when optimizing for $\alpha-$NDCG (second row) as well as MAP (third row), the optimal relevance threshold is 0.00. This is expected since neither $\alpha-$NDCG nor MAP have a notion of cost, *i.e.*, they both favor the longest possible ranked list. In other words, the $\alpha-$NDCG or MAP score of a ranked list can never improve by limiting the number of items in the ranked list, making them unsuitable for evaluation of filtering scenarios. However, $\alpha-$NDCG does penalize redundancy, as evident from the optimal novelty threshold of 0.40, while MAP is insensitive to novelty and redundancy, as evident from its optimal novelty threshold of 0.00, *i.e.*, no removal of redundancy content from the ranked list.

These differences in the characteristics of the three measures also explain the correlation (or lack thereof) of the scores in the three rows. EGU and MAP scores are completely opposite of each other: the highest EGU score of 0.4701 corresponds to the lowest MAP score of 0.3087 in the first row, and vice versa in the third row. EGU favors relevance as well as novelty-based filtering, while MAP favors none. On the other hand, $\alpha-$NDCG favors novelty-based filtering, but not relevance-based filtering (*i.e.*, limiting the length of the ranked list), which leads to worse apparent performance when both are enabled in the first row.

4.5 Related Work

Automatic Evaluation based on Answer Keys. Some methods have been proposed for automatic evaluation of question answering systems based on the idea of n-gram co-occurrences. Pourpre [Lin and Demner-Fushman, 2005] assigns a fractional recall score to a system response based on its unigram overlap with a given nugget's description. For example, a system response `A B C' has recall 3/4 with respect to a nugget with description `A B C D'. However, such an approach is unfair to systems that present the same information but using words other than A, B, C, and D. Another open issue is how to weight individual words in measuring the closeness of a match. For example, consider the question "How many prisoners escaped?". In the nugget 'Seven prisoners escaped from a Texas prison', there is no indication that 'seven' is the keyword, and that it must be matched to get any relevance credit. Using IDF values does not help, since 'seven' will generally not have a higher IDF than words like 'texas' and 'prison'. Also, redefining the nugget as just 'seven' does not solve the problem since now it might spuriously match any mention of 'seven' out of context. Nuggeteer [Marton and Radul, 2006] works on similar principles but makes binary decisions about whether a nugget is present in a given system response by tuning a threshold. However, it is also plagued by 'spurious relevance' since not all words contained in the nugget description (or known correct responses) are *central* to the nugget.

4.6 Summary

In this chapter, our goal was to understand the behavior of the proposed evaluation measure on real retrieval systems and datasets. We used past TREC runs to analyze the behavior of EGU under various settings including different tolerances towards redundancy and different user browsing models. We analyzed the correlation between EGU and other common performance measures like precision and $\alpha-$NDCG. We also compared the relative rankings of TREC runs under the two user browsing models proposed in Section 3.2.1, and found that the simpler persistence model can be used in place of the more complicated persistence-satisfaction model as long as only the rankings of retrieval systems is of concern.

We also focused on the task of information distillation, which showcases the full power of the proposed framework. To support experiments on this retrieval task, we developed a new testbed by extending the TDT4 dataset. We used a semi-automatic approach for creating "nugget-matching rules" that enable automatic evaluation of retrieval systems that return arbitrary spans of text. We conducted experiments using the CMU Adaptive Filtering Engine (CAFÉ) by tuning its parameters for different target evaluation measures, which highlighted the main differences in the properties of these measures with respect to EGU: While EGU is sensitive to both reading costs as well as redundancy, $\alpha-$NDCG is only sensitive to redundancy but not reading costs, whereas MAP is insensitive to reading costs as well as redundancy.

Chapter 5

Optimization – Computational Challenges

In Chapter 3, we focused on the problem of evaluating a retrieval system in terms of its expected utility. In this chapter, we focus on optimization, *i.e.*, how to rank a set of documents so as to maximize the expected utility as measured by `EGU`. The optimization problem can be divided into two sub-problems, based on whether the relevant nuggets that need to be covered are known:

1. **Nuggets are known.** This scenario arises when we wish to compute the best possible `EGU` score for a query based on the ground truth, *i.e.*, the nuggets that are relevant to the given query. As discussed in Section 3.5.1, the best possible score is used for normalizing the system score to make it comparable across different queries.

2. **Nuggets are unknown.** This scenario corresponds to the problem faced by the retrieval system, since it obviously does not have access to the ground truth. Instead, the system must depend on observable query and document features (*e.g.*, words and named entities) as surrogates for the true nuggets, and learn their relevance to the user's information need based on user feedback.

In this chapter, we focus on the case where the nuggets are known, and examine the computational issues associated with producing the optimal ranked list with respect to `EGU`. Like other metrics that are based on coverage of discrete elements (nuggets in our case), finding the optimal ranked list with respect to `EGU` turns out to be an NP-hard problem (Section 5.1.1). This fact is already known in other contexts like summarization [Lin and Bilmes, 2010] and diversity-based ranking [Agrawal et al., 2009]. However, all these settings are *set-based*, *i.e.*, they study the problem of finding a *set* of documents. The ranking of these documents is not of concern. In this thesis, we are interested in the *ranking* or ordering of the documents, which is an important function of many retrieval systems like search engines. Our main contribution in this chapter is the analysis of the computational challenges for the ranking problem, where the utility obtained by the user depends on his or her browsing behavior, which is captured using a probabilistic model in our framework.

In any case, the NP-hardness of the optimization problem forces us to use approximate algorithms for ranking. We show that EGU is a sub-modular function, which allows a simple greedy algorithm to achieve good performance (Section 5.2). Based on the sub-modularity property, we derive a lower-bound on the EGU score obtainable by the greedy algorithm (Section 5.4). We also conduct experiments on synthetic data to understand the behavior of the greedy algorithm under various conditions, *e.g.*, the user's tolerance towards redundancy, his or her persistence, *i.e.*, stopping probability (Section 5.5), as well as the presence of per-document reading costs. The optimization of performance across multiple ranked lists requires a subtle modification to the greedy algorithm due to the probabilistic nature of user's stopping position in each ranked list, which we describe in Section 5.6.

5.1 Computational Hardness

Unlike relevance-based ranking, where the best ranked list is obtained by ranking all relevant documents before irrelevant ones, finding the optimal ranked list with respect to EGU turns out to be a non-trivial problem. Before we formalize the optimization problem, let us begin with a simple example to illustrate the computational issues involved with optimizing EGU. Assume that our corpus consists of the following documents, where lower-case letters represent the nuggets present in these documents:

d_1: [a, b, c, d, e]
d_2: [a, b, f, g]
d_3: [c, d, h, i]

If we are required to select only one document for the user's attention, we can simply return document d_1, which contains five nuggets. If we are required to select two documents for the user, then we should return documents d_2 and d_3, which together contain eight nuggets. Evidently, the optimal ranked list of length two is not a subset of the optimal ranked list of length one. Moreover, a simple greedy approach of ranking documents by decreasing number of (new) nuggets does not necessarily produce the optimal ranked list whenever more than one document is to be presented.

Therefore, for novelty or diversity-based ranking, the optimal order of documents depends on the stopping position of the user. However, the system does not know in advance how many documents will be read by the user. Our evaluation framework solves this problem by imposing a probability distribution over the stopping positions of a population of users. Based on such a distribution, an optimal ranked list can be uniquely defined. Loosely speaking, if most users stop at the first rank, then the optimal ranked will place document d_1 at the first rank at the expense of the small number of users who stop at rank 2. Similarly, if most users stop at the second rank, then the optimal rank list will favor d_2 and d_3 instead of document d_1. In other words, the optimal placement of the documents will inevitably involve a trade-off between satisfying users who stop at different positions. This is unlike purely relevance-based ranking, where simply ranking documents by decreasing relevance levels maximizes the recall and precision at all stopping positions in the ranked list.

Now that we have anecdotally seen that a simple greedy ranking approach will not necessarily maximize EGU, let us formalize the computational hardness of the optimization problem by showing that maximizing EGU is indeed NP-hard.

5.1.1 NP-Hardness

To prove that optimizing EGU is NP-hard, we show that optimizing for a special case of EGU is NP-hard, so that the NP-hardness of optimizing EGU in general follows automatically.

Let us focus on a particular parameterization of $\gamma = 0$ *i.e.*, no tolerance towards redundancy, and $P(s = k) = 1$, *i.e.*, the user reads all documents from the top down and stops at a given rank, say k. Given a set of documents, all nuggets that appear in at least one of these documents are said to be *covered* by the set of documents. Then, finding a set of k documents that cover the most number of nuggets is exactly equivalent to the *Maximum Coverage Problem*, which is known to be NP-hard [Church and ReVelle, 1974].

Maximum Coverage Problem (MAX-COVER): Given a collection of sets $S = S_1, S_2, ..., S_m$, each containing a subset of elements, *i.e.*, $S_i \subseteq \{e_1, e_2, ..., e_n\}$, find the subset $S^* \subseteq S$ of size K such that the number of covered elements is maximized:

$$\begin{aligned} \arg\max_{S^* \subseteq S} & \left| \bigcup_{S_i \in S*} S_i \right| \\ \text{s.t.} \quad & |S^*| = k \end{aligned} \tag{5.1}$$

Our ranking problem can be reduced to MAX-COVER by mapping documents to sets and nuggets to elements.

5.1.2 Non-optimality at Each Rank

The second problem is that the optimal ranked list at any rank k is not necessarily a subset of the optimal ranked list at rank $k + 1$. In other words, it is theoretically impossible for a system to create a ranked list that is optimal (in the MAX-COVER sense) at all ranks. This follows from the NP-hardness of the problem: Otherwise a greedy algorithm that chooses documents by decreasing utility would lead to a globally optimal solution in polynomial time.

This non-optimality at each rank is unlike traditional relevance based ranking, where the independence of document relevance ensures the optimality of the *Probability Ranking Principle* [Robertson, 1977]: Ranking by decreasing probabilities of relevance maximizes the precision and recall at all ranks in the ranked list. Gollapudi and Sharma [2009] call this as lack of stability of diversity-based ranking criteria. Since most existing criteria [Zhai et al., 2003; Agrawal et al., 2009] are set-based, they do not directly address this problem, forcing the system designer to target a single rank for which to tune the system.

Therefore, unless a particular stopping rank is specified, the optimization criterion of MAX-COVER is not meaningful for our ranking problem. The definition of EGU naturally gets around this problem by taking an expectation over all ranks through the probability distribution over stopping positions. That is, our optimization problem may be viewed as an *expected* MAX-COVER problem, where the expectation is taken over the size of the set (k in Eq. 5.1). However, such a formulation of a covering problem has not been studied in literature, to the best of our knowledge.

5.2 Submodularity of the Utility Function

Our objective function admits additional structure that allows approximation algorithms to guarantee good performance. Specifically, the utility function used in EGU is *submodular*. Submodularity formalizes the intuitive property of diminishing returns, and is defined as follows:

Definition 5.1 ***Submodularity* *[Nemhauser et al., 1978].*** *A set function F is called submodular if and only if for all $A \subseteq B \subseteq V$ and $s \in V \setminus B$ it holds that $F(A \cup \{s\}) - F(A) \geq F(B \cup \{s\}) - F(B)$.*

By definition, utility is a linear combination of gain and cost (cf. Eq. 3.8). We show that gain is a submodular function, and cost is a supermodular function (*i.e.*, its negation is submodular). Then, it follows that utility is submodular since the difference between a submodular and supermodular function is submodular [Wolsey, 1982].

5.2.1 Submodularity of the Gain Function

The submodularity of the gain function follows from its concavity with respect to nugget counts, as shown in Figure 3.1. Intuitively, the increase in gain obtained by adding a document to a longer ranked list can never be larger than the increase obtained by adding the document to a subset of the ranked list.

The submodularity of the gain function can be shown mathematically as follows. Assume that L is a ranked list of documents, and L' is another ranked list such that $L \subset L'$. Let d be an arbitrary document such that $d \not\subset L'$. Let the nugget counts in the two ranked lists be denoted by $\eta(L)$ and $\eta(L')$, respectively. Thus, $\eta_\delta(L)$ denotes the number of times nugget δ appears in the documents of ranked list L, and so on. The marginal gain of adding d to the two ranked lists is equal to:

$$g(d|L) = \sum_{\delta \in d} \gamma^{\eta_\delta(L)} \tag{5.2}$$

$$g(d|L') = \sum_{\delta \in d} \gamma^{\eta_\delta(L')} \tag{5.3}$$

where $\delta \in d$ indexes all nuggets in document d. Now, we only need to show that $g(d|L) \geq g(d|L')$. Note that the number of times a nugget appears in the ranked list L can never be greater than the number of times the nugget appears in L', which is a superset of L. That is, $\eta_\delta(L) \leq \eta_\delta(L')$ for each δ. Therefore, it follows that $\gamma^{\eta_\delta(L)} \geq \gamma^{\eta_\delta(L')}$ since $\gamma \in [0, 1]$. This shows that $g(d|L) \geq g(d|L')$.

5.2.2 Supermodularity of the Cost Function

A function F is called supermodular if its negation, $-F$, is submodular. This corresponds to increasing returns, as opposed to diminishing returns in case of submodular functions. Intuitively, supermodularity means that the cost function should be superlinear, *i.e.*, the cost of adding another document to a ranked list L should be less than or equal to adding the same document a ranked list that is a superset of L. It is reasonable to imagine such a cost function: A user might get increasingly tired after reading each document in a ranked list, with the additional frustration increasing with every document.

In this thesis, we have limited our attention to simple cost functions that are defined per document (cf. Chapter 3). Such linear definitions of cost already satisfy the supermodularity requirement, since a linear function is submodular as well as supermodular.[1]

[1] Such a function is known as a modular function.

Algorithm 1 Greedy algorithm (Greedy)

```
/* **Input:** Documents to rank $D = \{d_1, d_2, ..., d_k\}$ */
/* **Output:** Ranked list $L$ */
$L \leftarrow \emptyset$
for $i = 1$ to $k$ do
    $j \leftarrow$ Choose-Next-Doc$(L, D)$
    $L \leftarrow L \cup d_j$
    $D \leftarrow D \setminus d_j$
end for

sub Choose-Next-Doc$(L, D)$ do
    /* **Input:** Current ranked list $L$, remaining documents $D$ */
    /* **Output:** Index of next document from $D$ to include in the ranked list */
    for $i = 1$ to $|D|$ do
        $s_0(i) =$ Marginal-Utility$(d_i, L)$
    end for
    return $\arg\max_i s_0(i)$
end sub
```

In the next section, we show that submodularity of the utility function ensures that a simple greedy algorithm achieves good performance with a constant approximation ratio.

5.3 Greedy Algorithm

The basic idea of the greedy algorithm for Max-Cover is to build the ranked list by iteratively selecting the document that contains the most number of previously uncovered nuggets. Since EGU uses a gain function with a diminishing returns property, we need to modify the greedy algorithm to select the next document with the highest marginal gain as defined in Eq. (3.2). Algorithm 1 lists the steps involved in the greedy algorithm.

We already showed that our objective function is submodular. A classic result shows that the greedy algorithm guarantees a constant approximation ratio:

Theorem 5.1 *[Nemhauser et al., 1978]. For any monotonic submodular function, the greedy algorithm achieves an approximation ratio of* $(1 - 1/e)$.

Moreover, no polynomial time algorithm can achieve a better approximation ratio unless P=NP. [Feige, 1998].

Note that when non-zero costs are involved, the utility function is no longer monotonic. However, in this case, the utility function resembles the uncapacitated facility location problem, for which Cornuejols et al. [1977a] derived the $1 - 1/e$ bound for the greedy algorithm, assuming the "shifted" approximation function, which we have also used in this thesis (see Section 3.5).

5.4 Improved Lower-bound

We develop a tighter bound on the performance of the greedy algorithm, as compared to the original bound of $1-1/e$, which is approximately 0.63. The main idea for deriving the new bound is that the lower bound of $1 - 1/e$ is too conservative: It is guaranteed irrespective of the size k of the covering problem. Cornuejols et al. [1977a] derived a tighter lower bound as a function of k:

$$\frac{g_k}{I_k} \geq 1 - (1 - 1/k)^k \tag{5.4}$$

where g_k is the greedy solution and I_k is the ideal solution. The bound of $1 - 1/e$ arises because:

$$(1 - 1/k)^k < 1/e \tag{5.5}$$

which approaches equality for large k. However, for smaller values of k, the gap is large. For instance, for $k = 1$, the expression $(1 - 1/k)^k$ is equal to zero, which corresponds to the fact that the solution of size 1 is always optimal. In other words, approximation factors better than 0.63 can be guaranteed for covering problems of smaller size. Since the total gain, say G, is calculated by taking an expectation over all stopping positions, *i.e.*, $k = 1, 2, ...$, we can therefore derive a tighter bound by taking the size-dependent bound into account. Specifically, the total gain of the ideal solution, say I, is equal to:

$$I = \sum \Pr(k) I_k \tag{5.6}$$

But due to Eq. (5.4), we have:

$$I_k \leq \frac{g_k}{1 - (1 - 1/k)^k} \tag{5.7}$$

Therefore, the desired bound is:

$$\frac{G}{I} \geq \frac{\sum \Pr(k) g_k}{\sum \left(\frac{\Pr(k) g_k}{(1-(1-1/k)^k)} \right)} \geq 1 - 1/e \tag{5.8}$$

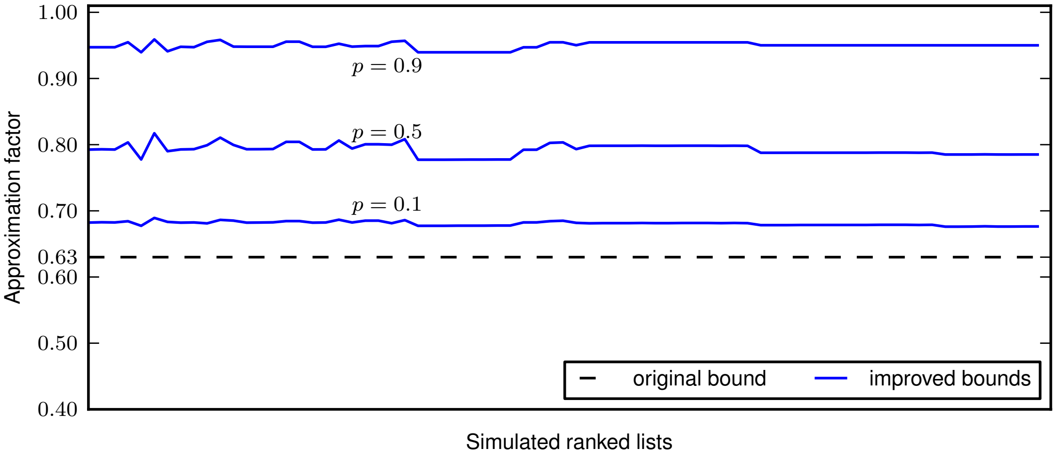


Figure 5.1: Comparison of the original $(1 - 1/e)$ and the improved bounds for various values of stopping probability p.

5.4.1 Experiments on synthetic data.

We compared this improved bound against the original bound of $(1 - 1/e)$ by running the greedy algorithm on 1000 synthetic ranked lists that were generated randomly. For each ranked list, we computed the following quantities: (i) the true approximation factor, *i.e.*, the ratio of the greedy algorithm's score and that of the ideal ranked list, which was obtained using exhaustive search, (ii) the improved lower bound as described above, and (iii) the original lower bound, which is simply equal to $1 - 1/e$, and hence, constant for all ranked lists.

Figure 5.1 shows the bounds obtained for various values of the stopping probability p. For ease of presentation, the ranked lists are arranged by decreasing values of the true approximation factor. Note that the improved bound is dependent on the greedy scores as well as the stopping probability, which is evident in Eq. (5.8). Also, the bound gets closer to 1.0 with increasing values of p, which is expected behavior because higher values of p correspond to higher likelihood of the user to stop at one of the top ranks, where the greedy algorithm guarantees better worst-case performance.

5.5 Performance of the Greedy Algorithm under Various Conditions

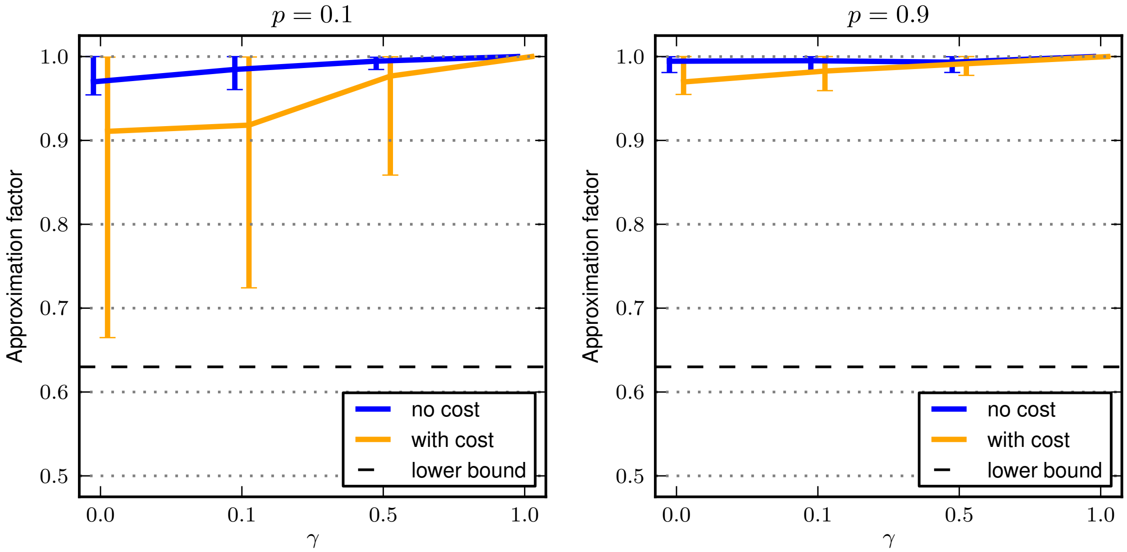
The greedy algorithm and its performance guarantees outlined in the previous section apply to the standard MAX-COVER problem. However, our problem of optimizing EGU is more general than MAX-COVER, since EGU incorporates diminishing returns from nuggets, and calculates the expectation over different stopping position. Moreover, our definition of utility, unlike MAX-COVER, also allows a per-document cost. Therefore, it would be interesting to study how the performance of the greedy algorithm is affected by parameters of the EGU metric that control the rate of diminishing returns (γ), the probability of stopping at different positions (p), and the presence or absence of per-document costs.

Tolerance for Redundancy (γ): The user's tolerance towards redundancy is parameterized by γ. The extreme case of $\gamma = 0$ corresponds to no tolerance towards redundancy, *i.e.*, each nugget is only counted once even if it is covered multiple times. This corresponds to the standard definition of MAX-COVER, where every element (nugget) is counted only once. In this case, the greedy algorithm achieves sub-optimal performance due to the NP-hardness of the MAX-COVER problem. At the other extreme, $\gamma = 1$ corresponds to complete tolerance for redundancy, and the system is credited for each presentation of a nugget. This corresponds to the relevance-based retrieval where the utility of each document is counted independently of others. In this case, the greedy algorithm is optimal. For intermediate value of γ, the greedy algorithm is sub-optimal.

Different Stopping Behaviors (p): The stopping behavior of the user also affects the optimality of the greedy algorithm. In the extreme case, if $P(s = 1) = 1$, *i.e.*, all users stop at the first position, the greedy algorithm is optimal: For $k = 1$, the objective function is maximized by choosing a document with the most number of nuggets, breaking ties arbitrarily. It is only when users read more than one document that the short-sightedness of the greedy algorithm can lead to a sub-optimal solution.

Per-document Costs: If a non-zero reading effort is associated with each document, then the greedy algorithm must optimize the ranked list by balancing the relevance of documents (measured in terms of gain) against the total reading effort (measured in terms of cost). We would like to compare two scenarios: (i) No reading effort, which corresponds to how retrieval performance is generally evaluated in literature, including web search, and (ii) Per-document reading effort, where a non-zero reading cost is associated with each document.

Let us observe these scenarios empirically by running the greedy algorithm on synthetic ranked lists, and comparing its performance against the best possible score obtained through exhaustive search.



(a) More reading persistence, $p = 0.1$

(b) Less reading persistence, $p = 0.9$

Figure 5.2: Approximation factors achieved by the greedy algorithm for different parameter settings (p, γ, and reading costs) of EGU on synthetic data. The "no cost" and "with cost" lines have been laterally shifted to avoid overlap.

5.5.1 Experiments on synthetic data.

We used 3000 randomly generated ranked lists to assess the effect of γ, p, and reading effort on the performance of the greedy algorithm. We experimented with four values of γ: 0.0, 0.1, 0.5, and 1.0, which represent increasing tolerances towards redundancy. We also compared two values of p: 0.1, and 0.9, which represent high and low browsing persistence, respectively. To understand the effect of reading effort, a randomly-generated reading cost between 0.0 and 2.0 was assigned to each document. We compared this "with cost" setting with the default setting of "no cost", where a reading cost of 0.0 was assigned to each document.

Figure 5.2a shows the performance obtained by the greedy algorithm for different values of γ, when $p = 0.1$, which corresponds to higher reading persistence, *i.e.*, the user prefers to read 10 documents on average. Similarly, Figure 5.2b shows performance for $p = 0.9$, which corresponds to lower reading persistence, *i.e.*, the user prefers to stop within the first two documents. We have plotted the mean (flanked by minimum and maximum) approximation factors, *i.e.*, the score of the greedy algorithm divided by the best possible score. Only those ranked lists where the greedy algorithm led to sub-optimal performance were included.

Notice that for a given value of p, the approximation factor improves (*i.e.*, gets close to 1) as the value of γ increases, which supports our analysis that the optimization problem is harder for smaller values of γ. Also, the approximation factors are better for $p = 0.9$, *i.e.*, when the user prefers to stop early in the ranked list. Again, this agrees with our analysis that the greedy algorithm performs better for smaller ranked list lengths. In both the graphs, the presence of per-document cost decreases the performance of the greedy algorithm, indicating that the optimization problem becomes slightly more difficult when relevance gains and reading costs need to be balanced to produce the optimal ranked list.

5.6 Optimization of Multiple Ranked lists

The greedy algorithm for producing a single ranked list is a variant of the following rule: Choose the next document that contains the most number of previously unseen nuggets. However, this rule cannot be used for optimizing performance across multiple ranked lists, since our probabilistic model of user browsing behavior assumes that the user does not read all documents in the previous ranked lists. For example, if a nugget was presented at a very low rank in a previous ranked list, it is less likely to be read by the user than a nugget that appeared in a top-ranked document in a previous list. Therefore, the contributions of these two nuggets should be treated differently when computing the marginal gain of documents for the purpose of producing a new ranked list.

In other words, instead of considering the integer counts of how many times each nugget has been presented to the user in previous ranked lists without regard to their ranks, the greedy algorithm should consider the fractional counts that represent the expected number of times each nugget would be seen by the user, taking into account the rank positions at which they appeared and the user's probabilistic model of browsing (*i.e.*, likelihood of stopping at various rank positions). We saw how to compute this quantity in Section 3.3. The marginal utility can then be calculated accordingly using the fractional counts instead of integer counts of nuggets from previous ranked lists.

Thus, the definition of marginal utility in the greedy ranking algorithm (Algorithm 1) is modified as follows:

$$\textsc{Marginal-Utility}(d_i) = \sum_{\delta \in \Delta_q} I(\delta, d_i)\mathbf{w}(\delta, q)\gamma^{\mathbb{E}\left[\eta_\delta(L' \cup d_{1:(i-1)})\right]} \tag{5.9}$$

where $\mathbb{E}\left[\eta_\delta(L' \cup d_{1:(i-1)})\right]$ represents the expected number of times nugget δ would be seen by the user in all documents present in all previously displayed ranked lists L' and all documents ranked higher in the current ranked list.

The expected nugget counts can be calculated efficiently as shown in Eq. 3.21. Therefore, the computational complexity of the greedy algorithm increases linearly with respect to the number of ranked lists in a search session.

5.7 Related Work

Computational complexity of finding optimal ranking. The NP-hardness of the problem of finding the optimal ranked list has been already known in the context of diversity-based ranking [Zhai et al., 2003; Agrawal et al., 2009]. Carterette [2009] provides an analysis of the NP-hardness of optimizing three metrics for diversity-based ranking: $\mathcal{S}$*-recall*, $\mathcal{S}$*-precision*, and α-NDCG. However, as we have seen in Chapter 2, these metrics have certain limitations that make them unsuitable for evaluating diversity-based retrieval systems. Moreover, EGU is more complicated due to the probability distribution over all stopping positions, which requires new analysis of the performance of greedy algorithms.

Sub-modular optimization. The sub-modular nature of certain diversity-based measures is already known [Agrawal et al., 2009]. However, our metric is more sophisticated due to the explicit inclusion of tolerance towards redundancy as well as a probabilistic model of user's browsing behavior, which requires the development new lower-bounds on the performance of greedy algorithms (as we have done in Section 5.4). Sub-modular functions also appear in many other domains including outbreak detection in networks [Leskovec et al., 2007], sensor placement [Krause, 2008], text summarization [Lin and Bilmes, 2010; Lin et al., 2010], and selecting the most informative subset of variables in graphical models [Krause and Guestrin, 2005].

5.8 Summary

In this chapter, we focused on the computational challenges associated with finding the optimal ranked list assuming perfect knowledge of the nuggets. This turns out to be an NP-hard problem, but we showed that the sub-modularity of EGU allows a simple greedy algorithm to guarantee good performance. We also developed a tighter lower-bound that takes into account the fact that EGU computes an expectation over multiple ranks.

Next, we empirically observed the performance of the greedy ranking algorithm on synthetic ranked lists under various conditions, *i.e.*, different tolerances towards redundancy, different levels of user persistence, and also compared the performance when each document is subject to a non-zero reading effort. Our experiments show that the greedy algorithm displays very good performance under a wide range of retrieval scenarios and therefore can be effectively used to optimize ranked lists with respect to the proposed evaluation measure, EGU. This greedy algorithm also forms the basis of our novelty-based ranking approach that we describe next in Chapter 6.

Chapter 6

Retrieval System Optimization

In the previous chapter, we discussed the problems associated with finding the optimal ranked list assuming perfect knowledge of the ground truth in the form of nuggets that are relevant to the given query. However, a real system does have knowledge of the relevant nuggets for an arbitrary query submitted by the user. Instead, the system must depend on observable query and document features (*e.g.*, words) to estimate the usefulness of each document, and then rank them accordingly to produce a ranked list.

In this chapter, we explore the effectiveness of various features, *e.g.*, words, named entities, and latent topics as surrogates for the actual nuggets. In certain scenarios, the retrieval system has access to explicit or implicit user feedback, *e.g.*, clicks on documents of interest. We propose an approach for learning from such feedback to improve the performance of the system in session-based search. Users provide feedback while taking into account other documents that they have already seen. Therefore, our approach interprets user feedback as indicative of the *marginal* utility of the document, instead of an absolute indicator of the relevance of the document (Section 6.2.4).

We conduct experiments on two datasets that have very different characteristics: The first dataset is a collection of news articles, while the second dataset is a collection of webpages. These two datasets help us in understanding the behavior of the proposed evaluation as well as optimization techniques on two different retrieval scenarios: the first involves long-lasting queries, and the second scenario involves ad hoc web queries.

Next, we describe our proposed novelty-based retrieval approach by first motivating the differences between two alternatives for novelty-based ranking in Section 6.1, followed by the technical details of the proposed approach in Section 6.2.

6.1 Direct vs. Indirect Novelty-based Retrieval

We have the following alternatives for designing a novelty-based retrieval system:

- An **indirect approach**, *i.e.*, maximize the novelty by avoiding redundancy between documents, which can be measured using cosine similarity, word overlap, KL-divergence, etc. Several such measures were investigated by Zhang et al. [2002] and Allan et al. [2003]. Indirect methods measure relevance and novelty separately, which must be combined in some way to rank documents. We have two choices:
    - **Linear combination** of relevance and novelty scores using a trade-off parameter. This corresponds to the MMR strategy proposed by Carbonell and Goldstein [1998], and adapted for aspect retrieval by Zhai et al. [2003].
    - A **filtering approach**, where documents are ranked by decreasing relevance scores, and the ones with novelty scores below a threshold are removed from the ranked list. Such an approach was used by Zhang et al. [2002] to perform novelty detection in an adaptive filtering setup. We also used this approach when designing a retrieval system that combines adaptive filtering, passage retrieval, and novelty detection [Yang et al., 2007; Yang and Lad, 2009].
- A **direct approach**, where the retrieval system hypothesizes nuggets in the documents in response to a query, and then ranks the documents so as to directly maximize the coverage of these nuggets. While such approaches have been applied to the task of diversity-based retrieval by several researchers [Zhai, 2002; Agrawal et al., 2009], our work [Lad and Yang, 2010] represents the first attempt to apply such approaches to the task of novelty detection across multiple ranked lists (*i.e.*, session-based retrieval), and evaluate their performance systematically under different tolerances for redundancy.

Our goal is to maximize the proposed measure, EGU, which is defined in terms of nuggets. However, the indirect methods, by definition, have no notion of nuggets, which also makes it difficult to optimize indirect methods for different tolerances towards redundancy (as defined in EGU in terms of repetitions of nuggets).

Both linear combination and the two-stage approach for combining scores also have certain problems: Linear combination, as mentioned in Chapter 2, treats relevance and novelty as compensatory criteria, and therefore may favor highly novel but irrelevant documents, which users are unlikely to find useful for their information need. On the other hand, the filtering approach fixes the order of the documents in the first step based on relevance scores. The second step merely removes the redundant documents, instead of moving them down in the ranked list. This might have a negative impact on the performance in the case when no other better documents are available to take their place.

Moreover, both the representative indirect approaches, `MMR` and redundancy filtering, model novelty in terms of pairs of documents. That is, they do not support the notion that a document can be deemed redundant due to two or more previous documents.

To avoid these problems, we propose a "direct method" for the task of relevance and novelty-based ranking. Since the retrieval system does not have knowledge of the nuggets, we propose an approach where the system uses observable query and document features as surrogates for the true nuggets, and then use the greedy algorithm described in Section 5.3 to produce a ranked list. Since the greedy algorithm ranks documents by decreasing marginal utilities, which in turn depends on the user's tolerance for redundancy (*i.e.*, γ in Eq. 3.2), such an approach can directly optimize the ranked list for different tolerances, unlike the indirect methods mentioned above. Moreover, we would also like to take advantage of user feedback, when available, to incrementally learn the user's preferences for nuggets. We explain our strategy in the following section.

6.2 Ranking Novel Documents based on Explicit Modeling of Nuggets

Given a query, our retrieval approach involves the following steps for maximizing the utility of the system:

1. Given a query, obtain a candidate set of documents using a standard retrieval approach.
2. Identify and assign weights to all nuggets that appear in the candidate set.
3. Re-rank the documents so as to maximize the weighted coverage of the nuggets.
4. Optionally, update weights based on user feedback. Repeat steps 3 and 4.

Let us look at each of these steps in detail.

6.2.1 Obtaining a Candidate Set of Documents

This step is accomplished using an off-the-shelf retrieval system, and serves to limit the number of documents that need to considered for novelty-based ranking. In a typical deployed system, this step would correspond to the default relevance-based retrieval ranker.

6.2.2 Identifying and Assigning Weights to Nuggets

The candidate set of documents is then used to extract nuggets. However, since the system does not have knowledge of the actual nuggets that are relevant to each query, it must use observable features as surrogates for the actual nuggets. We explore the following features:

- **Words:** The simplest and most straightforward approach is to use words as surrogates for nuggets. Then, our goal is to re-rank the initial ranked list so as to cover as many different words as possible, subject to an appropriate weighing scheme. This is similar in spirit to the diversification approached explored by Yue and Joachims [2008].

- **Named Entities:** Named entities are phrases that contain names of persons, organizations, locations, times, and quantities. They can be treated as units of factual information, and thus, have been used to support various natural language applications tasks like *e.g.*, retrieval [Caputo et al., 2009], novelty detection [Kumaran and Allan, 2004], and question answering, where a majority of *who-*, *where-*, and *when-* questions have answers in the form of person, location, and temporal entities, respectively [Prager et al., 2000; Srihari and Li, 2000].

- **Latent Topics:** Probabilistic topic models like Probabilistic Latent Semantic Indexing (PLSI) [Hofmann, 1999], and Latent Dirichlet Allocation (LDA) [Blei et al., 2003] have demonstrated good performance in modeling a corpus of text in terms of a small number of underlying topics, which can act as surrogates for nuggets. Both PLSI and LDA are admixture models, *i.e.*, each document can belong to multiple topics to different extents.

- **Document Source:** Another useful feature that has been successfully applied to diversification [Dou et al., 2009] is the website address or domain of the document. The goal is to diversify the ranked list with respect to the website domain, so as to prevent too many results from the same domain. Such a ranked list would favor diverse sources of information, and is more likely to address different interpretations of an ambiguous query.

Not all features are equally important towards addressing the user's information need. Certain frequently occurring words known as "stopwords" carry negligible information. Moreover, the importance of a feature depends on the user's query: *e.g.*, for a query like "BP oil spill", the system should focus on the coverage of nuggets that denote the occurrence, consequences, and containment efforts related to the oil spill. Furthermore, for broad or ambiguous queries, the importance of the features might depend on the intention or focus of the particular user, which can be determined based on explicit or implicit feedback. Therefore, the system uses the following scheme to assign weights to surrogate nuggets.

6.2.2.1 Assigning Weights to Words, Named Entities, and Document Sources

Surrogate nuggets like words, named entities, or document sources are either present or absent in each document retrieved by the system in response to a query. We assign a weight to each such nugget based on two factors:

1. The IDF (Inverse Document Frequency) of the term, which is negative logarithm of the fraction of documents in the entire corpus that contain that term. Favoring terms that are not too common in the entire corpus serves the purpose of identifying nuggets that are potentially relevant and discriminative with respect to the user's query.
2. The ranks of the documents (based on the initial ranking) that the nugget appears in. This serves the purpose of favoring those terms that appear in documents deemed more relevant by the retrieval engine.

That is, the weight assigned to nugget δ is:

$$\mathbf{w}_\delta = \mathrm{IDF}(\delta) \cdot \sum_{r=1}^{L} I(d_r, \delta) \exp(-r) \tag{6.1}$$

where r iterates over all ranks in the ranked list of length L, $I(d_r, \delta)$ indicates the presence of nugget δ in document d_r, and $\exp(-r)$ serves to exponentially discount the occurrence of the nugget in a lower-ranked document.[1]

6.2.2.2 Assigning Weights to Latent Topics

When using latent topics as nuggets, we must assign a weight to each latent topic. In the LDA model, a document belongs to each topic to different extents (as opposed to words or named entities that are either present or absent from each document), which is characterized by the posterior probability distribution $P(\theta|d)$ (see Blei et al. [2003] for details). Therefore, we assign a weight to each latent topic by modifying the above

[1] We published an earlier version of this weighting method in Lad and Yang [2010], which discounted the weights based on the document scores assigned by the initial ranker, instead of discounting based on the ranks. The latter puts more emphasis on the nuggets appearing in the top-ranked documents, and leads to better results. Therefore, we use this method in our experiments.

equation as follows:

$$\mathbf{w}_T = \sum_{r=1}^{L} P(\theta_T|d_r)\exp(-r) \tag{6.2}$$

where $\mathbf{w}_T$ represents the weight assigned to latent topic T and $P(\theta_T|d_r)$ is the posterior likelihood of document d_r belonging to latent topic T.

6.2.3 Ranking the Documents

Once we identify and assign weights to the nuggets, we must rank the documents so as to maximize the weighted coverage of the nuggets. This step is accomplished using the greedy algorithm described in Section 5.3.

Note that we are using four classes of features as surrogates for the true nuggets (see Section 6.2.2). Therefore, we must combine their contributions to determine the total gain of each document. Let the class of a nugget be denoted by $C(\delta) \in \{$`word`, `named-entity`, `latent-topic`, `doc-source`$\}$. Then, the gain of a document in terms of these features is defined as:

$$G(d_i|q) = \sum_{\delta \in \Delta_q} I(\delta, d_i) \cdot \mathbf{v}_{C(\delta)} \cdot \mathbf{w}_\delta \cdot \gamma^{\eta_\delta(d_{1:(i-1)})} \tag{6.3}$$

where $I(\delta, d_i)$ indicates the presence[2] of a nugget δ in document d_i. $\mathbf{v}_{C(\delta)}$ is the weight assigned to the class of the nugget δ. These weights reflect the relative importance of the nugget classes in determining the usefulness of a document. These weights are learned using cross-validation (see Section 6.3.4).

6.2.4 Learning From Feedback

The initial weights assigned to the nuggets may not accurately reflect the user's information need. In certain scenarios, the retrieval system has access to feedback from the user, *e.g.*, in the form of clicks on documents of interest. Our proposed solution is to leverage such user feedback to update the weights of these nuggets, and iteratively refine the ranked list produced by the system in a search session. Imagine a search session that involves multiple rounds of retrieval and user feedback. When presented with a ranked list, the user provides feedback on some of the documents. This feedback on the ranked list is used by the system to update the the user profile (characterized by the weights of nuggets), which in turn is used to produce a new ranked list for the user.

[2] Features like words and named entities are either present or absent from a document. For such features, $I(\delta, d_i)$ is either 0 or 1. On the other hand, a document belongs to one or more latent topics to varying degrees under the LDA model. In this case, $I(\delta, d_i)$ outputs values between 0 and 1, as determined by LDA's posterior distribution, $P(\theta|d_i)$.

Learning from user feedback poses two main challenges: First, user feedback is generally available at the document level: A user typically indicates interest in the entire document using either an explicit feedback button in the interface, or implicitly by taking an action to open the document, instead of indicating specific items of interest, which involves more effort. However, we need to learn the concept of usefulness at a much lower granularity of nuggets. Second, unlike traditional retrieval setup, where the relevance of each document is assumed to be independent of other documents in the ranked list, the usefulness of each document depends on other documents presented in the ranked list. Therefore, the user's feedback on a document can no longer be assumed to be independent of what he or she has seen before the current document. In other words, the user's feedback is an indicator of the *marginal* utility of a document, not its absolute utility.

To solve both these problems, we use a learning approach based on logistic regression, which models the user's feedback as a function of the marginal gain provided by each document. Specifically, the log-odds of receiving a positive feedback on a document is modeled as a linear combination of the marginal gain of each nugget in the document:

$$\Pr(f_i = 1|\mathbf{w}) = \frac{1}{1 + e^{-(g_i(\mathbf{w})+b)}} \tag{6.4}$$

where f_i is the feedback on the i^{th} document, and $g_i(\mathbf{w})$ is the corresponding marginal gain in terms of nuggets weighted by $\mathbf{w}$:

$$g_i(\mathbf{w}) = \sum_{\delta \in \Delta_{d_i}} \mathbf{w}_\delta \gamma^{\mathbb{E}[n_\delta(d_{1:i-1})]} \tag{6.5}$$

where $\mathbf{w}_\delta$ is the weight of nugget δ, and $\mathbb{E}\left[n_\delta(d_{1:i-1})\right]$ is the expected number of times nugget δ has been seen by the user before the current document.

Thus, each document in a ranked list is a training instance with label equal to the user's feedback (+1 or –1), and predictors equal to the marginal gain of each nugget. The goal of logistic regression is to find the weights for nuggets that best explain the observed user feedback. The optimal weights $\mathbf{w}^*$ are found through maximum a-posteriori (MAP) estimation, using a Normal prior whose mean is equal to the current estimate of the weights:

$$\Pr(\mathbf{w}) \sim \mathcal{N}(\mathbf{w}_0|\lambda\mathbf{I}) \tag{6.6}$$

where λ controls the strength of the prior. The use of a prior allows the system to adapt to the user's interests in an incremental manner by using the previous iteration's weights as the prior for the current step. In the first iteration (*i.e.*, the first ranked list produced in response to a query), $\mathbf{w}_0$ is initialized using the weighting scheme described in Section 6.2.2.1.

The optimal weights maximize the log-likelihood over all documents on which feedback is received:

$$\ell(\mathbf{w}) = -\sum_i \log(1 + exp(-f_i g_i(\mathbf{w}) - b)) - \frac{\lambda}{2}||\mathbf{w} - \mathbf{w}_0||^2 \tag{6.7}$$

which can be solved efficiently using conjugate gradient ascent [Minka, 2003].

6.3 Experiments

Our goal is to evaluate and compare various approaches for novelty-based ranking with respect to the proposed EGU metric. We conduct experiments using these approaches on two datasets: (i) the TDT dataset as described in Chapter 4, and (ii) the ClueWeb09 dataset (described below). These datasets lend themselves to two different information retrieval scenarios: The first scenario assumes long-lasting queries that involve multiple rounds of retrieval and user feedback. The second scenario corresponds to ad hoc search over queries that are characteristic of users' information-seeking behavior on the web.

In Section 6.3.1, we describe the datasets, followed by a description of the retrieval scenarios in Section 6.3.2. In Section 6.3.3, we provide details of the various ranking approaches that we compare.

6.3.1 Datasets

6.3.1.1 Topic Detection and Tracking (TDT) Dataset

TDT4 was a benchmark corpus used in Topic Detection and Tracking (TDT2002 and TDT2003) evaluations. It consists of over 28,000 English articles from various news sources published between October 2000 and January 2001. We extended this corpus for novelty-based evaluations by creating 120 queries. Each query is associated with multiple nuggets that reflect the various units of information needed to completely answer the query; see Chapter 4 for details. This dataset allows us to model long-lasting queries intended to track evolving news events.

6.3.1.2 The ClueWeb09 Dataset

The ClueWeb09 dataset comprises over 1 billion webpages gathered from the Internet in January and February, 2009 by researchers at Carnegie Mellon University[3]. ClueWeb09 was used as the benchmark dataset in the TREC 2009 Web Track [Clarke et al., 2009]. A

[3] http://boston.lti.cs.cmu.edu/Data/clueweb09/

subset of this dataset known as “Category B” consists of the first 50 million English webpages, including the full English Wikipedia collection. We use the Category B subset for our experiments. For the TREC 2009 Web track, NIST created a set of 50 topics that were used for the ad hoc as well as diversity task. For the diversity task, each topic was further divided into a representative set of subtopics (4.9 subtopics per query on average) to capture different information needs related to the main topic. Documents were manually annotated to mark the presence of nuggets for each query[4]. These subtopics can be treated as nuggets in our evaluation framework. This dataset allows us to model the information-seeking behavior of typical web searchers.

Next, let us look at how we use these two datasets to simulate two different retrieval scenarios.

6.3.2 Retrieval Scenarios and Experimental Setup

6.3.2.1 Long-lasting queries on the TDT Dataset

One of the retrieval scenarios of interest to us is session-based retrieval involving long-lasting queries so that we can evaluate the ability of the retrieval system to track events and rank documents that are relevant as well as novel with respect to the entire session, and can update its model based on user feedback. The long-lasting nature of queries is similar to the functionality provided by Google Alerts[5], which allows users to submit queries, whose results are sent to the user via email at regular intervals of time in the form of a ranked list of relevant webpages. We go further in two ways: First, we assume that the feedback provided by the user on these results in the form of clicks is available to the system and can be used to adjust subsequent rounds of retrieval. Second, we assume that the user is only interested in content that is relevant as well as novel with respect to the documents presented to him or her in previous ranked lists.

We use the TDT dataset to simulate such a retrieval scenario, targeted towards a hypothetical intelligence analyst who needs to gather relevant information on evolving news stories around the world by monitoring various sources of information. To simulate such a user-in-the-loop retrieval setting in an offline setup, we divide the 4-month span of the TDT dataset described in Chapter 4 into chunks of 5 consecutive days each, which leads to approximately 1,100 news articles per time chunk. At the end of each such time chunk, the retrieval system is expected to produce a ranked list of documents using the past 5 days of documents as the corpus. The system receives feedback from the user, and then produces a new ranked list for the next chunk, and so on. This setup simulates a user who is following an evolving news event over

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

[5] http:/www.google.com/alerts

an extended period of time—expecting the retrieval system to return a personalized ranked list of relevant and novel documents after every 5 days.

Simulation of Feedback. We are interested in analyzing the ability of the proposed approach to leverage user feedback to improve its performance in subsequent ranked lists. Therefore, we simulate a user in the loop as follows. In response to each ranked list presented by the system, we algorithmically simulate a user who browses the ranked list in a top-down manner, and at each rank r, randomly decides to provide feedback with probability equal to $P_R(r) \times P(feedback)$, where $P_R(r)$ is the probability of the user reading the document at rank r according to our persistence model of user browsing (Section 3.2.1), which uses a geometric distribution over stopping positions. $P(feedback)$ is a free parameter that we vary between 0 and 1 to simulate users who are willing to provide different amounts of feedback to the retrieval system.

6.3.2.2 Ad hoc Search on the ClueWeb09 Dataset

For completeness, we would also like to understand the behavior of the proposed evaluation and optimization methods on web-scale retrieval. Therefore, we also conduct experiments on the standard ad hoc retrieval scenario, where the system is expected to return a single ranked list of documents in response to each query. Novelty detection in this case amounts to avoiding redundancy among documents in a single ranked list. This is equivalent to the goal of diversity-based retrieval, assuming redundancy is defined in terms of aspects or categories, which are treated as nuggets in our framework. Novelty in terms of such nuggets leads to diversification of search results.

Unlike the TDT dataset, we do not explore the feedback-based retrieval scenario on the ClueWeb09 dataset due to the difference in the nature of this test collection: The information needs defined by TREC do not pertain to events that evolve over time. Instead, the TREC diversity-based retrieval task was targeted towards an ad hoc retrieval setting with a single ranked list per query. Moreover, no query chains are available, making it difficult to simulate a search session with multiple iterations of retrieval and user feedback.

6.3.3 Retrieval Approaches

We compare the following retrieval approaches on the two datasets:

6.3.3.1 Baseline Ranking

Our baseline (henceforth referred to as `Baseline`) is a purely relevance-based ranking approach, which does not attempt to perform novelty detection or result diversification. On the TDT dataset, we use Indri [Strohman et al., 2004] as our baseline ranker, which is a state of the art retrieval engine for relevance-based ranking.

On the ClueWeb09 dataset, which is a collection of web documents, it is non-trivial to create a well-performing ranker due to the challenges associated with estimating relevance at web scale and varying content quality (*e.g.*, presence of spam). Therefore, we use one of the top-performing retrieval systems in TREC 2009 Web Track on the ad hoc retrieval task, submitted by University of Maryland under the run ID `UMHOOsd`. The output of this run is treated as the baseline approach for the purposes of our experiments.

6.3.3.2 Maximum Marginal Relevance

Maximum Marginal Relevance (`MMR`) [Carbonell and Goldstein, 1998] is a well-known approach for diversifying search results. It starts with the baseline ranking as the candidate set of documents, and incrementally builds a diversified ranked list by choosing the next document with the highest marginal relevance, *i.e.*, high relevance to the query, and low similarity to already selected documents in the ranked list:

$$f(d_i|q, d_{1:(i-1)}) = \lambda_m \cdot \text{sim}(d_i, q) - (1 - \lambda_m) \cdot \max_{d_j \in d_{1:(i-1)}} \text{sim}(d_i, d_j) \tag{6.8}$$

where λ_m controls the relative importance of choosing relevant versus novel documents. In our experiments, λ_m is chosen using cross-validation. $\text{sim}(d_i, q)$ is the score assigned by the initial relevance-based ranker, and $\text{sim}(d_i, d_j)$ is defined as the cosine similarity between TF–IDF weighted document vectors.

6.3.3.3 Redundancy Filtering

Redundancy Filtering (`RedFilter`) goes through the baseline ranking from top to bottom and removes (*i.e.*, filters out) all documents that are too similar to one of the previous (higher-ranked) documents. This approach has been used previously for redundancy detection in adaptive filtering [Zhang et al., 2002] as well as non-redundant passage retrieval [Lad and Yang, 2007; Yang and Lad, 2009]. Specifically, the novelty score of a document is defined as follows:

$$\text{nov}(d_i|q, d_{1:(i-1)}) = 1 - \max_{d_j \in d_{1:(i-1)}} \text{sim}(d_i, d_j) \tag{6.9}$$

Documents with novelty score below a threshold, λ_r, are removed from the ranked list. The optimal value of λ_r is determined using cross-validation.

Note that the RedFilter does not modify the order of documents with respect to the original (Baseline) ranking, but simply removes redundant documents, unlike the MMR approach, which reorders documents so that documents with a higher balance of relevance and novelty appear at higher ranks.

6.3.3.4 Nugget-based Approach

This is the proposed approach that uses various document features as surrogates for the true nuggets, and then reorders the ranked list so as to maximize EGU. We compare the following combinations of features:

Nug[W]	Words
Nug[W+NE]	Words and Named Entities
Nug[T]	Latent Topics
Nug[S]	Source of the Document
Nug[W+NE+T+S]	All of the above

Named Entities: We use the Stanford Named Entity recognizer [Finkel et al., 2005] to extract three types of named entities: person names, locations, and organization names.

Latent Topics: We use LDA to extract latent topics since it has demonstrated performance superior to PLSI (in terms of lower perplexity) in modeling several text and collaborative filtering corpora [Blei et al., 2003]. We use the implementation of LDA made publicly available[6] by Blei et al. [2003]. LDA, like most clustering approaches, requires the number of latent topics as an input parameter. However, it is not clear how to determine the optimal number of topics for maximizing EGU. Therefore, we tried various values (see Figure 6.1) on a held-out set of queries and chose the best value: 10 latent topics.

Source of the Document: We treat the source of the document as a nugget. On the TDT dataset, this corresponds to the newswire source (*e.g.*, ABC, NBC, AP, and so on). Novelty detection in this case corresponds to avoiding many results from the same news agency.

On the ClueWeb09 dataset, this corresponds to the domain name in the webpage URL (*e.g.*, apple.com, microsoft.com, and so on). We tried two alternatives in the latter case: (i) the full host name (*e.g.*, www.lti.cs.cmu.edu), and (ii) the second-level domain name (*e.g.*, cmu.edu). The second alternative has the effect of mapping many sub-domains (*e.g.*, www.lti.cs.cmu.edu and www.ri.cmu.edu) to the same value (cmu.edu). On the ClueWeb09 dataset, we obtained better performance (+0.0120 in

[6] http://www.cs.princeton.edu/~blei/lda-c/

terms of EGU) with the second alternative on a held-out set of queries. Therefore, we use second-level domain names in our experiments on the ClueWeb09 dataset.

6.3.3.5 Upper Bound on Performance

All our novelty-based ranking methods are based on a two-step process: First, retrieve the initial set of documents using a relevance-based ranker. Second, re-rank the documents using one of the above-mentioned approaches. Obviously, the performance of the second step is dependent on that of the first step, *i.e.*, the ability of the relevance-based ranker to retrieve as many relevant documents as possible in the initial candidate set. If this candidate set does not cover many nuggets in the first place, there is little hope for the re-ranker to improve the quality of the ranked list.

To understand the effect of the relevance ranker's performance on the novelty-based re-ranker, we have included a special run called Upper Bound, which denotes the best possible re-ranking that can be achieved on the initially retrieved set of documents. Such a ranking is produced by accessing the ground truth of nuggets and therefore does not correspond to any real retrieval system. Nevertheless, it represents the best performance obtainable under the limitations of the relevance ranker's performance.

6.3.4 Cross-Validation for Parameter Tuning

All the novelty-based retrieval approaches that we wish to compare admit certain free parameters whose values must be manually determined for best performance. For instance, MMR uses λ_m to combine the relevance and novelty scores for each document (Section 6.3.3.2). RedFilter uses λ_r as the novelty threshold to filter our redundancy documents from the ranked list (Section 6.3.3.3). Also, the proposed nugget-based methods use parameters to control the relative weights of different features classes (Section 6.2.3).

We used a 5-fold cross-validation setup to choose the best values for these parameters in the most fair manner. Specifically, on each dataset, the queries were divided into 5 folds, each containing 20% of the queries. The first fold was used to optimize parameters and the performance was evaluated on the other four folds. This process was repeated—using the second fold for parameter tuning and evaluating on the other four folds, and so on. Thus, the results presented are averaged over 5 sub-experiments, each containing 80% of the queries. This ensures that all (100%) of the queries are represented in the final results.

6.3.5 Evaluation Measures

To provide a complete picture of the performance of various methods, we report three evaluation measures: EGU, $\alpha-$NDCG, and Prec@20. In EGU, we use $\gamma = 0.1$, which corresponds to low tolerance for redundancy. We are interested in ranking performance in these experiments as opposed to filtering performance (which we explored in Chapter 4 where the system was evaluated in terms of its ability to produce a finite list of useful documents for the user's attention). Therefore, here we set the reading cost to zero. For meaningful comparison, we set $\alpha = 0.9$ in $\alpha-$NDCG, which corresponds to the same redundancy tolerance as $\gamma = 0.1$ in EGU. Both $\alpha-$NDCG and precision are calculated at the depth of 20 results, which provides sufficient number of documents for reliable evaluation and comparison between the methods. Note that $\alpha-$NDCG is not suited for evaluating search sessions comprising multiple ranked lists. Therefore, we adapt $\alpha-$NDCG as described in Section 4.4.3.

6.4 Results

Table 6.1 shows the performance obtained by various novelty-based retrieval strategies using three evaluation measures on the two datasets. We performed paired t-tests to assess the statistical significance of the performance scores[7]. Statistically significant increases in performance over the Baseline (*i.e.*, p-value $< .01$) are marked with a "†" in the table.

Overall, novelty-based ranking leads to better performance in terms of EGU and $\alpha-$NDCG as compared to the baseline with no novelty detection. Let us focus on performance in terms of EGU: On the TDT dataset, the best-performing run (Nug[W+NE+T+S]) obtained an EGU score of 0.4612, which is an improvement of 15% over the no novelty-detection baseline, and 7% over the redundancy filtering approach. On the ClueWeb09 dataset, the best-performing run (Nug[W+NE+T+S]) obtained an EGU score of 0.4522, which is an improvement of 14% over the no novelty-detection baseline, and 5% over the redundancy filtering approach.

The use of novelty detection consistently leads to deterioration of performance in terms of Prec@20, which is expected since novelty detection methods inevitably lead to removal of some relevant documents that are redundant. However, precision does not differentiate between relevant-and-novel documents and relevant-and-redundant documents.

[7] The statistical assumptions of the t-test may or may not be satisfied for the evaluation measures that we are using. However, the t-test is known to be quite robust to violations of its assumptions [Lehmann and Romano, 2005]. Nevertheless, we also conducted the permutation test, which is a non-parametric test advocated by Smucker et al. [2007] for IR evaluations; no differences in significance outcomes were observed as compared to the t-test.

Now let us analyze the results in more detail. We mainly focus on performance in terms of `EGU`, and later analyze the correlation of `EGU` with $\alpha-$`NDCG` as well as `Prec@20` in Section 6.4.3.

6.4.1 Performance of **`MMR`** and Redundancy Filtering

On both datasets, the use of `MMR` and Redundancy Filtering (`RedFilter`) leads to better performance as compared to the baseline with respect to both `EGU` and $\alpha-$`NDCG`.

Redundancy Filtering performs better than `MMR` on both datasets. This is most likely because of the different modeling assumptions that underlie these two methods: `MMR` assumes that utility is a linear function of relevance and novelty and treats these two criteria as compensatory, which is unrealistic (see Section 2.4.2). In other words, `MMR` might favor a document that is highly novel but only marginally relevant to the query. On the other hand, the two stage filtering used in the Redundancy Filter only keeps those documents that are relevant *and* novel.

6.4.2 Performance of the Nugget-based method

The proposed nugget-based ranking method (`Nug[]`) admits various features as surrogates for nuggets. Let us analyze the effectiveness of each of these features for novelty and diversity-based retrieval.

Words and Named Entities as features. The use of words (`Nug[W]`) and named entities (`Nug[W+NE]`) leads to better performance over the traditional novelty-based approaches (`MMR` and `RedFilter`) on the TDT dataset. However, these features do not help in improving the performance on the ClueWeb09 dataset, and in fact lead to worse performance compared to the traditional novelty detection methods. This can be explained the terms of the nature of the datasets and their corresponding queries. The TDT dataset is a collection of news articles and the associated queries represent information needs related to news events. These queries generally ask for low-granularity information, *e.g.*, names of people, locations of events, and so on, which can be easily addressed by individual words or named entities. That is, the true nuggets for these queries can be effectively modeled using words and named entities.

On the other hand, the ClueWeb09 dataset consists of web pages and the associated queries represent typical web searches that may have multiple facets. These facets generally take the form of high-level topics, *e.g.*, “DJs specializing in hip hop music”, “hiring a DJ for wedding”, “jobs available for DJs”, and so on, which cannot be captured completely by individual words and named entities. In other words, these information needs are topical, while the information needs on the TDT dataset are factual.

Table 6.1: Performance of various approaches on the two datasets. The proposed nugget-based method is denoted as `Nug[]` with the following features: `W`: Words, `NE`: Named Entities, `T`: Latent Topics, and `S`: Document Source. † means statistically significant improvement with respect to `Baseline`.

(a) Performance on the TDT dataset (long-lasting queries).

System	`EGU`	$\alpha-$`NDCG`	`Prec@20`
`Baseline`	0.4016	0.2655	$\mathbf{0.2890}$
`MMR`	$0.4202^{\dagger}$	$0.3023^{\dagger}$	0.2690
`RedFilter`	$0.4311^{\dagger}$	$\mathbf{0.3141}^{\dagger}$	0.2670
`Nug[W]`	$0.4435^{\dagger}$	0.2742	0.2540
`Nug[W+NE]`	$0.4598^{\dagger}$	$0.2773^{\dagger}$	0.2480
`Nug[T]`	0.4029	0.2568	0.2670
`Nug[S]`	0.4018	0.2456	0.2650
`Nug[W+NE+T+S]`	$\mathbf{0.4612}^{\dagger}$	$0.2857^{\dagger}$	0.2640
`Upper Bound`	$0.6939^{\dagger}$	$0.5704^{\dagger}$	$0.5930^{\dagger}$

(b) Performance on the ClueWeb09 dataset (ad hoc search).

System	`EGU`	$\alpha-$`NDCG`	`Prec@20`
`Baseline`	0.3972	0.2459	$\mathbf{0.2800}$
`MMR`	$0.4168^{\dagger}$	$0.2659^{\dagger}$	0.2350
`RedFilter`	$0.4313^{\dagger}$	$0.2765^{\dagger}$	0.2300
`Nug[W]`	0.4098	0.2602	0.2420
`Nug[W+NE]`	0.4110	$0.2682^{\dagger}$	0.2410
`Nug[T]`	$0.4419^{\dagger}$	$0.2811^{\dagger}$	0.2290
`Nug[S]`	$0.4442^{\dagger}$	$0.2882^{\dagger}$	0.2250
`Nug[W+NE+T+S]`	$\mathbf{0.4522}^{\dagger}$	$\mathbf{0.2920}^{\dagger}$	0.2200
`Upper Bound`	$0.6674^{\dagger}$	$0.7008^{\dagger}$	$0.6650^{\dagger}$

Latent Topics as features. The difference in the nature of information needs on the two datasets is further supported by performance of latent topics (`Nug[T]`). On the TDT dataset, no (statistically significant) performance improvement is observed when using latent topics, whereas we observe a substantial improvement on the ClueWeb09 dataset. Again, this is due to the topical nature of information needs on this dataset, which is better captured using clustering techniques like LDA. These observations match with

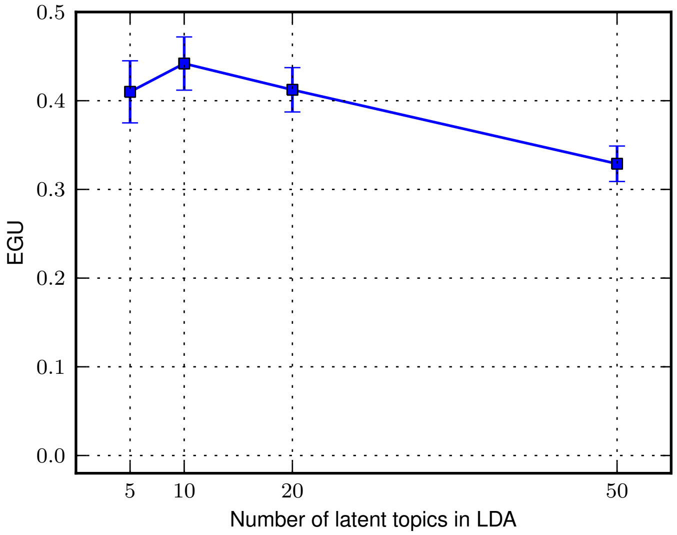


Figure 6.1: Effect of the number of latent topics on the system's performance on the ClueWeb09 dataset. The mean performance is flanked by one standard deviation, as measured across queries.

those of other researchers who have worked on result diversification on this dataset. [Dou et al., 2009; Li et al., 2009; Meij et al., 2009].

Effect of number of latent topics. When using latent topics as surrogates for the true nuggets, we must manually choose the number of latent topics to generate using LDA. We chose the optimal number using cross-validation on a held-out set of queries. Our experiments on the ClueWeb09 dataset indicated that the best performance is obtained when using 10 latent topics; see Figure 6.1. However, this is most likely dependent on the dataset and queries.

Document Source as feature. As explained in Section 6.3.3.4, we also used the source of a document as a surrogate for the true nuggets. Interestingly, this simple feature led to substantial performance boost on the ClueWeb09 dataset, and in fact is the best-performing feature on this dataset. A closer inspection of the queries revealed that the facets of most queries are often addressed by pages from different websites, *e.g.*, one of the TREC queries has the following facets: "restrictions on checked baggage during air travel", "sites that collect statistics on flight delays", "AAA's website", and "TSA's website". Moreover, diversifying in terms of the website also helps in improving performance on ambiguous queries, which appear in the TREC query set.

No (statistically significant) performance improvement was observed on the TDT

Table 6.2: Optimal weights of different feature classes on the two datasets with respect to EGU.

Dataset	Weights of Features			
	Words	Named Entities	Latent Topics	Source
TDT	0.4	0.5	0.1	0.0
ClueWeb09	0.1	0.1	0.4	0.4

dataset when document source was used as a nugget. This is primarily due to the factual nature of the queries. News events lead to more or less standard reporting, which does not vary to a large extent across newswire sources. This makes the source of the document irrelevant as far as answering factual queries is concerned.

Most effective feature combination. We also combined all the above-mentioned features as surrogates for nuggets. The combination weights were chosen using a held-out set of queries. In Table 6.2, it is evident that the optimal combination of features depends on the dataset. The relative weights of the features agree with our observations above on the individual contributions of the various features on the two datasets: Words and named entities are very effective on the TDTdataset, whereas latent topics and document source perform very well on the ClueWeb09 dataset.

Effect of User Feedback. On the TDT dataset, where we model a session-based retrieval scenario with long-lasting queries, we would like to analyze the effect of user feedback on the performance of the system. We simulated user feedback using the protocol described in Section 6.3.2. In Figure 6.2, we have plotted the system performance with respect to different values of $P(feedback)$ that correspond to users who are willing to provide different amounts of feedback to the retrieval system. The orange line corresponds to performance measured using EGU with $\gamma = 0.1$, *i.e.*, some redundancy tolerance, which is consistent with the rest of our experiments.

Even with 100% feedback, the performance does not improve substantially over the no-feedback variant: we only get an improvement of 5.3%. This can be explained by the reasoning that the benefits of feedback are nullified to some extent by the demand for novelty: Through feedback, the user indicates interest in specific items, but at the same time, expects the system to not retrieve the same (or very similar) items in the future, but instead, other items that are relevant *and* novel. Our evaluation merely shows that the benefits of user feedback may be overestimated if novelty (or redundancy) is not taken into account in the performance measure. To validate this hypothesis, we also measured the improvement in performance in terms of EGU with $\gamma = 1.0$ (blue

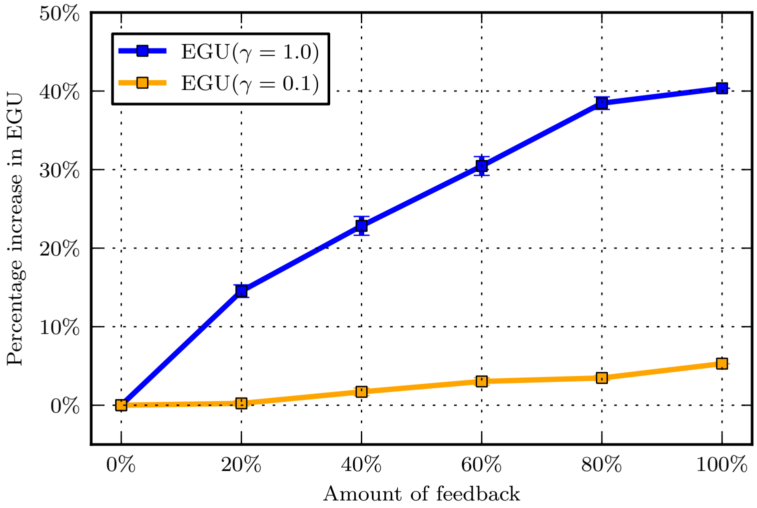


Figure 6.2: Effect of different amounts of user feedback on the performance of the nugget-based approach with all features (`Nug[W+NE+T+S]`) for two different redundancy tolerances (γ).

line in the graph), which corresponds to full tolerance for redundancy, and hence, the traditional relevance-based evaluation. In this case, we indeed observe a large increase in performance when feedback is provided.

6.4.3 Correlation of EGU with other metrics

To understand the behavior of the different evaluation measures, we calculated the correlation between the EGU scores of various runs (the first eight rows of Tables 6.1a and 6.1b, respectively) against the $\alpha-$NDCG scores, as well as EGU and Prec@20 scores, as shown in Table 6.3.

EGU and $\alpha-$NDCG. On the ClueWeb09 dataset, the EGU and $\alpha-$NDCG scores have strong positive correlation (Pearson's correlation coefficient $r = 0.97$). This is expected due to their highly similar form (see Section 3.6.1) for single ranked lists, which is indeed the retrieval scenario used on the ClueWeb09 dataset.

On the TDT dataset, the retrieval scenario consists of multiple ranked lists. The proposed measure, EGU, more accurately models utility (*i.e.*, relevance and novelty) across multiple ranked lists in a search session, which we claim is an important advantage over $\alpha-$NDCG. This difference appears as lower correlation ($r = 0.45$)

Table 6.3: Correlation between EGU, $\alpha-$NDCG, and Prec@20 scores of different novelty-based ranking methods on the two datasets.

Dataset	Correlation coefficients (r)	
	EGU vs. $\alpha-$NDCG	EGU vs. Prec@20
TDT	0.45	−0.69
ClueWeb09	0.97	−0.85

between EGU and $\alpha-$NDCG scores on the TDT dataset. The disparity is especially apparent when we compare the performance scores of the existing ranking methods (MMR and RedFilter) against the nugget-based methods (*e.g.*, Nug[W]). While MMR and RedFilter improve performance in terms of both EGU as well as $\alpha-$NDCG, Nug[W] improves performance in terms of EGU but decreases the $\alpha-$NDCG scores. This highlights the difference between the model of novelty inherent in MMR and RedFilter on the one hand, and the nugget-based methods on the other hand. MMR and RedFilter do not consider the rank of a previously-presented document when computing the novelty of a given document. This is also consistent with how $\alpha-$NDCG handles novelty across ranked lists (see Section 4.4.3). On the other hand, the nugget-based methods are sensitive to the ranks at which similar documents have appeared in the previous ranked lists, which is consistent with the model of novelty used in EGU. This causes the nugget-based methods, which are targeted towards optimizing EGU, to improve the EGU scores as expected, but decrease the $\alpha-$NDCG scores.

EGU and Prec@20. On both datasets, the use of novelty detection methods leads to decrease in the precision (Prec@20) scores, which appears as negative correlation between EGU and Prec@20 scores in Table 6.3. This is expected, as we already explained in Section 6.4: The use of novelty detection inevitably leads to removal of relevant (but redundant) documents, which is penalized by Prec@20, which does not differentiate between relevant-and-novel documents and relevant-and-redundant documents.

The negative correlation is stronger on the ClueWeb09 dataset ($r = -0.85$), where the retrieval scenario involves one ranked list per query, as compared to that on the TDT dataset ($r = -0.69$), where the multiple ranked lists per query scenario makes the EGU calculation substantially different from Prec@20, which simply averages over the individual precision scores of each ranked list in the search session.

6.5 Time Complexity

Let us estimate the time complexity of the proposed approach and compare it against other strategies for novelty-based ranking. Assume that the goal is to create a ranked list of length k, for which, we start with a candidate set[8] containing $O(k)$ documents[9]. Assume that each document contains m words on average.

The proposed nugget-based approach first processes the candidate set of documents to assign initial weights to all surrogate nuggets (*e.g.*, words), which requires O(km) operations. Then, the ranked list is built in k steps: At each step, score the remaining O(k) documents based on their marginal utility, which requires O(m) operations per document, assuming utility is calculated by treating words as surrogates for nuggets. This step leads to a time complexity of O(k^2m). Hence, the total time complexity of the proposed approach is O$(km + k^2m)$, or simply, O(k^2m).

The MMR-based approach is also based on iteratively building the ranked list: At each of the k steps, score the remaining O(k) documents by computing their cosine similarity with each of the O(k) documents already selected in the ranked list. A single cosine similarity computation takes O(m) operations. This leads to a time complexity of O(k^3m).

The redundancy filtering method processes the initial ranked list in a top-down manner: At each of the k steps, it calculates the cosine similarity of the document at the k^{th} rank with each of the previously-selected $O(k)$ documents. A single cosine similarity computation takes O(m) operations. This leads to a time complexity of O(k^2m).

In Figure 6.3, we have plotted the actual number of operations required for each of the three approaches against the ranked list length, assuming an initial candidate document set of size $3k$, and average number of words per document (*i.e.*, m) equal to 50.

Note that the proposed nugget-based approach also admits complex features that require additional processing, *e.g.*, running the Latent Dirichlet Allocation (LDA) inference to generate the latent topic features. For meaningful comparison, we have limited the time complexity computations to "words as nuggets" only, which are indeed the features used by all the three methods: the nugget-based method, MMR, and redundancy filtering.

[8] The retrieval of this candidate set is common to all novelty-based ranking methods, so we ignore it from the time complexity calculations.

[9] The initial set must be at least as big as the length of the final ranked list (*i.e.*, *k*). A standard choice in our experiments is to start with a candidate set of $3k$ documents.

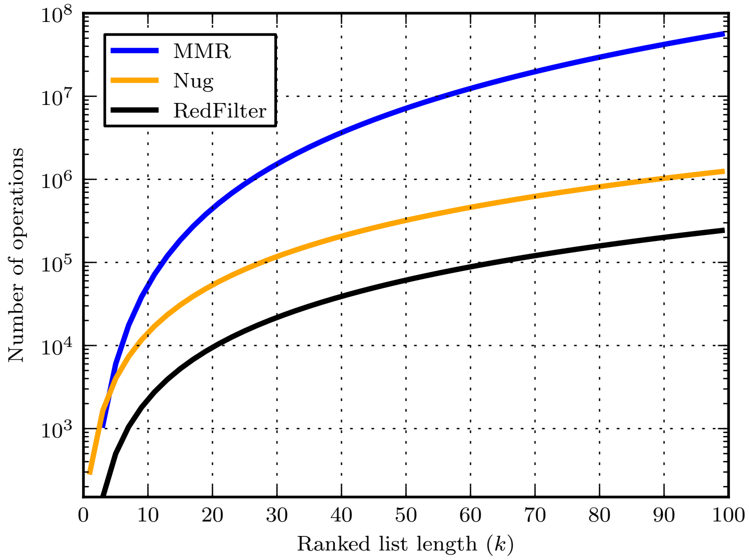


Figure 6.3: Time complexities of various novelty-based ranking approaches.

6.6 Related Work

Novelty and diversity-based retrieval. One of the earliest works that combined relevance and novelty as criteria for retrieval is the *Maximal Marginal Relevance* (MMR) strategy [Carbonell and Goldstein, 1998], which proposes a greedy algorithm for incrementally producing a diverse ranked list. However, MMR lacks a global objective function, and hence, does not explicitly maximize an appropriate measure of retrieval performance. Zhai et al. [2003] and Agrawal et al. [2009] extended the MMR algorithm by replacing the notion of marginal relevance with marginal utility that is defined in terms of aspects and categories, respectively. In Chapter 2, we discussed an inherent problem with all MMR-based approaches: They treat relevance and novelty as compensatory criteria, and therefore may favor highly novel but irrelevant documents, which users are unlikely to find useful for their information need. Our proposed approach is based on directly modeling the relevant and novel information by using surrogates for the true nuggets. Yue and Joachims [2008] proposed a diversification approach based on the weighted coverage of words. Zhang et al. [2002] explored several similarity measures for novelty detection. However, they used an adaptive filtering setup and focused on the problem of identifying novel documents (*i.e.*, making a yes–no decision) in a given set of relevant documents. Our focus in on ranked retrieval, which is a much more common mode of interaction with retrieval systems, and requires a careful combination of relevance and novelty criteria. Moreover, none of the above-mentioned approaches explicitly model the user's tolerance towards redundancy or the user's browsing behavior with respect to one or more ranked lists, which are essential components of our performance measure.

Mathematical frameworks for optimizing retrieval performance. Zhai [2002] proposed a risk minimization framework for information retrieval that can incorporate various loss functions, and reduce retrieval to a statistical decision problem. He proposed a specific form of loss function for sub-topic retrieval based on the coverage of "aspects". However, the risk minimization approach did not correspond to any explicit measure of retrieval performance that was based on ground truth. Zhai mentioned the challenges associated with combining relevance and novelty into a single measure of retrieval performance. As a compromise, he measured aspect coverage at a few arbitrarily chosen ranks and recall levels. Such a measure is not based on a user model, and does not allow different tolerances towards redundancy. Gollapudi and Sharma [2009] proposed an axiomatic approach to diversity-based retrieval by characterizing diversification approaches in terms of their relevance and novelty functions. However, their framework is based on explicit modeling of similarities between each pair of documents, which does not have a direct correspondence with how users derive utility from search results. Instead, our framework is based on modeling utility in terms of information units (*i.e.*, nuggets), which directly correspond to how users benefit from seeing each subsequent document in a ranked list. Moreover, their framework uses set-based objective functions, while we directly model ranking performance by taking the probabilistic browsing behavior of the user into account.

Learning from User's Feedback. El-Arini et al. [2009] proposed an approach for learning from user's feedback to provide a personalized set of diverse blogs to the user. They address the problem of non-independent feedback. However, their objective function is set-based, and hence, does not take ranking performance into account. Also, similar to the other approaches for diversity-based retrieval, different tolerances towards redundancy are not taken into account.

Learning from click-through data A recent line of work by Radlinski et al. [2008a] uses click-through data to optimize rankings. Since real users implicitly take all pertinent factors (relevance and novelty with respect to previously seen documents) into account when clicking on documents, such an approach can optimize for novelty without explicitly modeling it. However, such approaches are expensive since they require multiple interactions with real users to collect click-through patterns on different variations of ranked lists for each query. Therefore, such approaches do not provide an efficient means for offline testing of new and potentially risky algorithms.

6.7 Summary

In this chapter, we focused on the problem of system optimization, *i.e.*, finding the optimal ranked list in response to a query, without the knowledge of true relevant nuggets (ground truth). We described a general strategy for optimizing system performance through direct modeling of nuggets in terms of observable document features like words, named entities, latent topics, and document source. Leveraging user feedback in the presence of non-independent document utility also poses new challenges, for which we proposed a logistic-regression based learning algorithm.

We conducted experiments on two datasets to observe the performance of various novelty-based ranking methods on two retrieval scenarios. The first scenario was based on long-lasting information needs, which involve multiple rounds of retrieval and user feedback over an extended period of time. The second scenario was the standard web retrieval setup, where the goal was to retrieve a single ranked list in response to a query, while avoiding redundancy and thus maximizing the diversity of search results.

These experiments on the two datasets led to several interesting observations. On both datasets, all the novelty-based methods improved system performance in terms of EGU and $\alpha-$NDCG, while performance deteriorated in terms of Prec@20. This was expected, since Prec@20 does not distinguish between relevant-and-novel documents and relevant-and-redundant documents, making it unsuitable for novelty-based evaluation of retrieval systems. Redundancy filtering led to better performance than Maximum Marginal Relevance (MMR) on both datasets.

The performance of particular features in the proposed nugget-based method was dependent on the dataset. Words and named entities performed very well on the TDT dataset due to the factual nature of the information needs that was effectively captured by these features. On the ClueWeb09 dataset, latent topics performed very well due to the high-level or topical nature of the information needs, better captured by clustering methods. Interestingly, a simple feature, the web domain of the document, alone demonstrated very good performance on the ClueWeb09 dataset. An analysis of the information needs on this dataset indicated that most sub-topics are covered by web pages from different web domains or companies.

We also analyzed the correlation of EGU with two other commonly used metrics: $\alpha-$NDCG, and Prec@20. Since both these metrics are designed for evaluating single ranked lists, they were more correlated with EGU on the ClueWeb09 dataset (where we used a single ranked list retrieval setup) than on the TDT dataset (where we used a session-based retrieval setup involving multiple rounds of retrieval to satisfy long-lasting information needs).

Chapter 7

Conclusions and Future Work

7.1 Summary and Conclusions

In this thesis, we proposed a new evaluation and optimization framework for relevance and novelty based retrieval. Novelty based retrieval breaks the assumption of independent document utility, which leads to various challenges for evaluating as well as optimizing ranked lists: The dependence of novelty on other documents seen by the user means that it is not possible to build test collections based on offline per-document judgments. The non-independent nature of novelty further complicates the evaluation of session-based search comprising multiple ranked lists, which must be evaluated while taking the novelty (or redundancy) across ranked lists into account. Moreover, users might have different tolerances towards redundancy; this factor is not taken into account by existing evaluation metrics for novelty as well as diversity-based retrieval.

In Chapter 3, we proposed a new evaluation measure called Expected Global Utility (EGU) that realistically models the utility derived by a user going through one or more ranked lists in a search session. EGU models the utility of reading a document in terms of relevant and novel nuggets, with a diminishing returns property to model different tolerances towards redundancy. It also allows a notion of reading cost to support the evaluation of filtering scenarios. This notion of utility is then combined with a probabilistic model of user browsing behavior, which enables the formulation of the *expected* utility of a ranked list for an average user.

In Chapter 4, we analyzed the behavior of EGU on real retrieval systems and datasets. We conducted experiments on the diversity-based retrieval task using existing TREC runs, and analyzed the rankings of systems under various conditions including different tolerances towards redundancy and different user browsing models. We also analyzed the correlation with other well-understood measures like $\alpha-$NDCG and precision. We also examined the task of information distillation, which combines adaptive filtering, passage retrieval, and novelty detection. To support systematic and repeatable evaluations on this retrieval task, we developed a new testbed by extending the TDT4 dataset. We used a semi-automatic approach for creating "nugget-matching rules" that enable automatic evaluation of retrieval systems that return arbitrary spans of text. We conducted experiments using multiple parameter configurations of the CMU Adaptive Filtering Engine, which highlighted the main differences in the properties of EGU with respect to $\alpha-$NDCG and precision.

In Chapter 5, we focused on the computational challenges associated with finding the optimal ranked list assuming perfect knowledge of the nuggets. This turns out to be an NP-hard problem, but we showed that the sub-modularity of EGU allows a simple greedy algorithm to guarantee good performance. We also developed a tighter lower-bound that takes into account the fact that EGU computes an expectation over multiple ranks. We conducted experiments on synthetic ranked lists, which confirmed that the greedy algorithm displays very good performance under a wide range of retrieval scenarios and therefore can be effectively used to optimize ranked lists with respect to the proposed evaluation measure, EGU.

In Chapter 6, we focused on the problem of system optimization, *i.e.*, finding the optimal ranked list in response to a query, without the knowledge of true relevant nuggets (ground truth). We described a general strategy for optimizing system performance through direct modeling of nuggets in terms of observable document features like words, named entities, latent topics, and document source. Leveraging user feedback in the presence of non-independent document utility also poses new challenges, for which we proposed a logistic-regression based learning algorithm.

We conducted experiments on two datasets to observe the performance of various novelty-based ranking methods on two different retrieval scenarios. The first scenario was based on long-lasting information needs, which involve multiple rounds of retrieval and user feedback over an extended period of time. The second scenario was the standard web retrieval setup, where the goal was to retrieve a single ranked list while avoiding redundancy and thus maximizing the diversity of search results. Our experiments on the two datasets led to several interesting observations about the relative performance of existing novelty-based retrieval approaches like redundancy filtering and Maximum Marginal Relevance (MMR) and the proposed nugget-based method. We showed how the effectiveness of various features used for novelty or diversity-based ranking depend on the nature of the dataset and information needs. We also analyzed the correlation of EGU with two other commonly used metrics: $\alpha-$NDCG, and Prec@20, which highlighted a key difference between them and EGU, which is that EGU can seamlessly extend to evaluation of session-based retrieval, while $\alpha-$NDCG and Prec@20 are designed for evaluation of single ranked lists.

On the TDT dataset, our proposed nugget-based method led to an improvement of 15% in terms of EGU over the baseline approach with no novelty detection and an improvement of 7% over an existing novelty-based method, *i.e.*, redundancy filtering. On the ClueWeb09 dataset, our proposed nugget-based method led to an improvement of 14% in terms of EGU over the baseline approach and an improvement of 5% over redundancy filtering.

7.2 Contributions

The technical contributions of this thesis are:

1. **A framework for measuring retrieval performance with respect to relevance and novelty of information across one or more ranked lists.** Our evaluation metric goes beyond current evaluation measures for retrieval performance by flexibly modeling the user's desire for relevance and novelty, based on a probabilistic user model that seamlessly extends to multiple ranked lists. (Chapter 3)

2. **A testbed based on the TDT4 dataset comprising 120 queries, and answer keys in the form of nuggets with corresponding "nugget-matching rules".** This testbed enables more realistic evaluation and comparison of various types of retrieval systems (*e.g.*, adhoc retrieval, adaptive filtering, information distillation) in terms of relevance and novelty (Chapter 4).

3. **Mathematical and empirical analysis of the performance of approximate algorithms for finding optimal ranked lists with respect to the proposed measure.** Our analysis provides new insights into the performance of approximate algorithms for optimizing nugget-based performance measures under different conditions. (Chapter 5)

4. **An approach for directly optimizing the proposed measure through explicit modeling of nuggets and learning from user feedback in the presence of non-independent utility of documents.** We explored the most effective ways for novelty-based ranking based on explicit modeling of nuggets. We also proposed a new approach for learning from user feedback in the presence of non-independent utility of documents, which goes beyond traditional relevance feedback techniques that assume independent utility. (Chapter 6)

5. **Experiments on three retrieval scenarios using two datasets.** To demonstrate the effectiveness of the proposed framework on a variety of retrieval tasks, we conducted experiments on three retrieval scenarios: (i) information distillation on the TDT dataset, (ii) session-based retrieval with long-lasting queries on the TDT dataset, and (iii) diversity-based retrieval on the ClueWeb09 dataset. These

experiments led to better understanding of the behavior of the proposed measure, EGU, compared to other common measures like $\alpha-$NDCG and precision, and also new insights on how the effectiveness of various features (words, named entities, latent topics, etc.) used for novelty detection depends on the nature of the dataset and information needs.

7.3 Future Work

While novelty detection has been studied for many years as part of adaptive filtering, the evaluation and optimization of novelty and diversity-based ranking in the context of ranked retrieval has only recently received major interest from the IR research community. The Diversity Task of the TREC Web Tracks in 2009 and 2010 represent a promising start in this direction. This thesis also furthers the state of the art by developing a strong foundation for evaluation and optimization of retrieval systems. The proposed evaluation framework is very modular and can easily accommodate new definitions of nuggets, or new user browsing models based on advances in understanding user behaviors by other researchers. The proposed optimization framework also allows plugging in new features as surrogates for nuggets that can lead to better performance on specific retrieval tasks. Hence, there are many directions in which this thesis work can be extended. Moreover, novelty and diversity-based ranking is an interesting problem in itself, and there are many open challenges and unexplored subproblems that deserve attention in the near future:

More accurate user modeling. An important part of our evaluation measure is the probabilistic model of user browsing behavior. In this thesis, we restricted attention to two models that posit that users stop reading at some point either due to satisfaction or lack of persistence (*i.e.*, abandonment). However, these models do not take the query into account: The browsing behavior of the user may depend on nature or intent of the query. A simple example is navigational queries, where users are very likely to stop as soon as they find the one correct result, whereas in the case of information queries, users may read many documents before they are satisfied. Therefore, a more realistic user browsing model should take the nature of the query into account.

Modeling ambiguous information needs. Our nugget-based model of utility is appropriate for modeling system performance on informational queries: We posit that the utility of a document is the sum of the (marginal) utilities of individual nuggets. However, many queries on the web are ambiguous. An appropriate model to capture such information needs would impose a probability distribution over the set of nuggets to reflect the likelihood of the average user being interested in each of them. Such an approach was taken by Agrawal et al. [2009], who adapt popular measures like MAP and NDCG to account for the uncertainty associated with the true intention of the users.

EGU can similarly be modified to take ambiguous query intents into account.

Learning to rank for novelty-based ranking. There has been increasing interest in applying machine learning techniques to optimize ranking functions. However, such efforts have limited their focus on relevance-based ranking. The main challenge in applying these techniques to novelty or diversity-based ranking is that the learning algorithm must model the non-independent utility of documents, unlike current learning to rank formulations for relevance-based ranking that can be effectively reduced to standard classification and regression problems. One approach for applying learning to rank to novelty-based ranking would be to cast it as a structured learning problem [Bakir et al., 2007]. Such a formulation would also enable directly finding a diversified ranked list, as opposed to the two-step (relevance ranking followed by novelty-based re-ranking) process that we have used in this thesis, and has also been used by almost all approaches to diversity-based ranking so far [Carbonell and Goldstein, 1998; Yue and Joachims, 2008; Agrawal et al., 2009].

Detecting the need for novelty-based ranking. Our evaluation and optimization framework includes a parameter for controlling the user's tolerance towards redundancy, which might not be known in advance. In fact, the user's expectation of novelty and diversity might depend on the intent of her query. Certain information needs like product reviews might warrant more repetition, so that the user can perform fact-checking and assess the general consensus and popularity of product features. Other information needs like finding the Oscar nominees warrant no repetition since duplicate results in this case are likely to provide no additional utility to the user. One approach would be to use machine learning approaches that take the query class into account to determine the appropriate level of redundancy in the results. Another approach would be to leverage user's feedback (*e.g.*, clicks on various search results) to dynamically adjust the level of redundancy in subsequent interactions in the search session, in case such feedback is available.

Interdependent nuggets across information needs. In our framework, we have implicitly assumed that nuggets are independent across different information needs. In other words, while we have accounted for redundancy of information across multiple ranked lists that are part of a single search session, we have not allowed for the possibility that a user might be simultaneously engaged in multiple interrelated search sessions with the retrieval system. Such multitasking behavior has been commonly observed in user studies as well as search logs of web search engines [Spink et al., 2002; Ozmutlu et al., 2003; Spink et al., 2006]. Such simultaneous interactions with the search engine might lead to redundancy across different search sessions. To accurately model the utility obtained by the user in such scenarios, we must relax the nugget independence assumption across sessions and calculate the redundancy of nuggets for the entire user interaction, instead of individual sessions.

Validation of the Utility Model. Our framework in Chapter 3 is based on a new model that makes certain assumptions about how users browse one or more ranked lists and accrue utility by reading each document or passage. Certain parts of the model are supported by empirical justifications in literature, *e.g.*, the geometric distribution over stopping positions in a ranked list is supported by an eye-tracking user study conducted by Joachims et al. [2005]. However, in this thesis we did not fully investigate how well our user model overall reflects true utility. One approach for validating the model is to conduct user studies to assess the correlation of `EGU` scores with subjective measures of performance as elicited from human subjects interacting with real search engines. Such an approach would be similar to other attempts in literature to understand how well common metrics like Discounted Cumulative Gain (DCG) and precision correlate with users' preferences [Al-Maskari et al., 2007; Sanderson et al., 2010]. Another approach, which is less direct, but less expensive and time-consuming, is to assess the correlation of `EGU` with implicit measures of satisfaction, *i.e.*, users' clicks observed on search results in a deployed system. Such a click-metrics–based approach has also been used in literature to evaluate new measures like Expected Reciprocal Rank [Chapelle et al., 2009].

Appendix A

Miscellaneous Details

A.1 Normalized Geometric Distribution

In Section 3.2.1.1, we looked at the persistence model, which uses a geometric distribution to model the stopping probability of users. We proposed the truncated geometric distribution as the correct choice, which assigns the remaining probability mass to the last document in a ranked list. Let us also look at the normalized geometric distribution, which seems natural at first, but leads to non-intuitive behavior of the evaluation measure.

The normalized geometric distribution re-distributes the probability mass over the length of the system-produced ranked list. Thus, the stopping probability distribution for the k^{th} ranked list can be expressed as:

$$\Pr(S_k = s) = \frac{(1-p)^{s-1}p}{1-(1-p)^{|l_k|}} \tag{A.1}$$

The normalized geometric distribution corresponds to the behavior of users whose decision to stop—even at early ranks—is influenced by the length of the ranked list.

However, this leads to non-intuitive results, as we will show next. Consider a ranked list of three documents with the following gain and cost values (the first three columns):

Document	Gain	Cost	Normalized		Truncated	
			$P(S\|\ell=2)$	$P(S\|\ell=3)$	$P(S\|\ell=2)$	$P(S\|\ell=3)$
d_1	10	1	0.56	0.41	0.20	0.20
d_2	8	1	0.44	0.33	0.80	0.16
d_3	0	1	–	0.26	–	0.64
Expected Utility =			12.1	12.87	14.6	13.96

Assume that the system needs to decide between showing the first two vs. the first three documents. The expected utility values that would be obtained for these two decisions when the normalized and truncated geometric distributions are used for modeling stopping behavior are also shown in the table (columns 4–5 and 6–7, respectively).

Note that the third document has zero gain and unit cost, *i.e.*, it has a negative marginal utility. If the normalized geometric distribution is used, the system receives a higher expected utility score of 12.87 if it decides to return three documents, as opposed to a score of 12.1 if it returns the first two documents. This behavior is unreasonable, since the system would have an incentive to append useless documents to its ranked list. In other words, when a normalized geometric distribution is used for modeling stopping behavior, adding another document shifts the probability mass from the top ranks to the later ranks, thus unrealistically increasing the expected utility of a longer ranked list.

In case of the truncated geometric distribution, we get the reasonable behavior where returning three documents results in expected utility of 13.96, which is lower than that of returning two documents, *i.e.*, 14.6. Thus, the truncated geometric distribution leads to the intuitive behavior of the metric where the system cannot increase its score by adding a document with negative marginal utility to the ranked list. Therefore, we use the truncated geometric distribution in all our experiments.

A.2 Sample Queries in the Extended TDT Dataset

As described in Chapter 4, we extended the Topic Detection and Tracking (TDT) dataset with new information needs that reflect the retrieval scenario for an information analyst. For each query, we identified a set of nuggets and created corresponding nugget-matching rules.

We have listed a few sample information needs, and their corresponding nuggets and nugget-matching rules below:

- (Singapore Airlines crash: history of problems with aircraft)
  - Nugget 1: No major accident
    `(first & air & crash) OR (first & aircraft & disaster)`
  - Nugget 2: Great reputation of aircraft carrier
    `(good & flying & record) OR (safest & carrier)`
    `OR (prestigious & carrier)`
  - Nugget 3: Modern aircraft
    `(modern & aircraft) OR (new & passenger & aircraft)`
- (Singapore Airlines crash: information about black box)
  - Nugget 1: Mention of black box
    `(black & box) OR (cabin & voice & recorder)`
    `OR (cockpit & record & recovered)`
- (UN Climate Conference: topic of conference)
  - Nugget 1: Kyoto
    `(ratify & kyoto)`
  - Nugget 2: Global warming
    `(carbon) OR (emission) OR (rise & temperature) OR`
    `(climate & change) OR (global & warming)`
    `OR (reduce & pollution)`
- (UN Climate Conference: number of participants)
  - Nugget 1: 180 countries
    `(180 & countries) OR (180 & states)`
- (UN Climate Conference: position of United States)
  - Nugget 1: Carbon credits issue
    `(carbon & credits) OR (emissions & credits)`
    `OR (buy & emissions) OR (emission & trading)`
  - Nugget 2: Stance in general
    `(conditional & ratification) OR (america & refusal)`
    `OR (american & stance) OR (bush & not & convinced) OR (veto)`
- (UN Climate Conference: objections from other countries)

  - Nugget 1: European opposition
    `(europeans & reject) OR (europeans & officials & criticize) OR (europe & adamant) OR (EU & kyoto)`

The full set of information needs, nuggets, and nugget-matching rules are available for download at http://nyc.lti.cs.cmu.edu/downloads/etdt4.tar.gz.

References

Agarwal, S., Cortes, C., and Herbrich, R., editors (2005). *Proceedings of the NIPS 2005 Workshop on Learning to Rank*. (Section 2.3.1).

Agichtein, E., Brill, E., and Dumais, S. (2006). Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26. ACM. (Section 2.5.1.2).

Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM. (Sections 1, 2.4.2, 2.4.2, 3.4, 5, 5.1.2, 5.7, 5.7, 6.1, 6.6, 7.3, and 7.3).

Al-Maskari, A., Sanderson, M., and Clough, P. (2007). The relationship between IR effectiveness measures and user satisfaction. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 773–774. ACM. (Section 7.3).

Allan, J. (1996). Incremental relevance feedback for information filtering. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 270–278. ACM. (Section 2.2.2).

Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. (1998). Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA broadcast news transcription and understanding workshop*, volume 1998. (Sections 1 and 2.4).

Allan, J., Wade, C., and Bolivar, A. (2003). Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 314–321. ACM New York, NY, USA. (Sections 2.4, 3.6, and 6.1).

Arvola, P., Kekäläinen, J., and Junkkari, M. (2010). Expected reading effort in focused retrieval evaluation. *Information Retrieval*, pages 1–25. (Section 3.6).

Bakir, G., Hofmann, T., and Schölkopf, B. (2007). *Predicting structured data*. The MIT Press. (Section 7.3).

Bateman, J. (1998). Changes in Relevance Criteria: A Longitudinal Study. In *Proceedings of the ASIS Annual Meeting*, volume 35, pages 23–32. (Section 2.1).

Bates, M. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424. (Sections 1 and 2.2.1.1).

Blei, D., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022. (Sections 6.2.2, 6.2.2.2, 6.3.3.4, and 6.3.3.4).

Broder, A. (2002). A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM. (Section 2.2.1).

Buckley, C. and Voorhees, E. (2005). Retrieval system evaluation. *TREC: Experiment and Evaluation in Information Retrieval*, pages 53–75. (Sections 1 and 2.5.1.2).

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to rank using gradient descent. In *International Conference on Machine Learning*, volume 22, page 89. (Section 2.3.1).

Callan, J. (1998). Learning while filtering documents. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 224–231. ACM. (Section 2.2.2).

Caputo, A., Basile, P., and Semeraro, G. (2009). Boosting a Semantic Search Engine by Named Entities. In *Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*, page 250. Springer. (Section 6.2.2).

Carbonell, J. and Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM New York, NY, USA. (Sections 1, 2.4.2, 6.1, 6.3.3.2, 6.6, and 7.3).

Carbonell, J., Yang, Y., Lafferty, J., Brown, R., Pierce, T., and Liu, X. (1999). CMU report on TDT-2: Segmentation, Detection and Tracking. In *Broadcast News Workshop'99 Proceedings*, page 117. Morgan Kaufmann Pub. (Section 2.4).

Carterette, B. (2009). An Analysis of NP-Completeness in Novelty and Diversity Ranking. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory*, page 211. Springer. (Sections 1 and 5.7).

Carterette, B., Pavlu, V., Kanoulas, E., Aslam, J., and Allan, J. (2009). If I Had a Million Queries. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pages 288–300. Springer. (Section 2.5.1.2).

Chapelle, O., Metzler, D., Zhang, Y., and Grinspan, P. (2009). Expected Reciprocal Rank for Graded Relevance. *CIKM*. (Sections 3.6, 3.6, and 7.3).

Chapelle, O. and Zhang, Y. (2009). A dynamic bayesian network click model for web search ranking. In *Proceedings of the 18th international conference on World wide web*, pages 1–10. ACM. (Section 3.2.1.2).

Church, R. and ReVelle, C. (1974). The maximal covering location problem. *Papers in regional science*, 32(1):101–118. (Section 5.1.1).

Clarke, C., Craswell, N., and Soboroff, I. (2009). Overview of the TREC 2009 web track. (Sections 3.1, 4, 4.1, 4.1, and 6.3.1.2).

Clarke, C., Kolla, M., Cormack, G., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–666. ACM New York, NY, USA. (Sections 2.5.2.2, 2.5.2.2, 3.6.1, 4, and 4.1).

Cleverdon, C. (1993). On the inverse relationship of recall and precision. *Journal of Documentation*, 28(3):195–201. (Section 2.5.1.1).

Cooper, W. (1968). Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, 19(1):30–41. (Section 3.6).

Cornuejols, G., Fisher, M., and Nemhauser, G. (1977a). Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810. (Sections 5.3 and 5.4).

Cornuejols, G., Fisher, M., and Nemhauser, G. (1977b). On the uncapacitated location problem. *Studies in integer programming*, page 163. (Section 3.5.2).

Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B. (2008). An experimental comparison of click position-bias models. In *Proceedings of the international conference on Web search and web data mining*, pages 87–94. ACM. (Section 3.2.1.2 and 3.2.1.2).

Dang, H., Lin, J., and Kelly, D. (2006). Overview of the TREC 2006 question answering track. *TREC 2006*. (Sections 1 and 3.1).

Dou, Z., Chen, K., Song, R., Ma, Y., Shi, S., and Wen, J. (2009). Microsoft Research Asia at the Web Track of TREC 2009. *TREC 2009 Proceedings*. (Sections 6.2.2 and 6.4.2).

Dupret, G. and Liao, C. (2010). A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 181–190. ACM. (Section 3.6).

El-Arini, K., Veda, G., Shahaf, D., and Guestrin, C. (2009). Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298. ACM New York, NY, USA. (Section 6.6).

Feige, U. (1998). A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652. (Section 5.3).

Finkel, J., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics. (Section 6.3.3.4).

Fiscus, J. and Doddington, G. (2002). Topic detection and tracking evaluation overview. *Topic detection and tracking: event-based information organization*, pages 17–31. (Section 4.3).

Fiscus, J. and Wheatley, B. (2004). Overview of the TDT 2004 Evaluation and Results. *TDT Workshop. Dec*, pages 2–3. (Sections 2.2.2, 2.4.1, and 3.1.2).

Ge, J., Huang, X., and Wu, L. (2003). Approaches to Event-Focused Summarization Based on Named Entities and Query Words. *DUC 2003 Workshop on Text Summarization*. (Section 2.7).

Goldstein, J., Mittal, V., Carbonell, J., and Callan, J. (2000a). Creating and evaluating multi-document sentence extract summaries. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 165–172. ACM. (Sections 1 and 2.4.2).

Goldstein, J., Mittal, V., Carbonell, J., and Kantrowitz, M. (2000b). Multi-document summarization by sentence extraction. In *NAACL-ANLP 2000 Workshop on Automatic summarization*, pages 40–48. Association for Computational Linguistics. (Section 2.7).

Gollapudi, S. and Sharma, A. (2009). An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*, pages 381–390. ACM New York, NY, USA. (Sections 5.1.2 and 6.6).

Greisdorf, H. (2003). Relevance thresholds: a multi-stage predictive model of how users evaluate information. *Information Processing & Management*, 39(3):403–423. (Section 2.4.2).

Hakkani-Tur, D., Tur, G., and Levit, M. (2007). Exploiting Information Extraction Annotations for Document Retrieval in Distillation Tasks. In *Proceedings of International Conference on Spoken Language Processing (Interspeech) Antwerp, Belgium*. (Sections 1, 2.2.3, and 4.2).

Hennig, L. and Labor, D. (2009). Topic-based multi-document summarization with probabilistic latent semantic analysis. In *International Conference on Recent Advances in Natural Language Processing (RANLP)*. (Section 2.7).

Hersh, W. and Over, P. (2000). TREC-8 interactive track report. In *Proceedings of the 8th Text REtrieval Conference (TREC-8)*. (Sections 3.1 and 4).

Hofmann, T. (1999). Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. (Section 6.2.2).

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446. (Sections 1, 2.5.1.2, 2.5.2.2, and 3.6.1).

Järvelin, K., Price, S., Delcambre, L., and Nielsen, M. (2008). Discounted cumulated gain based evaluation of multiple-query IR sessions. In *Proceedings of the 30th European Conference on Information Retrieval*. (Sections 1, 3.6, and 3.6.1).

Joachims, T. (2002). Evaluating retrieval performance using clickthrough data. In *Proceedings of the SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval*, pages 12–15. (Section 3.6).

Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM. (Sections (document), 1, 3.2.1.1, 3.2, and 7.3).

Just, M. and Carpenter, P. (2002). A theory of reading. *Psycholinguistics: critical concepts in psychology*, 87(4):365. (Section 3.2.1.1).

Kent, A., Berry, M., and Perry, J. (1954). Machine literature searching II. Problems in indexing for machine searching. *American documentation*, 5(1):22–25. (Section 2.5.1.1).

Keskustalo, H., Jarvelin, K., and Pirkola, A. (2008). Evaluating the effectiveness of relevance feedback based on a user simulation model: effects of a user scenario on cumulated gain value. *Information Retrieval*, 11(3):209–228. (Section 3.6).

Krause, A. (2008). *Optimizing Sensing*. PhD thesis, Carnegie Mellon University. (Section 5.7).

Krause, A. and Guestrin, C. (2005). Near-optimal nonmyopic value of information in graphical models. In *Proc. of Uncertainty in AI*. (Section 5.7).

Kumaran, G. and Allan, J. (2004). Text classification and named entities for new event detection. *Proceedings of the 27th annual international conference on Research and developement in information retrieval*, pages 297–304. (Section 6.2.2).

Lad, A. and Yang, Y. (2007). Generalizing from relevance feedback using named entity wildcards. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 721–730. (Section 6.3.3.3).

Lad, A. and Yang, Y. (2010). Learning to Rank Relevant and Novel Documents through User Feedback. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. (Sections 6.1 and 1).

Lafferty, J. and Zhai, C. (2003). Probabilistic relevance models based on document and query generation. *Language modeling for information retrieval*, 13:1–10. (Section 2.3.1).

Lehmann, E. and Romano, J. (2005). *Testing statistical hypotheses*. Springer Verlag. (Section 7).

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 429. ACM. (Section 5.7).

Li, Z., Chen, F., Xing, Q., Miao, J., Xue, Y., Zhu, T., Zhou, B., Chen, R., Liu, Y., Zhang, M., et al. (2009). THUIR at TREC 2009 Web Track: Finding Relevant and Diverse Results for Large Scale Web Search. (Section 6.4.2).

Lin, H. and Bilmes, J. (2010). Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920. Association for Computational Linguistics. (Sections 5 and 5.7).

Lin, H., Bilmes, J., and Xie, S. (2010). Graph-based submodular selection for extractive summarization. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pages 381–386. IEEE. (Section 5.7).

Lin, J. and Demner-Fushman, D. (2005). Automatically evaluating answers to definition questions. *Proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, pages 931–938. (Sections 1, 3.1, and 4.5).

Mani, I., House, D., Klein, G., Hirschman, L., Obrst, L., Firmin, T., Chrzanowski, M., and Sundheim, B. (1998). The TIPSTER SUMMAC text summarization evaluation: Final report. *The MITRE Corporation, report number MTR 98W0000138*. (Section 2.7).

Manning, C., Raghavan, P., Sch
"utze, H., and Corporation, E. (2008). *Introduction to information retrieval*, volume 1. Cambridge University Press. (Section 2.5.1.2).

Marton, G. and Radul, A. (2006). Nuggeteer: automatic nugget-based evaluation using descriptions and judgements. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 375–382. (Sections 1, 3.1, and 4.5).

Meij, E., He, J., Weerkamp, W., de Rijke, M., and (NETHERLANDS), A. U. (2009). Topical Diversity and Relevance Feedback. (Section 6.4.2).

Minka, T. (2003). A comparison of numerical optimizers for logistic regression. *Unpublished draft*. (Section 6.2.4).

Mizzaro, S. (1998). Relevance: The whole history. *Historical studies in information science*, pages 221–244. (Sections 2.1 and 2.4.2).

Moffat, A. and Zobel, J. (2008). Rank-biased precision for measurement of retrieval effectiveness. (Sections 3.2.1.1 and 3.6).

Nemhauser, G., Wolsey, L., and Fisher, M. (1978). An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294. (Sections 5.1 and 5.1).

Ozmutlu, S., Ozmutlu, H., and Spink, A. (2003). A study of multitasking Web search. (Section 7.3).

Prager, J., Brown, E., Coden, A., and Radev, D. (2000). Question answering using predictive annotation. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR2000)*, pages 184–191. (Section 6.2.2).

Radlinski, F., Bennett, P., Carterette, B., and Joachims, T. (2009). Redundancy, Diversity and Interdependent Document Relevance. In *ACM SIGIR Forum*. (Section 2.6).

Radlinski, F., Kleinberg, R., and Joachims, T. (2008a). Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on Machine learning*, pages 784–791. ACM New York, NY, USA. (Section 6.6).

Radlinski, F., Kurup, M., and Joachims, T. (2008b). How does clickthrough data reflect retrieval quality? In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 43–52. ACM. (Section 3.6).

Raghavan, V., Bollmann, P., and Jung, G. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems (TOIS)*, 7(3):205–229. (Section 2.5.1.1).

Robertson, S. (1977). The probability ranking principle in IR. *Journal of documentation*, 33(4):294–304. (Sections 2.3.1 and 5.1.2).

Robertson, S. (2008). A new interpretation of average precision. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 689–690. ACM New York, NY, USA. (Sections 3.2.1.2 and 3.6).

Robertson, S. and Belkin, N. (1993). Ranking in principle. *Journal of Documentation*, 34(2):93–100. (Section 2.3.1).

Robertson, S., Maron, M., and Cooper, W. (1982). Probability of relevance: a unification of two competing models for document retrieval. *Inf. Technol*, pages 1–21. (Section 2.3.1).

Robertson, S. and Soboroff, I. (2002). The TREC-10 Filtering Track Final Report. *Proceeding of the Tenth Text REtrieval Conference (TREC-10)*, pages 26–37. (Sections 2.2.2, 2.4.1, and 3.1.2).

Robertson, S. and Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241. Springer-Verlag New York, Inc. New York, NY, USA. (Section 2.3.1).

Rose, D. and Levinson, D. (2004). Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19. ACM. (Section 2.2.1).

Sakai, T. (2005). Ranking the NTCIR systems based on multigrade relevance. *Information Retrieval Technology*, pages 251–262. (Section 2.5.1.2).

Sakai, T. and Robertson, S. (2008). Modelling A User Population for Designing Information Retrieval Metrics. In *Proceedings of the Second Workshop on Evaluating Information Access*. (Section 3.6).

Salton, G. (1971). The SMART retrieval system—experiments in automatic document processing. (Sections 2.3.1 and 2.5.1.1).

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24(5):513–523. (Section 2.3.1).

Salton, G. and McGill, M. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc. New York, NY, USA. (Section 2.3.1).

Salton, G., Wong, A., and Yang, C. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620. (Section 2.3.1).

Sanderson, M., Paramita, M., Clough, P., and Kanoulas, E. (2010). Do user preferences and evaluation measures line up? In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 555–562. ACM. (Section 7.3).

Shimizu, T. and Yoshikawa, M. (2007). A ranking scheme for XML information retrieval based on benefit and reading effort. *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 230–240. (Section 3.6).

Smucker, M., Allan, J., and Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632. ACM. (Section 7).

Soboroff, I. (2004). Overview of the TREC 2004 novelty track. In *Proceedings of the 13th Text REtrieval Conference (TREC 2004)*. (Sections 1, 2.4, and 2.5.2.1).

Spink, A., Ozmutlu, H., and Ozmutlu, S. (2002). Multitasking information seeking and searching processes. *Journal of the American Society for Information Science and Technology*, 53(8):639–652. (Sections 1, 2.2.1.1, and 7.3).

Spink, A., Park, M., Jansen, B., and Pedersen, J. (2006). Multitasking during Web search sessions. *Information Processing and Management*, 42(1):264–275. (Sections 2.2.1.1 and 7.3).

Srihari, R. and Li, W. (2000). A question answering system supported by information extraction. *Proceedings of the sixth conference on Applied natural language processing*, pages 166–172. (Section 6.2.2).

Strohman, T., Metzler, D., Turtle, H., and Croft, W. (2004). Indri: A language model-based serach engine for complex queries. *Proceedings of the International Conference on Intelligence Analysis*. (Section 6.3.3.1).

Toutanova, K., Brockett, C., Gamon, M., Jagarlamudi, J., Suzuki, H., and Vanderwende, L. (2007). The pythy summarization system: Microsoft research at duc 2007. In *the proceedings of Document Understanding Conference*. (Section 2.7).

Van Rijsbergen, C. (1979). *Information Retrieval*. Butterworth-Heinemann Newton, MA, USA. (Section 2.5.1.1).

Voorhees, E. (2003). Overview of the TREC 2003 Question Answering Track. *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*. (Sections 1, 3.1, and 1).

White, J., Hunter, D., and Goldstein, J. (2008). Statistical evaluation of information distillation systems. *Proceedings of the Sixth International Language Resources and Evaluation*, 8. (Sections 1, 2.2.3, 4, and 4.2).

Wolsey, L. (1982). Maximising real-valued submodular functions: Primal and dual heuristics for location problems. *Mathematics of Operations Research*, pages 410–425. (Section 5.2).

Yang, Y., Carbonell, J., Allan, J., and Yamron, J. (1997). Topic detection and tracking: Detection-task. In *proceedings of the Workshop of Topic Detection and Tracking*. (Sections 1 and 2.4).

Yang, Y. and Lad, A. (2009). Modeling Expected Utility of Multi-session Information Distillation. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory*, page 175. Springer. (Sections 4.4.1, 6.1, and 6.3.3.3).

Yang, Y., Lad, A., Lao, N., Harpale, A., Kisiel, B., and Rogati, M. (2007). Utility-based information distillation over temporally sequenced documents. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 31–38. (Sections 1, 2.2.2, 2.2.3, 2.5.2.2, 2.5.2.2, 4.2, 4.2, 4.4.2, and 6.1).

Yang, Y., Zhang, J., Carbonell, J., and Jin, C. (2002). Topic-conditioned novelty detection. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 688–693. (Section 2.5.2.1).

Yue, Y. and Joachims, T. (2008). Predicting diverse subsets using structural SVMs. In *Proceedings of the 25th international conference on Machine learning*, pages 1224–1231. ACM New York, NY, USA. (Sections 6.2.2, 6.6, and 7.3).

Zhai, C. (2002). *Risk minimization and language modeling in text retrieval*. PhD thesis, Carnegie Mellon University. (Sections 6.1 and 6.6).

Zhai, C., Cohen, W., and Lafferty, J. (2003). Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 10–17. (Sections 1, 1, 2.4.2, 2.4.2, 2.5.2.2, 2.7, 3.4, 3.6.1, 5.1.2, 5.7, 6.1, and 6.6).

Zhai, C. and Lafferty, J. (2006). A risk minimization framework for information retrieval. *Information Processing and Management*, 42(1):31–55. (Section 2.3.1).

Zhang, Y., Callan, J., and Minka, T. (2002). Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88. ACM New York, NY, USA. (Sections 2.2.2, 2.4.1, 2.4.1, 2.4.1, 2.4.1, 6.1, 6.1, 6.3.3.3, and 6.6).