Novelty and Diversity in Retrieval Evaluation

by

Maheedhar Kolla

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Queries submitted to search engines rarely provide a complete and precise description of a user's information need. Most queries are ambiguous to some extent, having multiple interpretations. For example, the seemingly unambiguous query "tennis lessons" might be submitted by a user interested in attending classes in her neighborhood, seeking lessons for her child, looking for online videos lessons, or planning to start a business teaching tennis. Search engines face the challenging task of satisfying different groups of users having diverse information needs associated with a given query. One solution is to optimize ranking functions to satisfy diverse sets of information needs. Unfortunately, existing evaluation frameworks do not support such optimization. Instead, ranking functions are rewarded for satisfying the most likely intent associated with a given query.

In this thesis, we propose a framework and associated evaluation metrics that are capable of optimizing ranking functions to satisfy diverse information needs. Our proposed measures explicitly reward those ranking functions capable of presenting the user with information that is novel with respect to previously viewed documents. Our measures reflects quality of a ranking function by taking into account its ability to satisfy diverse users submitting a query.

Moreover, the task of identifying and establishing test frameworks to compare ranking functions on a web-scale can be tedious. One reason for this problem is the dynamic nature of the web, where documents are constantly added and updated, making it necessary for search engine developers to seek additional human assessments. Along with issues of novelty and diversity, we explore one approximate approach to compare different ranking functions by overcoming the problem of lacking complete human assessments. We demonstrate that our approach is capable of accurately sorting ranking functions based on their capability of satisfying diverse users, even in the face of incomplete human assessments.

Acknowledgements

First, I would like to thank my supervisors Olga Vechtomova and Charlie L.A. Clarke for their patience, support and guidance towards successful completion of my thesis. I truly appreciate their commitment, help and thank them for giving me an opportunity to explore different research topics. I would also like to thank my committee members Gordon V. Cormack, Mark Smucker and Mariano Consens for their feedback at various stages of my thesis. Also, a special thanks to Mark Smucker for his valuable feedback at the final stages of my thesis.

I would like to thank all my colleagues and friends for a wonderful time spent from being part of the PLG/IR lab. I would like to thank Jun Chen, Hani Khoshdel Nikkhoo, Adriel Dean-Hall, Thomas Lynam, Adam Roegiest, Robert Warren, Mohammed Ahmadi, Nomair Naeeem, Ghulam Lashari for a great time in the lab. I would like to thank Krzysztof Borowski, Azin Ashkan, Seyed Ali Ahmedzadeh, Richard Bilson, Roy Krischer, Mona Mojdeh, Gaurav Baruah, Bahareh Sarrafzadeh, John Akinyemi, Ashif Harji, Stefan Büttcher, Bradley M. Lushman, Peter C.K. Yeung, and Fiona So, for game nights, PLG BBQ nights, volleyball, dodgeball, curling, hiking, casino, camping trips and making me part of their celebrations.

Outside the lab, I am grateful to have met people from different departments and backgrounds. In particular, I would like to thank Aaditeshwar Seth, Nabeel Ahmed, Sarvagya Upadhyay, Jalaj Upadhyay, Srinath Reddy, Krishnam Raju Jampani, Penny Dorka, Alma Juarez-Dominguez, Mike Patterson. I would also like to thank Sandra Céspedes for all early morning games/gym sessions followed by thoughtful discussions over breakfast. In addition, I would like to thank the friendly staff at the Tim Hortons, Russ and Marta for their friendly chat and warm wishes.

I would like to thank Margaret Towell, Paula Zister, Wendy Rush and Jessica Miranda for their help in getting my paperwork in order.

Outside the university, I would like to thank all my friends who have supported me during my stay in Canada. In particular, I would like to thank Dheeraj Achra, Ankita Mukherjee Achra and their family, Nimeesh Kaushal and Deepika Rajian, and Naina Patel, for welcoming me as one of their own family member and supporting me during my stay in Canada.

I would like to thank my parents and family for their extended support during my education in Canada. I would like thank my brother Vivek Kolla for his continuing support and guidance throughtout my career.

A special thanks to Arun Krishnakumar, Akshay Kiran Singh, Ashif Harji, Bradley M. Lushman, Bahareh Sarrafzadeh and Sandra Céspedes, Dheeraj Achra for their support during my time in Waterloo.

to my mom, dad, family and to all well-wishers.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Search engines have become an integral part of our day-to-day lives. People rely on search engines to follow news, research topics of interest, download music videos, search for airline tickets, etc. When using a search engine, we assume that users have a task or purpose in mind, such as following news, and that they interact the search engine to help them find information to complete their task or purpose. Users submit search requests in the form of *queries*. Depending on their underlying task or purpose (also called *intent*), queries may be broadly classified into three groups (Broder [2002]):*informational*, *navigational* and *transactional*. Users submit informational queries with the purpose of finding comprehensive information about a given topic. For example, a user submitting the query "machine learning" may intend to find more information about that field of computer science. Users submit navigational queries for the purpose of finding a particular web page or site. For example, a user submitting the query "university of waterloo" may intend to find the homepage of that university [1]. Users submit transactional queries for the purpose of downloading a file, making a purchase, etc. For example, a user submitting the query "2014 world cup tickets" may intend to purchase tickets for that tournament.

In order to satisfy user requests, search engines must:

- maintain an *index* of all documents that may be useful to its users,
- implement a scoring function to determine the likelihood that a specific document satisfies a specific query, and
- present a subset of documents containing information that could help users complete their tasks.

Search engines transform user requests into a representation that can be used to compute matching scores for each of the document in a given collection. In this thesis, we focus on ranking

[1] `http://www.uwaterloo.ca`

functions and their evaluation. We assume that there exists a collection that contains at least one document that a user could be interested in for a given query. Ranking functions are evaluated on two aspects: *efficiency* and *effectiveness*. While efficiency of a ranking function measures the speed at which the ranking function retrieves results for a given query, effectiveness of a ranking function is a measure of extent to which the ranking function satisfies the perceived information need of an user.

In this thesis, we are concerned with effectiveness of ranking functions in context of web search. We briefly introduce few ranking functions developed to retrieve results for a given query (Section 1.1). In Section 1.2, we we discuss the challenges of satisfying users' information needs in context of web search. Following that, we discuss our thesis objectives,contributions and thesis outline.

1.1 Ranking Functions

Several ranking functions have been proposed to to identify a subset of documents from a collection that are likely to satisfy an underlying information need. Gerald Salton [1971] proposed one of the first ranking methods, the Vector Space Model (VSM), to score documents of a collection with respect to given query. For each document d_i of a collection, a corresponding vector representation is created with respect to terms from entire collection $(t_1 \dots t_k)$

$$Vec(\vec{d_i}, t_{1\dots k}) =< f(d_i, t_1), f(d_i, t_2), \dots f(d_i, t_k) >$$

where $f(d_i, t_k)$ is equal to weight attributed by term t_k towards document d_i. This function can be binary (0 for term absence and 1 for term presence) or weighted, (*tf·idf* values), as proposed by Spärck Jones [1988]:

$$tf \cdot idf(d_i, t_k) = tf(d_i, t_k) \cdot log\frac{N}{n_{t_k}} \tag{1.1}$$

where $tf(d_i, t_k)$ is equal to frequency of term t_k in document d_i, N is equal to number of documents in the collection, and n_k represents the number of documents in the collection containing the term t_k. In general, $log\frac{N}{n_{t_k}}$ is referred to as the *inverse document frequency (idf)* value for term t_k. It is indicative of how rare (or common) the term t_k is in a collection of N documents. Similarly, each query q_j submitted by a user is transformed into its corresponding vector form:

$$Vec(\vec{q_j}, t_{1\dots k}) =< q_{j,t_1}, \dots q_{j,t_k} > \tag{1.2}$$

Salton then measured similarity between a given <query,document> pair based on cosine

distance between their corresponding vectors as shown:

$$\text{score}(d_i, q_j) = \frac{\vec{q_j}.\vec{d_i}}{||\vec{q_j}||||\vec{d_i}||} \tag{1.3}$$

where the numerator represents the cross product of the two vectors. For a given query q_j, all documents in the collection are sorted in decreasing order of their score().

Spärck Jones et al. [2000] proposed the family of *Best Match*, or *BM*, ranking functions, which sort documents in decreasing likelihood of satisfying an underlying information need. For the BM25 function, the most famous member of the family, the similarity between a document d and a query q is defined in terms of a *relevance weight* (RW), computed as follows:

$$RW(d,q) = \sum_{i=1}^{n} \frac{TF_i * (k_1 + 1)}{k_1 * ((1-b) + b * \frac{|dl|}{|avdl|}) + TF_i} \cdot log\frac{N}{n_i} \tag{1.4}$$

where TF_i is equal to the frequency of term i in current document, $|dl|$ equal to length of the document d, and $|avdl|$ is equal to average length of documents of a collection. The values of constants k_1 and b are usually take the default values of 1.2 and 0.75 respectively. Robertson et al. [2004] recently proposed a variant of this ranking function, BM25F, which incorporates a term frequency value that depending on the location of a term the document. For example, in a web page, terms appearing in a `<title>..</title>` field are considered more important than other terms in the document and are weighed appropriately.

Ponte and Croft [1998] proposed the first of *language model* based approaches, where in documents are ordered in terms of their ability to generate user submitted query. For a given query Q, documents are sorted based on their ability to generate Q computed as

$$\hat{p}(Q|M_d) = \prod_{t \in Q} \hat{p}(t|M_d) - \prod_{t \notin Q} (1.0 - \hat{p}(t|M_d)) \tag{1.5}$$

where M_d represents language model inferred from document d. Given such model, Ponte and Croft obtained term weight $\hat{p}(t|M_d)$ by

$$\hat{p}(t|M_d) = \begin{cases} \hat{p}_{ml}(t,d)^{(1.0-\hat{R}_{t,d})} \times \hat{p}_{avg}(t)^{\hat{R}_{t,d}} & \text{if } tf_{(t,d)} > 0 \\ \frac{cf(t)}{size} & \text{else} \end{cases} \tag{1.6}$$

where $p_{ml}(t,d)$ is the maximum likelihood estimate of the probability of term t in a given document d computed as

$$p_{ml}(t,d) = \frac{tf_{(t,d)}}{dl_d} \tag{1.7}$$

where $tf_{(t,d)}$ is frequency of term t in document d, with document length dl_d. Ponte and Croft

factored in term distribution across all documents the term t occurs in

$$\hat{p}_{avg}(t) = \frac{(\sum_{d_{(t \in d)}} p_{ml}(t|M_d))}{df_t} \tag{1.8}$$

where df_t is document frequency of term t. Ponte and Croft factored in risk factor, $\hat{R}_{t,d}$, associated with a term t (for term weight)

$$\hat{R}_{t,d} = (\frac{1}{(1+\bar{f}_t)}) \times (\frac{\bar{f}_t}{(1+\bar{f}_t)})^{tf_{(t,d)}} \tag{1.9}$$

based on its average frequency, $(\bar{f}_t)$ of the term in documents it occurs in (i.e. $p_{avg}(t) \times dl_d$). In cases where a given term t is not present in a given document (i.e. $tf_{(t,d)} = 0$), Ponte and Croft computed term weight to be equal to probability of term in entire collection, i.e. $\frac{cf(t)}{size}$, where $cf(t)$ is total frequency of term t across the entire collection, and $size$ represents the size of the collection.

Currently, both BM25 and query likelihood scoring functions serve as *de facto* baseline ranking methods against which any new proposal should be compared.

Since majority of ranking functions are based on term match between queries and documents, there is a chance that potential useful documents are assigned low scores on account of the document missing an important keyword. One way to overcome such term mismatch problem is by expanding user submitted queries with terms that are likely to be frequent be contained in documents satisfying information need (associated with a query). These terms could be selected either explicitly, through user interaction or implicitly from top ranked documents.

Several implicit approaches have been proposed over the years to expand the initial queries, using *pseudo-relevant documents*. Across these approaches, the top r documents (usually 10 or 25) are assumed to be relevant. Using statistical models, researchers then select terms that occur frequently in this pseudo-relevant set when compared to the rest of the collection. These terms are then added to the original queries, with appropriate scaling. Rocchio [1971] proposed the first relevance feedback method that modifies the term weights in a given query vector so as to be able to retrieve documents that are closer (in distance) to the perceived information need. This information is used to improve the results by expanding the initial query $\vec{q_i}$ to obtain a expanded vector $\vec{q_i}^{'}$, which better reflects the user's information need. Similarly, Billerbeck and Zobel [2004], and Carpineto et al. [2001] proposed different pseudo-relevant approaches based on different term selection criteria.

An alternate stream of ranking methods have been developed using machine learning techniques in information retrieval. In these approaches, commonly known as *learning to rank* methods, a training set is used to learn features of finding documents that satisfies the information

need associated with a given query (e.g. RankNet proposed by Burges et al. [2005], RankSVM proposed by Joachims [2002]).

As mentioned before, evaluation of ranking functions is carried out with respect to two aspects: *effectiveness* and *efficiency*. As expected, there exists a trade-off between effectiveness and efficiency of a ranking function. Effectiveness of a ranking function is the measure of likelihood of satisfying an underlying information need. In general, such information need can be satisfied by simply showing a *url* that the user is looking for (navigational) or a set of documents (informational) that user could be interested in.

While designing a ranking function, researchers compare and evaluate ranking functions in terms of their ability to satisfy information needs associated with a query. Throughout history, ranking functions are evaluated or compared against each other through experimentation (Harman [1993]). Usually, such experimentation is carried out offline, where several ranking functions are compared in terms of satisfying certain information needs. Researchers make use of several test collections to compare different ranking functions of their effectiveness in satisfying a set of information needs.

1.2 Challenges of Web Search

In web search, queries are comprised of a small number of keywords (Jansen et al. [2000]), which may not accurately reflect the complexity of the underlying information need. A simple query like "*ups*" could refer to either "United Parcel Service" (the shipping company) or "Uninterruptable Power Supply" (power device) or "University of Puget Sound" (an educational institution), etc. Each interpretation could, in turn, be associated with a bundle of intents as shown below:

```
United Parcel Service, - company history, nearest drop off location, ...
Uninterrupted Power Supply - product pricing, purchase outlets, ...
University of Puget Sound - homepage, admission deadlines, ...
```

where a user's purpose behind entering the query "ups" could depend on her geographic location, task on hand, etc. It therefore becomes a non-trivial task for search engines to identify the intent behind user queries. Such uncertainty or *ambiguity* in user intent is prevalent in web search. Recently, Song et al. [2009] reported that around 16% of queries submitted on the web are *ambiguous*. In addition, it has been observed that humans disagree among themselves while trying to determine the intent underlying a given query, without additional contextual information. In an experiment to measure the effectiveness of a commercial search engine, Huffman and Hochster [2007] requested human assessors to map web queries to their underlying information

need. Huffman and Hochster observed that for queries indicating name of an organization, e.g. *"Fisher Price"*, humans agreed that the query has a navigational intent such as

"*take me to the homepage of the toy company named Fisher Price*".

However, when asked for the query "*red envelope*", humans identified two different intents :

- *take me to the homepage of a well known gift site Red Envelope*
- *meaning of red envelope or red packet used in Chinese society.*

Where the former symbolizes a user searching for one particular gift site store (i.e., a navigational intent), the latter represents a user interested to find out the significance of a "red envelope", with reference to Chinese culture (i.e., an informational intent). In addition, one could argue that some users, entering the query "*Fisher Price*", may be interested in knowing more about the company, or find a store in their neighborhood.

Under such ambiguity, user expectations of search engine would vary according to their inherent information need. For example, a user submitting the query *ups* with an intent to purchase an "*uninterrupted power supply*" device would not be interested in browsing through documents about the nearest shipping office of the *United Parcel Service*. In absence of a rich contextual information to disambiguate user queries, search engines need to *diversify* retrieved results in order to satisfy a given user (Chen and Karger [2006]). Returning to our example, let us compare two (hypothetical) sets of results compiled for the query "ups", shown in Table 1.1. We observe that results belonging to "Result Set 1" appear to satisfy the information needs of users entering query *ups* with the purpose of carrying out tasks related the shipping company "United Parcel Service". On the other hand, results belonging to "Result Set 2" appear to satisfy diverse users submitting the query "ups" with different intent. In developing a ranking function for the web search scenario, it is necessary to take into account the inherent ambiguity in user queries.

Rank	Result Set 1	Result Set 2
1	UPS: Tracking Information	Welcome to UPS
2	Welcome to UPS	UPS: Tracking Information
3	UPS Store Canada	Uninterruptable power supply - Wikipedia
4	UPS Calculate Time and Cost	University of Puget Sound
5	UPS.com	Always On – UPS systems

Table 1.1: Two hypothetical result sets compiled for the query *ups*

Another challenge in designing a web search engine is the presence of duplicates or near duplicates. Let us consider a set of document titles retrieved by a generic search engine for the query "UPS" (Table 1.2) [2]. As observed, documents listed from rank $5 \ldots 7$ appear to be

[2] dated May 2012.

duplicates or near duplicates of each other. Further (not shown in the table), the search engine managed to return contact information of all UPS stores, based on geo-location, on a map located to the right of the results. It is unsure as to the added benefits, if any, which would persuade user to browse through lower documents. In designing a ranking function for the web, it is necessary avoid showing redundant information to a user.

Rank	Titles
1	Welcome to UPS
2	Shipping,Freight, logistics .. from UPS
3	UPS- Wikipedia, the free encyclopedia
4	Uninterruptible power supply - Wikipedia ..
5	The UPS Store - Find the UPS store ...
6	The UPS Store ....
7	Find the UPS Store Canada - ..

Table 1.2: UPS example:Web results for the query UPS

To summarize, when designing a new ranking function, it is necessary to optimize ranking functions to

- Satisfy diverse intents for a given (ambiguous) query.
- Display novel information by not promoting redundant information.

1.3 Problem

However, existing frameworks for evaluation, used in comparison of off-line ranking function development **completely** ignore the notion of ambiguity. Existing frameworks, instead, are centered around the notion of "*one interpretation per request*", where in each query is bounded to one particular intent (subjectively assigned by human assessors). In case of ambiguous queries, like the query *ups*, it is either discarded from the test set or subjectively mapped (disambiguated) to its most likely intent as shown below :

Query: UPS.
Need: User would like to navigate to tracking page of United Parcel Service
Type: Navigational

In the example topic, query "ups" is mapped to one possible information need, explained in the "Need" field. In addition, request explicitly mentions the type of information need associated with the query. In this instance, the query "UPS", is associated with a *navigational* request.

Any ranking function that is optimized towards catering to this particular information need, would not be ideal for other needs. As it stands, ranking functions would only be optimized to cater to one fraction of user population submitting the query *ups*. Almost all ranking functions are optimized with respect to either *navigational* queries, users searching for homepage of an entity, or *informational* queries, users wishing to synthesize information spread across multiple sources (Broder [2002]). Recent investigations Büttcher et al. [2006] and Hawking et al. [2000] highlight the drawbacks of building test collections for specific evaluation purposes, where they observed that ranking functions optimized for one kind of queries are not optimal for other kinds of queries. Having multiple intents for a given query should be a norm in optimizing ranking functions.

Another important factor that needs to be considered in the evaluation of a ranking function is the presence of duplicate and near-duplicate documents across the web. Bernstein and Zobel [2005], while investigating the **.gov** collection ($\sim$ 426GB) found that around 17% of the documents, judged relevant (to user's information need) were exact or near duplicates. Under such circumstances, it is unclear as to the utility gained by a user from reading a exact or even a near duplicate of the one that she has just read. It is important to **not** encourage ranking methods to retrieve redundant documents for a given query, without adding any benefit to users.

In terms of rewarding novelty based methods, existing framework and measures built on top of those frameworks do not factor in the novelty of information a user may come across in a document. This limitation stems from the notion of independence being imposed on document importance, in measuring its utility towards satisfying user's information need. Although recognized in Soboroff and Harman [2005], it is usually not incorporated for its operational difficulty. The implications of optimizing ranking functions using existing methodologies can be illustrated by considering a hypothetical set of results, Set3, compiled for the query *ups* (Table 1.2). Under existing methods of evaluation, where the query *ups* is constrained to one particular information need (from above example), Set3 would be rated as the best possible result set. It is unclear as to the added benefit, if any, which would persuade user to browse through the documents ranked 2 and lower.

The importance of presenting *novel* information, in comparison with previously seen information, is not restricted to web search engines. It could benefit a user browsing through a news archive, while researching about a particular topic of interest. It could also benefit users within a given enterprise, who have information spread across different sources such as mailing lists, discussion forms, meeting memos etc.

As it stands, there exists no large scale test collection that allows researchers to evaluate and optimize ranking functions factoring in ambiguity of user queries and redundancy of information. Few works have highlighted the shortcomings of not considering the underlying ambiguity. Recently, Spärck Jones et al. [2007] aptly highlighted the need for such test framework to evaluate

web scale ranking functions. Spärck Jones et al. suggest that a query be associated with a "*bundle of intents*". For example, for the query "Java", such bundle of intents consists of

- Java as a coffee
- Java as a language
- Java as an island

which could then be assigned a probability (say $0.4, 0.4, 0.2$ for each intent respectively). In judging the utility of a document, Spärck Jones et al. then weigh each document in terms of its benefits with respect to each individual intent. Spärck-Jones et al. suggest to make use of online encyclopedia's such as Wikipedia [3] to identify the bundles of intents associated with a query. Spärck-Jones et al. conclude their discussion stating the importance of having such test collection to promote research into ranking functions that are capable of withstanding challenges of web search.

1.4 Thesis Objectives and Contributions

In this thesis, we aim to overcome the lack of a existing framework and test collections as explained in previous section. The key contributions of this thesis are as follows:

Ambiguity of Queries

In this thesis, we aim to propose a framework that supports possibility of a query being associated with multiple intents. Further, we distinguished queries into ambiguous and underspecified groups based on the extent of their uncertainty. From our definition, when submitting an underspecified query, users' information need could vary in terms of different aspects or subtopics (e.g.: some of the facets associated with the query "University of Waterloo" are "homepage of the university", "contact information", etc.) In terms of ambiguous queries, we refer to those queries that could be associated with different interpretations (e.g.: *ups*). In such queries, each interpretation could possible refer to a different entity. It can be safely assumed that user interested in one interpretation would not be interested in information needs associated with different interpretations. We made use of such distinction in deriving effectiveness measures (Chapter 3).

[3] `en.wikipedia.org`

Novelty of Information

Building on our proposed framework, each document is independently judged with respect to various information needs covered in that particular document. It alleviates the burden on human assessors without requiring them to subjectively assess the novelty of information covered in a document. We then discount the utility gained from going through a information already seen in previous (higher ranked) documents.

Effectiveness Measures:α-nDCG, NRBP

By incorporating the notion of novelty and diversity, we proposed two effectiveness measures α-nDCG (Clarke et al. [2008]) and NRBP (Clarke et al. [2009a]), that reflects

- Probability of a user finding novel information while browsing through a ranked list of documents.
- Extent to which diverse information needs can be satisfied from a given ranked list.

Our proposed evaluation measures reward those ranking functions that display information that is novel to the user.

1.5 Thesis Outline

In Chapter 2, we present a detailed description of existing methods and measures in order to measure the effectiveness of ranking functions. We introduce the reader to the paradigm used in evaluating ranking functions and assumptions made in comparing different ranking functions. Along with retrieval evaluation, we explain the work done in in related fields such as summarization and question answering, that provide the inspiration for our work.

In Chapter 3, we outline the fundamental principles behind our framework of evaluation. We define two kinds of queries based on the extent of uncertainty. We then define relevance of a document in a probabilistic manner, reflecting the extent to which the given document would cater to the average information needs associated with a given query.

In Chapter 4, we describe an experiment carried out to demonstrate the functioning of our framework and its associated measures, on a simulated test collection. We took an existing collection to compare few well-known ranking functions (one baseline and two pseudo-relevant ranking functions). We then detail a few measures that were built on top of our framework of evaluation. We then detail a large-scale evaluation framework undertaken by NIST to evaluate ranking functions in context of a web-scale collection.

In Chapter 5, we explore approximate methods to estimate the probability of a document being judged relevant. In the follow-up exercise undertaken, we train a classifier by sampling some existing relevance assessments. Using this trained classifier, we estimate the probability of relevance of an unjudged document. We then propose a method to estimate the effectiveness of ranked results, in terms of expected utility gained by a user browsing through a retrieved list containing both judged and unjudged documents. We conduct preliminary experiments across different existing test collections and report the findings. In chapter 6, we conclude our thesis and discuss future work.

Chapter 2

Retrieval Evaluation

In this chapter, we provide the reader with background information regarding the evaluation of ranking functions. In general, retrieval evaluation is carried out in an experimental fashion, where several methods are compared with respect to some pre-defined tasks. Comparisons are made with respect to the speed of computing the retrieval result,i.e., *efficiency*, and the quality of the result set computed,i.e., *effectiveness*. Effectiveness measures are concerned with the quality of retrieved results, reflecting the extent to which user's information needs would be satisfied by browsing the retrieved results. In this work, we focus on evaluating retrieval systems with respect to their effectiveness. In particular, we are interested in challenges in the evaluation of ranking functions in the context of web search.

2.1 Paradigm

In general, the paradigm for evaluating ranking functions is based on the following use-case scenario:

> The user is seeking information that is necessary to complete a task at hand. She transforms her information need into a language that could be easily understood by the search engine (i.e., a *query*). The search engine employs a ranking function to select a subset of documents from the collection and presents them to the user in order of decreasing probability of satisfying the user's information need, i.e., according to expected relevance. The user then examines the documents and rates each document (useful, partly useful, not useful) in terms of its utility towards completing her task.

Broadly, user information needs could be classified into either *navigational* or *informational* requests Broder [2002]. Users submitting navigational queries are interested in navigating to some

particular document source. For example, user entering the query "**homepage of university of waterloo**" is likely seeking `http://uwaterloo.ca`. In case of informational queries, users are interested in (possibly in-depth) information about a given topic (event, place, thing, person, etc.), which may or may not be present in one single document/source. For example, user entering the query "*marine vegetation*" could be seeking information on the types of marine vegetation, their processing, potential applications etc., which she might need to compile from a variety of sources.

Consider a document set of D comprising of 10 documents retrieved for the query *"marine vegetation"* (Figure 2.1), effectiveness measures are meant to reflect the extent to which a user's information need is satisfied by reading the set D. In computing the effectiveness of a ranking function, it is assumed that search engines rank and present the documents in decreasing order of their ability to satisfy the user's information need and that the user browses the retrieved results in a top-down fashion, until her information need is satisfied. Under such assumptions, ranking functions are rewarded for retrieving documents capable of satisfying user information needs as close to the top ranks as possible. Several measures exist to compare the effectiveness of ranking functions. The two straightforward effective measures are *precision* and *recall*. Precision at a given rank, is defined as the fraction of documents retrieved at that rank that are relevant (Eq. 2.1). Recall value at a particular rank is defined as the fraction of relevant documents ($|R|$) that are retrieved at a given rank n (Eq. 2.2).

$$\text{precision@n(p@}n) = \frac{\text{\# of relevant documents}}{\text{number of documents seen at given rank } (n)} \tag{2.1}$$

$$\text{recall@n(r@}n) = \frac{\text{\# of relevant documents}}{\text{total number of relevant documents } (R)} \tag{2.2}$$

Let us assume that in our example, the user browses through the top 10 documents and finds the documents at ranks 1, 3, 4, and 8 relevant to her information need. In this case, the precision measure, computed under the assumption that user has read all top 10 documents, is 0.4. In order to construct recall value, let us assume that we know that there exists 14 relevant documents that a user looking for "marine vegetation" would be interested in reading. In such case, the recall value computed after the top 10 ranked documents is 0.28.

Recall is important in contexts where the user is interested in finding **all** possible related document about a given topic. Such criteria is true for legal domains, where law professionals are interested in providing all documents related to their legal battle. Similar demands are common across patent searches and medical search domain, where it is critical to identify all possible related documents. However, higher recall is usually associated with decreased precision.

1. 600 Ugandans Escape Rare Marine Disaster
2. Bangladesh Launches Bio-Diversity Management Project
3. Chinese Environment News: Weekly Highlights
4. Chinese Environment News: Weekly Highlights
5. Dazhai: Yesterday's Model Commune, Today's Travel Destination
6. BRET HELPS THE GULF COAST GET IN SHIPSHAPE
7. Report: Runoff Damages Coastal Areas
8. Feature: Kenyans Suffer as Lake Victoria Waters
9. S. Africa Launches Action Plan to Clear Alien Vegetation
10. Jiangsu Steps Up Afforestation Efforts

Figure 2.1: Top 10 documents displayed for the query "marine vegetation"

Existing evaluation frameworks have their roots in the two Cranfield experiments conducted by Cyril Cleverdon [1962] at the Cranfield College of Aeronautics. In Cranfield1, Cleverdon compiled a test collection, where queries were composed from the source document. Several indexing systems were compared under the criteria of being able to retrieve the document from which the original query was composed from. This approach was meant to avoid seeking human judgements for the results retrieved for a given query. In Cranfield2 experiment, Cleverdon constructed queries from source documents and discarded those source documents from the collection. Documents retrieved by the search systems for the query are then judged for relevance by human judges. Research into evaluation methods went hand in hand with that of advances in ranking functions. However, most of the evaluation efforts were individual in nature where researchers construct a test collection catering to a specific evaluation objective. Karen Spärck Jones, in the process of proposing the inverse document frequency (IDF) measure to identify key-terms for a given document in a given collection (Spärck Jones [1988]), highlighted the lack of supporting test collections to easily compare the effectiveness with respect to existing methods. She further emphasized the need to construct an "*ideal*" test collection, (Spärck Jones and Van Rijsbergen [1975]), which could allow researchers to easily compare several ranking functions against one common collection and information needs (queries). Karen Spärck Jones, along with Donna Harman, worked at establishing TREC for the experimental comparisons of ranking functions.

TREC

The Text REtrieval Conference (TREC) [1], is an annual large-scale effort across by researchers to establish test collections for evaluate ranking functions over a variety of contexts and settings Harman [1993]. Organized by the U.S. National Institute of Standards and Technology (NIST), it forms the *de facto* means to evaluate the benefits of any "novel" ranking function, in

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

comparison with existing ranking functions. Following the Cranfield paradigm Cleverdon [1962], each test collection compiled consists of three major components:

- A static document collection (*corpus*)
- A set of information needs (*query topics* or *topicset*)
- A set of relevance judgements (*qrels*)

In general, human assessors/judges employed by NIST construct query topics and assess/judge the relevance of retrieved documents. Query topics constructed by assessors are meant to reflect the typical kinds of information needs that a user might have while browsing the document collection. Each query is carefully mapped to a specific information need, explicitly stated in form of *description* and *narrative* fields, as seen in Figure 2.2 for the query "marine vegetation". In addition, assessors check that the actual information need could be satisfied from the collection. Usually, this check is performed by submitting the the query to an existing state of art search system and checking the retrievability of relevant document. Both description and narrative fields are useful in defining the criteria upon which assessors judge the relevance of each document during the judging phase.

```
<num> Number: 314
<title> Marine Vegetation

<desc> Description:
Commercial harvesting of marine vegetation such as
algae, seaweed and kelp for food and drug purposes.

<narr> Narrative:
Recent research has shown that marine vegetation
is a valuable source of both food (human and animal)
and a potentially useful drug.  This search will
focus primarily on these two uses.  Also to be
considered relevant would be instances of other
possible commercial uses such as fertilizer, etc.
```

Figure 2.2: Sample TREC topic for the query "*Marine Vegetation*"

Several test collections have been compiled over the years to compare retrieval functions under various information seeking scenarios. Most common of the behavior emulated is the "adhoc search" scenario, where a user is interested in knowing more information about a certain topic (e.g. *marine vegetation* above) while browsing the web, a news archive, or a library collection. In addition to the adhoc search scenario, several collections have been compiled over the years

emulating several other information seeking scenarios. They include the scenario of a user seeking information from documents in a intranet (*Enterprise Search*), a user seeking opinionated email discussions (*Discussion Search*), users interested in finding homepage of a particular entity (*Namepage Finding*), users looking for an expert about the topic in a given organization (*Expert Search*), users interested in finding all documents related to the legal case on hand (*Legal Search*), users searching information across web scale (*Web Search*), users searching blogs (*Blog Search*), and users looking for information across a given governmental site **.gov** (*Terabyte track*).

For each query in a given topicset, participating systems are required to submit a ranked set (typically 1000 documents), sorted in the decreasing order of their likelihood of satisfying the underlying information need. Unlike the initial Cranfield experiments, Cleverdon [1962], where each and every document in the collection is judged with respect to each and every query topic, it is impractical to expect human judges to evaluate all query topics with respect to all documents in the collection. NIST adopts the *pooling* approach, proposed by Spärck Jones and Van Rijsbergen [1975], to construct a document pool for each query in a given topicset. This pool is created from top ranked documents retrieved by participating systems for that particular query. It is based on the hypothesis that, collectively, all participating systems would identify (almost) all of the documents that a user would be interested in. Typically, assessors are provided with a pool obtained from the union of top 100 documents from each participant system.

Each document in the pool is then independently judged in terms of its relevance with respect to the given topic. Since the assessor is impersonating a user, with some underlying some information need, it prompts a heavily debated question

"*What makes a document relevant?*".

Several attempts have been made to identify the criteria or factors that influences an assessor's relevance judgements. Cooper [1971] defined the notion of relevance of a document as the *correspondent in context between an information requirement statement and an article*, in other words, the extent to which the article covers material that is appropriate to the requirement statement (description or narrative) .

Recently, Chen and Xu [2005] grouped various factors that were proposed by previous researchers into five major categories: *topicality*, *reliability*, *understandability*, *novelty*, and *scope*. From their work, the *topicality* of a document is a subjective assessment of (the extent of) overlap between the topic of a given document and that inferred from a given query. For example, a document describing the events on the day of "Obama's inauguration" is indicated as being "on topic" with respect to a query such as "Obama's Swearing in Ceremony". It is possible for a document to cover several topics. The *reliability* of a document is the predefined notion of "trustworthiness" or "credibility" any user associates with the source. For example, current web

user often consider Wikipedia [2] to be a credible source for information than a blog post by an unfamiliar writer. The *scope* of a document (for a given topic) refers to fraction of the document that is about the given topic. I was hypothesized that users would prefer documents where the scope of user's query topic is higher (i.e., most of the information in the document is related to the topic). The *understandability* of a document is the extent to which the information or language used by the author of the document can be understood by the user. It was hypothesized that the understandability of a document would be directly proportional to user relevance assessments. The *novelty* of a document is the the amount of information in the document that is new in comparison with what the user already knows prior to reading the document. Intuition dictates that users would prefer documents that provide novel information rather than documents that are redundant. Chen and Xu [2005] carried out user studies to verify these hypotheses. They reported that assessor judgements of document relevance positively correlated with the topicality and novelty of a document, and had not correlation with the remaining factors such as scope, understandability and reliability.

Until now, in most collection building exercises the relevance of a document is judged solely based on its topicality. Early assessments were binary in nature, classifying each document as either relevant (1) or non-relevant (0) (based on its topicality).

$$P(r|d_i, q) = \begin{cases} 1 & \text{if topical} \\ 0 & \text{else} \end{cases} \tag{2.3}$$

where $P(r|d_i, q)$ is the probability of relevance of the document d_i to query q. In this fashion, assessors would judge the relevance of all the pooled documents and compile a qrels file, where each qrel is a quadruple of the form: <*topic, temp, document, relevance*> (as shown below):

```
314 0 FBIS3-41339 0
314 0 FBIS3-41483 0
314 0 FBIS3-41666 0
314 0 FBIS3-42467 1
314 0 FBIS3-42590 0
314 0 FBIS3-43115 0
```

Since its inception, TREC has developed into an annual conference/competition, and played an important role in improving retrieval algorithms. Archived TREC collections are still being used in the design of new ranking functions. Similar ideas have been adopted for evaluating

[2] `http://en.wikipedia.org`

focused retrieval in structured documents at INEX [3] , and for multilingual retrieval at the Cross-Language Evaluation Forum (CLEF) [4].

Although, TREC provides an ideal testbed to easily compare different ranking systems, it does have its own shortcomings (Blair [2002]). First, TREC and Cranfield paradigm focuses on system effectiveness, i.e. effectiveness by which the system could retrieve the relevant information. This approach fails to consider the extent, or ease, with which the user is able to access the relevant information. However, considering this kind of evaluation may be too cumbersome and expensive. Second, NIST employed assessors (or judges), involved in construction of query topics and assessing the relevance of pooled documents, are experts in that particular topic. It does not reflect the systems ability to aid users who are not familiar with that particular topic, who cannot compose the appropriate query, and who may not be capable of assessing the relevance of a document. Third, it is ironic to rely on the effectiveness of participating systems to collectively identify all possible relevant documents available for a given query, in a given document collection [5]. Nonetheless, TREC continues to have a positive effect in the field of information retrieval.

2.2 Measures

Over the years, researchers have investigated different effectiveness measures in parallel with their investigations of ranking functions. In the first TREC conference, ranking functions were compared based on precision and recall values (Eq. 2.1, Eq. 2.2). The basic precision measure (for example, precision at rank 10) when computed at a given rank (say 10) does not reflect the distribution of relevant documents up to rank 10. To demonstrate this, let us consider two hypothetical systems, systemA and systemB, retrieving documents in the following order

systemA = y,y,y,y,y,n,n,n,n,n
systemB = n,n,n,n,n,y,y,y,y,y

where 'y' indicates that the document is relevant, and 'n' indicates that the document is not relevant (from assessor judgments). When compared using precision at rank 10 (P@10), it is the same for both systems (= 0.5). However, we can guess that a user is more likely to be satisfied by the result from systemA than by the result from systemB. In addition to precision and recall values, systems were compared using, '**precision at r**', at specific r recall values ($r = 0.1, 0.2, 0.3 \ldots 1.0$) [6]. In the first TREC conference, systems were compared by means

[3] `http://www.inex.otago.ac.nz`
[4] `http://www.clef-campaign.org`
[5] Ref Section 2.3. for detailed explanation
[6] recall is computed in terms of $|R|$, i.e. the number of known relevant documents for a given query topic.

of precision-recall curves (also referred to as recall/precision plots). The x-axis of the curve has recall values at fixed values of recall $r = 0.1, 0.2 \ldots 1.0$, and the y-axis of the curve has the precision values ranging from $0 \ldots 1$. For each topic, system effectiveness is represented by plotting the precision values at each fixed recall value (x-axis). The precision at each recall point are then averaged across all the topics in the given topicset, to obtain system-level precision-recall curves. Harman [1993] report that the recall/precision curves for several systems were too close to determine if one system is better (in terms of statistical significance) than the other system by just viewing the plots. In addition, due to the limitation on the number of results participating systems were asked to retrieve (200), recall/precision curves were not reliable beyond certain point of recall (0.4 of R).

Average Precision

At TREC 2 researchers introduced the average precision measures (AP) in non-interpolated form in order to summarize both precision and recall values into one value. The average precision value of a system can be defined as the "average of precision at each relevant document retrieved by the system", normalized by the total number of relevant documents known to be exist for a given query topic in the collection (Eq. 2.4).

$$\text{AP} = \frac{1}{|R|} \sum_{i=1}^{n} \text{Precision@}i.r_i \tag{2.4}$$

where R is the total number of relevant documents available for the particular topic, r_i is the relevance of a document, as assessed by human assessors (Eq. 2.3), and n indicates the total number of ranked documents retrieved by a system for the query. If the document at rank i is relevant then $r_i = 1$; otherwise $r_i = 0$. By substituting, Precision@i, we obtain

$$AP = \frac{1}{|R|} \sum_{i=1}^{n} r_i \cdot \sum_{j=1}^{i} \frac{r_j}{i} \tag{2.5}$$

Returning to the two ranking system output (Eg. 2.2), and assuming that there are 15 relevant documents in total for the topic in the collection, the AP values for system A and B are 0.33 and 0.12 respectively. Average Precision has become one of the frequently reported measuring for comparing the effectiveness of ranking functions. Average Precision favors ranking methods that tend to retrieve relevant documents at the top Buckley and Voorhees [2000]. Average Precision is also found to be stable with respect to randomly sampled sub-collections Hawking and Robertson [2003]. Recent work by Webber et al. [2008] has found the Average Precision measure to be a better predictor of system's effectiveness than measures such as P@10 etc. Mean of Average Precision (MAP) computed over a given set of topics is one of the official measures used while

reporting the effectiveness of ranking functions at TREC.

Reciprocal Rank

In tasks such as Namepage Finding, where user is assumed to be seeking the homepage of a given entity, or Known-Item Search, where the user is assumed to be seeking a particular document that know to exist in the collection, there may only be a single relevant document in the collection ($|R| = 1$) or a small number of equivalent documents. For example, for the query "facebook" only one page might be considered relevant, the page **http://facebook.com**, or several others might be considered equivalent, like **http://facebook.ca**. For such queries, it becomes difficult to differentiate between systems with measures such as average precision, as the user is interested in only one relevant document. For these tasks, researchers have the proposed *reciprocal rank* (rr) measure, Craswell et al. [2003] ($rr(q)$),

$$rr(q_i) = \frac{1}{f_i} \tag{2.6}$$

where f_i is the first rank at which a relevant document occurs in the ranked list for query q_i. For example, if a relevant document is retrieved at four different positions, $i = 1, 2, 3, 4, \ldots$, by four different systems, corresponding $rr(i)$ values are $1, 0.5, 0.33, 0.25 \ldots$ respectively. The mean of Reciprocal Rank (MRR) could then be computed for a given set of $|Q|$ queries as average of individual $rr(q_i)$ (Eq. 2.7).

$$\frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{f_i} \tag{2.7}$$

It is to be noted that the reciprocal rank measure is suitable for tasks such as Namepage Finding and Known-item Search, but not standard adhoc retrieval tasks.

Graded Relevance Measures

In earlier evaluation experiments, document relevance is considered as a binary variable. Documents were either relevant or not relevant to user's information need. Unfortunately, such dichotomous relevance assessments leads to a false assumption that all relevant documents are equally likely to satisfy the given information need. Robertson [1977a] proposed an alternative scheme, wherein relevance assessments are carried out over a continuous scale of relevance. In their proposal, one extreme of the scale contains documents that are either "not relevant", "harmful", etc., while the other extreme contains documents that are likely to be "highly relevant" to the users information need. Generally, documents are placed under levels of relevance such as *not relevant*, *marginal relevant*, *relevant*, and *highly relevant*, Robertson and Belkin [1978], reflecting the extent of their overlap with user's topic of interest. Sormunen [2002] carried out an

experiment where humans were asked to re-assess documents that have been judged as relevant from previous evaluation exercises. Unsurprisingly, Sormunen reported that of all the documents judged as relevant, only 16% of them were re-judged as highly relevant, while the majority (50%) were re-judged as marginally relevant.

Järvelin and Kekäläinen [2002] proposed *gain* based effectiveness measures, where the multi-level graded relevance value of a document is equated to the utility gained by the user from viewing the document. Given a ranked list of documents $[d_1, d_2, d_3, d_4, d_5 \ldots]$, Järvelin and Kekäläinen construct a gain-based vector $G = [g_1, g_2, g_3, g_4, g_5 \ldots]$, where $g_i \in [0, 3]$ corresponds to the relevance (graded) value of the document at rank i, as judged by an assessor. Here 0 corresponds to *non-relevant* documents, and 3 corresponds to *highly relevant* documents. need. Given such gain vector, Järvelin and Kekäläinen first proposed the *Cumulated Gain* measure ($cg@n$), a straightforward means to compute effectiveness in terms of information accumulated after the top n ranked documents.

$$CG(n) = \sum_{i=1}^{n} g_i \tag{2.8}$$

Example: Given a set of 5 documents $D = [d_1, d_2, d_3, d_4, d_5]$ with their corresponding graded relevance assessments $G = [3, 1, 0, 2, 0]$. Let A and B be two different ranking functions, who ranks the document set D, as shown follows:

$A =< d_2, d_1, d_3, d_4, d_5 >$
$B =< d_1, d_4, d_5, d_2, d_3 >$

Computing $CG@5$ as in Eq. 2.8 , both ranked sets A and B are found to be equally effective. However, the Probability Ranking Principle Spärck Jones et al. [2000] dictates that ranking functions are most effective if they are able to order the documents in decreasing order of their likelihood of being relevant. Under such circumstances, ranked set B should be more effective than ranked set A.

Järvelin and Kekäläinen proposed a variant of CG measure, *Discounted-Cumulated Gain* (DCG) , where the utility of a relevant document is *discounted* in terms of the rank at which it is shown to the user. Any discount is meant to reflect the underlying effort, on user's side, to actually reach to the document. $DCG@n$ can be computed as follows:

$$DCG@n = \sum_{i=1}^{n} \frac{g_i}{log_b(1+i)} \tag{2.9}$$

where g_i is the relevance (graded) of the document at rank i and $log_b(1+i)$ is the factor by which the relevance (or utility) of a document at rank $i > b$ is discounted. Assuming $b = 2$, i.e. utility of any document from rank 2 onwards, we obtain $DCG@5$ values of 3.79 and 4.76 for A and B

respectively. Several discount factors have been proposed in order to aptly reflect the effort in the user's side. Burges et al. [2005] were critical about the rank-based discount factor, and proposed a different discount factor

$$\frac{2^{g_i} - 1}{2^{g_{max}}} \tag{2.10}$$

where g_i is equal to the graded relevance value of the document d_i judged on a multi-level scale of relevance, g^{max} is the maximum relevance assessment that could be assigned to any document on the multi-graded scale of relevance.

Järvelin and Kekäläinen proposed normalized variants for both DCG and CG measures to compare across different topics. In their proposal, system ranking output is compared against an *ideal* ranked list, using all the documents that are judged in the pool. An ideal ranked list can be obtained by ranking all the highly relevant documents at the top, followed by relevant documents, and then marginally relevant documents before non-relevant documents. In our example, the ideal ranked list constructed from the 5 judged documents is

$$I = [3, 2, 1, 0, 0] \tag{2.11}$$

Given I, one can compute Ideal (Discounted) Cumulated Gain ($I(D)CG$) as shown in Eq. 2.9 and Eq. 2.8. Finally, normalized metric, normalized Discounted Cumulated Gain ($nDCG@n$) can be obtained as follows:

$$nDCG@n = \sum_{i=1}^{n} \frac{DCG(n)}{IDCG(n)} \tag{2.12}$$

Similarly $nCG@n$ can be obtained as follows

$$nCG@n = \sum_{i=1}^{n} \frac{CG(n)}{ICG(n)} \tag{2.13}$$

Of both normalized metrics, Normalized Discounted Cumulated Gain ($nDCG@n$) is commonly reported while comparing different ranking functions. Al-Maskari et al. [2007], while investigating into correlation between user satisfaction and gain based metrics, surprisingly reported that DCG correlates better with user satisfaction than $nDCG$.

Sakai [2004] defined an alternative $Q - measure$ similar to Average Precision (AP) that can accommodate both binary and graded relevance values.

$$\text{Q-measure} = \frac{1}{R} \sum_{1 \leq r \leq L} \text{isrel(r).BR(r)} \tag{2.14}$$

where $isrel(r)$ is either binary or graded relevance of document r, $BR(r)$ refers to the *blended ratio*, which factors the extent to which the system output deviates from an ideal ranked output

(similar to NDCG).

$$BR(r) = \frac{cg(r) + count(r))}{(cg_I(r) + r)} \tag{2.15}$$

where $cg(r)$ indicates the cumulated gain of top r documents in the list (L), and $cg_I(r)$ indicates the cumulated gain after traversing the top r documents in an ideal ranked list.

Although graded relevance assessments allow researchers to distinguish between highly relevant and marginally relevant documents, there is no clear guide to determining the number of relevance levels that should be used in a particular circumstance.

Preference based measures (ppref)

Carterette et al. [2008] suggested an alternate means to obtain relative ordering between documents, without the need to artificially construct a grading schema. Instead, they display a pair of documents side-by-side and ask the assessor to "choose" one of the document, which is closer to satisfying the underlying information need for the given topic. Carterette et al. then sort the documents in decreasing order of preference as captured directly from the assessor. For example, let us assume that three documents, a,c,d were shown to a assessor, who expressed following preference:

$a > d$, document a is preferred over document d
$a > c$, document a is preferred over document c
$d > c$, document d is preferred over document c

Based on observed preferences, the relative ordering of documents is established as $a > d > c$. From the retrieved set of documents, a pair of documents, say a,d are said to be *ordered* pairs, if the retrieved set contains both of the documents before a defined threshold rank, say k. Among ordered pairs, a pair is termed as *correctly ordered* if the system retrieved order (of those two documents) matches with that of the assessor's preference. For example, say a system X retrieves all three documents (not necessarily contiguous), $\ldots a \ldots c \ldots d$, then pair (a, c) and (a, d) are correctly ordered, and the pair (d, c) is incorrectly ordered. Carterette et al. [2008] then defined a measure "*precision of preferences*", which is equal to the fraction of correctly ordered pairs in the retrieved set of results.

A naive implementation would require assessors to judge a total of $\frac{n*(n-1)}{2}$ document pairs, in order to obtain the preference order among n documents. It would become tedious to carry out such experiment for a large topic set. Careterette et al. eliminate such requirement using the property of preference transitivity. For example, any document that has been tagged as "bad", could be assumed to be least preferred and need not be paired up again. Though this property

```
1 G21-65-1713985 NA -2
2 G44-69-3668666 G09-27-3412367 -1
2 G09-27-3412367 G01-15-3190453 -1
2 G01-15-3190453 G08-92-2189977 -1
1 G18-17-2457040 G40-16-0903600 -1
1 G40-16-0903600 G02-53-3076277 -1
1 G02-53-3076277 G00-03-1898526 -1
```

Figure 2.3: Preference qrels sample

may not be 100% accurate, it eliminates the need to capture preference between every documents pair.

One major advantage of preference based assessments is their direct applicability in training pair-wise learning to rank functions (Figure 2.3). In each line, the -1 indicates the document on the left hand side is preferred (relevant 1 or highly relevant 2). Carterette and Bennett [2008] further extended the standard measures such as recall preference ($rpref$) , weighted precision of preferences ($wpref$), normalized weighted precision of preference ($nwppref$) etc. that could be used with preference based qrels.

Expected Search Length

Cooper [1968] was critical of the notion of using two complimentary measures such as precision and recall while reporting the system's effectiveness. Further, Cooper criticized the lack of a user model, in everyday usage of search engines, where recall is necessary. Instead, Cooper introduced the notion of *search length*, which is equal to the number of non-relevant documents user has to go through before completely satisfying his/her information need. In computing search length, Cooper did not include the number of relevant documents that user might come across.

Cooper [1968] proposed a probabilistic metric *Expected Search Length* (ESL), which is the expected length a user might have to scan to satisfy his/her information need:

$$esl = \sum_{m=1}^{M} pr(l_m).l_m \tag{2.16}$$

where M is all search lengths that are possible for a given query q, l_m indicates the length of such search length m, and $pr(l_m)$ is the associated probability of the search length l_m. Tang and Sun [2003] report that expected search length was ideal for evaluation in the context of web searches.

Rank-Biased Precision

Zobel [1998] carried out a thorough investigation concerning the extent to which existing pooling approaches (e.g., the union of top 100 documents from each system) includes all relevant documents for a given query. They estimate that at best around 50% to 70% of the relevant documents are judged for a given topic and that a recall-based measure (which assumes that every relevant document is known) is highly uncertain.

Later, Moffat and Zobel [2008] proposed an alternate measure wherein the utility of a ranking function is determined, not by the number of relevant documents present, but by the depth to which user is willing to proceed in order find relevant information They based this measure on the notion of *user persistence* ($0 < p \leq 1$), which is probability with which a user at the current document rank would read next document (p) or would leave and end the search session ($1 - p$). It is assumed that user is bound to read the first document. Under such behavior, users would end up reading the second document with a probability p, the third document with probability of p^2, and so on. The total number of documents a user is likely to go through would depend on his/her persistence. Lower persistence $p = 0.1$ simulates an impatient user, while higher persistence $p = 0.9$ simulates a patient user, who is willing to go through more of the retrieved documents.

On average, the number of documents users are expected to read is

$$\sum_{i=1}^{\infty} p^{i-1} = \frac{1}{1-p}. \tag{2.17}$$

Now let r_i be the relevance (binary or graded) values of a document with respect to a given topic. Utility gained by a user from reading a document could be obtained as a product of likelihood of a user reading the document and relevance of the document

$$r_i.p^{i-1} \tag{2.18}$$

Overall, the total utility, or Rank-Biased Precision(RBP), for a list of d documents is defined as

$$RBP = (1-p).\sum_{i=1}^{d} r_i.p^{i-1}. \tag{2.19}$$

Moffat and Zobel explored different persistence values between $0 < p \leq 1$, and identified $p = 0.85$ as providing the best correlation with existing effectiveness measures and a good correspondence with user behavior. Although such a method of computing utility is straightforward, user persistence is not factored based on the relevance of a document. That is, Moffat and Zobel do not consider the persistence of a user based on the relevance of the current document being

viewed.

Normalized Cumulative Precision

Average Precision, as mentioned before, has always been subjected to the criticism for the lack of any clear user model. Robertson [2008] hypothesized a simple user model (inspired by search length Cooper [1968]), to provide a probabilistic interpretation of Average Precision. As in Cooper [1968], Robertson assumes that a user would traverse the ranked list until their information need is satisfied. Robertson computed AP @n, the rank at which the user information need is satisfied.

$$AP_n = \frac{1}{n} \sum_{m=1}^{n} \delta_{m,n} \tag{2.20}$$

where $\delta_{m,n}$ is the cumulated precision at rank n which is equal to 1 if $m < n$, and

$$\delta_{m,n} = \begin{cases} 1 & \text{if } i_m > 0 \text{ and } i_n > 0 \\ 0 & \text{else} \end{cases} \tag{2.21}$$

Assuming that the user could stop reading at each ranked document with a probability of $p_s(n)$, the utility gained by the user, defined as Normalized Cumulative Precision (NCP), can be computed as

$$NCP = \sum_{n=1}^{\infty} p_s(n) AP_m \tag{2.22}$$

NCP is equivalent to Average Precision (AP) by assuming $p_s(n) = \frac{1}{|R|}$, i.e., the user is likely to stop at each relevant document encountered in the ranked list with equal probability. When $p_s(1) = 1$, i.e. user is bound to stop reading after the first relevant document, NCP is equal to rr. A common underlying assumption across all these measures is that the user is bound to view the first document in all instances. Further, NCP assumes that users tend to browse through ranked lists until their information need is satisfied, and usually stop after a relevant document.

S-DCG

Järvelin et al. [2008] extended the existing (n)DCG measures (Järvelin and Kekäläinen [2002]) to evaluate queries in "sessions" (SDCG), simulating users who submit multiple queries with one underlying information need. The SDCG measure discounts the relevance of a document retrieved for later queries to reflect the effort on user's side to reformulate and submit a new request.

$$sDCG(q) = (1 + log_{bq} q)^{-1} . DCG \tag{2.23}$$

where q indicates the sequence number of the query, $bq(1 < bq < 1000)$ the patience of a user to re-formulate their initial query requests (2 = impatient, 10=patient).

%no metric & GMAP

After the first few years of TREC, it was observed that there was great variance in system's performance across the set of topics in a topicset. In particular, it was observed that even the top performing systems failed to retrieve many or any relevant documents for some topics (Harman [2000]). In 2004, NIST organized the Reliable Information Access (RIA) workshop, Harman and Buckley [2009], and invited several research groups to investigate the various causes for such variance. Several features associated with such "hard" query topics, queries that the systems did not fare well, were reported in their report. One can summarize them into one of the following instances (Carmel et al. [2006]) :

- Focus on one aspect and are missing on other terms. For example for the query "incidents of stolen art or forged art", it is observed that majority of the systems focus on one aspect (say art) and not on the modifier "stolen", which is important to the information need.
- Focus on one aspect or other, but not both. For example, for the query "disasters occurred in tunnels used for transportation", majority of the systems focused on one aspect, tunnel disasters, or transportation disasters, but not both.
- Focused on an irrelevant aspect and not the main concept. For example, for the query "Spotted Owl incident in America", systems retrieved documents that focused on Owl but not Owl spotting in America.
- Requires relationship analysis between entities. In the query "Quantity of sugar exported by Cuba", a relevant documents is the one that connects the quantity sugar being exported from Cuba to any other country. To retrieve a relevant document, system needs to establish the following template <cuba, export sugar, X>, where X could be other country, and score accordingly.
- Systems need human help to identify difficult aspect. These instances refer to those queries, where domain specific knowledge needs to be provided to identify the information need. For example in the query "New methods of producing Steel?", systems need to know infer if a given text actually is talking about the novel method of producing steel.

To overcome these issues, NIST introduced Robust Track (Voorhees [2004a]), to promote the development of ranking functions that could answer "hard" queries from previous TREC exercises. Hard topics are defined in terms of median of systems' performance on the topic in

previous TREC exercises. One of the measure used to evaluate the effectiveness if $\%no\ metric$, which is fraction of queries for which the system did not retrieve even a single relevant document at a given rank.

In parallel, researchers investigated means by which retrieval systems could benefit from a *clarification* step Allan [2004], in which systems are provided a one-time interaction with the human assessors. This proved useful in enabling systems get more information about the queries that require deeper analysis. In this track, baseline runs are submitted by each system before initiating human interaction . Systems then submit a "final run" using the knowledge gained from human interaction.

Robertson [2006] proposed the Geometric Mean Average Precision (GMAP) measure, where the overall system effectiveness is computed by the *geometric mean* of the system effectiveness measure for each individual topic.

$$AL(X_1, X_2 \dots X_n) = \frac{1}{n} \sum_{i=1}^{n} log(X_i) \tag{2.24}$$

$$GMAP = \exp^{AL(X_1, X_2 \dots)} \tag{2.25}$$

where X_i is the system (X) performance for a given topic i. Robertson claim that GMAP is ideal when comparing pseudo-relevant feedback runs with their initial (baseline) run, since GMAP is capable of differentiating systems that improve (in the final run) over poorly performing topics (in the baseline) .

As mentioned before, current approaches create a document pool from the union of top N, usually $N = 100$, ranked documents from each participating system. It is presumed that all known relevant documents for a given query are obtained this way. Worse, any document not part of the original pooled set is considered not-relevant. Such claims were challenged by Zobel [1998]. Zobel observed the rate at which participating systems continued to retrieve relevant document past the depth beyond the depth where all documents are pooled. Based on the rate of returning relevant document, Zobel reported that at best only $50\% - 70\%$ of the relevant documents could be identified for a given topic. Büttcher et al. further investigated the bias, either against or for, retrieval functions that did not contribute documents to the construction of the original pool.

Cormack et al. [1998] aim to solve this problem through a *greedy pooling* approach to maximize the chance of gathering (almost) all relevant documents for a given topic in the corpus. In their approach, participating systems are selected in round robin fashion. For each selected system, Cormack et al. would continue to pool documents from the system in a *greedy fashion* — i.e. contiguously add documents retrieved by the system. After judging the document added to the

pool (if not already judged), the precision measure for all the systems retrieving this document at the current rank are updated simultaneously. The process continuous until the precision of system from which documents are currently being pooled decreases beyond certain threshold. Next, the best system, based on updated precision scores, from the priority queue is selected. This process continuous till precision value across all systems in the queue fall below the threshold.

2.3 Measures based on Incomplete Relevance Judgements

Buckley and Voorhees [2004] observed that a majority of the retrieval measures like Mean Average Precision, R-precision, Precision at 10, are unstable in the presence of incomplete relevance assessments. In the absence of relevance assessments, these measures are heavily biased against the systems that did not contribute documents towards original pool construction process. Buckley and Voorhees then proposed an alternate measure, *bpref*, which takes into consideration the presence of unjudged documents in the ranked lists.

$$\text{bpref} = 1 - \frac{1}{R}\sum_{r}(1 - \frac{\text{number of } n \text{ above r}}{R}) \tag{2.26}$$

where R refers to the number of relevant documents from the judged pool, r and n refers to relevant documents and non-relevant documents respectively. They further propose a variant *bpref-10* measure, to accommodate topics that have few (1 or 2) relevant documents in the pooled documents.

$$\text{bpref-10} = \frac{1}{R}\sum_{r}(1 - \frac{\text{n greater than r}}{10 + R}) \tag{2.27}$$

Buckley and Voorhees observed that the *b-pref* measure correlates well with established measures like Mean Average Precision (MAP) in the presence of relevant judgements.

Similarly, Yilmaz and Aslam [2006] propose three approximate measures named *Induced Average Precision (IndAP)*, *Subcollection Average Precision (SubAP)* and *Inferred Average Precision (InfAP)* that could be used to estimate effectiveness in light of incomplete and imperfect relevance assessments. In *IndAP*, only documents that are judged are considered while comparing the ranking functions.

$$indAP = \frac{1}{R}\sum_{r}\frac{\text{number of relevant upto rank r}}{rank(r)} \tag{2.28}$$

where $rank(r)$ refers to the rank of a document, (judged relevant). In *SubAP*, Yilamz and Aslam obtain a smaller qrel by sampling from the original set of relevance judgements. They report that *SubAP* was found to be a closer approximate to AP than IndAP.

Yilmaz and Aslam proposed the third measure *infAP* as an expectation of outcome of the

following random experiment. First, a document, d, is randomly picked from an existing relevant set and let its corresponding rank be i in the retrieved list. Next, Yilmaz and Aslam select a document at random from the set $(1 \ldots i)$, and output the relevance of that particular document. An expected precision value at rank k (infAP) is then calculated as

$$E[\text{precision at k}] = \frac{1}{k}.1 + \frac{k-1}{k}(\frac{|d100|}{k-1}.\frac{n_{rel}}{n_{rel}+n_{nonrel}}) \tag{2.29}$$

Büttcher et al. [2007] approached the problem of assessing relevance values for a given document based on the established set of relevance assessments. They train a SVM-classifier [7] using the relevance judgements and use the trained classifier into predicting the relevance judgement of an unseen document. They report reasonable success in predicting the relevance of a document.

2.4 Limitations of Cranfield Paradigm

Although the Cranfield paradigm allows us to compare and evaluate different ranking functions, it does make some oversimplifications:

- Relevance of a document is judged in an independent fashion.
- Each query is associated with one specific information need.

In this section, we discuss the consequences of this simplification and also highlight several papers that attempted to overcome this problem.

2.4.1 Novelty and Redundancy

Under the Cranfield paradigm of evaluation, documents pooled for a given query are judged in an independent fashion. This approach could lead to a situation where a ranking function falsely benefits from retrieving the same content across several near-duplicate documents in the ranked list. For example, as shown the Figure 2.1, the documents at rank 3 and 4 appear to be the same documents (perhaps with minor edits). However, both are judged as relevant by assessors. Under existing paradigm, the system is given credit for retrieving the same content across two documents. However, in reality, a user browsing through the documents would end-up viewing the same document content without any added benefit. Cooper [1971] emphasized the need to distinguish between *topicality* or *topic-relatedness* and the actual utility that could be attributed to a given document. Cooper questioned the benefit of showing a user information she has already

[7] `http://svmlight.joachims.org/`

seen and the utility gained by such a user. Cooper proposed that the relevance of a documents should reflect the absolute utility, in terms of the added information that user would gain, over what she already knows prior to reading the document, in defining the utility of a given document. Similar arguments were put-forth by Goffman [1964]. Recently, Chen and Xu [2005] carried out an experiment to identify the factors that induce the relevance assessments from the assessors. Chen and Xu report that assessors judgement of relevance of a given document positively correlates with both topicality and novelty of information in a given document. Irrespective of this, all the measures introduced in previous section do not penalize ranking functions for retrieving redundant information.

Carbonell and Goldstein [1998] proposed a scoring function based on Maximal Marginal Relevance (MMR), which comprises a linear combination of a document's similarity with the query and dis)similarity of the given document with the information already known to be seen by the user.

$$\lambda * Sim(d_i, Q) - (1 - \lambda) * \arg\max_{d_j \in S} Sim2(d_i, d_j) \tag{2.30}$$

where $Sim(d_i, Q)$ is the similarity (probability of relevance) of a document d_i with respect to a given query q, $Sim2(d_i, d_j)$ is the similarity between the document d_i and document d_j, which is part of the set S of documents already assumed to be read by the user. The value $\lambda \in [0, 1]$ is used to balance the weight given to relevance with the query ($\lambda = 1$) and how distant is the document from all the previously seen documents ($\lambda = 0$).

Carbonell and Goldstein proposed a greedy re-ranking approach, where the next document selected is expected to maximize the following function

$$\arg\max_{d_i \in R \setminus S} [\lambda * Sim(d_i, Q) - (1 - \lambda) * \arg\max_{d_j \in S} Sim2(d_i, d_j)] \tag{2.31}$$

where R indicates documents belonging to total documents in the collection, S indicates the set of documents already selected (higher rank). In a pilot experiment, Carbonell and Goldstein report that a majority of the users (%80) preferred the results compiled from MMR approach in comparison with those of standard retrieved results. Carbonell and Goldstein adapted the MMR measure for sentence extraction process in order to construct extractive summaries. Radlinski and Dumais [2006] proposed a method to diversify the results for a given query in the context of web searches. They approach this task with the aim of personalizing retrieved results for a given query. For each query q, they select a set of $k = \{0, 2, 4, 9, 19\}$ valid query reformulation (q_j), where a valid reformulation is the a query that is submitted thirty minutes subsequent to q. For each valid re-formulations and the original query, they then select around $\frac{100}{k+1}$ results retrieved and combined to form a pool of results.

In TREC Novelty Track Soboroff and Harman [2005], researchers investigated the task of

finding sentences from a document that are both relevant and also novel to the topic. That is, systems were compared by their ability to identify both relevant and novel sentences from a given set of documents, judged to be relevant to the query. System effectiveness is measured in terms of the set-based precision and recall measures, and a combined F-measure based on the precision and recall measures. Soboroff and Harman [2005] report difficulty in obtaining a reliable set of assessments concerning novel sentences across different assessors.

2.4.2 Intent Ambiguity

In traditional adhoc retrieval evaluation, as established under Cranfield paradigm, each query is carefully mapped onto an information need. This definition is applied while judging the relevance of a document. However, users (particularly web search users) may have different intents associated with the same query. These differences could, in part, be due to the lack of clarity in user queries, which masks their underlying information needs. Teevan et al. [2005] observed that, when asked to compose queries, users attributed different intents to similar queries. Teeval et al. also report that, even in those instances when similar intents were explicitly expressed by the users for a given query, there exists a variation in relevance assessments associated with documents judged across several users. They attribute such behavior to the possibility of users not being explicit enough in their queries.

In TREC Interactive Track, Over [1997], researchers explored this concept of user interest comprising a different set of *instances*. For example, a topic defined for the query "British Chunnel Impacts" (Figure 2.4), some of the instances that users could possibly be interested, as identified by assessors were

```
1  environmental impact
2  financing of high-speed rail line
3  cost of additional safety standards
4  merger (rationalization) of ferry companies
......
```

in which users could possibly be interested in few or all instances depending on their particular information needs. In the Interactive Track, systems were judged by *instance recall*, a measure of fraction of unique instances covered in a given duration. Since the track was aimed at studying the interaction of a user with the system, more emphasis was given to the instances identified by the users in a given time limit.

Zhai et al. [2003] further extended the analysis and proposed that each topic comprises of a set of sub-topics or *aspects* . As with the definition of *instances*, each aspect is meant to

```
Title:
       British Chunnel impacts
Description:
       Impacts of the Chunnel - anticipated or actual - on the British
       economy and/or the life style of the British
Instances:
       In the time alloted, please find as many DIFFERENT impacts of
       the sort described above as you can. Please save at least one
       document for EACH such  DIFFERENT impact.
       If one document discusses several such impacts, then you need
       not save other documents that repeat those, since your goal
       is to identify as many DIFFERENT impacts of the sort described
       above as possible.
```

Figure 2.4: Sample query for interactive track

be representative of one possible intent user might have while submitting the query. Zhai et al. proposed, S-recall@k, a recall-based measure in terms of the sub-topics covered at a given rank,K

$$\text{S-recall@}K = \frac{|\bigcup_{i=1}^{K} \text{subtopics}(d_i)|}{n} \tag{2.32}$$

where n indicates the number of sub-topics or aspects (known) for the query, subtopics(d_i) indicates the sub-topics covered in document d_i, as judged by assessors. A normalized variant S-precision@r, to compare across different topics, could then be obtained as follows

$$\text{S-precision@}r = \frac{\text{minRank}(S_{opt}, r)}{\text{minRank}(S, r)} \tag{2.33}$$

where minRank(S, r) is the rank at which the given ranking function would cover r unique sub-topics. Similarly, minRank(S_{opt}, r) is the minimum rank at which an ideal ranked list would have covered r sub-topics. In this measure, $minRank(S_{opt}, r)$ computation is NP-Hard and they proposed a greedy approximate algorithm that could be used to compile an approximate set.

Zhai et al. further proposed a re-ranking approach, inspired by MMR measure, using language model approach where weight of a term w_i based on previously seen documents

$$l(\lambda|d, \theta_O) = \sum_{i=1}^{n} log((1-\lambda)p(w_i|\theta_O) + \lambda * p(w_i|\theta_b)) \tag{2.34}$$

where $p(w_i|\theta_O)$ is the term weight under "old model" θ_O, obtained by the concatenation of documents read so far.

Chen and Karger [2006] highlighted the shortcomings of current optimization (based on evaluation measures) in dealing with ambiguity of web searches. For example, an ambiguous query such as *Java* could be subjected to multiple interpretations and therefore a variety of intents. Standard ranking methods are optimized to retrieve the most likely or most probable document at each rank. This might fail in retrieving any relevant documents, in cases where the query terms are ambiguous, and all the documents. Similar observations were made by Harman and Buckley [2009] in RIA workshop.

Chen and Karger proposed a simple evaluation measure, $k - call$ at n,

$$k - call@n = \begin{cases} 1 & \text{if count(rel)} \geq K \\ 0 & \text{else} \end{cases} \tag{2.35}$$

i.e. value of the metric is 1, if only the ranking function could retrieve atleast K relevant document at n. When $k = 1$, 1-call, the metric would be equal to the $(1 - nometric)$. For $k = n$, the system needs to obtain perfect "precision at n". The objective function corresponding to evaluation measure would then to be

$$Pr(r_0 \cup r_1 \cup r_2 \dots r_{n-1} | d_0, d_1, d_2 \dots d_{n-1}) \tag{2.36}$$

which could be used as an optimization function to rank documents for a given query. Chen and Karger proposed a greedy approximation method to compute the probability of a document being relevant

$$P(r_1 | d_0, d_1, \neg r_0) \tag{2.37}$$

that is, the likelihood of the current document d_1 being relevant given that document d_0 is not relevant. In general, document at rank i is scored by as follows:

$$Pr(r_i | d_0, d_1 \dots d_{i-1}, \neg r_0, \neg r_1 \dots \neg r_{i-1}) \tag{2.38}$$

that is documents are iteratively selected under the assumption that the previously seen documents so far are not relevant. This is in part could be similar to "blind negative feedback". Chen and Karger report that their greedy approach 1-greedy outperforms probability ranking principle, when measured under 1-call measure. They observe that a side-effect of such greedy ranking method is would implicitly diversify the retrieved results.

2.4.3 Word Sense Disambiguation

In past, researchers have attempted to use word sense disambiguation approaches to negotiate with the underlying ambiguity in user queries. Word disambiguation approaches could be grouped

into two kinds:

- Lexical Disambiguation
- Semantic Disambiguation

where the former contains the instances where the term belongs to different syntactic categories: the term "read" could be a noun or as a verb. The latter scenario arises, when the term could be used in different scenario. For example, the term "bank" (in noun form) could be used in one of the following interpretation [8]):

- bank – (sloping land (especially the slope beside a body of water)
- bank – (a financial institution that accepts deposits and channels the money into lending activities)

Automatic disambiguation approaches are based on heuristics based on either syntactic parse of the contextual sentences or the contextual terms surrounding the ambiguous term. For example, the term "bank" could probably being used in financial institution sense in the presence of terms such as "financial loans", "revenue", "cash deposit" etc. Several works in the past have explored the advantages of adopting disambiguation approaches towards information retrieval.

Krovetz and Croft [1992] investigated into the correlation between relevance of a document retrieved for a given query and the (mis-)match of sense of terms common to both . They observed a strong correlation between the relevance of a document and sense match. Similarly, the detected a strong correlation between mismatch of sense between the terms in query/document pairs and non-relevant documents. Another conclusion from their work is that disambiguation does not add any benefit for those documents that contain a lot of query terms. Voorhees [1993] studied the benefits of adopting existing disambiguation approaches towards both document and query representation, towards information retrieval. In her experiments, terms present in both documents and the queries are disambiguated. Voorhees, surprisingly, observed that disambiguation of query terms and using sense-based vectors instead of word stem vectors actually degraded the effectiveness of the retrieval systems. Although, it improved the effectiveness for few queries, it degraded the performances for majority of the query topics. Voorhees summarized such surprising results into the following points:

- Query statements are too short, to provide contextual information for accurate disambiguation

[8] from WordNet `http://wordnet.princeton.edu`

- Disambiguation methods are not accurate enough, and end-up identifying the wrong sense or not finding any sense at all.

Sanderson [1994] further explored the implications of presence of ambiguous terms towards the effectiveness of a ranking function. He compiled an ambiguous collection using the concept of "pseudo-words", where two words such as "banana" and "Kalashnikov" are replaced by ambiguous words "banana/Kalashnikov" all over the test collection.

Overall, it has been reported that the disadvantages of disambiguation far outweigh the advantages of using them. In recent times, researchers have begun to explore into post-hoc disambiguation, where the results retrieved for a given query are grouped into separate clusters to enhance the information seeking behavior of the user. Artiles et al. [2007] initiated the drive to construct test collections that can be used to evaluate disambiguation approaches in context of people search — search for people names. The problem is prevalent with respect to web searches, where a name is common to several persons spread across the world and across different eras. For example, the name "*Peter Jackson*" is common to several people as shown in Table 2.1. Disambiguation methods are then required to identify and group the documents that are related to one particular named entity. System effectiveness is then measured in comparison with gold standard clusters identified by human judges.

Name	Identity
Peter Jackson	a New Zealand-born film maker
Peter Jackson	black heavyweight boxer
Peter Jackson	an English cricketer
Peter Jackson	English Rugby union footballer
Peter Jackson	English footballer and football manager
Peter Jackson	British Silver medalist in rowing

Table 2.1: Multiple Entries for the name "Peter Jackson" (from Wikipedia)

2.5 Thesis Problem Statement

As reported in Voorhees [1993], if the disambiguation method fails to identify the correct sense of the query terms, it would have negative effect towards retrieved results. A feasible alternative is then to design ranking functions that considers queries to be ambiguous and implicitly aims to diversify the results to satisfy different interpretations Chen and Karger [2006]. Existing test frameworks (and associated evaluation measures) do not aid in such design process, which could lead to scenarios where ranking methods that are optimized under one particular collections do not adapt to different kinds of intents Büttcher et al. [2006] Hawking et al. [2000].

Spärck Jones et al. [2007] emphasized the need for such test collections in order to develop methods that could maximize the probability of user finding relevant information. Spärck Jones et al. associate each query with "bundles of intents" that user could possibly have while submitting the query. For example, a person searching for the query *Ubuntu Netbook remix* could possibly be interested in either the latest download-able version of the software, developers forums, hardware specs supported etc. Alternatively, bundle of intents could span across different un-related entities. For example, for the query "Peter Jackson", it could be possible that the user is interested in one or more persons listed in Table 2.1. Ranking functions should be optimized to be able to cater to various interpretations possible for the given query, so as to maximize the probability of user finding relevant information

2.6 Related Evaluations

In this section, we introduce to evaluation efforts in related fields such as summarization and question answering that motivate our framework (as described in Chapter 4).

Summarization

Summarization is the task of distilling the most salient information from a single document or multiple documents, and present the information in the manner specific to particular task. Summaries could be either generic or query-biased, bounded by the number of words or sentences etc. In case of multi-document summarization tasks, it is necessary to avoid reporting duplicate information that could make the summaries redundant.

System-generated summaries are usually compared against existing gold standard summaries, generated by human assessors. Given a system-generated summary and a gold-standard summary, quality of a summary is measured in terms of overlap of sentences between them.

Nenkova et al. [2007] proposed a Pyramid Approach of evaluation in order to provide an unified framework of evaluation that allows comparison of system-generated summaries and gold standard summaries on a semantic level, rather than mere term overlap. In their approach, each sentence from gold-standard summary is divided into Semantic Content Units (SCU's), which are semantically motivated, need not be consequent, and around sentential clause length. For example, given a sentence

> **Computers are used** to keep track of the immense volumes of patient data amassed, to obviate mountains of paperwork, and to cut approval times.

some of the SCU's that could be obtained are

```
Computers are used
Computers are used to keep track of immense volumes of patient data
Computers are used to obviate mountains of paperwork
Computers are used to cut approval times
```

Humans proceed to identify all possible SCU's from a given gold-standard summaries. Once all the summaries are annotated, humans then group the SCU's that are semantically equivalent to one another. As shown in Figure 2.5, each SCU is provided with a label, indicative of the information content common across each of its contributor(s). Each *contributor* in the shown example are semantically equivalent to the their label " *Computers are increasingly used to speed up clinical trial phase* ". In general SCU's labels, are assessor generated and verbose enough to minimize human error in assessment. In addition, each SCU is weighted based on the number of times it was included in all the human-generated summaries (e.g. weight = 3 in our example). This would reflect the importance of a SCU in view of the human summarizers.

```
<scu label="Computers are increasingly used to speed up clinical trial phase">
<contributor label="Here, also, computer technology is ... and more to speed up the process">
 <part label="Here, also, computer technology is used more and more to speed up the process" />
</contributor>
<contributor label="new timesaving computer processing equipment is being installed">
 <part label="new timesaving computer processing equipment is being installed" />
</contributor>
<contributor label="Computers are used...to cut approval times">
 <part label="to cut approval times"/>
 <part label="Computers are used" />
</contributor>
</scu>
```

Figure 2.5: A Summary Content Unit (SCU) along with contributors.

All weighted SCU's are arranged in a pyramid tier-style, where each tier consists of SCU's that have the same weight (indicating their prominence in terms of human summarizers). An *optimal* summary for a particular pyramid is obtained by selecting all the SCU's starting from the top most tier (say t_i) before including the SCU's from the next-most tier (t_{i-1}) and so on, till the summary limit is reached. Effectiveness of a system-generated summary is then compared against the optimal summaries, in terms of the number of top-tier SCU's that are covered by the system, over those covered by an ideal summary of same length.

Question Answering

Question Answering systems aim to provide the user with a direct answer to user query, contrary to the standard approach of displaying top 10 documents (snippets, headlines and hyperlinks). For example, user entering the query "capital of Canada?" could be satisfied from the response "Ottawa", without the need to scan through retrieved documents.

In general, question answering systems are designed to respond to *factoid* style of questions, which require clause-like response (or one sentence) or *list* style of questions such as

Query : Countries that neighbour Brazil ?
Answer: Argentina, Bolivia, Colombia ...

where the response consists of list of entities gathered from different document sources. In addition to the response, QA systems provide the supporting document(s) for the compiled result page. Human judges then rate the correctness of the answer along with the source they are extracted from. Most QA systems employ information retrieval techniques to extract a set of documents, and use deeper analysis methods like template extraction, syntactic parsing etc. to identify the exact answer from the retrieved document(s).

Since 2004, researchers have begun exploring *question series* (Voorhees [2004b]), which consists of a mixture of both factoid and list style questions (Table 2.2). Each question series is focused on a particular target (Christopher Reeve in the example shown). In general, target is either a named entity such as name of a place or person, or an event from the history. Each question in the series represents a *facet* or *aspect* of information that the user, in a dialogue with a QA system, would be interested to know about the topic. In addition to the factoid and list style of questions, researchers included "other" questions, which should be read as "find me any other facet of information about the target", whose response is bounded to certain length.

Type	Question
FACTOID	What year was Christopher Reeve paralyzed?
FACTOID	How many "Superman" movies did he make?
FACTOID	During what years were these "Superman" movies made?
FACTOID	Which actress co-starred in the most "Superman" movies with Reeve?
FACTOID	What year did Reeve commence his theatrical career?
LIST	List titles of movies, other than "Superman" movie that Christopher Reeve acted in
Other	-

Table 2.2: Question Series topic for the target: "Christopher Reeve"

In judging phase, human assessors were asked to judge the correctness of response for factoid and list kinds of questions. For "other" responses, judges first filter out the responses that provide information redundant with the factoid and list style questions. Next, assessors judge each response as either "vital", which are meant to provide important information, or "non-vital", whose information is not very important.

In addition to the judgements for the response of each question, NIST releases a file containing the regular expression patterns of the correct responses of each question. These patterns are constructed from the responses that have been judged as correct by human judges. Such patterns could be useful in approximately evaluating the correctness of retrieved responses for later experiments.

```
154.1 1995
154.2 (4|four)
154.6 Rear Window
...
```

Although, question series topics are compiled for the purpose of evaluating QA systems, they could symbolize the information needs users could have while searching about the target topic. For example, a user entering the query "Christopher Reeve" in a standard web search engine, could be interested in any one or more of the facets listed in Table 2.2. Lin [2007] made use of the existing test collections in order to compare the performance of a generic retrieval system with that of top-performing question answering systems. Lin compared the effectiveness of all the systems based on the number of unique facets that could be answered after browsing through top set of documents. Jimmy Lin concatenate the text from the top retrieved documents until a character limit ($\sim$ 1000 characters) is reached. Similarly, they retrieve the sentences containing the responses provided by QA systems and concatenating them until the maximum limit ($\sim$1000 characters) is reached. They proceed to compare both kinds of systems in terms of the recall of the facets at regular interval lengths ($\sim$50 character lengths). Jimmy Lin used the regular expression patterns to identify the facets covered from a given text. Jimmy Lin report that the top performing QA systems (based on official measures), would out-perform the IR-baseline system. However, it is reported that the IR baseline system outperforms the median QA system effectiveness.

2.7 Summary

In this chapter, we summarized existing methods and measures for determining effectiveness of ranking functions. We introduced various measures and also highlighted the shortcomings: lack

of support for novelty of information covered, lack of support for diversity of intents. As such, these measures cannot be used to evaluate ranking functions in context of web search. Our aim is to establish a framework capable of evaluating ranking functions in view of query ambiguity, and one that can accommodate novelty of information in a given document. In next chapter, we introduce one such framework that can be used to evaluate ranking functions in context of web search, by extending existing Cranfield paradigm to accommodate diversity among intents, and novelty of information retrieved.

Chapter 3

Evaluating Ranking Functions with Novelty and Diversity

In this chapter, we introduce a nugget-based framework for evaluating effectiveness of retrieval functions. Through our framework, we aim to provide researchers

- Ability to measure utility in terms of novel information retrieved, where novelty is measured in terms of lack of redundancy.
- Diversity in terms of different intents, associated with a given query, covered in the document. Diversity of intents covered in a given document is necessary in an attempt to tackle the ambiguity associated with user queries.

Our framework of evaluation is centered around an idea that information can be represented in terms of nuggets. Similar ideas were proposed by (as explained in Section 2.6) Nenkova et al. [2007] and Lin [2007]. Extending their idea of nugget, we consider a nugget in a broader perspective to encompass any binary property associated with a given document or information need. In terms of document properties, these could include features such as document size, content type, spamminess etc. Any feature could be represented as a nugget, as long as it can be judged on a binary scale. In terms of representing a query, each nugget is mapped to an underlying information need associated with that particular query.

We begin by explaining means to adapt nuggets to represent queries (along with their respective intents) and documents along with their feature set in our framework. Built on top of our framework, we then proceed to compute the probability of relevance of a document, accounting for novelty of information presented in a document (in comparison with previously seen documents), and diversity of intents (covered). Next, we derive two different effectiveness

measures based on established user models: gain-based model Järvelin and Kekäläinen [2002] and user persistence based model Moffat and Zobel [2008].

3.1 Query Representation

As mentioned before, we first consider our nugget framework to model a given query along with its set of associated nuggets. A query Q can be associated with infinite number of nuggets

$$Q = [Q_1, Q_2, \ldots \infty]$$

where each Q_i represents an underlying information need (or intent), a user U would associate with Q. Each information need can be of different type: informational, navigational or transactional (Broder [2002]).

For better understanding, let us consider consider an example query "**Harry Potter**" with associated information needs (Table 3.1). Any user entering the query could be interested in one or all of the associated intents. As listed in Table refex31, user could be interested in Harry Potter (Book series), and would like to find official site of the publisher. In another possible context, a user entering the "Harry Potter" could be interested in finding more about movies

Type	Intent
N	Official site for Harry Potter, the book series
N	Official site for Harry Potter, the movie
N	Box-office report on the 7^{th} movie
I	Critique of the book series
I	Critical reception of the 6^{th} movie
I	Author of the story book series
..	..
N	Fan club site

Table 3.1: Few possible intents associated with the query "Harry Potter". Here N indicates a *navigational* intent and I indicates an *informational* intent of need.

corresponding to the book series. Similarly, users could be interested in the critique of the book writings, or more information of the author of book series. Lacking any contextual information, to identify the underlying intent for an instance, it is reasonable to assume that a user submitting the query could be interested in some or all intents. It is therefore important to optimize and evaluate ranking function with an objective function to satisfy several intents to maximize the likelihood of satisfying majority of users.

However, it is not easy to obtain such intents under existing paradigm of evaluation. Under

existing paradigm, human assessors (considered renowned experts in a given topic) are invited to compose a topic for a given query. Extending would require human assessors to carry out extensive analysis to find different information needs possible for a given query. This could be tedious and cumbersome, and could not be replicated frequently. A viable alternative would be to use services such as Mechanical Turk [1], to seek possible intents from a large group of users, spread across the world. A third alternative is to mine existing query logs to study user interaction behavior to identify the various intents [2]. For now, we proceed to define our framework under the assumption that there exists a finite number of N intents that user would be interested in. Under such assumption(s), each query is represented as transforming the query Q into

$$Q = \{n_1, n_2, \ldots, n_N\} \tag{3.1}$$

where each nugget n_i represents a unique information need (Q_i). Next, we assign a probability value $Pr(n_i|Q,U)$, which is equivalent to the probability that a user U, representative of certain user population, would submit query Q with an underlying information need represented as n_i in our framework. Further, we assume that the set of nuggets identified are complete (or exhaustive) indication of information needs associated with a query Q.

$$\sum_{i=1}^{N} Pr(n_i|Q,U) = 1 \tag{3.2}$$

In absence of any information identifying the relative importance of a given nugget for a given query, we assume that each intent is of equal importance (γ) for a given query.

$$Pr(n_i|Q,U) = \gamma \tag{3.3}$$

In a given set of nuggets associated with a given query, user interest in one nugget may or may not be indicative of his/her interest in others. For example, user entering the query "Harry Potter" with an intent to find out more information of the 7^{th} movie — such as the critical reception, box office report, etc., may also be interested in knowing about Harry Potter book series, and its associated author.

3.1.1 Ambiguity and Underspecification

In general, user entered queries are short and rarely convey their underlying information need Jansen et al. [2000]. In this part, we classify query topics into

- Underspecified Queries

[1] `http://mturk.com`
[2] future work.

- Ambiguous Queries

based on extent of uncertainty in defining query intents. We then explain differing user expectations associated with both types of uncertainty.

By *underspecified queries*, we refer to those which can be subjected to one particular interpretation. However, there exists several *subtopics* or *aspects* associated with given query topic. A typical scenario would be a user submitting a query "Christopher Reeve", without being specific of an exact information, being interested in one of the possible intents shown in Table 2.2. One can assume that each question in the series represent a facet/aspect/subtopic of information that user entering "Christopher Reeve" could possible be interested in. We can therefore map each question from Table 2.2 to an nugget reflecting variety of underlying intents associated with users entering the query (Table 2.2).

```
<topic=1>
<query> Christopher Reeve </query>
    <subtopic=1> Year that Christopher Reeve was paralyzed? </subtopic>
    <subtopic=2> How many ‘‘Superman’’ movies did Reeve act in? </subtopic>
    <subtopic=3> Years when the ‘‘Superman’’ movies were made ?  </subtopic>
    <subtopic=4> Actresses starring with Reeve in ‘‘Superman’’ movies ? </subtopic>
    <subtopic=5> Year in which Reeve started his theatrical career ?  </subtopic>
    <subtopic=6> Other movies in which Reeve acted?  </subtopic>
</topic>
```

Figure 3.1: Example of an *underspecified query*, Christopher Reeve, along with its intents.

Any user entering the query "Christopher Reeve" could possibly be interested in few or all possible nuggets as shown in Figure 3.1. Similarly, a document retrieved for the query by a retrieval function could be relevant to few or all possible facets of information.

By *ambiguous queries*, we refer to those queries where a user-submitted query could be subjected to one or more interpretations. A typical query such as "**UPS**" could mean any one of the interpreted in one of the following interpretations:

1. United Parcel Service
2. Uninterrupted Power Supply
3. University of Puget Sound

where first interpretation refers to a package delivery service, while second interpretation refers to a device that provides power in case of emergency scenarios, where there is no input power. The

third interpretation refers to an educational institution, located in Tacoma, Washington. Each interpretation could contain a different set of intents as shown in Table 3.2. These could vary from finding the nearest drop-off point by or interested in tracking certain packet, and freight charges (for the first interpretation). In case user implies Uninterrupted Power Supply while submitting **UPS** she could be interested to find out about underlying technology involved in the working of a device, or a vendor to purchase it from in her locality, or comparison of different products available etc. The third one reflect users interested in information related to the educational institution, such as exam schedule, programs offered etc. User interest in a particular interpretation and its associated intents could depend on the context of usage.

Although, both ambiguous and underspecified queries represent underlying uncertainty in user queries, there exists difference in user expectations with differing level of uncertainty. In case of ambiguous queries, user interest in one particular interpretation could indicate lack of interest in other interpretations (or intents) associated with the query. In our previous example, this could mean that user interested in finding more information about an uninterrupted power supply, would not be interested in pages/documents related to parcel delivery service (UPS). However, user interest (in nuggets related to Uninterrupted Power Supply) could be independent of each other — i.e. user interested in finding out nearest store to purchase a device could also be interested in documents comparing different products (features vs price). On the other hand, while considering the various intents associated with a given query, *Christopher Reeve*, user interest in one of intent could be independent of her interest in other related intents. User may be interested in all of the intents or selective subset of intents. We consider this distinction in Section 3.7, in deriving effectiveness measures for ambiguous queries.

Interpretation	id	intent
United Parcel Service	n_1	finding the nearest UPS store
	n_2	UPS tracking number site
	n_3	freight charges ..
	n_4	homepage
Uninterrupted Power Supply	n_1	variety of brands available
	n_2	information about the technology
	n_3	comparison price vs features
University of Puget Sound	n_1	university home page
	n_2	degree programs
	n_3	industry tie-ups

Table 3.2: Different interpretations and intents associated with the query “ups”

3.2 Document Representation

In previous section, we have adopted our nugget based framework to represent queries along with their associated intents. We now adopt the notion of nuggets to represent various features associated with documents present in a collection. As mentioned at the beginning of the chapter, each nugget represents a feature associated with a particular document, which can be indicated on a binary scale, indicating its presence or absence. For example, following nuggets

```
Is the document size  greater than 200KB? [y|n]
Is the document spam ? [y|n]
```

can be used to represent if a given document is spam (or not) and if the same document is larger than 200KB. In addition to features such as document size, document type etc., one could expand the feature set to represent topicality (defined in Chapter 2 as the document is about a particular topic) of a document. The notion of topicality is similar to that followed by Cranfield Paradigm (Cleverdon [1962]), and adopted by various collection building exercises. For example, for the query " **Christopher Reeve**", one could assess topicality of a document by answering

```
Is the document about Christopher Reeve ?
```

which falls in-line with notion of topicality. Modifying each subtopic or intent associated with a given query topic, into a binary question, we transform each of intent shown in Figure 3.1, as following:

```
provide the year that Christopher Reeve was paralyzed ?
answer the number of ‘‘Superman’’ movies did Reeve act in ?
answer the years when the ‘‘Superman’’ movies were made ?
contain the list of actresses who co-starred with Reeve in ‘‘Superman’’ movie ?
contain the year in which Reeve started his theatrical career ?
...
```

which reflects extent to a particular document could satisfy user interested to know about "Christopher Reeve". In theory, a document feature set could include all attributes for all possible query topics. In current proposal, we consider only intents associated with a particular query topic for which a document is retrieve (i.e. bounded by the number of intents possible for a given query, N).

As mentioned above, features with respect to a given document can be obtained from human assessments, carried out on a binary scale. Assessor judgement ($J(d, n_i)$) of the of nugget n_i with respect to a document d

$$J(d, n_i) = \begin{cases} 1 & \text{if true} \\ 0 & \text{else} \end{cases} \tag{3.4}$$

reflects the presence of absence of a particular nugget in the given document. Previous studies Sormunen [2002] observed that relevance judgements, carried out by humans, are highly subjective in nature. Sormunen [2002] reported that only a fraction of documents judged as relevant were judged highly relevant in a follow-up experiment. Voorhees [1998] observed that assessors agrees less than 50% of the time in their judgements about the topicality of a given document. Given the subjective nature of human relevance assessments, we incorporate the notion of probability α into assessor judgments of nugget coverage in a given document.

$$Pr(n_i \in d) = \begin{cases} \alpha J(d, n_i) & \text{if true} \\ 0 & \text{else} \end{cases} \tag{3.5}$$

where any "true" assessment made by an assessor is factored with the certainty, α. It should be noted we associate such uncertainty with only "true" judgements of the assessors and not for the false ones. The value of α could be taken between $[0, 1]$ inclusive, where 1 implies that the assessor judgements can be taken with absolute certainty. In later derivations, we tweak the notion of α to signify the extent to which user information need associated with a given query, represented by nugget n_i is satisfied in document d.

Based on our framework to represent queries and documents in terms of nuggets, we proceed to derive an effectiveness measure, to reflect utility gained by a user while browsing through a ranked document list. Our measures are firmly rooted with principle of *Probability Ranking Principle* (PRP) (Robertson [1977b]), which suggests that overall effectiveness of a retrieval system is best obtainable if the system is capable of ranking documents in decreasing order of their likelihood of being relevant to user's information need. In judging relevance of a document with respect to an information need, we adapt notion that relevance of a document is proportional to overlap between information stored in and information being sought (Cooper [1971]). In the following sections, we first derive probability of relevance of a given document with respect to different information needs. We then derive a novelty-induced means to measure utility gained by a user, where utility is measured in terms of relevant information shown to a user that is not redundant.

3.3 Probability of Relevance

In our framework, we define probability of relevance based on two well established norms:

- A document is considered relevant, if it contains *any* information related to the topic of interest.
- Taken on a multi-grade notion of relevance, a document's probability of relevance reflects the extent to which a user information need could be satisfied from that particular document.

Based on the first norm, we derive probability that a document d is relevant ($Pr(r = 1, d|q, u)$, if there exists a user u entering query q with an intent i (represented as n_i) that is covered in by the document d.

$$Pr(r = 1, d|u, q) = P(\exists n_i (u \in U), n_i \in u \cap d)$$

which is equivalent to 1 *minus* probability that there exists no intersection between nuggets covered in a document and those related to the topic, i.e.

$$\begin{aligned} Pr(r = 1, d|q, u) &= 1 - \prod_{i=1}^{N} (1 - Pr(n_i \in d).Pr(n_i \in u)) \\ &= \sum_{i=1}^{N} (Pr(n_i \in d).Pr(n_i \in u)) - O(max(Pr(n_i \in u).Pr(n_i \in d)^2)) \\ &\approx \sum_{i=1}^{N} Pr(n_i \in u).Pr(n_i \in d) \end{aligned} \tag{3.6}$$

Substituting the values of $Pr(n_i \in u)$ from Eq. 3.3 and $Pr(n_i \in d)$ from Eq. 3.5, we obtain

$$Pr(r = 1, d|q, u) \approx \sum_{i=1}^{N} \gamma.\alpha.J(d, i) \tag{3.7}$$

i.e. the relevance of a document is proportional to the extent to which nuggets associated with the query are covered in the document. This conforms with notion of relevance of a document as proposed by Cooper [1971] and by Robertson [1977b], where document relevance is defined in expected to be proportional to overlap between information stored in a document and that user is currently interested in.

As a working example, let us consider top six documents retrieved by a hypothetical ranking function for the query "Christopher Reeve" (as shown in Table 3.3). In Table 3.4, we have each document marked with their nuggets covered, related to the the topic as listed in Figure 2.2. These nugget coverage were taken from human assessments compiled as part of assessor evaluation

Docno	Title
APW19980627.0871	Paralyzed 'Superman' still rescuing people
APW19981128.0831	Christopher Reeve returns to acting in Rear Window
APW19990514.0279	Reeve still unable to Walk
APW19990414.0334	Reeve to Speak at Williams College
APW19990604.0086	Actor Reeve Named to Research Board
NYT20000612.1332	Top New Stories

Table 3.3: Top 6 documents retrieved for query "Christopher Reeve"

in official QA track. The third column of the table, titled "topicality", reflects the value if the document contains at least one nugget related to the topic. This is similar to dichotomous nature of relevance employed in standard retrieval evaluation exercise, where a document is judged into either relevant or not.

In the last column, we compute $Pr(r = 1, d|q, u)$ for each document. In computing the probability of relevance, we assume that user is equally likely to be interested in all the nuggets. Taking γ and α to 0.2 and 1 respectively in Eq. 3.7, we compute probability of relevance of each document with respect to query topic, "Christopher Reeve". As expected, probability of relevance of each document is proportional to the number of nuggets covered in them.

Rank	Docid	Topicality	Nuggets					$Pr(r = 1, d\|u, q)$
			n_1	n_2	n_3	n_4	n_5	
1	APW19980627.0871	1	0	1	0	0	0	0.2
2	APW19981128.0831	1	1	1	0	0	0	0.4
3	APW19990514.0279	1	1	1	1	0	0	0.6
4	APW19990414.0334	1	1	0	0	0	1	0.4
5	APW19990604.0086	1	0	1	1	0	0	0.4
6	NYT20000612.1332	1	1	0	0	0	0	0.2

Table 3.4: Documents with nuggets (covered) with respect to the query "Christopher Reeve"

In computing the probability of relevance for each document, we reflect the extent to which a document would satisfy the information needs associated with a given query. However, so far, we have considered each document in an independent fashion. Cooper [1971] stated that treating documents retrieved in a independent fashion, would imply that user interest in a given information does not decrease irrespective of the number of times she has come across it. Further, Cooper stressed need to differentiate between relevance (logical) and utility towards a user. As defined by Cooper, logical relevance demonstrates the extent to which a document is related to topic information. Utility, as mentioned by Cooper, should reflect added benefit from a user's perspective in terms of information/knowledge gained to complete her information need. Since

it is impossible to measure knowledge gain, Cooper measured relative gain of information in comparison of the previously seen (higher ranked) documents.

Xu and Yin [2008], in their user studies report that human assessors notion of document relevance correlates well with respect to topicality and novelty of the information covered in the document. Previous attempts in quantifying the novelty of information present in a given document was carried out on a sentence level. Human assessors were asked to identify the novel sentences in a given document with respect to previously seen sentences. Next, we derive a novelty-induced means to compute utility gained by a user.

3.4 Novelty Induced Utility

> *It is their (retrieval systems) utility, not their beauty, complexity or modernity, which counts* — Cooper [1971]

In our framework, we measure novelty induced utility of a document in terms of number of non-redundant nuggets user may come across in it. Here, we define probability of a user being interested in a given nugget n_i,

$$Pr(n_i \in d_k | d_1, d_2, d_3, \ldots d_{k-1})$$

in context of previously seen $(k-1)$ ranked documents.

Modifying Eq. 3.6, we compute probability of relevance of a document d_k, taking into account the fact that user has gone through the previous $k-1$ documents as

$$Pr(r, d_k | Q, U) = \sum_{i=1}^{N} Pr(n_i \in U).Pr(n_i \in d_k | d_1 \ldots d_{k-1}) \tag{3.8}$$

where $Pr(n_i \in d_k | d_1 \ldots d_{k-1})$ is meant to reflect interest (or lack of it) in a nugget n_i covered in document d_k, under assumption that user has read through previous $k-1$ documents.

Following Cooper [1971], we assume that user interest in a particular information decreases depending on the number of times she has seen it, while browsing through the ranked list of documents. We reflect this by discounting utility accumulated by an user, from viewing a document covering a particular nugget. If information is *novel*, i.e. none of the previous documents contain it, it contributes more towards the utility gained from that particular document, as compared to the document covering a nugget already seen in previous documents.

$$Pr(n_i \in d_k | d_1, d_2 \ldots d_{k-1}) \propto Pr(n_i \notin d_1 \ldots d_{k-1}) \tag{3.9}$$

which could be simplified (given that probability of a nugget covered in a document is independent) into

$$\prod_{j=1}^{k-1} Pr(n_i \notin d_j) \tag{3.10}$$

and further simplifying it, in terms of the probability that the nugget n_i is covered in previous documents

$$\begin{aligned} Pr[n_i \in d_k | d_1, d_2, d_3, d_4 \dots d_{k-1}] &= \prod_{j=1}^{k-1} Pr(n_i \notin d_j) \\ &= \prod_{j=1}^{k-1} (1 - Pr(n_i \in d_j)) \end{aligned} \tag{3.11}$$

Substituting the value of $Pr(J(d_j, n_i))$ in Eq. 3.11, we obtain the probability of relevance attributed to the document d_k from covering a nugget n_i as

$$\begin{aligned} Pr[n_i \in d_k | d_1, d_2, d_3, d_4 \dots d_{k-1}] &= \prod_{j=1}^{k-1} (1 - \alpha) \\ &= (1 - \alpha)^{C(k, n_i)} \end{aligned} \tag{3.12}$$

where $C(k, n_i)$ is equal to the frequency of the occurrence of nugget n_i in top $k-1$ ranked documents, assumed to have been gone through by a user.

$$C(n_i, d_1 \dots d_{k-1}) = \begin{cases} \sum_{j=1}^{k-1} J(d_j, n_i) & \text{if k } > 1 \\ 0 & \text{k} = 1 \end{cases} \tag{3.13}$$

i.e. we discount utility gained from having a nugget covered in document, by a factor of $(1 - \alpha)$. Let us assume that a nugget n_1 is covered in both documents d_1 and d_2. Taking the value of $\alpha = 0.5$, utility gained by documents d_1 and d_2 from covering the nugget would be 1 and 0.5 respectively.

On closely observing the discount factor $(1 - \alpha)$, one could equate it to extent to which a user would be interested in a nugget n_i after current document. If α is equal to 1, which is to state that we are absolutely certain of user judgement of nugget coverage in the document. In such case, $1 - \alpha$ would be 0 — i.e user is no longer interested in the nugget and that any document covering it would not provide additional utility gain.

Substituting values from equations Eq. 3.13 and Eq. 3.3 in Eq. 3.11, we obtain utility or relevance of a document d_i with respect to a given user U, who have submitted the query Q with with an assumption that the user has gone through the previous $(k-1)$ documents ($[d_1, d_2 \dots d_{k-1}]$)

as

$$= \gamma \sum_{i=1}^{N} (1-\alpha)^{C(k,n_i)} \tag{3.14}$$

Returning back to the previous example, we re-compute the probability of relevance of a document taking into account novelty based utility, for those documents retrieved for the query "Christopher Reeve". Taking $\gamma = 0.2$ as before, and $\alpha = 0.5$, i.e. the user is 50% interested in the nugget after the first occurrence, 25% after the second occurrence and so on, the utility based probability values are shown in Table 3.5 .

Rank	Doc	topicality	Nuggets					$Pr(d_k \mid d_1 \dots d_{k-1})$
			n_1	n_2	n_3	n_4	n_5	
1	APW19980627.0871	1	0	1	0	0	0	0.2
2	APW19981128.0831	1	1	0.5	0	0	0	0.3
3	APW19990514.0279	1	0.5	0.25	1	0	0	0.35
4	APW19990414.0334	1	0.25	0	0	0	1	0.25
5	APW19990604.0086	1	0	0.125	0.5	0	0	0.03
6	NYT20000612.1332	1	0.125	0	0	0	0	0.02

Table 3.5: Probability of Relevance taking into account the Novelty if information covered

Using novelty-based-utility relevance values of each document in retrieved set, we proceed to compute two effectiveness measures based on the following two user models. They are:

- α- Normalized Discounted Cumulated Gain (α-nDCG@k), where effectiveness of a retrieved set is measured in terms of user browsing through top k ranked documents following gain based model Järvelin and Kekäläinen [2002].
- Novelty-and Ranked-Biased Precision (NRBP), where effectiveness is factored based on user persistence Moffat and Zobel [2008].

3.5 α-nDCG

Our first approach proposal to compute effectiveness measures is to adapt gain based measures as proposed by Järvelin and Kekäläinen [2002].

Since our notion of document relevance is synonymous with the relative utility gain from going through the document, we we could easily plug-in the gain vectors from a given ranked set into measures proposed in Järvelin and Kekäläinen [2002]. We first construct α-g vector

$$\alpha\text{-g} = [\alpha\text{-g}_1, \alpha\text{-g}_2, \dots \alpha\text{-g}_k]$$

where α-g$_i$ indicates the relative gain of the document in rank i. In the previous example, α-g vector for the top 6 retrieved documents (Table 3.5) is

$$\alpha\text{-g} = [0.20, 0.30, 0.35, 0.25, 0.03, 0.02]$$

Corresponding to cumulated gain at rank k, as proposed in Järvelin and Kekäläinen [2002], we compute α-cg@k value as

$$\alpha\text{-cg@}k = \sum_{i=1}^{k} \alpha\text{-g}_i$$

as sum of α-g values for first k ranked documents. The values of such vector computed for previous example is

$$\alpha\text{-cg@}k = [0.20, 0.50, 0.85, 1.10, 1.13, 1.15]$$

Corresponding to DCG@k measure, which aims to discount the relevance gained from documents seen at lower ranks, we propose α-dcg@k as

$$\alpha\text{-dcg@}k = \sum_{i=1}^{k} \frac{\alpha\text{-g}_i}{log_2(1+i)} \tag{3.15}$$

here α-g$_i$ represent utility of a document at rank i and $log_2(1+i)$ is the discount factor to reflect the measure of work done by users in reaching the document. Computing the values of α-dcg@$k[k = 1, 2 \ldots 6]$ for our example values from Table 3.5

$$\alpha\text{-dcg@}k = [0.20, 0.39, 0.57, 0.68, 0.69, 0.76]$$

3.5.1 Ideal Gain ($\alpha DCG'$)

Similar to DCG, α-dcg@k computed above is unbounded and needs to be normalized to be able to average the value across topics. Järvelin and Kekäläinen [2002] constructed an *Ideal* ranked result set $\mathcal{I}$,

$$\mathcal{I} =< r_1, r_2, r_3, r_4 \ldots r_R >$$

constructed based on known R relevant documents (judged by the assessors) for a given query topic, sorted in descending order of their relevance, i.e. $r_i \geq r_j$, $\forall i > j$. Järvelin and Kekäläinen then computed corresponding CG'@r DCG'@r for such Ideal ranked set $\mathcal{I}$ to be used for normalizing ranked set of documents. Identifying such an ideal ranked set$\mathcal{I}'$, and computing $\alpha DCG'$@k measures is however not straightforward, as described below.

An ideal ranked result set taking into account the various nuggets covered by a document, and covered over a group of documents, would be the minimum number of documents that would

cover all the nuggets. Let us construct a graph $G = (V, E)$ from the known set of assessor judgements, where $V = \{v_1, v_2 \dots v_n\}$ are represent those documents that have been judged as relevant with covering at least one related nugget. Edge set $E = \{e_1 \dots ep\}$ consists of edges where e_i connecting two vertices $\{u, v\}$ such that $(u, v) \in V$ and there exists a nugget $n_i \in n_{1 \dots N}$ associated with the topic covered by both documents represented by (u, v). The task of finding an ideal ranked set would be equivalent to identifying the minimum set of vertices $V' \subseteq V$, $|V'| \leq K$, where $K \leq |V'|$ such that for each edge $\{u, v\} \in E$, at least one of u or v are belongs to V'. The task is equivalent to the well known "Minimum Vertex Cover" problem, which is NP-Complete.

There exists a greedy approximation algorithm to construct such minimum vertex cover. The approach starts with selecting the document with the highest incidence number, in terms of nuggets shared with other documents. Once the vertex v is selected, all other vertices $u \in V$ that share an edge with v (i.e. $(u, v) \in E$) are removed from the vertex set i.e. $V - \{u\}$. In next step, the algorithm selects the vertex with maximum incidence among the remaining. Iteratively, we select the vertices until all the edges are covered. The approximation ratio of such an greedy-approximation algorithm, measured in terms of

$$\frac{|V'| \text{from greedy approximation}}{|V'| \text{obtained by optimial}}$$

is $1 + log|V|$.

Let us assume that the known relevant document set for our running example "Christopher Reeve" consists of the 6 documents, as shown in Table 3.5. Based on the nuggets covered in each of the document, the ideal result set computed would be as follows: In the first iteration, the document with most number of nuggets covered i.e. docid "APW19990514.0279". Once added to the ideal ranked set, we recompute the α-dcg values for the remaining documents . The values for the remaining documents would then be

```
APW19980627.0871  0.5/(ln3)
APW19981128.0831  1/(ln3)
APW19990414.0334  1.5/(ln3)
APW19990604.0086  1/(ln3)
NYT20000612.1332  1/(ln3)
```

where the denominator $ln3$ reflects the discount factor corresponding to rank 2. By greedy approach, we select $APW19990414.0334$ and compute the $\alpha DCG@3$ values of the remaining 4 documents

```
APW19980627.0871 0.50/ln4
```

```
APW19981128.0831 0.75/ln4
APW19990604.0086 1/ln4
NYT20000612.1332 0.25/ln4
```

where we select the document $APW19990604.0086$. Continuing in this fashion, we pick until all relevant documents in decreasing order of their αDCG values.

```
APW19990514.0279  0.60
APW19990414.0334  0.19
APW19990604.0086  0.10
APW19981128.0831  0.05
APW19980627.0871  0.01
NYT20000612.1332  0.01
```

3.5.2 α-nDCG@k

Using the above computed α-DCG'@k values computed based on an ideal ranked result constructed in a greedy fashion, we compute α-nDCG@k values by

$$\alpha\text{-nDCG@}k = \frac{\alpha\text{-DCG@}k}{\alpha\text{-DCG}'\text{@}k} \tag{3.16}$$

which for our example would come $\frac{0.76}{0.90}$ to 0.84.

Sub-modularity of αDCG@k The greedy approximate algorithm to construct an ideal result set is *sub modular*. As defined in Nemhauser et al. [1978], assuming that there exists a real-valued function $Z(S)$ of th type

$$\max_{S \subseteq N} \{Z(S) : |S| \leq K\} \tag{3.17}$$

which is aimed finding the subset S of cardinality $|S| \leq K$, where K is a value $< n$. The above mentioned function is defined as *sub modular* function if it satisfies the two properties as shown in Eq. 3.18 and Eq. 3.19 respectively.

$$Z(S \cup \{k\}) - Z(S) \geq 0, \forall S \subset N \wedge k \in (N - S) \tag{3.18}$$

The first constraint requires that the function $Z(.)$ is always a non-decreasing function when adding a new element to existing subset.

$$Z(R \cup \{k\}) - Z(R) \geq Z(S \cup \{k\}) - Z(S), R \subset S \subset N, k \in N - S \tag{3.19}$$

The second constraint indicates that the relative gain of adding a new element k to an existing subset $R \subset N$ is either greater than or equal to the marginal gain for adding the same element k to another subset $S \subset N$, when $R \subset S \subset N$.

α-dcg is non-decreasing The function α-dcg'@k is a nondecreasing function. The value always increases by adding any relevant document or stays the same for non-relevant document.

Constraint 2 Let us assume two subsets of relevant documents set I_1 and set I_2, with the condition that $I_1 \subset I_2 \subset Ir$, where Ir represents the total number of relevant documents for the given topic. Let document d_p , $d_p \notin I1 \wedge d_p \notin I2$, be the next document being selected. Since I_1 and I_2 are constructed on a greedy approximation methods, based on their α-dcg scores, it relative gain in α-dcg from adding d_p to I_1 is greater than or equal to relative gain from adding same document to set I_2.

3.6 NRBP

In this section, we follow the user model introduced by Moffat and Zobel [2008] to compute utility gained by a user in terms of *user persistence*. Moffat and Zobel defined *user persistence* (β) as patience exhibited by a user while browsing through a ranked set of documents. After browsing through each document, users are expected to jump to the next document, in the ranked list, with a probability of β or abandon her interaction with a probability of $(1 - \beta)$. User persistence varies from 0, representing a highly impatient users interested only in top ranked document, to 1, representing a highly patient user willing to browse through infinite number of documents. Moffat and Zobel explored different values of β and observed that the values of 0.85 to agree with system comparisons done in previous TREC exercises. It should be pointed that user persistence, as assumed by Moffat and Zobel, is static in nature and does not vary in terms of the documents seen or relevance of current document.

Based on their user persistence model, Moffat and Zobel [2008] measured utility gain from going through a document d_k as a product of document relevance, r_k, (with respect to the given topic) and probability that user would actually read the document at rank k (β^{k-1}) (i.e. $r_k.\beta^{k-1}$). Total utility gain from browsing through large infinite number of documents is

$$\sum_{k=1}^{\infty} r_k.\beta^{k-1} \tag{3.20}$$

Moffat and Zobel then normalize total utility gained, from going through all retrieved documents,

by maximum number of relevance one can encounter .

$$\frac{1}{1-\beta} \tag{3.21}$$

Combining both Eq. 3.20 and Eq. 3.21, would give rise to Rank Biased Precision (RBP)

$$(1-\beta).\sum_{k=1}^{\infty} r_k.\beta^{k-1} \tag{3.22}$$

In our approach, we attempt to derive the effectiveness measures via alternate means. Unlike Moffat and Zobel [2008], we normalize expected utility gain by computing expected utility gain from going through an ideal result set. As in previous section, let us assume that there is an ideal ranked result set, which is comprised of all relevant documents (to that particular topic) belonging to the corpus ahead of the rest (non relevant).

$$\mathcal{R}' =< 1, 1, 1, ..., 1, 0, 0, 0, ... > \tag{3.23}$$

Let us assume that there are R relevant documents for a given topic in entire corpus, which imposes the following condition on set $\mathcal{R}$

$$\mathcal{R}'[k] = \begin{cases} 1 & \text{if } k \leq R \\ 0 & \text{otherwise} \end{cases} \tag{3.24}$$

Computing the expected utility gain for such an ideal result set $\mathcal{R}$, where the top R documents are relevant

$$\sum_{k=1}^{R} \beta^{k-1} = \frac{1-\beta^R}{1-\beta}. \tag{3.25}$$

which is equal to the total expected utility while browsing an ideal ranked list, with top R documents being relevant. Using total expected utility obtained from an ideal ranked result, Eq. 3.25, to normalize, the expected utility of a system generated ranked list, Eq. 3.20, we get a normalized expected utility gain

$$\frac{1-\beta}{1-\beta^R}\sum_{k=1}^{\infty} r_k\beta^{k-1}. \tag{3.26}$$

However, one needs to know the exact number of relevant documents R present in a given corpus, for the given query topic, to simplify $1-\beta^R$ in the denominator. Given the number of documents present in current day test collections, it is impractical to assume that all the (relevant) documents could be identified. There exists different methods to estimate the number of relevant documents via selective sampling approaches. However, denominator $1-\beta^R$ would

be significant if and only if the value of R is small. We consider the presence of an "ideal ideal" result list $\mathcal{R}''$ consisting of infinite number of relevant documents.

$$\mathcal{R}'' = \langle 1, 1, 1, ... \rangle. \tag{3.27}$$

which when used in computation of ideal gain as in Eq. 3.26, the denominator becomes $\simeq 1$ and our expected utility gain is similar to RBP as derived by Moffat and Zobel [2008]

$$\text{RBP} = (1 - \beta) \sum_{k=1}^{\infty} r_k \beta^{k-1}, \tag{3.28}$$

The advantage of normalizing system utility gain using expected utility gain from an ideal result is that such framework could easily be extended to graded or probabilistic relevance values. In this case, expected utility gain in Eq. 3.20 could then modified to

$$\sum_{k=1}^{\infty} r_k . \beta^{k-1}, \ (0 \leq r_k \leq 1) \tag{3.29}$$

An ideal result, with $0 \leq r_k \leq 1$, would be to have a ranked result set where the documents are sorted in decreasing order of their probability of being relevant

$$\mathcal{R}' = \langle r'_1, r'_2, r'_3, ... \rangle, \tag{3.30}$$

where ${r_k}' \geq {r_{k+1}}', \forall {r_k}'$. Normalizing Equation 3.20, with the ideal ranked set (sorted in decreasing order) we get RBP computed for probabilistic relevance values as

$$rbp = \frac{\sum_{k=1}^{\infty} r_k \beta^{k-1}}{\sum_{k=1}^{\infty} {r_k}' \beta^{k-1}}. \tag{3.31}$$

As before, we assume the presence of an "ideal ideal" ranked list (Eq. 3.27), where there exists an infinite number of relevant documents, the above equation Eq. 3.31 becomes equivalent to RBP value computation as proposed by Moffat and Zobel Eq. 3.28.

3.6.1 Novelty-based RBP (NRBP)

As mentioned while deriving system effectiveness in terms of user persistence based user model, our approach of normalizing using ideal result set would allow us to extend the value to probabilistic values of relevance. We now proceed to compute relevance of a document in terms of relative gain of utility (factoring in novelty of information covered) and discount factor depending on

user persistence

$$= g_k.\beta^{k-1} \tag{3.32}$$

where g_k is equal to utility gained from going through document k, as computed by Eq. 3.14 and β^{k-1} represents discount in terms of user persistence to read through top k ranked documents.

Returning to our "Christopher Reeve" example, we present in Table 3.6 utility gained from browsing through each document, computed with values of α, β and γ as 0.5 and 0.85 respectively. Total utility gain by a patient user, after the top 6 documents is 0.8799 in our example.

To normalize utility gained from browsing through system retrieved documents, we substitute g_k in place of r_k in Eq. 3.31, to derive (canceling γ, common to both numerator and denominator)

$$\frac{\sum_{k=1}^{\infty} g_k \beta^{k-1}}{\sum_{k=1}^{\infty} g_k{}' \beta^{k-1}}, \tag{3.33}$$

where the denominator $g_k^{'}$ indicates utility gain from browsing through an ideal ranked result, taking into account novelty of information covered. As mentioned in previous section, task of compiling an ideal ranked result is NP Complete. In previous section, we obtained ideal result in a greedy manner, by sorting documents in decreasing order of their utility gain.

We now derive based on assumption of an "ideal ideal" result set that, considering the notion of nuggets, would refer to a set of documents $\mathcal{I}''$, where each document covers all possible nuggets related to the given query topic.

$$\mathcal{I}'' = [N, N, N, N, N, \ldots]$$

Under such circumstances, the gain of relevance for an ideal ideal document at rank k, computed by Eq. 3.14 would be

$$= (1-\alpha)^{k-1}.N \tag{3.34}$$

where N is the number of the nuggets for the query. Constructing a gain vector for such an ideal ideal list,

$$\mathcal{G}'' = \langle g''_1, g''_2, g''_3, ... \rangle = \langle N, (1-\alpha)N, (1-\alpha)^2 N, ... \rangle. \tag{3.35}$$

Now,

$$\sum_{k=1}^{\infty} g_k{}'' \beta^{k-1} = N \sum_{k=1}^{\infty} \left((1-\alpha)\beta\right)^{k-1} = \frac{N}{1-(1-\alpha)\beta}. \tag{3.36}$$

Substituting the above equation (Eq. 3.36) and Eq. 3.14 in Eq. 3.33, we obtain Novelty and Rank-Biased Precision (NRBP).

$$\text{NRBP} = \frac{1-(1-\alpha)\beta}{N} \sum_{k=1}^{\infty} \beta^{k-1} \sum_{i=1}^{N} J(d_k, i)(1-\alpha)^{C(k,i)}. \tag{3.37}$$

Returning to our "Reeve Example" (Table 3.4), NRBP value of the shown 6 ranked documents is equal to 0.538 (Table 3.6).

Rank	Doc	rel	Nuggets					$g'_k.\beta^{k-1}$
			n_1	n_2	n_3	n_4	n_5	
1	APW19980627.0871	1	0	1	0	0	0	0.2
2	APW19981128.0831	1	1	0.5	0	0	0	0.25
3	APW19990514.0279	1	0.5	0.25	1	0	0	0.2528
4	APW19990414.0334	1	0.25	0	0	0	1	0.1535
5	APW19990604.0086	1	0	0.125	0.5	0	0	0.0156
6	NYT20000612.1332	1	0.125	0	0	0	0	0.008

Table 3.6: Utility gain corresponding to top 6 documents retrieved for the query "Christopher Reeve", taking into account user persistence.

3.7 Ambiguous Queries

Until now, we have derived effectiveness measures to measure system performance for underspecified queries, i.e, we have been working under the assumption that user interest in one intent does not reflect any influence over her interest in related intents. Such underlying assumption would not apply for *ambiguous queries*, where user interest among given set of intents is not independent of each other. As introduced in Section 3.1, a simple query like "UPS", can be interpreted in one of the following ways:

- United Parcel Service
- Uninterrupted Power Supply
- University of Puget Sound

where each interpretation is associated with possibly divergent information needs, as shown in Table 3.2. It is safe to assume that users with information needs associated with one interpretation would not be interested in information needs associated with different interpretations. For example, user submitting the query "UPS" to track a particular delivery (he is expecting) would not be interested in finding information about pages relevant to Uninterrupted Power Supply. An ideal result page should balance the results from different interpretations, in decreasing order of their interest among users.

Intent Aware Measures

Agarwal et al. [2009], prior to our NRBP measure, proposed to optimize ranking function on the web taking into account ambiguity associated with queries. In their model, both documents and queries are associated with one or more categories, of a taxonomy. Each category in the taxonomy is indicative of a particular user intent. For instance, given the query "UPS", there are three categories named "United Parcel Service", "Uninterrupted Power Supply" and "University of Puget Sound". Agarwal et al. do not distinguish between the various intents further associated with the UPS as "United Parcel Service" interpretation. In other words, both ups tracking page and page containing the local drop-off store are considered as relevant to the "United Parcel Service" category ($c(q)$).

Agarwal et al. [2009] assume that there exists a taxonomy listing all categories that could be associated with a given query. Assuming that a query is associated with $1 \dots M$ categories, and $P(c_j|q)$ is probability that user entering the query q, would be interested in category represented as c_j.

$$\sum_{j=1}^{M} P(c_j|Q) = 1 \tag{3.38}$$

In a similar fashion, each document is assumed to cover different categories, $c(d)$. Assuming the $V(d|q, c_j)$ as quality of a document d with respect to category c_j, Agarwal et al. aim to optimize the ranking function by maximizing

$$\sum_{c} P(c|q).(1 - \prod_{d \in S}(1 - V(d|q, c))) \tag{3.39}$$

Agarwal et al. measure effectiveness of a retrieved document set, with respect to each category associated with the query. Agarwal et al. then combine them with category probabilities $P(q_j|Q)$ to obtain Intent-Aware(IA) version of corresponding adhoc measure. From their definition, Intent-Aware MAP (MAP-IA), is obtained by

$$MAP - IA = \sum_{j} P(q_j|Q) MAP(Q, k|j) \tag{3.40}$$

where $MAP(Q, k|j)$ refers to the standard MAP value computed at rank k, by treating only those documents that covered category j as relevant. Similarly, Intent-Aware NDCG(NDCG-IA) can be obtained by

$$NDCG - IA = \sum_{c} p(c|q).NDCG(Q, k|c) \tag{3.41}$$

However, Agarwal et al. do not distinguish beteween different sub-topics, aspects of a categories. Next, we derive our NRBP measure for ambiguous queries.

Let us consider the query "UPS" and its associated set of intents as seen in Table 3.2. In adopting user model proposed by Agarwal et al., we map each interpretation to one *category* and each nugget associated with each category as *subtopic* associated with that category, as shown in Figure 3.2. Similar to Agarwal et al., we assume that there exists a way to assign relative weights to each of the category. In our UPS example, this means that user entering the query UPS has a 50% chance of being interested to find out about "United Parcel Service", 30% likely to be interested in intents associated with "Uninterrupted Power Supply", and around 20% of users could be interested in "University of Puget Sound" related intents.

In judging relevance of a document, each retrieved document is first assessed as to what category the document belongs to. Once a document is found relevant with respect to a given category, the document is not expected to be relevant with respect to other categories (exclusive). Following the identification of the category, we then seek relevance of the document with respect to intents belonging to that particular category. We believe that this would be of same effort as the underspecified query, since we restrict each document to only one category.

Assuming that the ambiguous query has $1 \dots M$ categories, we compute NRBP value as

$$NRBP = \sum_{j=1}^{M} P(j|Q).NRBP_j \tag{3.42}$$

where $NRBP_j$ is the value with respect to the j^{th} category obtained as

$$NRBP_j = \frac{1-(1-\alpha)\beta}{N_j} \sum_{k=1}^{\infty} \beta^{k-1} \sum_{i=1}^{N_j} J(d_k, j, i)(1-\alpha)^{C(k,j,i)}. \tag{3.43}$$

where N_j represents number of nuggets present for j^{th} interpretation of the query Q, $J(d_k, j, i)$ represents binary value, which is equal to 1 if document d_k covers i^{th} nugget of j^{th} interpretation. As observed, utility gain function (g_k) Eq. 3.14 would now be equivalent to

$$\sum_{i=1}^{N_j} (1-\alpha)^{(C(j,k,i))} \tag{3.44}$$

$C(k, j, i)$ represents frequency of times a nugget i of the j^{th} interpretation is covered in top $k-1$ ranks. Combining both Eq. 3.43 and Eq. 3.42, we obtain NRBP as

$$(1-(1-\alpha)\beta) \sum_{k=1}^{\infty} \beta^{k-1} \sum_{j=1}^{M} \frac{p_j}{N_j} \sum_{i=1}^{N_j} J(d_k, j, i)(1-\alpha)^{C(k,j,i)}. \tag{3.45}$$

The above value of NRBP Eq. 3.45 would however never be equal to one, as the interdepen-

```
<topic>
  <query> UPS </query>
  <category number=1 probability=0.50>
    <description> United Parcel Service </description>
    <subtopic number=1> Nearest UPS store </subtopic>
    <subtopic number=2> UPS parcel tracking site</subtopic>
    <subtopic number=3> Homepage of UPS store Canada </subtopic>
    <subtopic number=4> Freight charges for within Canada shipping </subtopic>
  </category>
  <category number=2 probability=0.30>
    <description>Uninterrupted Power Supply </description>
    <subtopic number=1> How does it work ? </subtopic>
    <subtopic number=2>Where can one buy UPS device in Waterloo?</subtopic>
    <subtopic number=3> Price versus feature comparison of different </subtopic>
  </category>
  <category number=3 probability=0.2>
   <description> University of Puget Sound </description>
    <subtopic number=1> Homepage of the university </subtopic>
    <subtopic number=2> Degree programs offered </subtopic>
    <subtopic number=3> Application deadlines </subtopic>
 </category >
</topic>
```

Figure 3.2: Hypothetical evaluation topic for the query "UPS", with its various categories and subtopics.

dency (or mutual exclusiveness — i.e. user is interested in only one category and her interest in one category equates to the lack of interest in other categories). Unlike Agarwal et al., who overlook such normalization, we construct an ideal ranked result to construct the normalization factor. Since ideal document would not consist of all nuggets with respect to all possible interpretations, we construct is as follows: Starting with first rank, we select ideal documents that are relevant to most likely interpretation of a query. Once the top ranked document is selected the next document is chosen

$$\underset{1 \le j \le M}{\operatorname{argmax}} \left(p_j (1-\alpha)^{D(k,j)} \right), \tag{3.46}$$

where p_j is probability that user entering query q would choose j^{th} interpretation, $D(k,j)$ is number of "ideal ideal" documents from the category j seen in ranks $1 \ldots (k-1)$. From our UPS example, we have three categories, say $U1$, $U2$ and $U3$, with probabilities 0.50, 0.3 and 0.2 respectively, the ideal result set could then obtained as follows: At rank 1, an ideal document with respect to $U1$ is selected, followed by a document from $U2$ at rank 2, followed by another

document from $U1$ at rank 3, and so on.

$$\mathcal{I}'' =< U1, U2, U1, U3, U2, U1 \ldots > \tag{3.47}$$

Let $I(\alpha, \beta, p_1, ..., p_M)$ be the ideal value calculated by the procedure above. The final version of our NRBP measure is then:

$$\text{NRBP} \;=\; \frac{1-(1-\alpha)\beta}{I(\alpha,\beta,p_1,...,p_M)} \sum_{k=1}^{\infty} \beta^{k-1} \sum_{j=1}^{M} \frac{p_j}{N_j} \sum_{i=1}^{N_j} J(d_k, j, i)(1-\alpha)^{C(k,j,i)}. \tag{3.48}$$

3.8 Discussion

In this chapter, we outlined a framework of evaluation and proposed evaluation measures built on top of our framework, to evaluate effectiveness of ranking function. Our framework is built around the notion of *nuggets* to represent both queries and documents of a collection. On query side representation, a nugget represents an underlying intent or information need that a user could have while submitting the query. On document side, a nugget represents attributes of features of a document.

We then adopted the classic probability ranking principle to define probability of a document proportional to nuggets covered in a document that user entering the query could potentially be interested in. We then augmented the notion of information novelty in terms of the nuggets covered by a document, in context of previously seen documents. Utility gained by a user from going through a document would then reflect the extent to novel information a user could encounter. In our measures, we reward novelty, by discounting utility gained from redundant information.

Next, we describe preliminary exercise carried out while defining our framework and evaluation measures. We then propose means to compute expected effectiveness measures to overcome the problem of missing qrels (as explained in Chapter 2).

Chapter 4

Preliminary Exercise, Web Track, and Related Measures

In previous chapter, we introduced our nugget-based framework that can accommodate multi-intent representation for a query. In our framework, nuggets are meant to represent an underlying intent of possible interest to a user submitting a query. We then derived two effectiveness measures, $\alpha-$nDCG and NRBP, which reflects utility gained by a user, while browsing through a given set of retrieved documents. In this chapter, we report a preliminary exercise conducted using a simulated test collection to demonstrate functioning of our framework and associated effectiveness measures (Section 4.1). In Section 4.3, we aim to capture loss of novelty in pseudo-relevant feedback approaches. We then explain a large-scale evaluation exercise currently being undertaken to compare ranking functions across web, modeled on our framework.

4.1 Preliminary Exercise

Due to lack of a large-scale test collection (prior to our work), we transform an existing test collection [1] to simulate evaluation of few well-known ranking functions, in terms of our proposed measures (α-nDCG and NRBP). Our objective behind this exercise is to demonstrate the functioning of our proposed framework of evaluation. We wish to demonstrate effect of system effectiveness of few well known baseline and pseudo-relevant feedback approaches, with varying discount factors to penalize for redundant information.

For this exercise, we choose data (topics and relevance assessments) compiled for Question Answering 2006 Track (QA2006) Dang et al. [2006] [2]. A total of 75 topics were developed for that

[1] constructed for a Question Answering Track(s)

[2] `http://trec.nist.gov/data/qa/t2006_qadata.html`

particular track. Each topic consists of a series of questions all related to particular *target*. In general, the target chosen is either a named entity like person, place, organization or an event supposed to have taken place over certain time period. Each topic is constructed to simulate dialogue between a user, who is interested about a target, and an automated question answering system. For example, in Figure 4.1, we can see different questions centered around a target "Christopher Reeve". Each question is representative of certain aspect of information that user will be interested when seeking information about "Christopher Reeve".

Depending on type of questions, user information need can be answered in form of few words or phrases (*factoid*) or in form of a list of related entities (*list*). In addition, systems have an option to retrieve *other* information, which was not explicitly questioned by a user. System response for *other* question is expected to be prose form, and is expected to be bounded sentence length .

```
<target id = "154" text = "Christopher Reeve">
       <q id = "154.1" type="FACTOID">
        What year was Christopher Reeve paralyzed?
       <q id = "154.2" type="FACTOID">
        How many ‘‘Superman’’ movies did he make?
       <q id = "154.3" type="FACTOID">
        During what years were these "Superman" movies made?
       <q id = "154.4" type="FACTOID">
       Which actress co-starred in the most "Superman" movies with Reeve?
       <q id = "154.5" type="FACTOID">
       What year did Reeve commence his theatrical career?
       <q id = "154.6" type="LIST">
       List titles of movies, other than ‘‘Superman’’ movies
       that Christopher Reeve acted in
       <q id="151.7", type="other">
       other
</target>
```

Figure 4.1: Sample questions centered around a target query *Christopher Reeve*.

For our exercise, we transform QA2006 collection as follows: For each query topic, we assume the target ("Christopher Reeve" in above example) to be a query submitted by a unknown user. User intent behind submitting query is not explicit from her query itself. We assume that user submitting query could be interested in subset or all of questions listed under the target. Such topics corresponds to *underspecified* kind of queries as defined in Chapter 3. In this exercise, we do not distinguish between *factoid* and *list* style of questions. Further, we treat each response associated with list-style question as a distinct nugget that user could be interested in. In assessing relevance of a document, we make use of patterns compiled from human assessments

(and released by NIST). Previously, these patterns were used to construct Question Answering test collections Lin and Katz [2006].

4.2 Reverse Ideal Gain

Before we compare the effectiveness of ranking function, we compute the "*Reverse Ideal Gain*", which is equal to the utility gain of a ranked list obtained by the reverse of an "ideal result list". To refresh, an ideal ranked list is obtained (in a greedy fashion) by picking the document that increases the gain of utility the most. To obtain a reverse variant of the list, we pick, at each attempt, the document that contributes the least gain of utility for an user browsing through the list. In Figure 4.2, we plot reverse of ideal gain for pattern-matched qrels compiled as explained in previous section. Each document used contains or covers at least one nugget associated with a query. Under binary notion of relevance, each document could be considered as relevant, based on their topicality [3]. However, when effectiveness is measured in terms of α-nDCG, we observe that utility gained at a given rank decreases, on varying the value of α from $0 \ldots 1$. The instance, $\alpha = 0$, in the plot corresponds to standard nDCG proposed by Järvelin and Kekäläinen [2002], where there exists no discount for redundancy. As we increase the value of α, effectiveness (measured in terms of α-nDCG) decreases.

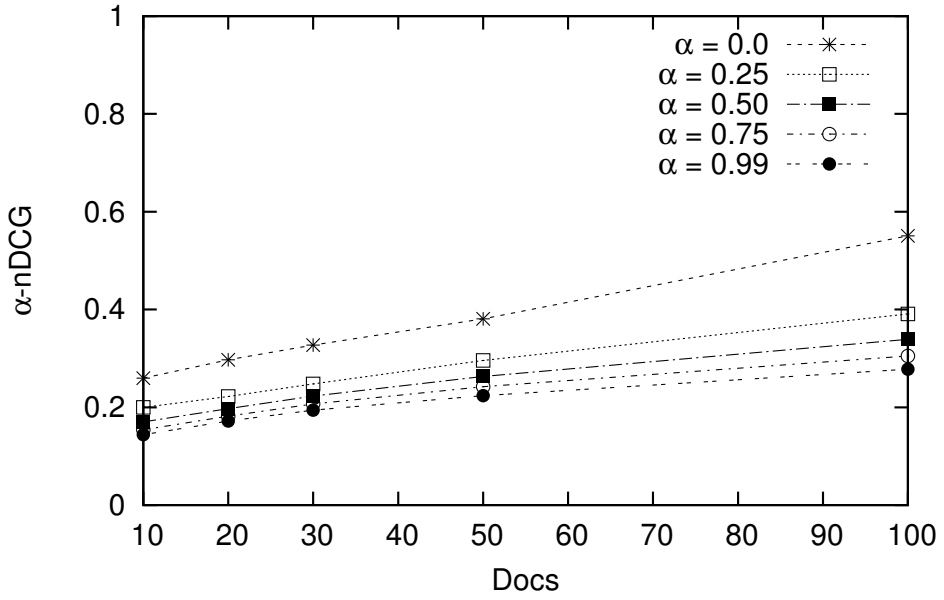


Figure 4.2: Reverse Ideal for $\alpha - nDCG$ for QA 2006 test collection

[3] a document is relevant if it consists of *any* information relevant to the topic

4.3 Query Expansion and Novelty

Over the years, researchers have considered pseudo-relevant feedback approaches to overcome term mismatch problem [4]. In these approaches, it is assumed that top ranked document retrieved by a ranking function are pseudo-relevant to underlying information need. Based on statistical measures, researchers proposed methods to identify terms that are prevalent to query topic. A new ranked list is then compiled using expanded queries (original query + feedback terms). Chen and Karger [2006], first highlighted problems encountered while applying pseudo-relevant feedback approaches in context of web search. Chen and Karger discuss loss of novelty in terms of pseudo-relevant feedback approaches, whereby (existing) ranking functions have been optimized to retrieve redundant information.

We wish to demonstrate the ability of our measures to capture such loss of novelty in pseudo-relevant feedback approaches. For our exercise, we consider a baseline ranking function (BM25) and two standard pseudo-relevant feedback approaches: Okapi Feedback, variant proposed by Billerbeck and Zobel [2004] and KL-divergence based approach proposed by Carpineto et al. [2001]. Both approaches employ statistical measures to select feedback terms that are prevalent in documents relevant to user information need. We carried out experiments using ranking functions implemented in the Wumpus retrieval engine [5]. We conduct our experiments on the QA2006 test collection. We initially query the collection (indexed in Wumpus), using the targets as query terms (e.g. Christopher Reeve). We select BM25 to compile our baseline set (bl) (our baseline set is obtained using BM25 ranking function). We then make use of top 25 documents to expand the initial query using term selection criteria corresponding to each pseudo-relevant ranking approach. We then retrieve a feedback based document set using expanded query (fb). As before, we ran expanded queries against QA2006 collection and retrieve top 100 documents for comparisons. Since, we lacked actual assessor judgments, we ran the pattern matching approach (similar to Lin and Katz [2006]) to identify various nuggets covered in a given document.

We measure the relative effectiveness (Δ) of each of the pseudo-relevance feedback approaches as

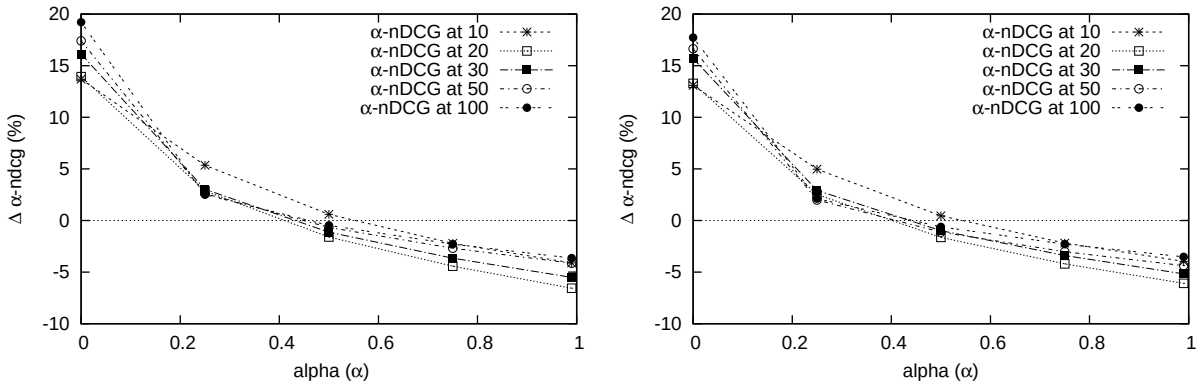
$$\Delta = \frac{\alpha\text{-nDCG}_{fb}@r - \alpha\text{-nDCG}_{bl}@r}{\alpha\text{-nDCG}_{bl}@r}$$

where α-nDCG$_{fb}@r$ represents α-nDCG value at rank r for a feedback retrieval run (fb), α-nDCG$_{bl}@r$ represents α-nDCG value at rank r for a baseline retrieval run. In Figure 4.3, we plot relative gain of utility from a pseudo-relevance feedback run, measured in terms of α-nDCG at pre-defined ranks, over corresponding baseline retrieval run. As observed in the plots, relative gain of utility is around 20% for instances when we do not penalize redundancy $\alpha \simeq 0$. As we increase the value of *alpha*, pseudo-relevant feedback approaches get penalized for retrieving redundant

[4] mismatch between terms used by a user and those occurring in a document

[5] `http://wumpus.org`

information under our effectiveness measure. Intuitively, this could be credited to the nature of pseudo-relevant feedback approaches that tend to pull information similar to the top ranked documents.



(a) KLD (b) Okapi Feedback

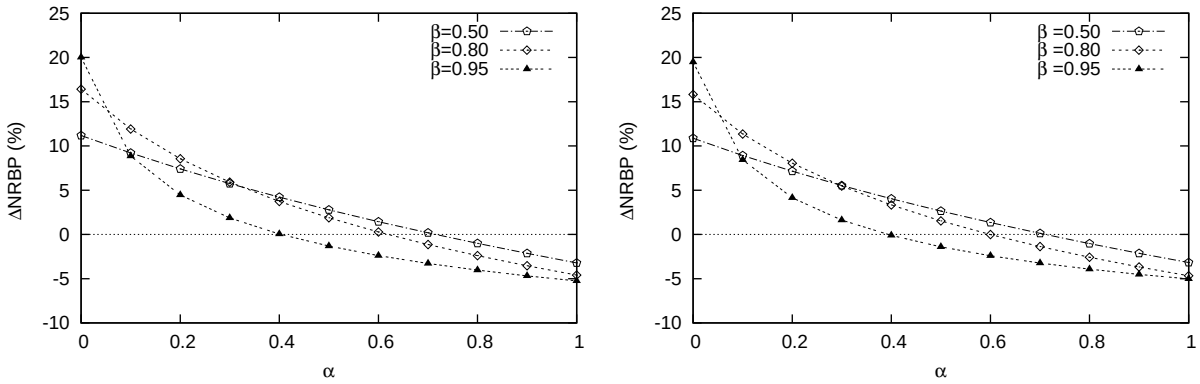
Figure 4.3: Comparison of *gain over baseline* in terms of $\alpha - nDCG$ for KLD-based feedback approaches (left) and Okapi-feedback (right). None of the measures are significantly different across all experiments.

In Figure 4.4, we plot similar "relative gain over baseline" values for our other proposed measure NRBP, computed at rank 100. We select three different persistence values $\beta = 0.50, 0.85, 0.95$ indicative of depth to which a user is willing to read. Similar to α-nDCG plots (Figure 4.4), relative effectiveness of pseudo-relevant ranking approaches decreases as we increase α value from 0 to 1. When $\beta = 0.95$, it is indicative of a patient user browsing through a lot of documents. When $\alpha = 0$, user is expected to gain over documents retrieved in baseline ranking (not significant). As we increase the value of α towards 1, relative effectiveness of pseudo-relevant drops below zero. In fact the drop in relative effectiveness gain is steep when compared with a utility gained by a less persistent user (say $\beta = 0.5$).

Next, we focus on comparing similarity between pairs of documents, as measured in terms of nugget overlap (i.e. nuggets in common) versus compression based similarity measures.

4.4 Nugget Overlap and Normalized Compression Distance

In this section, we investigate into nugget overlap (number of nuggets in common between document pairs), and compression-based similarity measures. There exists several other means to compute similarity between two documents, such as cosine distance, etc. In this section, we compare the document similarity as indicated in terms of nugget overlap and that of *Normalized Compression Distance* (NCD), a universal metric, as proposed by Li et al. [2003].



(a) KLD (b) Okapi Feedback

Figure 4.4: Comparison of *gain over baseline* in terms of NRBP for two pseudo-relevance feedback approaches, with varying persistence β and penalty for redundancy information α.

Li et al. introduced Information Distance based on Kolmogorov Complexity ($K()$). Kolmogorov complexity ($K(x)$) is defined as the length of a smallest binary program that can output string x. Li et al. defined Information distance ($E(x, y)$) between two strings x and y as *smallest* binary program that computes x from y and vice-versa.

$$E(x, y) = max\{K(y|x), K(x|y)\}$$

where $K(y|x)$ is Kolmogorov Complexity to generate the sequence y, given x as an auxiliary input. Given two sequences x and y, information distance could benefit from the presence of any redundant information. A Normalized Information Distance between between two strings can then be defined as follows:

$$NID = \frac{max\{K(x|y^*), K(y|x^*)\}}{max\{K(x), K(y)\}} \tag{4.1}$$

To overcome the non-computability of Kolmogorov Complexity Li et al. [2003] approximate NID through Normalized Compression Distance (NCD), defined as

$$NCD(x, y) = \frac{C(x, y) - min\{C(x), C(y)\}}{max\{C(x), C(y)\}} \tag{4.2}$$

where $C(x)$ and $C(y)$ refers to lengths of strings x and y respectively, $C(x, y)$ represents compressed length of a string obtained by concatenation of both x and y. The value of $NCD(x, y)$ is varies $0 \leq NCD(x, y) \leq (1 + \epsilon)$, which is indicative of the how *different* the two strings are: similar strings have NCD() value closer to 0.

Using QA2006 test collection, we compare NCD() values and number of nuggets in common for pairs of documents that have been judged relevant for a given topic. Let C(A) represents size

of document A in compressed [6] format, C(B) represents size of document B in compressed format, C(A,B) represents size of both texts concatenated in compressed form. We obtain NCD(A,B) as

$$NCD(A,B) = \frac{C(A,B) - min\{C(A), C(B)\}}{max\{C(A), C(B)\}}$$

For each topic, we consider pairs of documents that have been judged to answer at least one question. In Figure 4.5, we plot (Box Plots) NCD values (y-axis) and nuggets in common between pairs of documents (x-axis). First we bucket all document pairs (from QA2006 test collection) based on the number of nuggets they have in common. Next, we plot corresponding NCD() value for each pair of documents in a corresponding bucket. In a given box plot, we show (within the box) the median of corresponding NCD() values, and where box boundaries represent lower quartiles (25%) and upper quartile points (75%). We also plot lower fence and upper fence values. Points lying outside of those fences (represented by +) are deemed outliers. From observed median values, it is hard to distinguish between the relation between number of nuggets shared in common and similarity in terms of NCD() values.

In addition to the NCD measure, we also compute similarity based on Compression-based Dissimilarity Measure (CDM) (Keogh et al. [2004]), computed as

$$CDM = \frac{C(A,B)}{C(A) + C(X)} \tag{4.3}$$

However, unlike NCD(), the value of CDM is bounded between 0.5 (for redundant, similar documents) and 1.00 (for documents that are quite diverse). In Figure 4.6, we plot CDM values(y-axis) along with number of nuggets shared (x-axis) between document pairs of documents from QA2006 test collection. As in previous plots, we bucket document pairs

In next section, we explain in detail work proposed after our initial framework. We first present in detail Expected Reciprocal Rank (ERR) measure, which similar to our proposed measures attempts to discount utility gained in a document relative to previously seen documents.

4.5 Expected Reciprocal Rank (ERR)

Craswell et al. [2008] introduced a *cascade model* to understand user interaction with results retrieved by a search engine. In their model, users are browse retrieved results in a top-down fashion, estimating relevance of documents displayed. Users would then select those document(s) (i.e. click for further reading) that most likely satisfy their information need. In their model, user is expected to click on a given document (d_i), with a probability of r_i or skip it, with a probability of $(1-r_i)$. The probability of a user clicking on a document d_i is factored on likelihood of user not

[6] We use LZMA tool to compress our documents `http://www.7-zip.org/sdk.html`

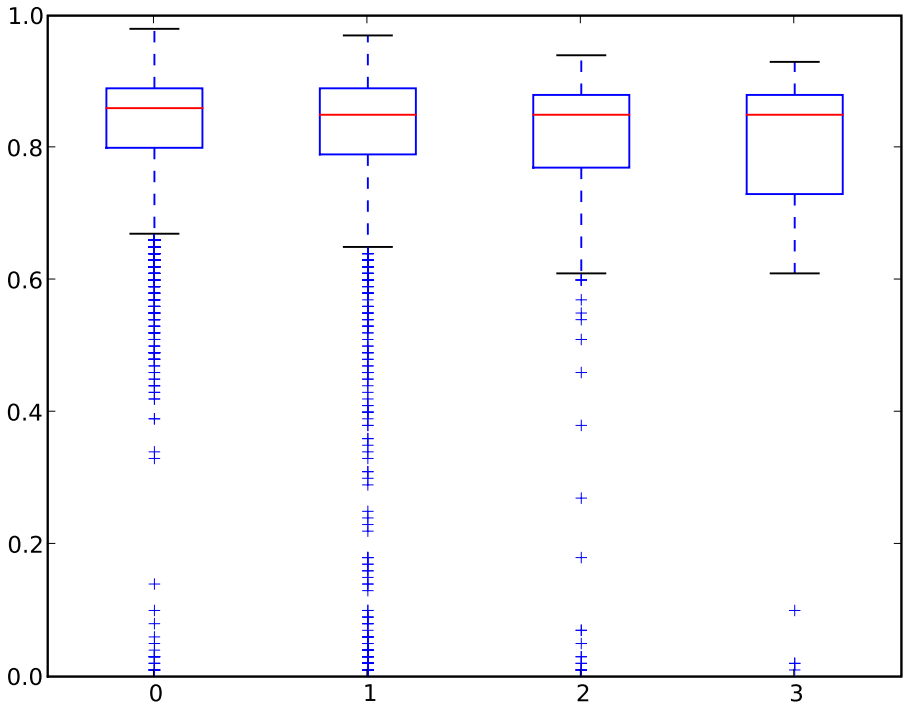


Figure 4.5: QA2006 data: Box plot of NCD vs Nugget overlap

finding similar (relevant) document at higher ranks. For a document at given rank i, probability of user being satisfied by that document is

$$r_i . \prod_{j=1}^{i-1} (1 - r_j) \tag{4.4}$$

User clicks on a document and is expected to never return back to results page.

Chapelle et al. [2009] derived an effectiveness measure based on similar model of user interaction while browsing through a retrieved set of results. Chapelle et al. measured utility gained by a user from a document d_i, at rank i as

$$R_i . \prod_{j=1}^{i-1} (1 - R_j) \tag{4.5}$$

where R_i was defined as probability of document at rank i being relevant. Chapelle et al. com-

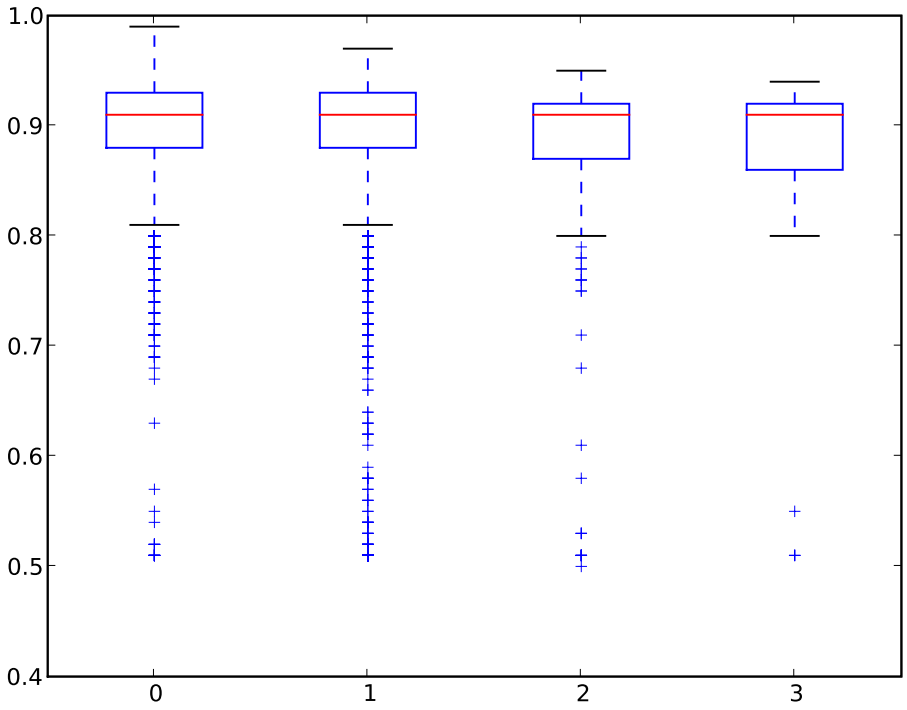


Figure 4.6: QA2006: Box plot of CDM vs Nugget overlap

puted probability of being judged relevant from graded relevance measures as follows:

$$R(i) = \frac{2^{g_i} - 1}{2^{g_{max}}} \tag{4.6}$$

where g_i is the relevance of the document d_i assessed between $0 \ldots g_{max}$ (usually $0 \ldots 3$). Finally, Chapelle et al. factored in rank i into discounting relevance gained by a factor of $\frac{1}{i}$, which is meant to reflect effort on user part to skim down to rank i. Chapelle et al. defined an effectiveness measure, Expected Reciprocal Rank (ERR) to reflect total utility gained by a user under such model.

$$ERR = \sum_{i=1}^{n} \frac{R_i}{i} \prod_{j=1}^{i-1} (1 - R_j) \tag{4.7}$$

In case of binary relevance, ERR would be equivalent to reciprocal rank(rr) Craswell et al. [2003], which is the first rank at which user encounters a relevant document.

ERR-IA

Chapelle et al. extended their ERR measure into Intent-Aware ERR measure, (ERR-IA), to accommodate possibility of diverse intents associated with a given query q

$$\sum_{i=1}^{n} \frac{1}{i} \sum_{t \in q} P(t|q) \prod_{j=1}^{i-1} (1 - R_j^t) R_i^t \tag{4.8}$$

where $P(t|q)$ refers to probability of user interested in intent t with query q, R_i^t indicates the "relevance of the document at rank i, with respect to t^{th} intent associated to query q".

ERR and α-nDCG

Looking back at our definition of probability of relevance (Eq. 3.5)

$$R_i = \begin{cases} \alpha J(d, n_i) & \text{rel} \\ 0 & \text{non-rel} \end{cases} \tag{4.9}$$

where $J(d, n_i)$ is a binary assessment of document d being relevant to an intent represented by nugget n_i. Taking into account the notion of α, in terms of the extent to which a user is satisfied of the information need in the given document. Substituting the values in Eq. 4.5, and taking $J(d, k)$ to be binary (i.e. 0 or 1), we observe that gain of utility across both measures is the same.

$$\alpha \prod_{j=1}^{i-1} (1 - \alpha) \tag{4.10}$$

4.6 Web Track: Novelty and Diversity Task

Starting 2009, NIST (in context of TREC) enabled researchers to experimentally compare ranking functions on a significantly large test collection: Clueweb Corpus [7]. The corpus consists of approximately one billion English and non-English documents crawled from the web. Several tracks have been proposed based on Clueweb corpus. Under Web Track Clarke et al. [2009b], researchers have defined two tasks

- adhoc task
- diversity task

[7] `http://lemurproject.org/clueweb09.php`

where adhoc task follows the standard cranfield paradigm in which each query is associated with one particular information need. Each topic defined for this tasks comprises of a query along with its description (e.g. Figure 4.7). The second task, diversity task, follow our framework of evaluation where each query is augmented with notion of sub-topics [8], as shown in Figure 4.8.

Topic Creation

As in previous exercises, NIST employed human assessors to create query topics. In addition to standard topic definition, assessors were required to identify various intents for a given query. Possible intents were identified from mining query logs of a commercial search engine by Radlinski et al. [2010b]. For a query q, Radlinski et al. identified query pairs $< q, q^{'} >$ that satisfy the following constraints:

- Query $q^{'}$ is submitted to the search engine within a window of 10 minutes following q, by at least two unique users.
- Of all queries submitted to the search engine (for the query q) $< q, . >$, there is a significant chance δ (> 0.001) of $q^{'}$ following q.

Radlinski et al. then construct a bi-partite graph from all filtered $< q, q^{'} >$ pairs and group queries sharing common urls (co-clicks) by carrying out two-step random-walk Craswell and Szummer [2007]. Radlinski et al. then clustered all possible intents associated with a query and assigned weight $w(q^{'})$ for each query intent in a cluster

$$w'_q = w_q . \frac{N(q, q')}{\sum_{r \in R(q)} N(q, r)} \tag{4.11}$$

where $N(q, q^{'})$ indicates frequency of $q^{'}$ following q, and $\sum_{r \in R(q)} N(q, r)$ is equal to the number of times that the query q was modified. Clusters are weighted as sum of weights of each intent member.

NIST assessors were then shown clusters constructed as explained above, to aid in topic construction. Assessors picked intents from clusters and labeled them as sub-topics for a given query. Topics constructed can be classified into two kinds:

- Faceted topics: query topics in which intents associated are not independent to each other. For example, query "Horse Hooves" (Figure 4.8) (Underspecified).
- Ambiguous topics: query topics in which the intents are independent of each other and corresponds to different interpretations (Figure 4.9) (Ambiguous).

[8] sub-topics, aspects, nuggets and intents are equivalent

Human assessors were given the cluster of queries found for the given query, and were asked to define topics for each query. Similar to adhoc test collections, assessors composed *description* field for each query. For example, for the query **Horse Hooves**, topic compiled for the query along description is as shown in Figure 4.7.

```
<query>horse hooves</query>
  <description>
  Find information about horse hooves, their care, and diseases of hooves.
</description>
```

Figure 4.7: TREC 2010 Web Track (ad-hoc) sample topic: *horse hooves*

Unlike previous evaluation exercises, participants for this track do not receive complete set of intents prior to actual submission of results. This is meant to simulate scenarios where search engines are unaware of actual intent behind a query submitted. Systems are therefore required to diversify their results with aim to maximize likelihood of users finding some relevant information.

Relevance Assessments

A total of 32 systems that took part in Web 2010 diversity task. Since the topics were shared between both adhoc and diversity tasks, documents retrieved from each system that participated in either tasks were pooled together during judging phase. Documents were judged following a two-tier process: First documents were judged for their relevance in either binary (2009) or on graded scale basis (2010) with respect to the adhoc topic (and its given description). For those documents that have been judged as relevant on a topic basis, for a given query, assessors then identified different intents/subtopics, associated with the particular topic, for which the document could be relevant to.

Effectiveness Measures

In 2009, systems taking part in the Diversity Task (Clarke et al. [2009b]), were compared using α-nDCG (Clarke et al. [2008]), and MAP-IA (Agarwal et al. [2009]) measures at three different ranks, ($N = 5, 10, 20$). In addition, all ranking functions were compares with two combined precision and recall measures:

- combined precision: Number of documents relevant to *any* of subtopics for a given query.
- sub-topic recall - Fraction of subtopics covered in top N ranked documents Zhai et al. [2003].

```
 <topic number="51" type="faceted">
  <query>horse hooves</query>
  <description>
    Find information about horse hooves, their care, and diseases of hooves.
  </description>
  <subtopic number="1" type="inf">
    Find information about horses' hooves and how to care for them.
  </subtopic>
  <subtopic number="2" type="nav">
    Find pictures of horse hooves.
  </subtopic>
  <subtopic number="3" type="inf">
    What are some injuries or diseases of hooves in horses, and how
    are they treated?
  </subtopic>
  <subtopic number="4" type="inf">
    Describe the anatomy of horses' feet and hooves.
  </subtopic>
  <subtopic number="5" type="inf">
    Find information on shoeing horses and horseshoe problems.
  </subtopic>
</topic>
```

Figure 4.8: TREC 2010 Web Track diversity task example (faceted): *horse hooves*

```
<topic number="52" type="ambiguous">
  <query>avp</query>
  <description>
    Find information about events sponsored by AVP, the Association of
    Volleyball Professionals.
  </description>
  <subtopic number="1" type="nav">
    Go to the homepage for the AVP, sponsor of professional beach
    volleyball events.
  </subtopic>
  <subtopic number="2" type="inf">
    Find information about pro beach volleyball tournaments and events
    sponsored by AVP.
  </subtopic>
  <subtopic number="3" type="nav">
    Find the homepage for AVP antivirus software.
  </subtopic>
  <subtopic number="4" type="inf">
    Find reviews of AVP antivirus software and comparisons to other products.
  </subtopic>
  <subtopic number="5" type="inf">
    Find information about the Avon Products (AVP) company.
  </subtopic>
  <subtopic number="6" type="nav">
    Find sites devoted to the "Alien vs. Predator" movie franchise.
  </subtopic>
  <subtopic number="7" type="inf">
    Find information about Wilkes-Barre Scranton International Airport
    in Pennsylvania (airport code AVP).
  </subtopic>
</topic>
```

Figure 4.9: TREC 2010 Web Track diversity task example (ambiguous): *avp*

Clarke et al. [2011] carried out extensive comparison of various effectiveness measures that have been defined in context of the *diversity task* of TREC 2009 Web Track. Clarke et al. broadly grouped effectiveness measures into two categories: *cascade* (ERR, α-nDCG, NRBP) and *non-cascade* measures (MAP-IA).

Across all cascade measures, utility gained is measured as

$$g_i^k = \alpha.g_i^k(1-\alpha)^{c_j^k} \tag{4.12}$$

where g_i^k is relevance (binary) of the document i with respect to nugget k. Each cascade measure, however, has its unique way of discounting utility gained in terms of rank at which document is shown to user (as shown below).

$$discount(k) = \begin{cases} log_2(1+k) & \alpha\text{DCG} \\ k & err \\ (\frac{1}{\beta})^{k-1} & \text{NRBP} \end{cases} \tag{4.13}$$

In non-cascade measures, Clarke et al. measured utility in terms of MAP-IA measure proposed by Agarwal et al. [2009].

$$MAP - IA = \sum_{i=1}^{M} P_i.Si^{map} \tag{4.14}$$

From their experiments, Clarke et al. observed that there exists strong correlation between system rankings based on cascade measures. Unsurprisingly, it was observed that cascade measures do not correlate with MAP-IA measure (non-cascade). However, Clarke et al. observed that non-cascade measures have better discriminative power in comparison to cascade measures.

Discriminative Power

Sakai [2006] proposed a bootstrap-sampling based approach to determine *discriminative power* of an effectiveness measure. Based on bootstrap sampling process, system performance is measured for a collection of topics obtained from random sampling (with repetition) topics from existing topicset. A sample b^* topicset of $n = |Q|$ topics is compiled by re-sampling original topicset (Q). This process if repeated a large number (1000) of times, and effectiveness of pair of systems are compared. Discriminative power of an effectiveness measure is then defined as fraction of pairs that are found to be significant under this multiple number of sample set. Clarke et al. [2011] compared different effectiveness measures and found that non-cascade measures such as MAP-IA (and even MAP) have higher discriminative power than cascade measures.

Nugget Overlap Versus Compressionability

As in QA2006 exercise, in Figure 4.10, we report using box-plot nugget overlap and NCD() values. In this exercise, we select documents that have atleast one subtopic or intent covered in a given topic.

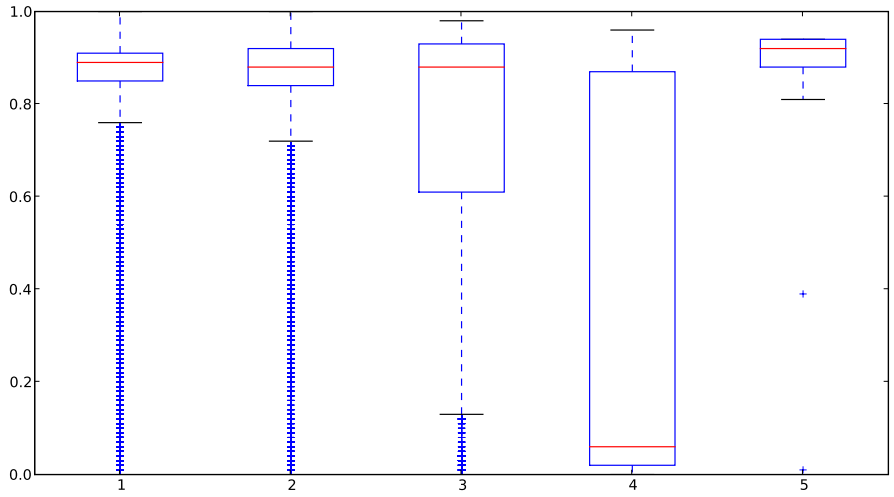


Figure 4.10: Web 2010 (Box Plot): Nugget overlap vs NCD

We next discuss few related works that were built similar to our evaluation measures.

4.7 Idiv Measures

Sakai et al. [2010] proposed *Idiv* measures (read as Intent Diversity measures) for evaluating ranking functions taking into account the diversity of intents available for a given query, and those that are covered by a retrieved set of results. Sakai et al. proposed following two measures by augmenting notion of diversity to existing adhoc measures:

- Idiv-nDCG — Intent Diversity version of standard nDCG measure Järvelin and Kekäläinen [2002]
- Idiv-Q — Intent Diversity version of Q measure Sakai [2007]

Though lacking a clear user model, Sakai et al. defined above measures to be "simple", easy to understand. Both measures can be broken down into two components:

1. Intent recall measure (I-recall).

2. Div measures (div-nDCG for Idiv-nDCG and div-Q for Idiv-Q).

4.7.1 I-recall

The first component is I-recall or intent-recall at rank l, which similar to S-recall proposed by Zhai et al. [2003] is a count of the number of unique intents covered at rank l.

$$\text{I-recall@}l = \frac{\cup_{r=1}^{l} I(d_r)}{n} \tag{4.15}$$

where n represents total number of intents for a given query topic across all interpretations and $I(d_r)$ represents the intents covered in the document d_r retrieved at rank r.

4.7.2 div-nDCG, div-Q

Similar to our proposed gain of utility measure Eq. 3.14 (in Chapter 3), Sakai et al. compute defined "global gain" of a document d with respect to all intents associated with query q as

$$\sum_{i \in I_q} P(i|q).g_i(d) \tag{4.16}$$

where $P(i|q)$ indicate the probability of the intent i being associated with the query q and $g_i(d)$ is equal to gain of relevance of document d for intent i. Gain value g_i, is attributed to the multi-level grade assessment of human assessors, on a scale $0 \ldots 3$, where 0 refers to non-relevance and 3 represents the document being highly relevant.

Similar to Järvelin and Kekäläinen [2002], Sakai et al. compute *cumulative global gain CGG* at rank l and divide it with the *ideal global cumulative gain* GG^* at rank l to obtain *normalized cumulative gain* (*div-nDCG*) at rank l

$$\textit{div-nDCG@}l = \frac{\frac{\sum_{r=1}^{l} GG(r)}{log(r+1)}}{\frac{\sum_{r=1}^{l} GG^*(r)}{log(r+1)}} \tag{4.17}$$

where the GG^* values are computed over an ideal ranked result obtained sorted by their global gain values Eq. 4.16. The value of GG and GG^* were also plugged into the Q-measure Sakai [2007] to obtain *div-Q* measure as shown below:

$$div - Q = \frac{1}{R} \sum_{r=1}^{L} J(r).divBR(r) \tag{4.18}$$

where $divBR(r)$ is obtained as (similar to their Q-measure)

$$\frac{\sum_{k=1}^{r} J(k) + \beta \sum_{k=1}^{r} GG(k)}{r + \beta \sum_{k=1}^{r} GG^*(k)} \tag{4.19}$$

The final $Idiv\text{-}nDCG$ and $Idiv\text{-}Q$ measures are then obtained by linear interpolation of two of its corresponding components – for $Idiv - nDCG$ measure

$$Idiv\text{-}nDCG@l = \gamma I - recall@l + (1-\gamma) div - nDCG@l \tag{4.20}$$

and for $Idiv - Q$ measure

$$Idiv\text{-}Q@l = \gamma I - recall@l + (1-\gamma) div - Q@l \tag{4.21}$$

where γ is a constant $\in [0, 1]$ to tune influence of $I - recall$ and $Idiv$ components. Sakai et al. were unclear as to motivation behind such interpolation.

In computing global gain value for a given document, query topic pair, Sakai et al. [2010] do consider the possibility of a document being judged on a graded multi-level scale of relevance, as opposed to our binary nature. They do not however consider the novelty of information retrieved in a given document. Further, we treat in our computation Eq. 3.14 that each intent is equally important for a given query and treated it as constant. Sakai et al. let go such assumption in obtaining the measure. In fact, Sakai et al. falsely claim that $\alpha\text{-}nDCG$ metric does not allow consider the weight of an intent. Instead, Sakai et al. arbitrarily chose the weight of an intent , for intent j as

$$\frac{2^{n-j+1}}{\sum_{k=1}^{n} 2^k} \tag{4.22}$$

where n is the total number of intents possible for the given query topic. Such computation is not based on any user model and is at best arbitrary (personnel opinion).

4.8 Subtopic Metrics

Radlinski et al. [2010a] proposed the following four measurable properties to measure quality of a topicset compiled for a diversity test collection.

- Coherence of an intent.
- Distinctness of an intent.
- Plausibility of an intent.

- Completeness of an intent.

Radlinski et al. base their attributes on relevance assessments carried out by judges for documents pooled for a query q.

- R_i represents set of documents relevant to intent i
- R_i^j represents documents judged relevant by judge j for intent i
- $R_u(q)$ represents a document set judged relevant by a user submitting query q
- $I(q)$ represents intent set possible for a given query q

similarity between two intents was measured, based on Jacardian Index, was measured as

$$sim(i_1, i_2) = J(R_1, R_2) = \frac{|R_{i1} \cap R_{i2}|}{|R_{i1} \cup R_{i2}|} \tag{4.23}$$

Radlinski et al. then defined the four qualities based on the $sim()$ measure:

Coherence An intent i associated with query q is said to be coherent if the following holds true

$$sim(R_i^{j1}, R_i^{j2}) > \alpha$$

where R_i^{j1} and R_i^{j1} represents the documents being judged relevant to the intent i of the query q by two independent judges $j1$ and $j2$ respectively. Radlinski et al. proposed the measure of coherence as means to avoid the accidental association of an intent to a given query, by virtue of subjective assessments made by one human judge.

Distinctness Two intents i_1 and i_2 are known as $\alpha-$distinct, if the similarity value $sim(i_1, i_2)$ is $\leq (1 - \alpha)$. Any intents $i1$ and $i2$ that do not satisfy such condition are indeed redundant (i.e. same intent different form).

Plausibility An intent i is known as $\alpha\beta-$plausible if there exists at least α fraction of user who issued the query q being satisfied by the $sim(R_u, R_i) \geq \beta$

Completeness Given a query q, along with a set of intents $I(q)$, the set is known as $\alpha\beta-$complete if at least a fraction α of users satisfy the condition $\arg\max_{i \in I(q)} sim(R_u, R_i) \geq \beta$

4.9 Discussion

In this chapter, we reported on one of the preliminary exercise carried out to demonstrate functioning of our framework. We then explained a large-scale evaluation workshop undertaken by NIST to compare ranking functions in context of Web Search. It was reported that cascade measures, measures that discount for redundant information, correlate with each other.

Although our framework and measures proposed facilitates comparative evaluation of ranking functions with respect to diversity and novelty, it would require

a) human assessors to identify intents associated with a query q

b) assess relevance of a document on a multi-intent scale

which increases effort required to construct a test collection. In following chapters, we wish to investigate into methods that overcome second constraint.

Chapter 5

Estimating Effectiveness in Retrieval Evaluation

So far we, have outlined our framework and associated measures taking into account both novelty and diversity in retrieved results. In deriving our measures, we assume qrels are complete. However, as explained in Chapter 2, it is impossible to identify and compile all possible relevant documents for a given topic, via pooling methods based on Spärck Jones and Van Rijsbergen [1975]. On the other hand, standard evaluation measures implicitly assume the opposite — i.e. assume presence of complete relevance judgments for a given topic. Worse, any unjudged document retrieved by a system, possibly employing a novel ranking method, are simply treated as non-relevant. Such considerations and presumptions were strongly contested by Zobel [1998], who based on observed rate of relevant documents estimate that at best current methods find $50\% - 70\%$ of total relevant documents for a given topic. Further Büttcher et al. [2007] observed certain "bias" against new systems, which did not take part in original pooling process.

Broadly, different approaches that have been proposed to overcome the problem of incomplete qrels can be grouped as

- Measures robust enough to withstand presence of unjudged documents.
  - bpref by Buckley and Voorhees [2004]
  - subAP, indAP, infAP by Yilmaz and Aslam [2006]
  - Q measure by Sakai [2004], etc.
- Methods to complete such incomplete qrels by estimating/predicting relevance of unjudged documents. e.g. Büttcher et al. [2007], and Carterette and Allan [2007].

In this chapter, we follow the latter approach by predicting the probability of an unjudged document being judged relevant (Section 5.2). Next, in Section 5.3, we propose a dynamic programming based approach to compute expected number of relevant documents at a given rank. In Section 5.4, we build on our dynamic programming approach to estimate Average Precision (AP) for a ranked list. In Section 5.5, we carry out experiments to compare our estimate of MAP with that of true MAP values across four different ad-hoc test collections.

5.1 Predicting Relevance

In order to predict relevance (of an unjudged document), a classifier is first trained on documents that have been judged by human assessors for a given topic. Such trained classifier is then used to estimate or predict the relevance of unjudged documents. Büttcher et al. [2007] proposed two threshold-based approaches to estimate (binary) relevance of an unjudged document. In one approach, Büttcher et al. first compiled a relevance language model by concatenating documents judged relevant by human assessors. Büttcher et al. then classify an unjudged document into either relevant or not relevant based on its KLD measure with that of the relevance language model.

$$KLD(M_j, M_R) = \begin{cases} < \delta & \text{relevant} \\ > \delta & \text{not relevant} \end{cases} \tag{5.1}$$

where M_j represents unigram language model of an unjudged document d_j and M_R represents relevance model. Büttcher et al. experimentally select a threshold δ such that an unjudged document d_j with $KLD(M_j, M_R)$ less than δ is considered relevant. In an alternate approach, Büttcher et al. used existing relevance assessments to train a SVM^{light} classifier. Using the trained classifier, Büttcher et al. classified an unjudged document as being relevant or not. Büttcher et al. were able to overcome the "bias" of a new ranking method for returning an unjudged document using trained classifiers for each topic.

Carterette and Allan [2007] proposed a semi-automatic approach to estimate relevance of a document, in a probabilistic manner. Carterette and Allan constructed a *relevant cluster* for a given topic, from all documents judged as relevant for that particular topic. Following the cluster construction, any unjudged document is compared with each document in the relevant cluster. Carterette and Allan defined similarity between two documents d_i and d_j, based on cosine similarity

$$cos(d_i, d_j) = \frac{\sum_{t \in V} w_{i,t} \cdot w_{j,t}}{\sqrt{\sum_{t \in V} w_{i,t}^2} \sqrt{\sum_{t \in V} w_{j,t}^2}} \tag{5.2}$$

where V represents all vocabulary terms, $w_{i,t}$ and $w_{j,t}$ refers to weights $(tf \cdot idf)$ of the term t in documents d_i and d_j, respectively. Document similarities are then regularized through a

logistic regression model, where log-odds of relevance is modeled as a weighted sum of similarity for an unjudged document with each of relevant document. Carterette and Allan reported higher correlation of system rankings in comparison with true rankings (based on full qrels) while using only a smaller pool of documents, and estimating the relevance for the rest of the qrels using their semi-automatic approach.

In an alternative approach, Soboroff et al. [2001] explored possibility of evaluating ranking systems without any relevance judgements from humans. For a given test collection, Soboroff et al. obtained number of documents judged relevant from existing TREC collection building exercise. Based on the number of relevant documents present, Soboroff et al. went ahead to construct a relevant document pool by sampling the same number of documents from the pooled set, and terming them as relevant. Soboroff et al. observed that system ordering achieved using their pseudo-rels positively correlates ($\tau = 0.45$) with system rankings obtained using full qrels. Although, the value of τ is less than 0.8 (indicative of weak correlation), Soboroff et al. reported that they were able to distinguish or isolate best and the worst performing systems from the rest.

In next section, we detail our approach to estimate or predict the relevance of an unjudged document using the classifier proposed in Cormack et al. [2011].

5.2 Probability of (judged) Relevant

Given a set of judged documents, we predict the relevance of an unjudged document using a classifier trained on existing judged documents. In our preliminary experiments, we implemented the logistic regression classifier, as outlined in Cormack et al. [2011] to obtain a probability of relevance given the document's features. The classifier comprises of two functions: `spamminess()` (Listing 5.1) and `train()` (Listing 5.2) and is trained using gradient-descent approach. The classifier transforms input text into overlapping 4-byte grams as feature vectors. For example, feature set created for a phrase "Hello World" consist of "Hell" , "ello", "llo ", "lo W","o Wo"," Wor", "Worl" and "orld" respectively.

As explained in Cormack et al. [2011], the classifier is trained by making a single pass over all (judged) documents [1]. After going through each training document(d), the feature vector (β) is updated by

$$\beta \leftarrow \beta + \delta \cdot X_d(\text{isRel(d)} - \frac{1}{1 + e^{-score(d)}}) \tag{5.3}$$

where X_d is the feature vector associated of document d, δ is learning rate parameter (set to 0.0002), $score(d)$ is equal to

$$score(d) = \beta \cdot X_d \tag{5.4}$$

[1] We did not impose any ordering by which the classifier is trained.

and finally the function $isRel(d)$ represents relevance, assigned by human assessors, for a given query (on binary scale):

$$\text{isRel(d)} = \begin{cases} 1 & \text{d is relevant, highly relevant} \\ 0 & \text{d is not relevant,spam etc.} \end{cases} \tag{5.5}$$

Following Cormack et al., we truncate input documents at 35000 bytes. Cormack et al. interpreted the $score(d)$ (returned by `spamminess()`) as equivalent to log-odds estimate of document d being relevant

$$score(d) \approx log \frac{Pr(d \text{ is relevant})}{Pr(d \text{ is not relevant})} \tag{5.6}$$

Once the classifier is trained, we obtain the probability of any unjudged document ($d^{'}$), being relevant as

$$P(rel|d^{'}, \beta) = \frac{1}{1 + e^{-score(d^{'})}} \tag{5.7}$$

where $score(d^{'})$ is obtained as given in Eq. 5.4, using $X_{d'}$ is overlapping 4-gram feature vector associated with the document $d^{'}$, and trained model β.

Listing 5.1: Spamminess function of the logistic regression classifier (Cormack et al. [2011]).

```
function spamminess(unsigned char *page, int n) {
        p = 1000081 ;
        δ = 0.0002 ;
        PREF = 35000 ; float w[p];
        unsigned i,b,h ;
        cookie++;
        if (n > PREF )
                n= PREF ;
        score = 0.0 ;
        b = (page[0] << 16) | (page[1] << 8) | (page[2]) ;
        for(i=3 to n){
              b = (b << 8) | page[i] ;
              h = b % P ;
              if(dun[h] == cookie)
                        continue ;
               dun[h] = cookie;  score += w[h] ;
        }
        return score ;
}
```

Listing 5.2: Train function of the logistic-regression classifier (Cormack et al. [2011]).

```
function train(unsigned char * page, int n, int isRel){
        p = 1000081 ;
        δ = 0.0002 ;
        PREF = 35000 ; float w[p];
        unsigned dun[p], cookie;
        unsigned i,b,h;
        if (n > PREF )
                n = PREF ;

        float p = 1/(1+exp(-spamminess(page,n))) ;
        cookie++;
        b = (page[0] ≪ 16) | (page[1] ≪ 8) | (page[2]) ;
        for (i=3 to n ){
                b = (b ≪ 8) | page[i] ;
                h= b % P;
                if( dun[h] == cookie)
                        continue ;

                dun[h] = cookie;
                w[h] += (isRel - p) * δ;
        }
}
```

5.2.1 Classifier Effectiveness (ROC)

We use ROC curves to demonstrate classifier effectiveness in terms of its ability to rank relevant documents higher than non-relevant documents. Following ROC curve generation from classification literature, we measure the following values:

- True Positive (TP): if true relevance of a document is equal to classifier's predicted relevance.
- False Positive (FP): if human assessment is not relevant, while predicted class is relevant.
- True Negative (TN): if human assessment and classified result are both not relevant.
- False Negative (FN): classifier tagged is not relevant, while humans judged it as relevant.

Using the four observed values, we compute the True Positive Rate (tpr) as

$$\text{tpr} = \frac{TP}{P} \tag{5.8}$$

where P represents the total number of relevant documents for a given topic. In addition to tpr, we compute False Positive Rate (fpr)

$$\text{fpr} = \frac{FP}{N} \tag{5.9}$$

where N represents total number of non relevant documents for a given topic. Effectiveness of a classifier is then presented through ROC Curves obtained by plotting the $< fpr, tpr >$ values for a given classifier.

Unlike binary classifiers that output only one $< fpr, tpr >$ point, we follow a sliding threshold-based method to generate a roc-curve as outlined in Fawcett [2006]. First, we sort documents in decreasing order of their probability of being relevant (Table 5.1). Starting at top-most document, we pick and select each data point as a threshold such that documents that have higher probability values are considered relevant, and those below the threshold as not-relevant. For each such threshold, we then compute corresponding $< fpr, tpr >$ values. Finally, the ROC Curve is obtained by joining all the corresponding $< fpr, tpr >$ values.

Docid	Prob	Rel (qrel based)
GX030-88-15296047	0.960561	2
GX048-40-13813813	0.956475	1
GX066-83-4439678	0.949342	2
GX030-89-11609923	0.892371	2
GX048-60-15735375	0.878694	2
GX266-64-1487215	0.000037	0
GX233-48-0127164	0.000013	0

Table 5.1: Example: documents sorted in decreasing order of their probability of being judged relevant

5.2.2 Experiments (ROC)

In order to measure the extent to which our approach of using a trained classifier to sort document in decreasing order of their probability values, we simulate incomplete qrels scenario as follows (in line with existing methods Büttcher et al. [2007]) :

1. For each topic in a given topicset, we randomly sample (without replacement) $n\% = 10\%, 20\% \ldots 90\%$ of qrels compiled for that particular topic. We call this training set or *training qrels* or *incomplete rels*.

2. For a training sample drawn for a given topic, we consider the rest as *test qrels* or *unjudged qrels*

3. We train a logistic regression based classifier (Section 5.2) using the training qrels sampled for a given topic [2].

4. Using the trained classifier, we estimate probability of being judged value for each of the unjudged qrels only. As in Büttcher et al., we refer to this final set of qrels with both absolute and probability relevance values as *completed qrels*.

We carry out our simulation experiments on four different ad-hoc datasets:

TREC 2006 **Terabyte Track:** We select data compiled for ad-hoc task of TREC 2006 Terabyte track (Büttcher et al. [2006]). This dataset consists of documents crawled from **.gov** domain, and a topicset consisting of 50 (*informational*) query topics. These 50 topics were picked from query logs of a commercial search engine, for which a user has clicked on one of a document belonging to **.gov** domain. A total of 31984 qrel tuples were compiled for this track across all 50 topics. While constructing *qrels*, assessors judged each document on a graded scale of relevance $(0, 1, 2)$. In our experiments, we treated documents judged under graded relevance value of 1 and 2 as relevant.

TREC-7 **ad-hoc track:** We select data compiled for TREC 7 ad-hoc track (Voorhees and Harman [1998]), defined over **disk 4** and **disk 5** (minus CR records) datasets. In this track, a total of 80345 qrel tuples were compiled across 50 topics. In this track, all relevance assessments were carried out in a binary scale (0 for non relevant and 1 for relevant).

TREC-8 **ad-hoc track:** Similarly, we pick data compiled for TREC 8 ad-hoc track (Voorhees and Harman [2000]). As in TREC 7 ad-hoc track, this track made use of *disk4* and *disk5* dataset. In this track, a total of 86830 qrel tuples were pooled across 50 topics.

TREC 2010 **Web track:** Lastly, we investigate into methods to overcome incomplete qrels for data compiled for TREC 2010 Web track (Clarke et al. [2010]). A total of 25329 qrel tuples were compiled in construction of qrels for this track. In this track, human assessors judged relevance on a graded scale, where 0 is not-relevant, 1 is relevant and 2 is highly relevant. As in TREC 2006 Terabyte data, we treated them on a binary scale of relevance.

[2]We do not enforce or verify to see the order in which the classifier scans the training documents. Through personal communication, we learnt that it could have an impact on the classifier effectiveness. We wish to explore into it in near future.

5.2.3 ROC Curves and AUC Values

We plot ROC curves and compute corresponding Area Under the Curve values for unjudged qrels only (obtained after step 2 of the process explained in previous section) following methods outlined in Fawcett [2006]. For each dataset considered, we plot the curves for each level of training level sample considered. Each ROC curve labeled ($s10, \ldots s90$) represents fraction of actual qrels (i.e. $10 \ldots 90$) used for training. In Figure 5.1, we plot ROC curves for data compiled for TREC 2006 Terabyte track. We plot the corresponding $< fpr, tpr >$ values at various training sample levels considered. Similarly, in Figure 5.2 and Figure 5.3, we plot the ROC curves for TREC 7 adhoc track and TREC 8 adhoc track. In Figure 5.4, we plot the ROC curves for the data compiled for TREC 2010 Web track (adhoc).Since, the focus of current work is not to compare different classifiers, we plot ROC curves for only one (of 25) training sample compiled for a given topicset. [3]

In addition, we report the effectiveness in terms of Area Under the Curve (AUC) values as explained in Fawcett [2006] . We compute AUC values as follows:

- We first compute topic-level AUC value for each topic in a given topicset.
- We then average those topic-level AUC values to obtain topicset-level AUC values.
- For a given sample level, we then compute mean of topicset-level AUC values across 25 samples

In Table 5.2, we present the AUC values obtained for the classifier across all four ad-hoc datasets. For each dataset, we report mean (along with its standard deviation, σ) topicset-level AUC values for each dataset considered. We observed that on an average, we are able to score or rate relevant documents higher than that of non-relevant documents [4]. The average AUC values (for a given topicset) encouraged us to explore into estimating effectiveness measures using probability values obtained for unjudged documents (in qrels).

Next, we outline our approach to estimate effectiveness of a ranking function in terms of estimating Average Precision (estAP).

5.3 Estimating Average Precision (Adhoc)

In this section, we derive a dynamic programming based approach to estimate Average Precision. We make use of two NXN matrices ($N = 1000$), expectedSP$_{[N][N]}$ and $Pr_{[N][N]}$, to compute

[3] Further, it has been found that the method applied to plot ROC curves, across different topics is not truly apporpriate to draw conclusion from observing the curves.

[4] We did not report per-topic AUC values for each dataset considered in this thesis. We did observe that some topics had better AUC values than others. We wish to thoroughly explore into topic-level AUC values later.

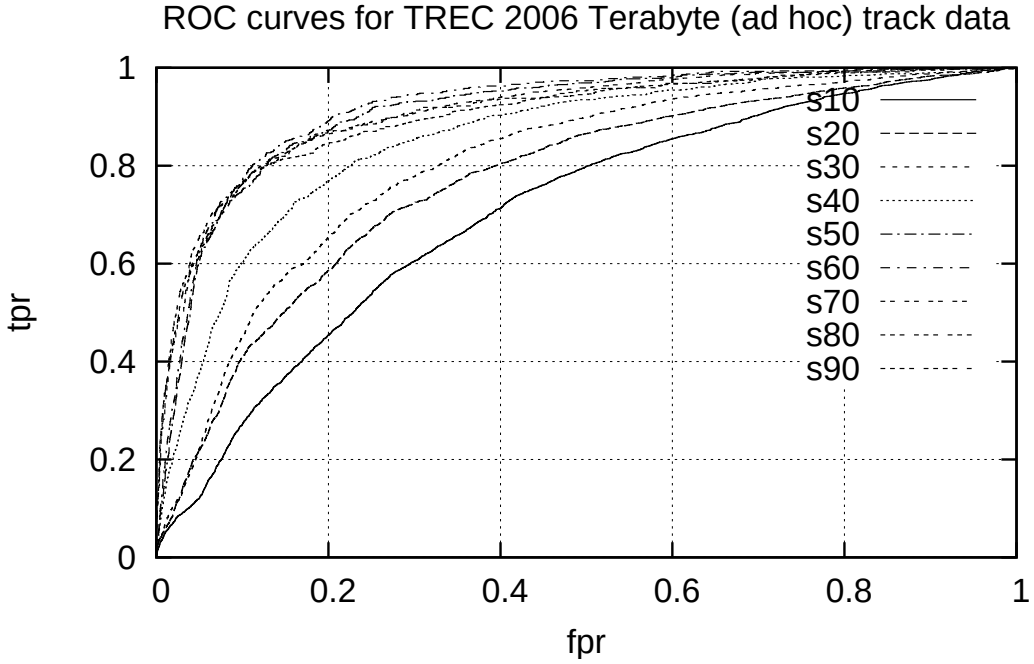


Figure 5.1: TREC 2006 Terabyte Track: ROC curves showing various training samples.

Training	TREC 2006 Terabyte	TREC 7 ad-hoc	TREC 8 ad-hoc	TREC 2010 Web
(%)	AUC (mean) , (σ)	AUC (mean), (σ)	AUC (mean), (σ)	AUC (mean), (σ)
10	0.714 ,($\pm$ 0.013)	0.576, ($\pm$ 0.012)	0.650, ($\pm$ 0.010)	0.763 ,($\pm$ 0.0128)
20	0.787, ($\pm$ 0.010)	0.663, ($\pm$ 0.010)	0.718, ($\pm$ 0.009)	0.822,($\pm$ 0.010)
30	0.827, ($\pm$ 0.009)	0.718, ($\pm$ 0.010)	0.756, ($\pm$ 0.008)	0.849,($\pm$ 0.008)
40	0.850, ($\pm$ 0.008)	0.755, ($\pm$ 0.009)	0.787, ($\pm$ 0.008)	0.865 ,($\pm$ 0.009)
50	0.866, ($\pm$ 0.006)	0.787, ($\pm$ 0.014)	0.819, ($\pm$ 0.009)	0.874,($\pm$ 0.007)
60	0.881, ($\pm$ 0.007)	0.802, ($\pm$ 0.010)	0.835, ($\pm$ 0.012)	0.884 , ($\pm$ 0.008)
70	0.893 ,($\pm$ 0.008)	0.826, ($\pm$ 0.010)	0.848, ($\pm$ 0.012)	0.889 ,($\pm$ 0.008)
80	0.899, ($\pm$ 0.008)	0.846, ($\pm$ 0.014)	0.863, ($\pm$ 0.012)	0.896, ($\pm$ 0.008)
90	0.912, ($\pm$ 0.008)	0.857, ($\pm$ 0.017)	0.878, ($\pm$ 0.014)	0.900, ($\pm$ 0.013)

Table 5.2: Area Under Curve (AUC) across four ad-hoc test datasets. Foreach dataset, we present mean AUC and σ values computed over 25 samples.

expectedSP@N values for a user browsing through top N ranks. Previous work by Ashkan and Clarke [2011], adopted similar dynamic programming approach to determine *informativeness* of several of effectiveness measure. In their work, however, Ashkan and Clarke considered an NXR matrix, where R is equal to total number of relevant documents present in the corpus for a given topic.

First, we outline our approach to compute $Pr_{[N][N]}$ matrix, where the value in i^{th} row and j^{th} column, $Pr_{[i][j]}$, corresponds to the probability of user finding j ($0 \leq j \leq i$) relevant documents

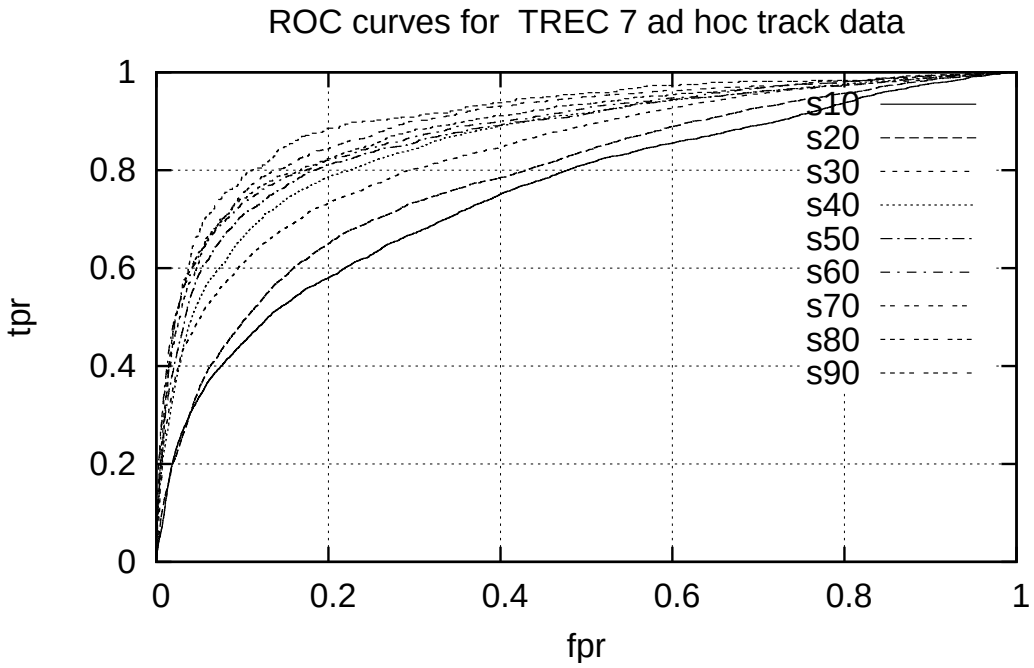


Figure 5.2: TREC 7 ad-hoc: ROC curves at various training sample levels

.

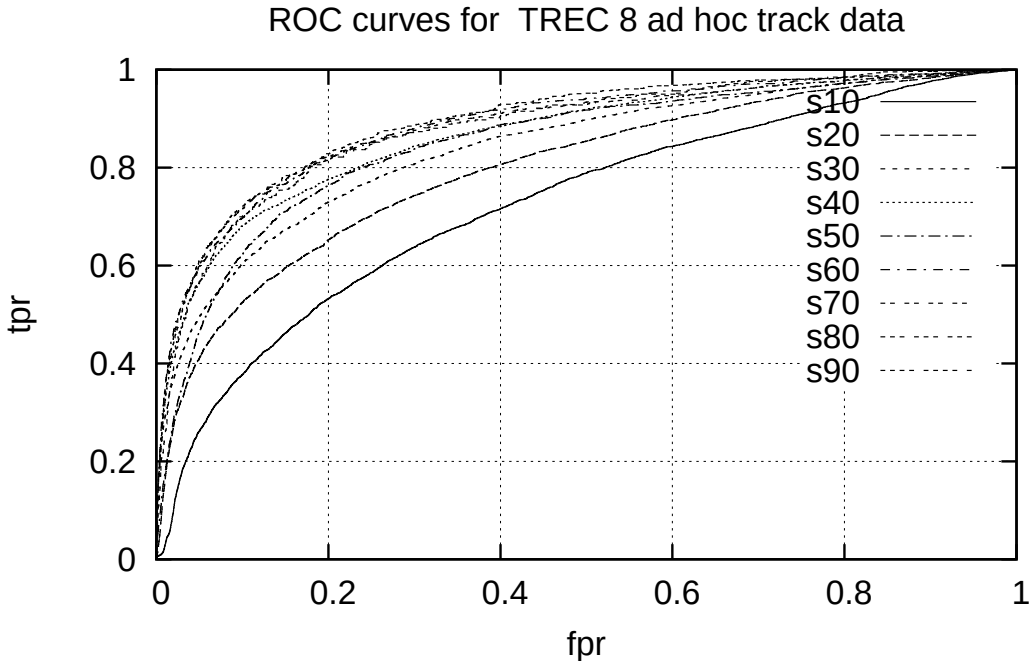


Figure 5.3: TREC 8 ad-hoc: ROC curves at various training samples.

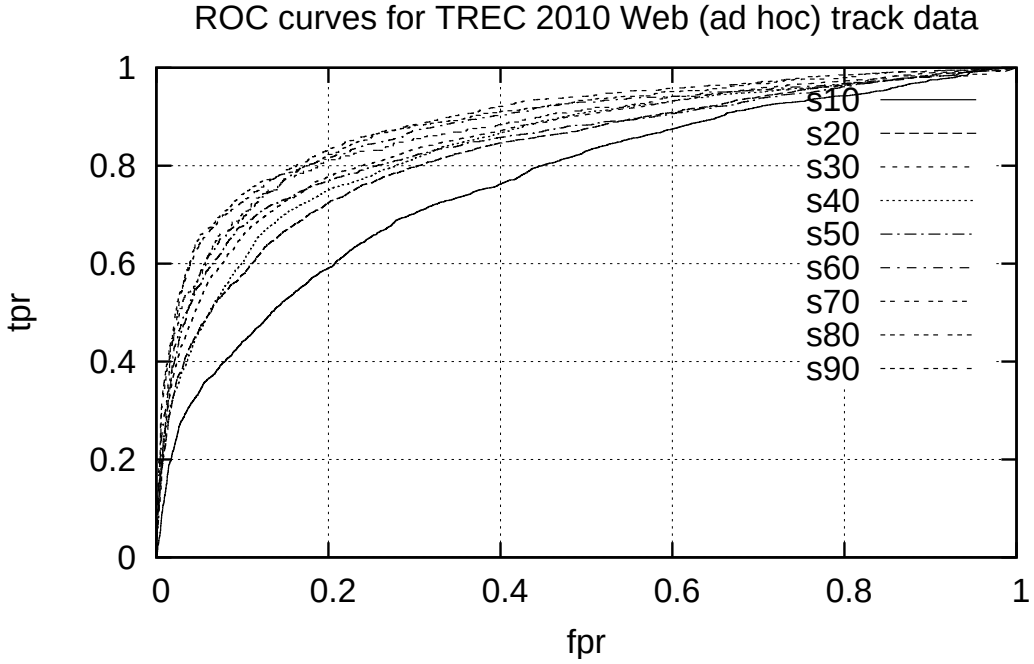


Figure 5.4: TREC 2010 Web Track: ROC curves for various training samples

at rank i.

Let D be a ranked set of documents

$$D =< d_1, d_2, d_3, d_4 \ldots d_n >$$

retrieved by a system for a given query q, with associated probability of relevance values

$$P =< p_1, p_2 \ldots p_n >$$

obtained from human assessments or output of the classifier.

At rank 1 (i.e. $i = 1$), with probability of relevance p_1, user can find either 0 ($Pr_{[1][0]}$) with a probability $(1 - p_1)$, or find 1 relevant ($Pr_{[1][1]}$) with a probability p_1.

At rank 2, there are three possibilties:

0 relevant: Both documents are not relevant:

$$Pr_{[2][0]} = (1 - p_1) \cdot (1 - p_2) \tag{5.10}$$

1 relevant: Either d_1 or d_2 is relevant, but not both

$$Pr_{[2][1]} = (((1 - p_1) \cdot p_2) + ((1 - p_2) \cdot p_1)) \tag{5.11}$$

2 relevant: Both d_1 and d_2 are relevant

$$Pr_{[2][2]} = (p_1 \cdot p_2) \tag{5.12}$$

Example: Let us consider two hypothetical document sets, set_A= $[1, 1]$ and $set_V = [0.5, 0.5]$. Substituting the values in above, for set_A, we obtain

$$Pr_{20} = 0, Pr_{21} = 0, Pr_{22} = 1 \tag{5.13}$$

i.e. at rank 2, user is expected to find $((0 \times 0) + (0 \times 1) + (1 \times 2) = 2$ relevant documents. Similarly, for set_V, it is

$$Pr_{20} = 0.25, Pr_{21} = 0.5, Pr_{22} = 0.25 \tag{5.14}$$

of finding $((0.25 \times 0) + (0.5 \times 1) + (0.25 \times 2) = 1$ relevant document.

Since document d can have two possible outcomes, it would require 2^i exponential computations to compute all possible probability values at rank i. We simplified it to $O(i \times i)$ computations by adopting dynamic programming principle.

At any given rank i, for a user to find j relevant documents, it could mean

- User is expected to have found $(j - 1)$ documents relevant at rank $(i - 1)$ ($Pr_{[i-1][j-1]}$) and the probability of user finding j relevant rank i is equal to $Pr_{[i-1][j-1]} * p_i$
- User is expected to have found j relevant documents at rank $(i - 1)$, and is likely to find j relevant at rank i with a probability of $(1 - p_i) * Pr_{[i-1][j]}$

Applying this principle, we fill the $Pr_{N][N]}$ matrix from top to bottom one row at a time. In each row (say i), we fill out the matrix from left to right ($j \leq i$) and update each element using Eq. 5.15.

$$Pr_{[i][j]} = \oplus \begin{cases} (Pr_{[i-1][j-1]} * p_i) & \text{rel} \\ (1 - p_i) * (Pr_{[i-1][j]}) & \text{nonrel} \end{cases} \tag{5.15}$$

Example: In Table 5.3 we show the computation of expected number of relevant documents using dynamic programming approach for a ranked list $D = [1, 1, 0.8, 1, 0.1]$

Next, we built upon the dynamic programming approach in measuring contribution towards Sum of Precision (SP) value from going through each document in the ranked list.

rank	R= 0	=1	2	3	4	5
1	0.00	1.0	-	-	-	-
2	0.00	0.0	1.0	-	-	-
3	0.00	0.0	0.2	0.8	-	-
4	0.00	0.0	0.0	0.2	0.8	-
5	0.00	0.0	0.0	0.18	0.74	0.080

Table 5.3: Pr for a hypothetical ranked list $D = [1, 1, 0.8, 1, 0.1]$

In estimating Sum of Precision, we consider the non-interpolated definition of AP (inspired by Robertson [2008]). For each document at rank i, it contributes the following (towards SP)

$$P@i \cdot r_i$$

where $P@i$ is precision at rank i and r_i represents the relevance of document at rank i. Let this be referred to as δ_i.

In our approach, at rank i, we compute for each j ($0 \leq j \leq i$) the corresponding expectedSP$_{[i][j]}$ values. For each document at rank i, with probability of being judged relevant p_i (either from human assessments, or from classifier output), it contributes the following

$$((Pr_{[i-1][j-1]} * \delta_{[i][j]}) + (expectedSP_{[i-1][j-1]})) * p_i \tag{5.16}$$

where $Pr_{[i-1][j-1]}$ represents the probability of user finding $j-1$ relevant documents at rank $i-1$, expectedSP$_{[i-1][j-1]}$ represents the Sum of Precision value from observing $j-1$ relevant documents at rank $i-1$. Here $\delta_{[i][j]}$ indicates the relative gain (in SP) from observing a relevant document at rank i.

Now, considering the possibility of the document at rank i not being relevant

$$(Pr_{[i-1][j]} * expectedSP_{[i-1][j]}) * (1 - p_i) \tag{5.17}$$

Combinining, both, we obtain the value of expectedSP$_{[i-1][j-1]}$ as

$$expectedSP_{[i][j]} = \oplus \begin{cases} ((Pr_{[i-1][j-1]} * \delta_{[i][j]}) + (expectedSP_{[i-1][j-1]})) * p_i & \text{(rel)} \\ (Pr_{[i-1][j]} * expectedSP_{[i-1][j]}) * (1 - p_i) & \text{(non rel)} \end{cases} \tag{5.18}$$

where $\delta_{[i][j]}$ is obtained as

$$P@ij \cdot p_i \tag{5.19}$$

where $P@ij$ represents precision for finding j relevant at rank i (i.e. $\frac{j}{i}$).

We then obtain expectedSP at rank i $expectedSP@i$ as follows:

$$expectedSP@i = \sum_{j=0}^{i} expectedSP_{[i][j]} \tag{5.20}$$

Example: Consider two hypothetical ranked sets $set_A = [1, 1]$ and $set_V = [0.5, 0.5]$. Computing exectedSP at rank 2 ($expSP_2$) for set_A

$$expectedSP@2 = 0 * 0 + (0 * 1 + 0 * 0.5) + 1(2) = 2$$

and similarly, expectedSP (at rank 2) for set_V would be

$$expectedSP@2 = (0.25 * 0 + (0.25 * (0.25) + 0.5 * (0.25)) + 0.25 * (1 * 0.5 + 0.5 * 1)) = 0.4375$$

As observed in case of set_A, in case of binary relevant values, our estimate of sum of precision values would be equivalent to regular sum of precision values .

To summarize, for a given ranked list of documents, starting at the top-row, we fill both matrices in a top-down fashion one row at a time (expectedSP$_{[0][0]}$, to expectedSP$_{[N][N]}$ and $Pr_{[0][0]}$ to $Pr_{[N][N]}$). In each row, i, we we traverse from left to right and fill out all j column values ($j \leq i$). Each element, expectedSP$_{[i][j]}$ can be obtained as

$$expectedSP_{[i][j]} = \oplus \begin{cases} p_i * (expectedSP_{[i-1][j-1]}) + (p_i * (Pr_{[i-1][j-1]}) * (P@ij * p_i)) & (\text{ rel }) \\ ((1 - p_i) * (Pr_{[i-1][j]})) * expectedSP_{[i-1][j]} & (\text{ non rel}) \end{cases} \tag{5.21}$$

where p_i represents probability of document at rank i being relevant (obtained from human assessor or classifier). The value $Pr_{[i][j]}$, likelihood of a user finding j relevant documents at rank i, is updated in parallel as shown in Eq. 5.15.

For a given ranked result, we estimate the sum of precision value at rank i can be obtained as a sum of values in each column in the row, i.e.

$$expectedSP@i = \sum_{j=0}^{i} expectedSP_{[i][j]} \tag{5.22}$$

Example 1: In Table 5.4, we demonstrate the process of tabulating the values for an hypothetical ranked result set $AB = [1, 1, 0, 1, 0]$. Each column represents the number of relevant documents user may find at given rank. In Table 5.5, we show corresponding Pr values. As observed, we only populate lower triangle of the matrices.

rank	R= 0	=1	2	3	4	5
1	0.0	1.0	-	-	-	-
2	0.0	0.0	2.0	-	-	-
3	0.0	0.0	2.0	0.0	-	-
4	0.0	0.0	0.0	2.75	0.0	-
5	0.0	0.0	0.0	2.75	0.0	0.0

Table 5.4: expectedSP for a hypothetical ranked list AB= $[1, 1, 0, 1, 0]$

rank	R =0	=1	=2	=3	=4	=5
1	0.0	1.0	-	-	-	-
2	0.0	0.0	1.0	-	-	-
3	0.0	0.0	1.0	0.0	-	-
4	0.0	0.0	0.0	1.0	0.0	-
5	0.0	0.00	0.0	1.0	0.0	0.0

Table 5.5: Pr for the ranked list AB = $[1, 1, 0, 1, 0]$

Example:2 Let us assume that we have another hypothetical set of ranked result $AB_{e1} = [1, 1, 0.1, 1, 0.8]$. Here, we have a mixture of binary judgements (from humans) and real-values (from classifier output). In Table 5.6 and Table 5.7, we show the corresponding expectedSP and Pr values.

rank	R= 0	=1	2	3	4	5
1	0.0	1.00	-	-	-	-
2	0.0	0.00	2.0	-	-	-
3	0.0	0.00	1.8	0.21	-	-
4	0.0	0.00	0.0	2.475	0.31	-
5	0.0	0.00	0.0	0.495	2.5028	0.312

Table 5.6: *expectedSP* for a hypothetical ranked list $AB_{e1} = [1, 1, 0.1, 1, 0.8]$

rank	R= 0	=1	2	3	4	5
1	0.0	1.0	-	-	-	-
2	0.0	0.0	1.0	-	-	-
3	0.0	0.0	0.9	0.1	-	-
4	0.0	0.0	0.0	0.9	0.10	-
5	0.0	0.0	0.0	0.18	0.74	0.080

Table 5.7: Pr matrix for a hypothetical ranked listi $AB_{e1} = [1, 1, 0.1, 1, 0.8]$

Example 3: Let us consider a third hypothetical ranked list $AB_{e2} = [1, 1, 0.8, 1, 0.1]$. In Table 5.8 and Table 5.9 , we show the corresponding expectedSP and Pr matrices values.

rank	R= 0	=1	2	3	4	5
1	0.0	1.00	-	-	-	-
2	0.0	0.00	2.0	-	-	-
3	0.0	0.00	0.4	2.24	-	-
4	0.0	0.00	0.0	0.55	3.04	-
5	0.0	0.00	0.0	0.495	2.796	0.312

Table 5.8: expectedSP for a hypothetical ranked list $AB_{e2} = [1, 1, 0.8, 1, 0.1]$

rank	R= 0	=1	2	3	4	5
1	0.00	1.0	-	-	-	-
2	0.00	0.0	1.0	-	-	-
3	0.00	0.0	0.2	0.8	-	-
4	0.00	0.0	0.0	0.2	0.8	-
5	0.00	0.0	0.0	0.18	0.74	0.080

Table 5.9: Pr for a hypothetical ranked list $AB_{e2} = [1, 1, 0.8, 1, 0.1]$

Now, computing the expectedSP@5 values for all three hypothetical ranked lists, which is equal to sum of expectedSP$_{[5][.]}$ values shown for corresponding expectedSP matrices:

$$\begin{aligned} AB@5 &= 2.75 \\ AB_{e1}@5 &= 3.309 \\ AB_{e2}@5 &= 3.5996 \end{aligned}$$

TREC 2006 **Terabyte example:** To illustrate the functioning of our approache, we plot our estimated values expectedSP@1000 (y-axis) vs true [5] SP@1000 values for a topic (Topic number: 842) from Terabyte2006 track(Figure 5.5). In first case (top plot), we train our classifier using 30% random sample and use our trained classifier to estimate the probability for rest of qrels. We then compute expectedSP@1000 value for each participant system. Similarly, in bottom plot, we plot expectedSP@1000 values against SP@1000 values when our classifier was trained with 50% of qrels. We observe that our estimated values expectedSP@1000 under-estimate true SP@1000 for almost all systems across both samples.

5.3.1 Normalization

Under standard evaluation, when computing Average Precision (AP), the sum of precision (SP) value at a given rank is normalized by using $|R|$, the total number of relevant documents identified through pooling process during collection compilation. However, the notion of total relevant documents being bounded by those identified from pooling exercise has been challenged several

[5] Computed using trec_eval -m 1000 option

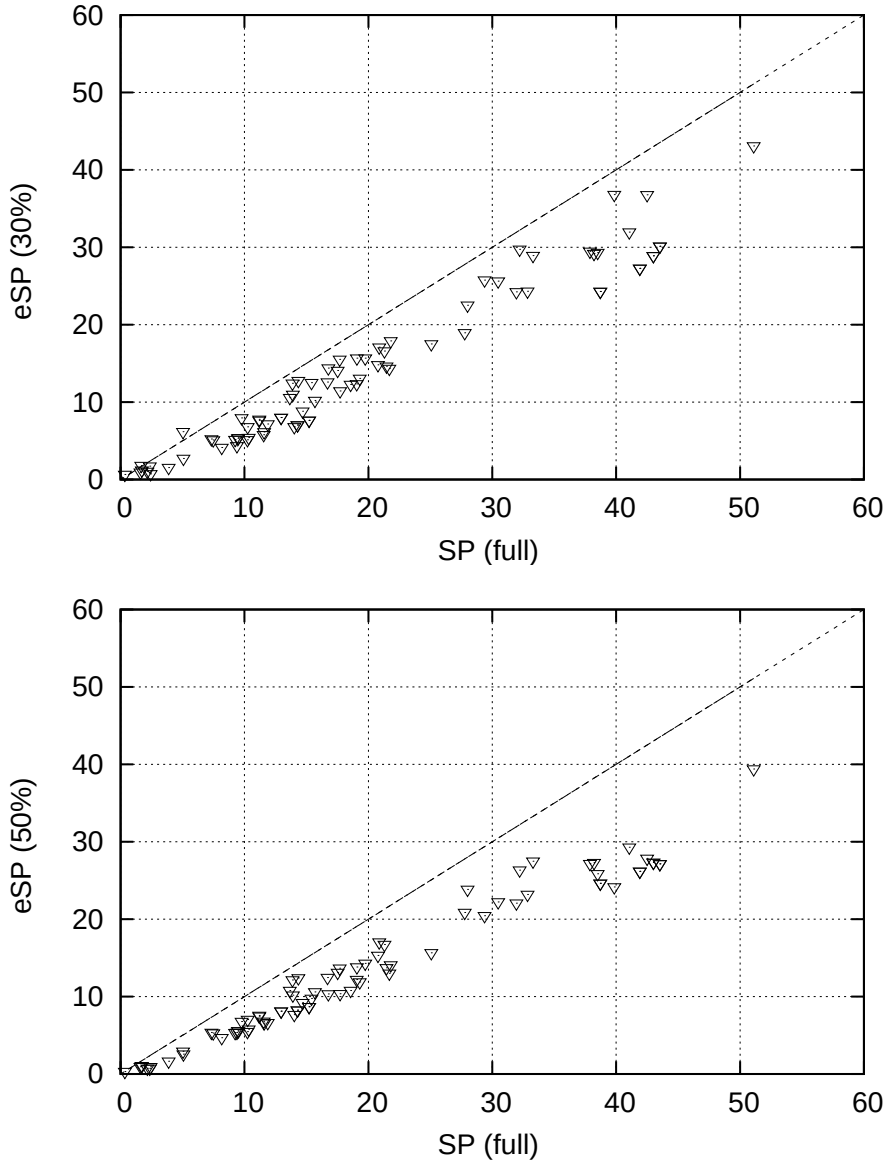


Figure 5.5: Topic 842: expectedSP@1000 vs. SP@1000 for two different training samples - 30% (top) and 50% (bottom)

times in the past. Zobel [1998] experimentally proved that it is impossible for current pooling methods to identify all possible relevant documents for a given topic [6].

In our approach, we normalize the expectedSP values by considering an ideal ranked list I' and using it to normalize expectedSP obtained from system ranked list. Given the set of documents

[6]Recently, research into evaluating ranking functions in context of legal search domain, has focused on estimating the number of relevant documents for a given topic Tomlinson and Hedin [2011].

pooled for a given topic (through pooling process), we transform it into an ideal ranked list $I^{'}$

$$I^{'} =< p_1, p_2, p_3 \ldots p_k, 0, \ldots > \tag{5.23}$$

where $p_i \geq p_j, \forall i < j$. That is, we sort documents in decreasing order of their probability values. In case of binary values, ideal ranked list would then be equivalent to a list of documents where top $|R|$ documents are relevant followed by non-relevant documents.

$$I =< 1, 1, 1, 1, \ldots 1(R), 0, 0, 0 \ldots >$$

whose sum of precision (SP) value is equal to $|R|$, i.e., total number of relevant documents identified for the topic.

Example: First, let us consider an ideal ranked list I_1 containing only binary relevance values

$$I_1 = [1, 1, 1, 0, 0]$$

In Table 5.10 and Table 5.11, we show corresponding expectedSP and Pr matrices for for set I_1. As mentioned before, the expectedSP(I_1) (containing binary values only) is equal to total number of relevant documents (i.e. 3). Now, we consider a second ideal ranked list I_2 containing both absolute and predicted probability of relevance values

$$I_2 = [1, 1, 1, 0.8, 0.1]$$

In Table 5.12 and Table 5.13, we show the computation of expectedSP and Pr values for set I_2. From Table 5.12, we compute expectedSP(I_2)) for an ideal ranked list I_2 ($[1, 1, 1, 0.8, 0.1]$) is

$$= 2.796 + 0.312 + 0.54 = 3.6496$$

rank	R= 0	=1	=2	=3	=4	=5
1	0.0	1.0	-	-	-	-
2	0.0	0.0	2.0	-	-	-
3	0.0	0.0	0.0	3.0	-	-
4	0.0	0.0	0.0	3.0	0.0	-
5	0.0	0.0	0.0	3.0	0.0	0.0

Table 5.10: expectedSP matrix for an ideal documents $I_1 = [1, 1, 1, 0, 0]$.

Given an ideal list, we measure expectedSP($I^{'}$) to measure expectedSP for a user browsing through the ideal ranked list $I^{'}$. On computing expectedSP($I^{'}$) values, we obtain estimate of

rank	R= 0	=1	2	3	4	5
1	0.0	1.0	-	-	-	-
2	0.0	0.0	1.0	-	-	-
3	0.0	0.0	0.0	1.0	-	-
4	0.0	0.0	0.0	1.0	0.0	-
5	0.0	0.00	0.0	1.00	0.0	0.0

Table 5.11: Pr matrix for an ideal order of absolute relevant list $I_1 = [1, 1, 1, 0, 0]$.

rank	R= 0	=1	2	3	4	5
1	0.0	1.00	-	-	-	-
2	0.0	0.00	2.0	-	-	-
3	0.0	0.00	0.0	3.00	-	-
4	0.0	0.00	0.0	0.6	3.04	-
5	0.0	0.00	0.0	0.54	2.796	0.312

Table 5.12: expectedSP matrix for an ideal set $I_2 = [1, 1, 1, 0.8, 0.1]$

rank	R= 0	=1	2	3	4	5
1	0.00	1.0	-	-	-	-
2	0.00	0.0	1.0	-	-	-
3	0.00	0.0	0.0	1.0	-	-
4	0.00	0.0	0.0	0.2	0.8	-
5	0.00	0.0	0.0	0.18	0.74	0.080

Table 5.13: Pr matrix for an ideal set $I_2 = [1, 1, 1, 0.8, 0.1]$

Average Precision ($estAP$) value for a ranking function s as

$$estAP(s) = \frac{expectedSP(s)}{expectedSP(I^{'})} \tag{5.24}$$

Returning to our two hypothetical ranked lists from previous section, AB_{e1} and AB_{e2}, we normalize their expectedSP@5 values with that of expectedSP($I^{'}$) values to their corresponding estAP values

$$\begin{aligned} AB_{e1} &= 0.9066 \\ AB_{e2} &= 0.9862 \end{aligned}$$

TREC 2006 **Terabyte Example:** In Figure 5.6 we plot estimate Average Precision ($estAP$) (y-*axis*) values with Average Precision (AP) (x-*axis*) values for all systems for one topic (Topic 842 of Terabyte 2006 track). In top plot, we show $estAP$ vs AP values, when 30% of qrels sampled (for training) for topic 842. In bottom plot, we show $estAP$ vs AP values for the same topic when the classifier is trained using 50% of qrels. As observed in the plots, our approach over-estimates $estAP$ values of few (top ranked) systems in comparison to their actual AP values,

when 50% of sample is used for training.

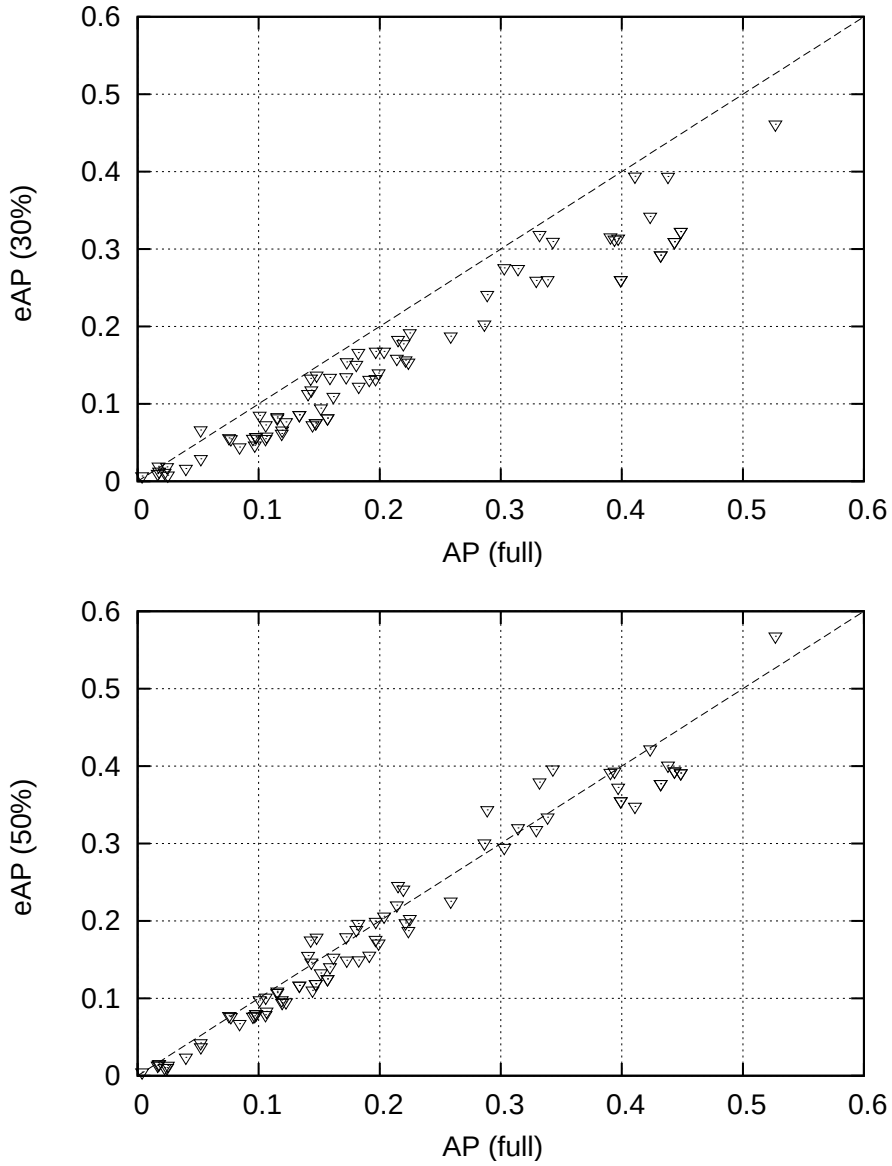


Figure 5.6: Topic 842: estAP vs AP for two different training samples - 30% (top) and 50% (bottom)

Given a topicset containing t topics, we can compute estimate Mean Average Precision (estMAP) values by taking the average of eAP value for each topic

$$estMAP(s) = \frac{1}{t}\sum_{k=1}^{t} estAP_k(s) \tag{5.25}$$

5.4 Adhoc Experiments

In this section, we compare the extent to which our estimate of MAP (estMAP) with that of true MAP [7]. The outline of our experiments is as follows:

1. Using *completed qrels* , i.e. documents with binary (training) values and probability estimates (obtained from the classifier), we compute estMAP as explained in previous section.
2. We compute estMAP across 25 SAMPLEs compiled at each sample level and compare the extent to which the estimate values compare with that of true MAP.

In our experiments, we compare estMAP and MAP values based on two factors: Kendall Tau (τ) value and root mean square error(rmse) value. Kendall Tau, τ, computed as shown below:

$$\tau = \frac{P - N}{(P + N)} \tag{5.26}$$

reflects the extent to which system orderings based on our estimated values (estMAP) are in accordance with their ordering based on their true (MAP) values. In computing τ, P represents number of system pairs whose ordering is in accordance and N represents number of pairs whose ordering is not. In general, τ values greater than or closer to 0.9 are preferred.

The second factor ,rmse value, computed as :

$$rmse = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}(T_i - E_i)} \tag{5.27}$$

measures difference between true (MAP) value and estMAP values of each system. In above equation, N_s represents total number of systems considered, T_i is true (MAP) value of system i, and E_i represents (estMAP) value of system i. In general, smaller rmse indicates that our estimated values are closer to the true measure.

TREC 2006 Terabyte track:

In total, 80 systems took part in this track. As explained before, we train our our classifier using different qrel samples [8] ($10\%, 20\% \ldots 90\%$) and use trained classifier to estimate probability for remaining qrels (i.e. test qrels). Any unjudged document retrieved that is not part of official qrels (compiled as part of TREC 2006 workshop) is considered as non-relevant. We then compute $estMAP$ values (at rank 1000). We repeat the whole process 25 times and present the

[7] Obtained by full qrels and trec_eval software.
[8] samples from Section 5.2.2.

mean Kendall τ and rmse values across the 25 samples (along with 95% confidence intervals) in Table 5.14. As observed in the table, our measures are comparable with only 40% of actual qrels.

In Figure 5.7 we plot true MAP values against $estMAP$ values for all (80) participating systems. On X-axis, we report participating systems in decreasing order of their true MAP value [9]. For each system, we then plot their true MAP value (with + symbol) and $estMAP$ values (over 25 samples, with 'x' mark representing the median value) on Y-axis. The top plot corresponds to case when 10% of qrels are sampled for training, and in the bottom one, we plot system values when 50% of qrels are sampled for training (all samples are reported in Appendix A).

Sample	Kendall Tau (τ)	rmse
10	0.7411 (0.7222 — 0.7598)	0.1148 (0.1103 — 0.1191)
20	0.8292 (0.8160 — 0.8424)	0.0947 (0.0914 — 0.0980)
30	0.8687 (0.8545 — 0.8828)	0.0775 (0.0745 — 0.0804)
40	0.8981 (0.8877 — 0.9085)	0.0624 (0.0598 — 0.0649)
50	0.9242 (0.9177 — 0.9306)	0.0486 (0.0461 — 0.0511)
60	0.9373 (0.9317 — 0.9429)	0.0381 (0.0368 — 0.0392)
70	0.9520 (0.9488 — 0.9551)	0.0263 (0.0246 — 0.0279)
80	0.9651 (0.9613 — 0.9687)	0.0185 (0.0171 — 0.0198)
90	0.9739 (0.9723 — 0.9755)	0.0088 (0.0078 — 0.0096)

Table 5.14: TREC 2006 Terabyte Track: True MAP vs estMAP: Kendall τ and rmse values (with 95% confidence intervals)

TREC-7 ad-hoc track

In next experiment, we compared true MAP values against our estimate of MAP (estMAP) values for systems taking part in TREC 7 ad-hoc track. A total of 103 ranking systems took part in this 7 track. Using a trained classifier on those sampled documents, we computed $estMAP$ values for each instance. In Figure 5.8, we plot estMAP versus actual map for two different training samples created. At top, we plot system effectiveness when 10% of documents are used for training. In bottom plot, we present estimate values when 50% of qrels were sampled for training. Along X-axis, we have systems sorted in decreasing order of their true MAP values. We then plot true MAP values (represented by +) and the estimated MAP (estMAP) values (over 25 samples, with x representing the median) values for each system along Y-axis. The corresponding mean (with 95% confidence intervals) Kendall Tau τ and rmse values over 25 samples are shown in Table 5.15. As observed in the table, the system orderings correlate well ($\tau > 0.9$) when only 30% of qrels are used for training.

[9] calculated using trec_eval software : "trec_eval -M1000 qrelsFile input.sys"

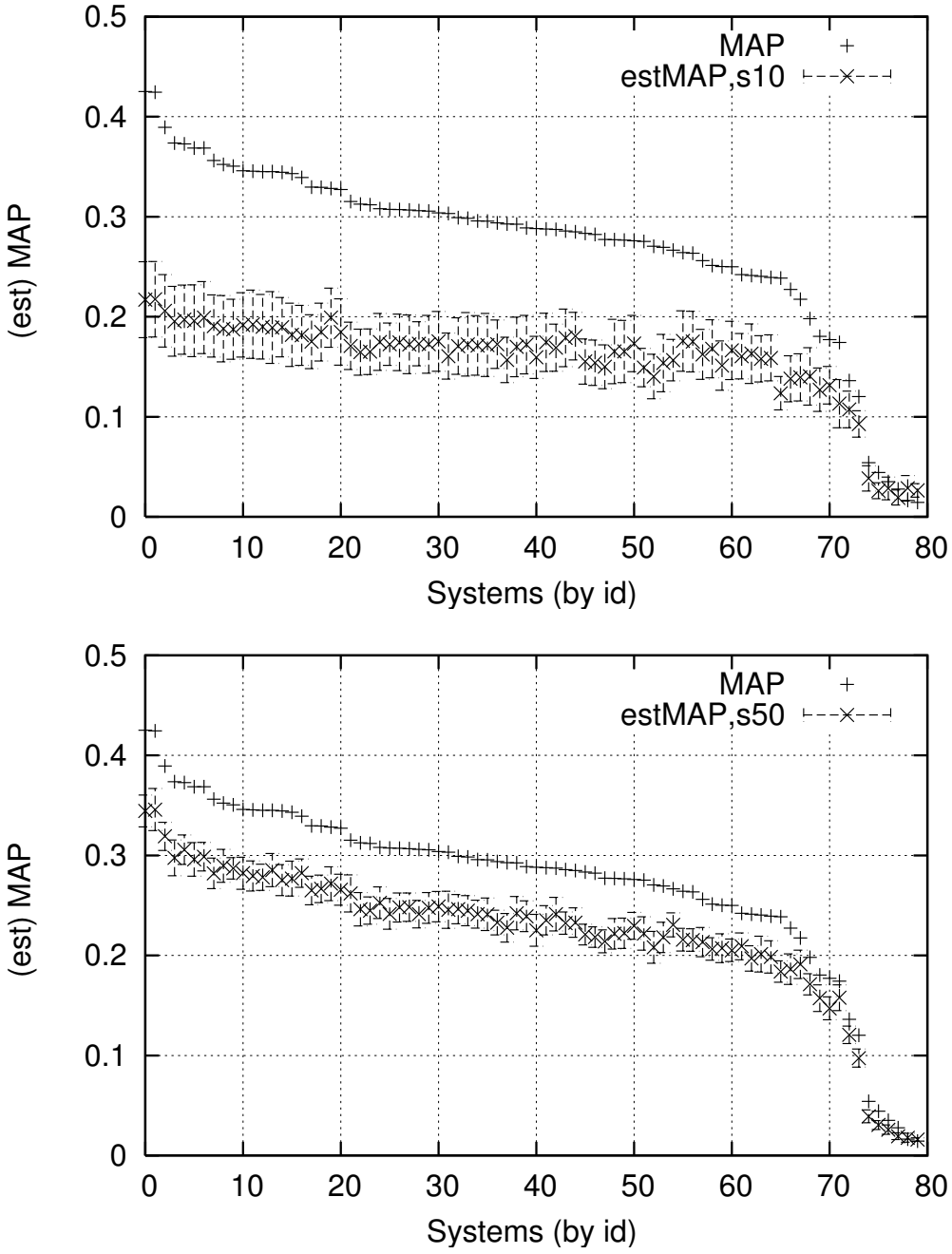


Figure 5.7: TREC 2006 Terabyte plots: system MAP and estMAP values when trained using 10% of qrels(top), when trained using 50% of qrels (bottom).

TREC-8 ad-hoc track

Similarly, in Figure 5.9, we plot estMAP versus actual map for two different training samples from TREC 8 ad-hoc track. A total of 129 systems took part in this track. At top, we plot system

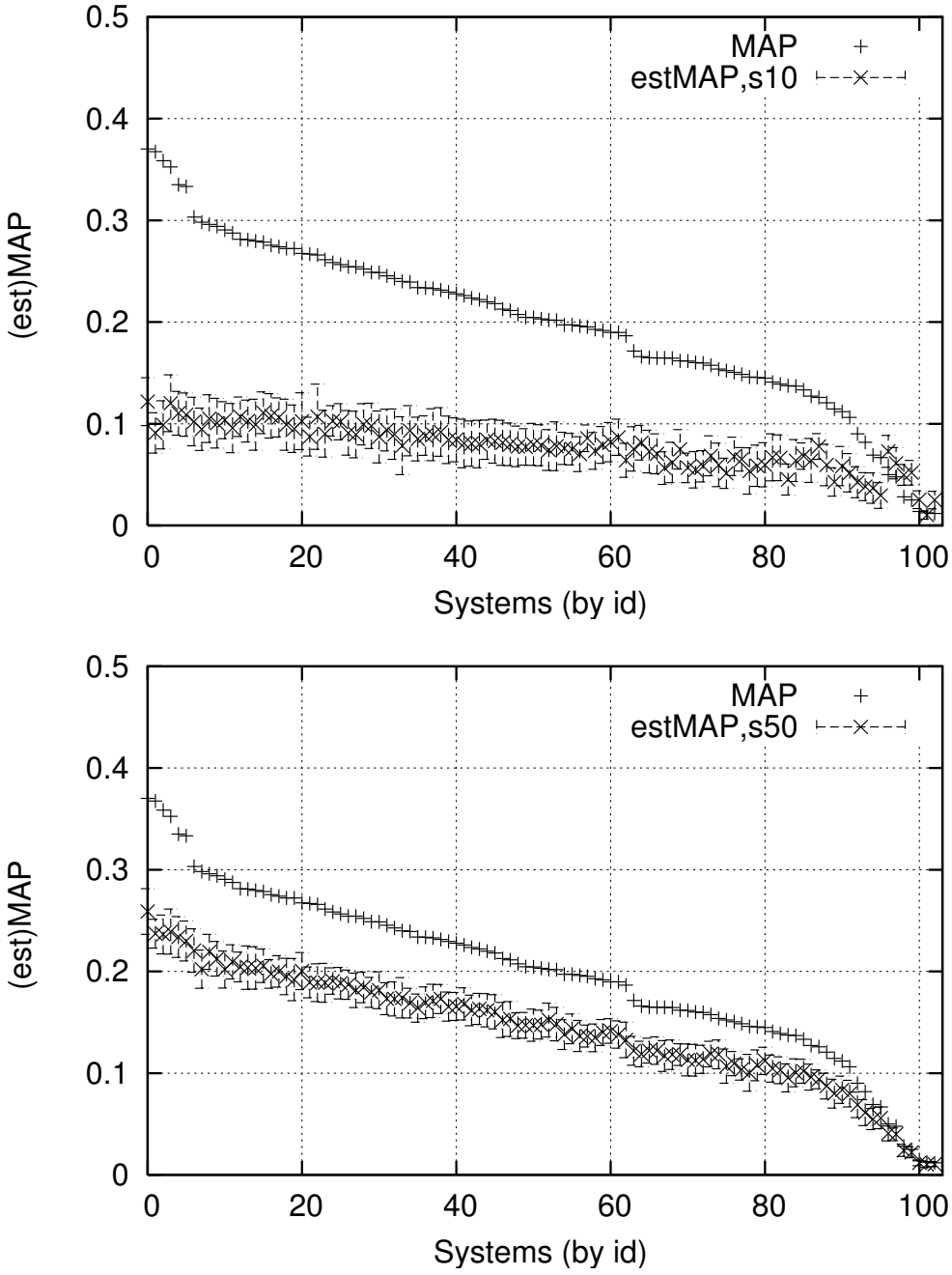


Figure 5.8: TREC7 ad-hoc plots: True MAP versus estimated MAP values: (Top) when trained using 10% of qrels(top), when trained using 50% of qrels (bottom).

effectiveness when 10% of documents were used for sampling to train and estimate relevance. In bottom plot, we present estMAP values when 50% of qrels were sampled for training. As before, the systems were sorted (on X-axis) in sorted order of their true MAP values. We plot the true

Sample	Kendall tau (τ)	rmse
10	0.731 (0.7138 — 0.7480)	0.135 (0.1319 — 0.1381)
20	0.865 (0.8539 — 0.8767)	0.109 (0.1047 — 0.1120)
30	0.912 (0.9039 — 0.9210)	0.093 (0.0906 — 0.0950)
40	0.931 (0.9268 — 0.9365)	0.078 (0.0746 — 0.0799)
50	0.944 (0.9410 — 0.9484)	0.062 (0.0589 — 0.064)
60	0.957 (0.9542 — 0.9614)	0.049 (0.0458 — 0.0512)
70	0.964 (0.9627 — 0.9668)	0.035 (0.0332 — 0.0370)
80	0.976 (0.9745 — 0.9787)	0.023 (0.0211 — 0.0248)
90	0.985 (0.9835 — 0.9872)	0.010 (0.009 — 0.0116)

Table 5.15: TREC-7 ad-hoc results: Kendall Tau (τ) and rmse values (with 95% confidence intervals)

MAP (represented by +) and the estMAP values (over 25 samples, with x representing the median) values on Y-axis. The corresponding Kendall Tau τ and rmse values over 25 iterations are shown in Table 5.16. Similar to previous results, we observe that our estMAP values are closer to true MAP (based on τ) with only 30% of qrels sample.

Sample	Kendall Tau (τ)	rmse
10	0.6998 (0.6787 — 0.7208)	0.166 (0.1639 — 0.1687)
20	0.8429 (0.8337 — 0.8520)	0.137 (0.1331 — 0.1407)
30	0.8835 (0.8759 — 0.8910)	0.114 (0.1106 — 0.1168)
40	0.9092 (0.9021 — 0.9163)	0.097 (0.0942 — 0.1005)
50	0.9259 (0.9214 — 0.9303)	0.075 (0.0721 — 0.0785)
60	0.9449 (0.9401 — 0.9495)	0.059 (0.0564 — 0.0615)
70	0.9566 (0.9526 — 0.9605)	0.043 (0.0404 — 0.0452)
80	0.9673 (0.9647 — 0.9698)	0.030 (0.0287 — 0.0320)
90	0.980 (0.9782 — 0.9816)	0.015 (0.0139 — 0.0163)

Table 5.16: TREC-8 ad-hoc values: τ and rmse values (with 95%confidence intervals)

TREC 2010 **Web Track**

As final adhoc experiments, we picked data compiled for TREC 2010 Web (adhoc) track. The track explored adhoc retrieval on terabytes of data. In total, 56 systems took part in this track. Unlike previous collections, where top 100 documents from each system were pooled, only top 20 documents are pooled in this collection. In Figure 5.10, we plot estMAP versus actual MAP computed for all 56 participant systems. In top plot, we plot estMAP values when 10% of qrels are used for training. Similarly, in bottom plot, we plot estMAP versus actual MAP values for all systems when 50% of qrels are sampled for training. In Table 5.17 we report the extent to

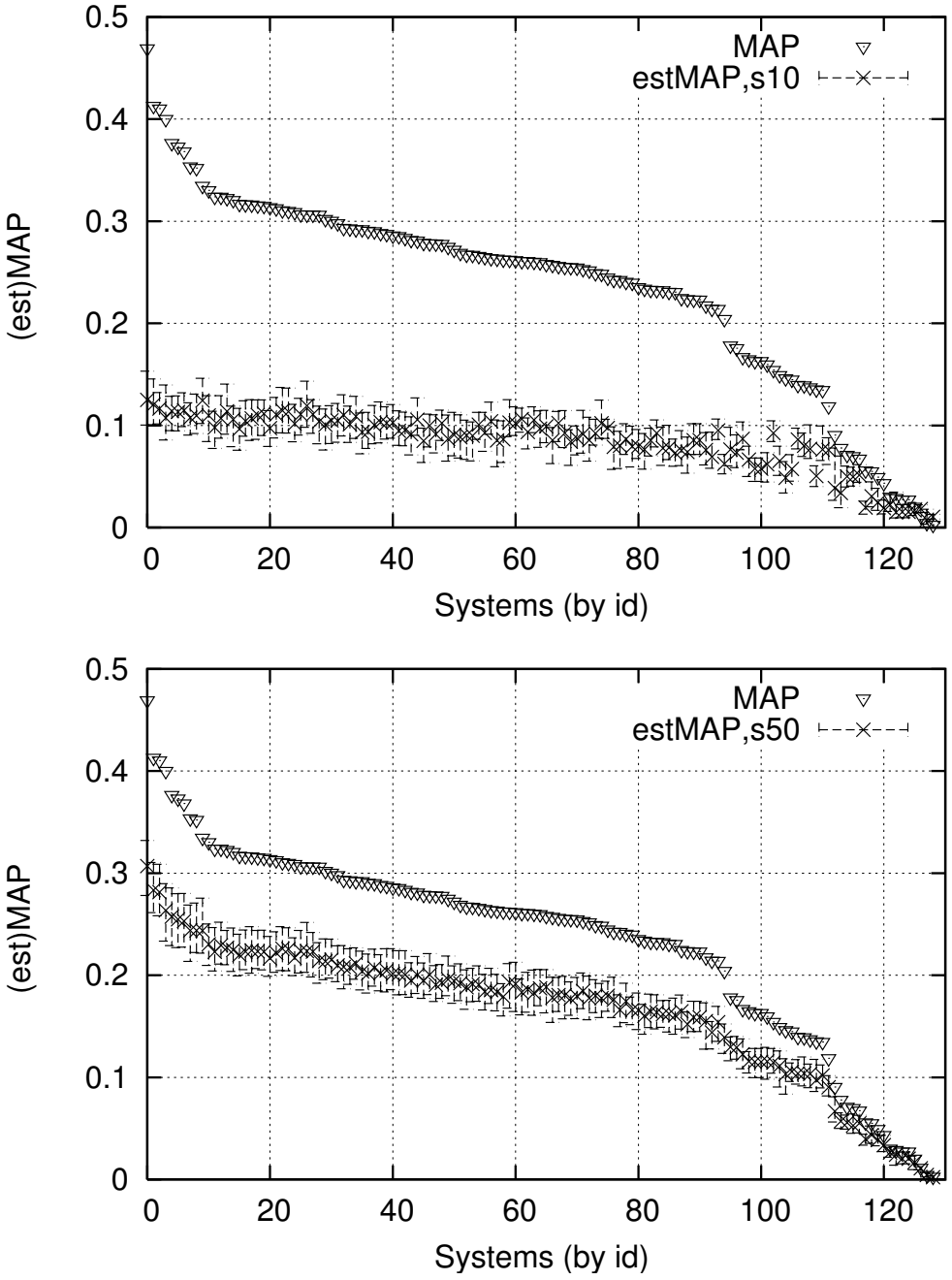


Figure 5.9: TREC-8 ad-hoc plot: system MAP and estMAP values when trained using 10% of qrels(top), when trained using 50% of qrels (bottom).

which estMAP values are comparable to true MAP values in terms of Kendall Tau(τ), and RMSE values.

A key observation from the rmse values obtained for this track is that the rmse values are

Sample	KendallTau (τ)	rmse
10	0.8391 (0.8215 — 0.8566)	0.0139 (0.0123 — 0.0155)
20	0.8759 (0.8661 — 0.8856)	0.0090 (0.0079 — 0.0099)
30	0.8925 (0.8847 — 0.9001)	0.0076 (0.0069 — 0.0082)
40	0.9164 (0.9100 — 0.9228)	0.0060 (0.0054 — 0.0066)
50	0.9307 (0.9242 — 0.9371)	0.0046 (0.0041 — 0.0050)
60	0.9400 (0.9355 — 0.9444)	0.0041 (0.0037 — 0.0045)
70	0.9525 (0.9500 — 0.9549)	0.0032 (0.0029 — 0.003)
80	0.9600 (0.9574 — 0.9625)	0.0029 (0.0026 — 0.0030)
90	0.9716 (0.9696 — 0.97362)	0.00230(0.0022 — 0.00238)

Table 5.17: Web 2010 adhoc track estMAP vs MAP: τ and rmse values (with 95% confidence intervals)

smaller even when only 10% of the sample is used for training. One possible explanation could be due to the smaller number of documents pooled per topic (20 versus the usual 100). We wish to further explore into this hypothesis by simulating pooling only to a smaller depth (20) using previous collections (TREC 7 ad hoc data).

5.5 Discussion

In this chapter, we carried out preliminary exercise into estimating or predicting relevance of an unjudged document using a logistic-regression trained classifier. Our objective behind this exercise is to explore into methods to estimate effectiveness in the presence of unjudged documents in results retrieved by a ranking function. Our preliminary experiments indicates that we are able to estimate or predict the relevance of an unjudged document. Further, we are able to obtain system ordering closer to ordering obtained by complete qrels with only 40% of the qrels used.

We now wish to carry out the following experiments :

- Comparison with approaches such as Büttcher et al. [2007] to explore the efficiency of the approach in overcoming "bias" against novel ranking functions.
- Measure the re-usability of such collection obtained Carterette et al. [2010]
- Incorporate the notion of novelty and diversity Clarke et al. [2008] into estimating effectiveness.

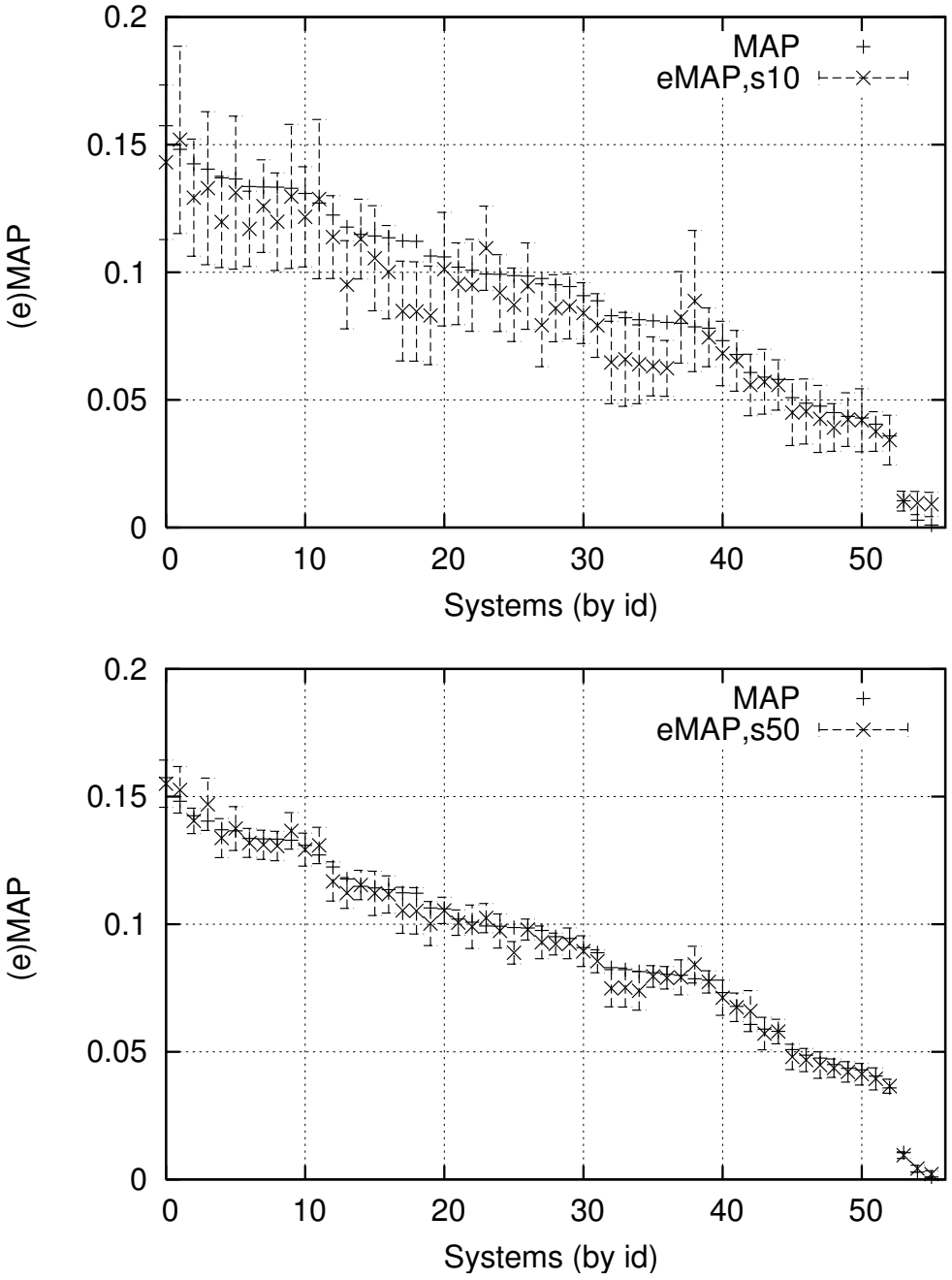


Figure 5.10: TREC 2010 Web (adhoc) track: true MAP and estMAP values when trained using 10% of qrels(top), when trained using 50% of qrels (bottom).

5.6 Summary

In this chapter, we outlined our approach to compare ranking functions in context of incomplete qrels. We first train a classifier and use the trained classifier to predict relevance of an unjudged document. We then outlined an approach to estimate AP value of a ranking function. Through our experiments, we observed that our estimatedAP measures orders ranking functions closer to their true ordering, with less than 50% of qrels.

Chapter 6

Conclusion and Future Work

It has been observed that user queries on the web are terse (Jansen et al. [2000]) and rarely express their true underlying information need. Depending on user context of usage, different queries can be interpreted in different ways. Search engines face a challenging task of satisfying differing information needs associated to satisfy different user expectation(s). One solution is to optimize ranking functions to satisfy diverse sets of information needs. Prior to this thesis, evaluation efforts focused on measuring effectiveness of ranking functions have rewarded for satisfying the most likely intent associated with a given query. In this thesis, we highlighted shortcomings of such approach. We outlined and presented an approach that supports evaluation of ranking functions taking into account the diversity of intents possible for each query.

6.1 Contributions

The contributions of this thesis are as follows:

Ambiguous and Underspecified Queries

As explained in several works prior to this thesis, queries submitted by users are ambiguous. In current thesis, we clearly distinguished between ambiguous and underspecified queries based on the extent of their uncertainty associated with their underlying intents. From our definition, an underspecified query is a query associated with one interpretation. However, an underspecified query could be associated with several subtopics or aspects that user may be interested while entering the query. In terms of ambiguous queries, we refer to those queries that could be associated with different interpretations (e.g.: *ups*). In these queries, each interpretation could possible refer to a different entity. Our motive behind such distinction is to establish the extent

to which a ranking function is expected to satisfy a given user. It can be safely assumed that user interested in one interpretation would not be interested in information needs associated with different interpretations. We made use of such distinction in deriving effectiveness measures in Chapter 3.

Novelty

Building on our proposed framework, each document is independently judged with respect to various information needs (represented as nuggets associated with a given query) covered in that particular document. It alleviates the burden on human assessors without requiring them to subjectively assess the novelty of information covered in a document. We then discounted the utility gained from going through a nugget already seen in previous (higher ranked) documents

Effectiveness Measures:α-nDCG, NRBP

By incorporating the notion of novelty and diversity, we proposed two effectiveness measures α-nDCG (Clarke et al. [2008]) and NRBP (Clarke et al. [2009a]), that reflects

- probability of a user finding novel information while browsing through a ranked list of documents
- extent to which diverse information needs can be satisfied from a given ranked list.

Our proposed framework and measures have been widely adopted and used in evaluating ranking functions in context of web search evaluation. Several works have been proposed that make use of our evaluation method to compare the effectiveness of their implemented methods.

Predicting Relevance and Estimating Effectiveness

In final part of this thesis, we followed methods (Büttcher et al. [2007]) in predicting relevance of an unjudged document, using a classifier trained from existing human assessments. Using predicted probabilities, we then proposed a approach to estimate Average Precision for a given topic. Our preliminary experiments show promising results in being able to estimate effectiveness of a ranking function (in terms of Average Precision) with less than half of actual qrels available.

6.2 Limitations

One of the limitations of our framework of evaluation and measures arises from the need to identify different information need for a given query. Such analysis, if not impossible, is tedious to be

carried out on a regular basis, as it requires exhaustive analysis of all possible intents associated with a given query and identify the most important of those. Song et al. [2010] and Liu et al. [2009] compiled a test collection for evaluation in for ambiguity, based on Wikipedia Disambiguation pages. For example, a query such as **TREC** does have following disambiguation entries in the Wikipedia [1]. as shown in Table 6.1. It is evident that there are certain limitations with respect to the number of entries in the Wikipedia's Disambiguation pages. In order to extend collection, and identify different interpretations that were not listed in Wikipedia's disambiguation pages. Human assessors were given a option to list other intents, other those listed in initial disambiguation entries, and tag them as additional candidate intents for a given query (as shown in Table 6.2).

#	Intent
1	Text Retrieval Conference
2	Texas Real Estate Commission
3	Trans-Mediterrean Renewable Energy Co.
4	T-cell receptor Excision Circles

Table 6.1: Various Intents for the query *TREC*

#	Intent
5	Tennessee Real Estate Commission
6	TREC-UK sport of TREC
7	trec horse rider
...	...
24	Tenderloin Reflection Education Center

Table 6.2: Additional Intents for the query *TREC*

As mentioned before, even with presence of structured databases such as Wikipedia, Song et al. used human assessors to extend and identify different intents associated with a query.

6.3 Future Work

Missing Nuggets

To overcome our limitations of possibly missing few intents associated with a given query, we wish to

- first explore the effects of having missing nuggets towards the stability of our measures. We wish to study these effects by simulating on existing test collections.

[1] `http://en.wikipedia.org/TREC` , as accessed in early 2010

- next identify means to automatically find different information needs by mining query logs.

Incomplete Qrels Experiments

We wish to extend on experiments carried out in Chapter 5 to overcome instances where an unjudged document is retrieved by a system. In Chapter 5, we created random samples of documents judged for a given topic. We wish to investigate into cases of biased qrels, where documents unique to a given system are removed from qrels. This will cover second kind of incomplete collections, where systems not part of original exercise have a bias either for or against them Büttcher et al. [2007]

In addition we wish to explore designing experiments such that smaller samples of documents would suffice for training a classifier. As a preliminary exercise, we wish to explore work carried out by by Sanderson et al. [2010], who investigated into assessor consistency based on the distance between two judgements (between pairs of judged documents). As a preliminary exercise, we wish to investigate into making use of such document pairs for training and its impact in effectively predicting relevance of an unjudged document.

APPENDICES

Appendix A

Adhoc Plots

In this Appendix, we present results of our experiments as outlined in Chapter 5.

A.1 TREC 2006 Terabyte track

In Figure A.1 and Figure A.2 we plot estMAP vs. true MAP values for systems that took part in TREC 2006 Terabyte track.

A.2 TREC 7 ad-hoc data

In Figure A.3 and Figure A.4 we plot estMAP vs. true MAP values for systems that took part in TREC 7 ad-hoc track.

A.3 TREC 8 ad-hoc

In Figure A.5 and Figure A.6 we plot estMAP vs. true MAP values for systems that took part in TREC 8 ad-hoc track.

A.4 TREC 2010 Web track

In Figure A.7 and Figure A.8 we plot estMAP vs. true MAP values for systems that took part in TREC 2010 Web (ad-hoc) track.

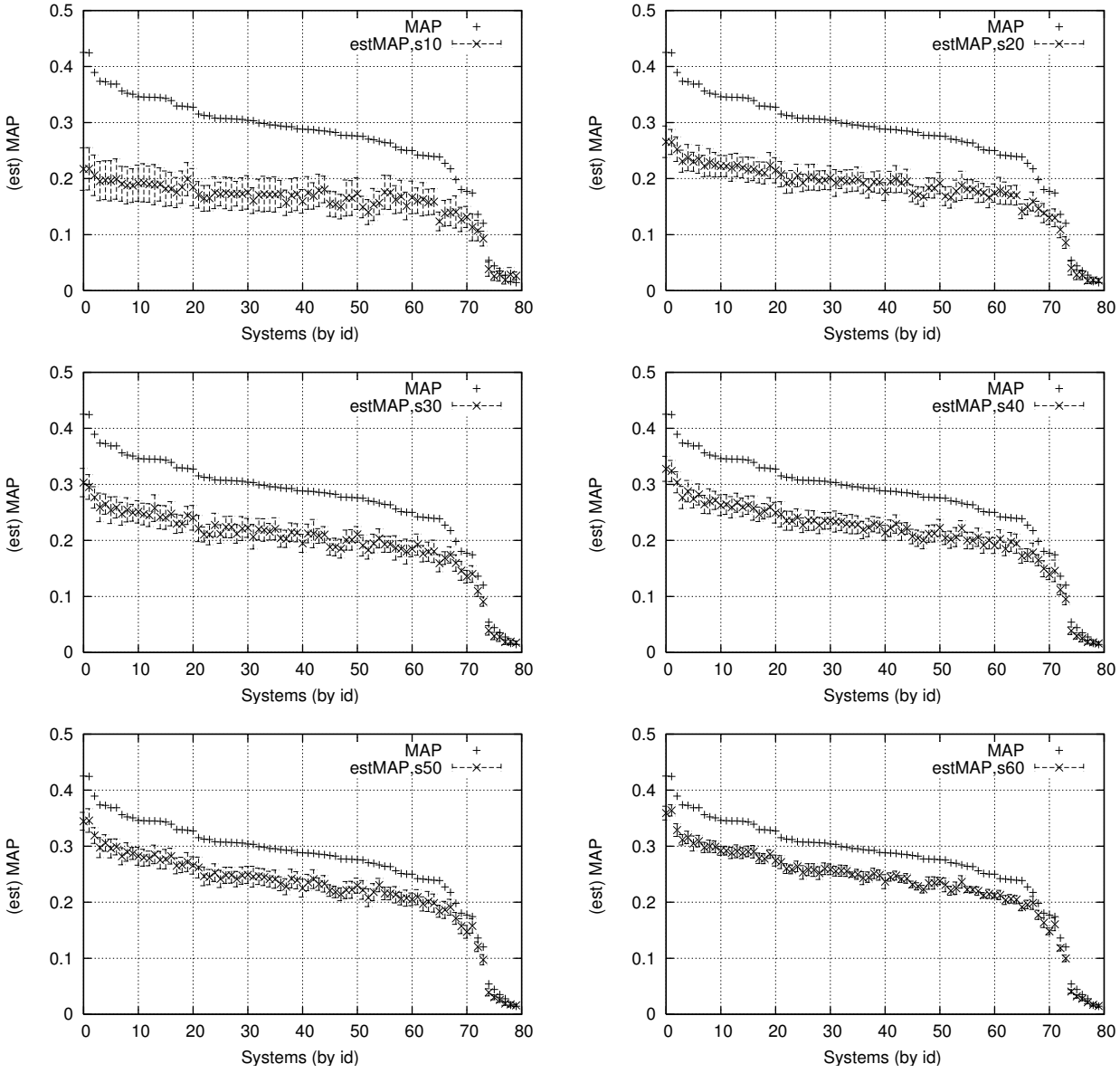


Figure A.1: TREC 2006 Terabyte: estMAP vs MAP values at different training levels (10 . . .60)

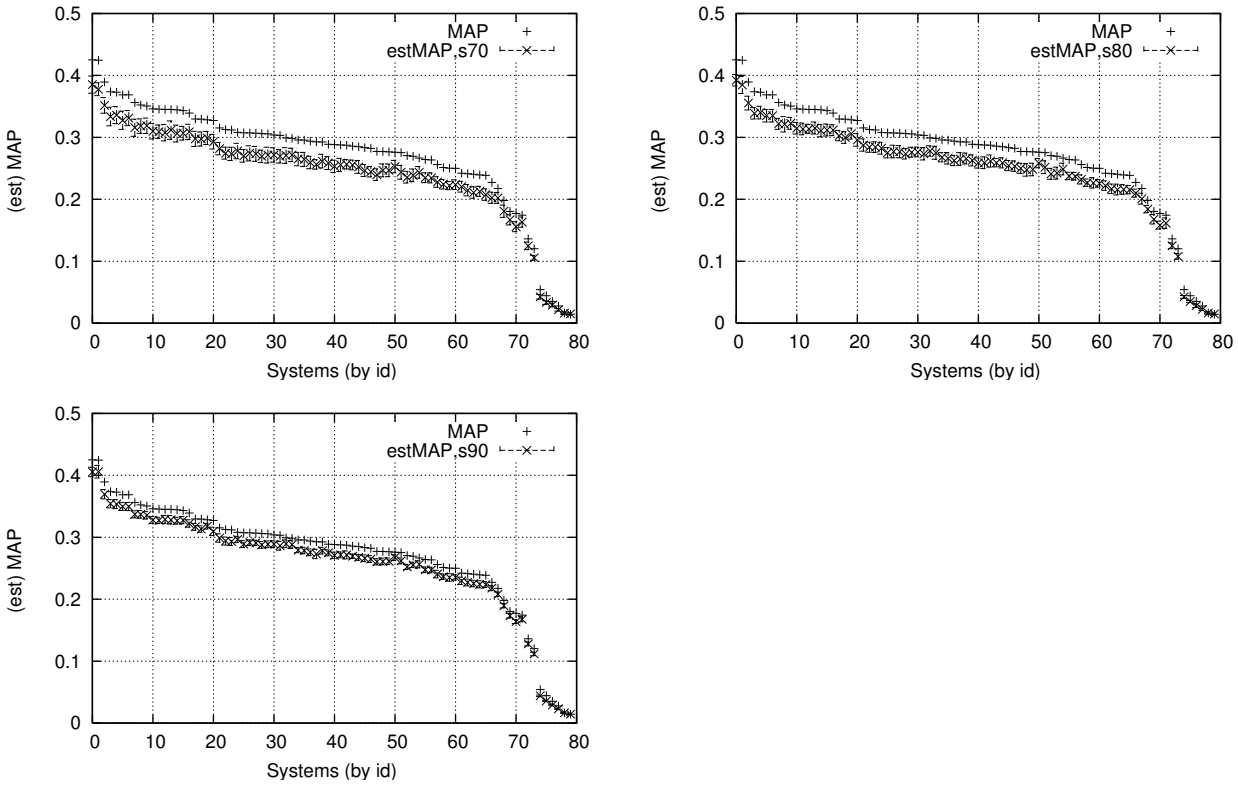


Figure A.2: TREC 2006 Terabyte : estMAP vs MAP values at different training samples (70, 80, 90)

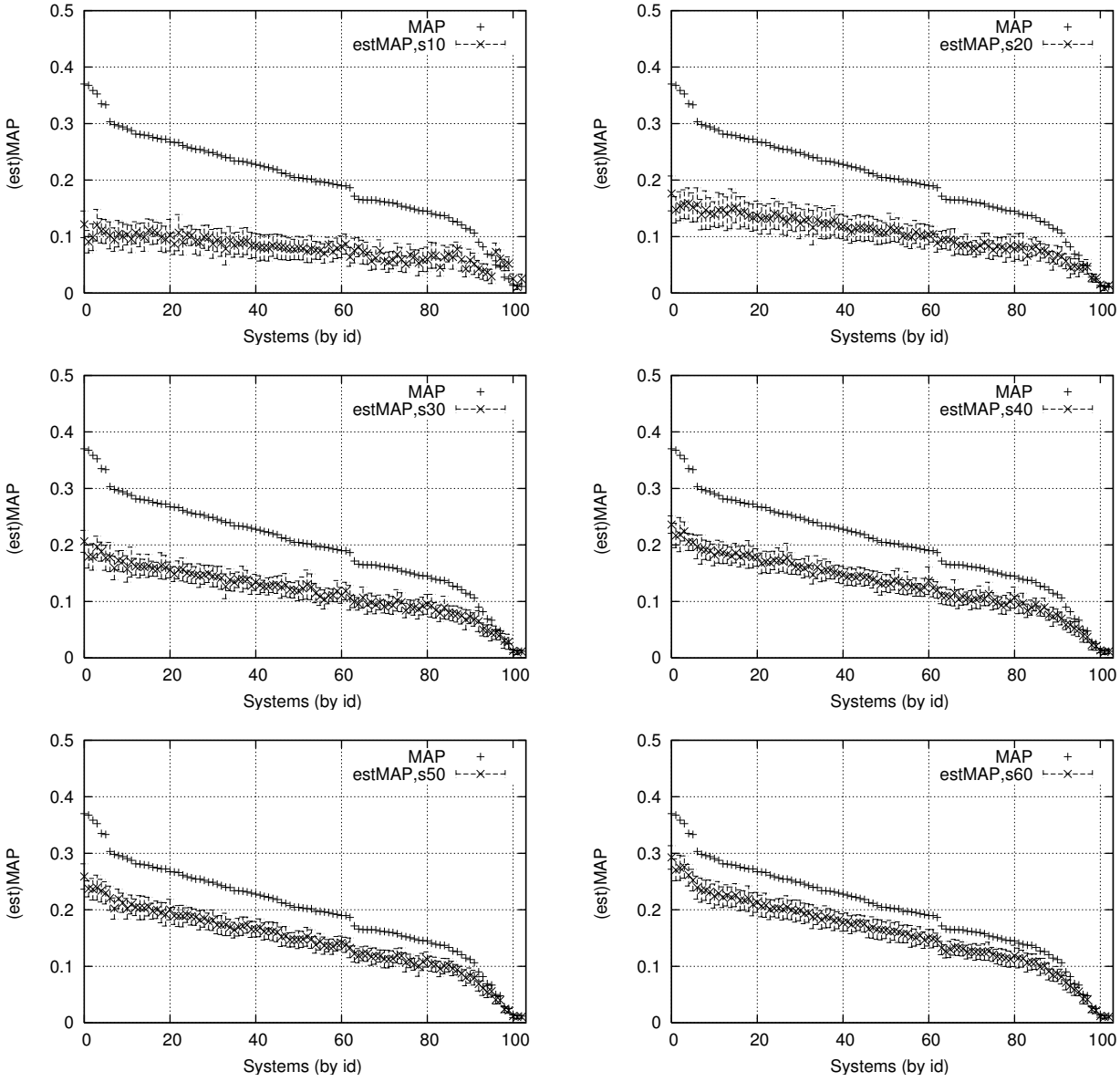


Figure A.3: TREC 7 ad-hoc: estMAP vs MAP values at different training sample size.

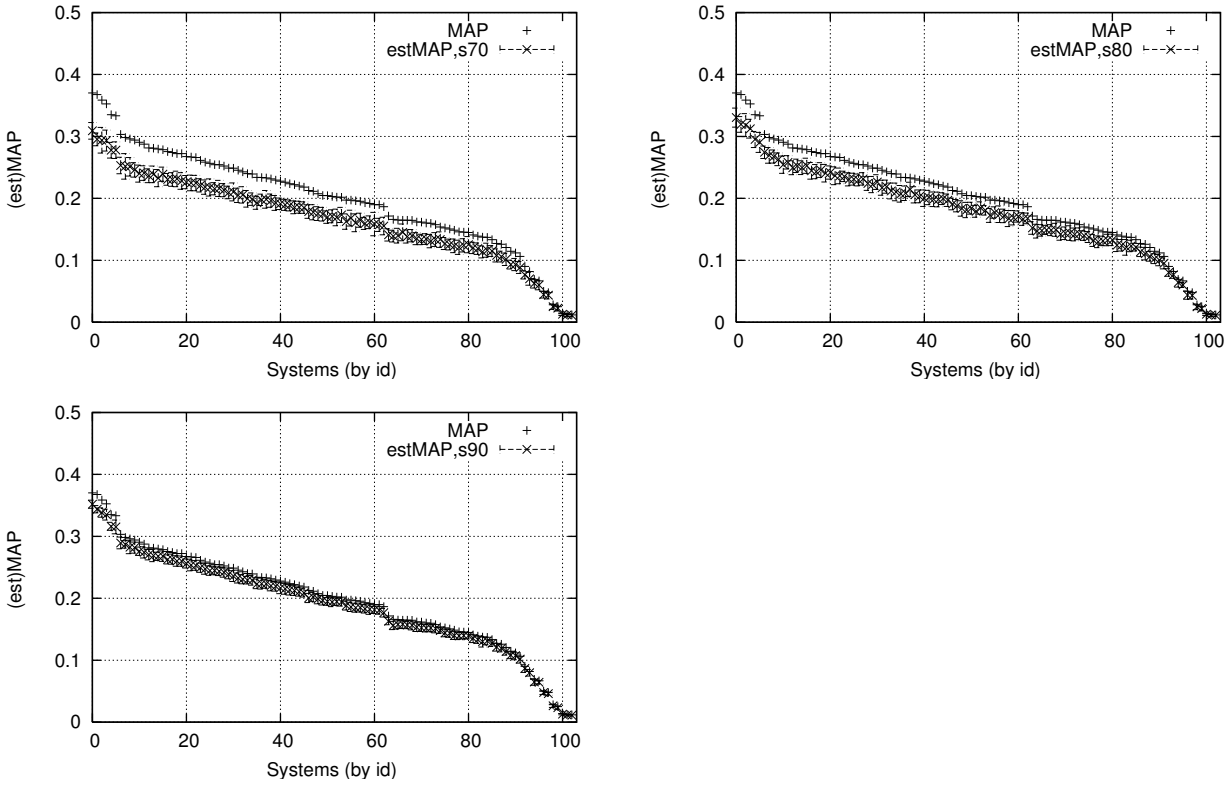


Figure A.4: TREC 7 ad-hoc: estMAP vs MAP values at different training sample size (cont'd).

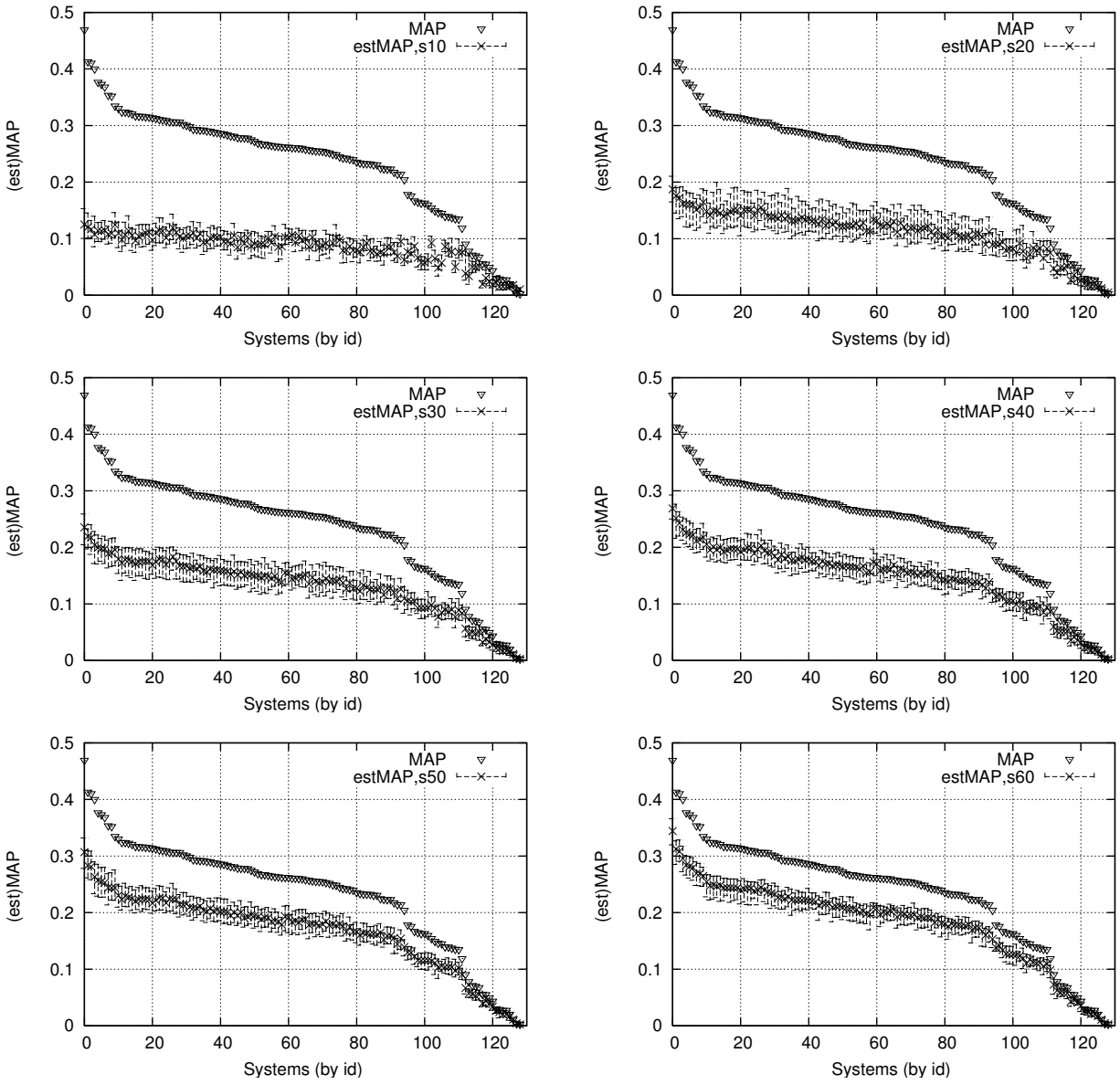


Figure A.5: TREC 8 ad-hoc: estMAP vs MAP values at different training sample size

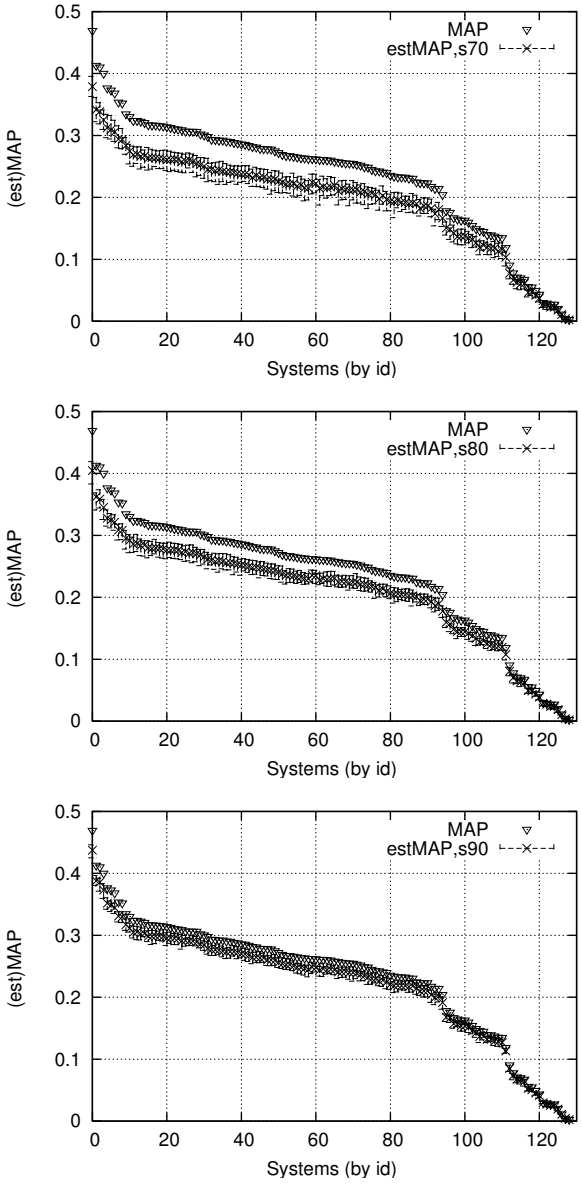


Figure A.6: TREC 8 ad-hoc: estMAP vs MAP values at different training sample size (Cont'd)

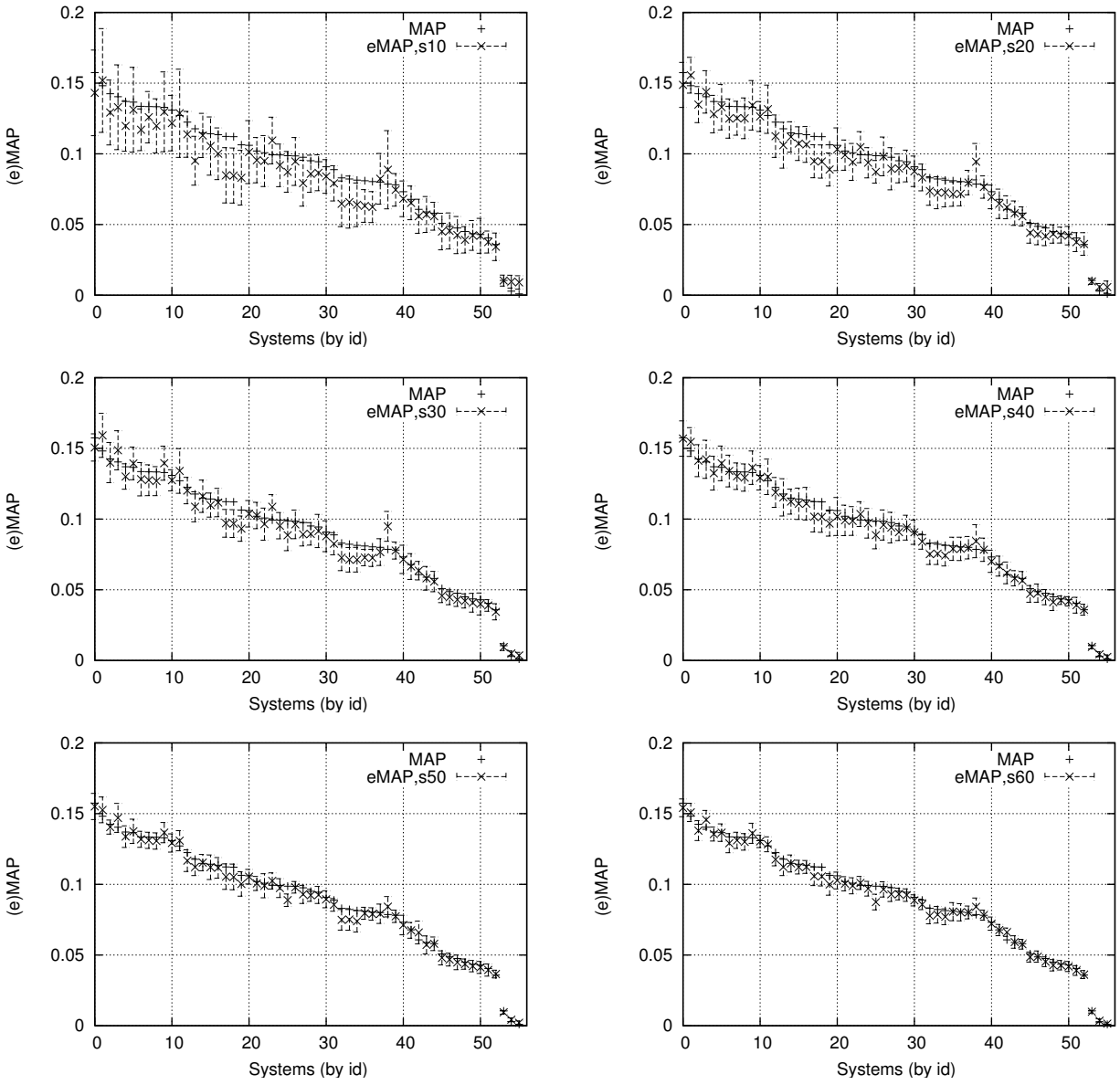


Figure A.7: TREC 2010 Web:estMAP vs MAP values at different training samples considered

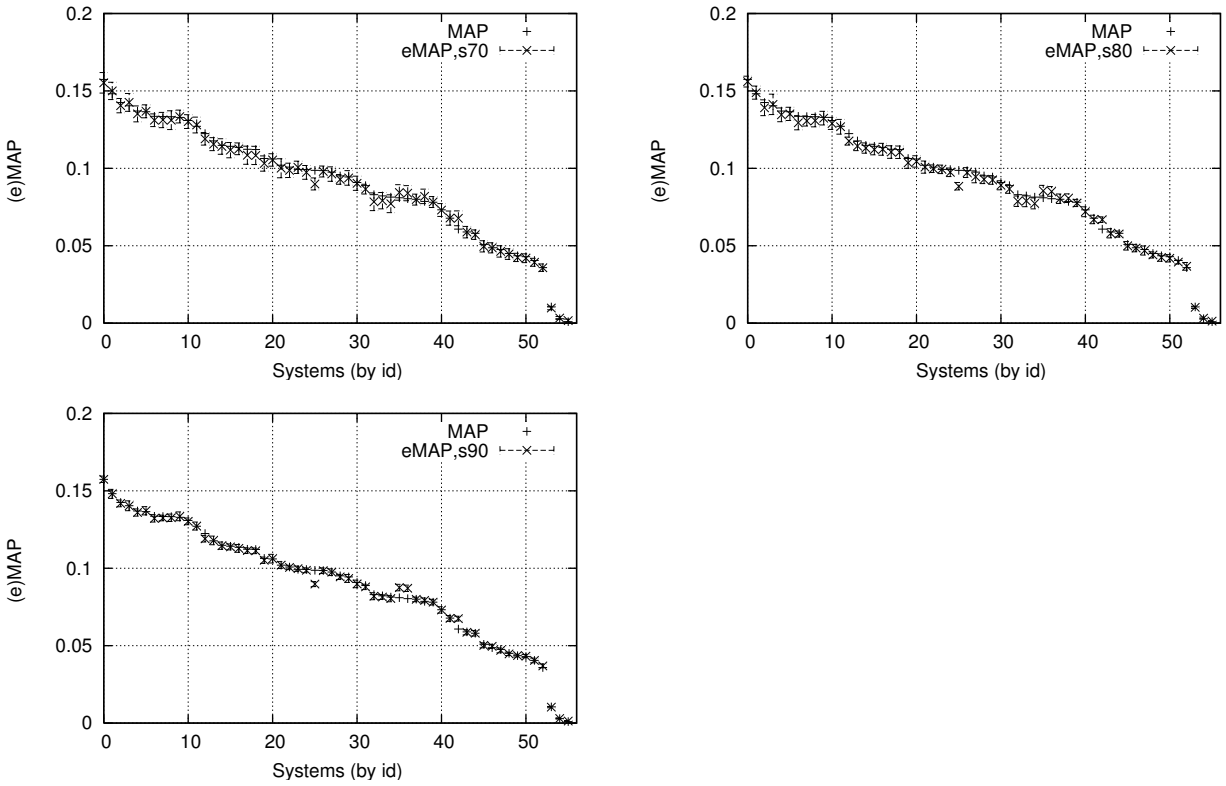


Figure A.8: TREC 2010 Web:estMAP vs MAP values at different training samples considered (Cont'd).

Bibliography

Rakesh Agarwal, Sreenivas Gullapudi, Alan Halverson, and Samuel Leong. Diversifying search results. In *WSDM*, pages 15–21, 2009. 62, 77, 80

Azzah Al-Maskari, Mark Sanderson, and Paul Clough. 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, 2007. 22

James Allan. HARD track overview in TREC 2004 - High Accuracy Retrieval from documents. In *proceedings of the Thirteenth Text REtrieval Conference*, Maryland, U.S.A, 2004. 28

Javier Artiles, Julio Gonzalo, and Satoshi Sekine. The SemEval-2007 WePS Evaluation: Establishing a benchmark for the web people search task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 64–69, Prague, Czech Republic, June 2007. Association for Computational Linguistics. 36

Azin Ashkan and Charles L. A. Clarke. On the informativeness of cascade and intent-aware effectiveness measures. In *WWW*, pages 407–416, 2011. 94

Yaniv Bernstein and Justin Zobel. Redundant documents and search effectiveness. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 736–743, 2005. 8

Bodo Billerbeck and Justin Zobel. Techniques for efficient query expansion. In *SPIRE*, pages 30—42, 2004. 4, 69

David C. Blair. Some thoughts on the reported results of TREC. *Information Processing & Management*, 38(3):445–451, 2002. ISSN 0306-4573. 18

Andrei Broder. A taxonomy of Web search. *SIGIR Forum*, 36(2):3–10, 2002. 1, 8, 12, 43

Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, New York, NY, USA, 2000. ACM. ISBN 1-58113-226-3. 19

Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA, 2004. ACM. 29, 86

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 89–96, New York, NY, USA, 2005. ACM. 5, 22

Stefan Büttcher, Charles L. A. Clarke, and Ian Soboroff. The TREC 2006 Terabyte Track. In *proceedings of the Ffteenth Text REtrieval Conferencei (TREC 2006)*, Maryland, U.S.A, 2006. 92

Stefan Büttcher, Charles L. A. Clarke, and Peter C. K. Yeung. Index pruning and result reranking: Effects on adhoc retrieval and named page finding. In *15th Text REtrieval Conference*, Gaithesburg, USA, 2006. 8, 36

Stefan Büttcher, Charles L. A. Clarke, Peter C. K. Yeung, and Ian Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 63–70, New York, NY, USA, 2007. ACM. 30, 86, 87, 91, 112, 116, 118

Jaime Carbonell and Jade Goldstein. 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, 1998. 31

David Carmel, Elad Yom-Tov, Adam Darlow, and Dan Pelleg. What makes a query difficult? In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 390–397, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. 27

Claudio Carpineto, Renato de Mori, Giovanni Romano, and Brigitte Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001. 4, 69

Ben Carterette and James Allan. Semi-automatic evaluation of retrieval systems using document similarities. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 873–876, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-803-9. 86, 87

Ben Carterette and Paul N. Bennett. A test collection of preference judgments. In P. Bennett, B. Carterette, O. Chapelle, and T. Joachims, editors, *In SIGIR 2008 Workshops: Beyond Binary Relevance: Preferences, Diversity, and Set-Level Judgments*, 2008. 24

Ben Carterette, Paul N. Bennett, David Maxwell Chickering, and Susan T. Dumais. Here or there: Preference judgments for relevance. In *30th European Conference on IR Research*, pages 16–27, Glassgow, Scotland, 2008. 23

Ben Carterette, Evangelos Kanoulas, Virgil Pavlu, and Hui Fang. Reusable test collections through experimental design. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 547–554. ACM, 2010. ISBN 978-1-4503-0153-4. 112

Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *CIKM '09: Proceedings of the 18th ACM conference on Information and knowledge management*, pages 621–630, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. 73

Harr Chen and David R. Karger. Less is more: Probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 429–436, 2006. 6, 33, 36, 69

Zhiwei Chen and Yunjie Xu. User-oriented relevance judgment: A conceptual model. In *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4*, page 101.2, Washington, DC, USA, 2005. IEEE Computer Society. 16, 17, 31

Charles L. A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and Diversity in Information Retrieval Evaluation. In *Proceedings of the 31st ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2008)*, Singapore, July 2008. 10, 77, 112, 116

Charles L. A. Clarke, Maheedhar Kolla, and Olga Vechtomova. An effectiveness measure for ambiguous and underspecified queries. In *ICTIR*, pages 188–199, 2009a. 10, 116

Charles L. A. Clarke, Nick Craswell, Ian Soboroff, and Azin Ashkan. A comparative analysis of cascade measures for novelty and diversity. In *WSDM*, pages 75–84, 2011. 80

Charles L.A. Clarke, Nick Craswell, and Ian Soboroff. Overview of the TREC 2009 Web Track. In *proceedings of the Eighteenth Text REtrieval Conference (TREC 2009)*, Maryland, U.S.A, 2009b. 75, 77

Charles L.A. Clarke, Nick Craswell, Ian Soboroff, and Gordon V. Cormack. Overview of the TREC 2010 Web Track. In *Proceedings of the Nineteenth Text REtrieval Conference (TREC 2010)*, Maryland, U.S.A, 2010. 92

Cyril W. Cleverdon. Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems, 1962. 14, 15, 16, 47

William S. Cooper. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, 19(1):30–41, 1968. 24, 26

William S. Cooper. A definition of relevance for information retrieval. *Information Storage and Retrieval*, 7(1):19 – 37, 1971. 16, 30, 48, 49, 50, 51

Gordon Cormack, Mark Smucker, and Charles Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval*, pages 1–25, 2011. 88, 89, 90

Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. Efficient construction of large test collections. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 282–289, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. 28

Nick Craswell and Martin Szummer. Random walks on the click graph. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 239–246, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. 76

Nick Craswell, David Hawking, Ross Wilkinson, and Mingfang Wu. Overview of the trec 2003 web track. In *proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, pages 78–92, Maryland, U.S.A, 2003. 20, 74

Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 87–94, New York, NY, USA, 2008. ACM. 72

Hoa Trang Dang, Jimmy Lin, and Diane Kelly. Overview of the TREC 2006 question answering track. In *Proceedings of the Fifteenth Text REtrieval Conference*, Gaithersburg, Maryland, 2006. 66

Tom Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27:861–874, June 2006. ISSN 0167-8655. 91, 93

William Goffman. On relevance as a measure. *Information Storage and Retrieval*, 2(3):201 – 203, 1964. 31

Donna Harman. Overview of the first TREC conference. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 36–47, New York, NY, USA, 1993. ACM. ISBN 0-89791-605-0. 5, 14, 19

Donna Harman. What we have learned, and not learned, from TREC. In *BCS IRSG 2000 PROCEEDINGS*, pages 2–20, 2000. 27

Donna Harman and Chris Buckley. Overview of the reliable information access workshop. *Information Retrieval*, 12(6):615–641, 2009. ISSN 1386-4564. 27, 34

David Hawking and Stephen Robertson. On collection size and retrieval effectiveness. *Inf. Retr.*, 6(1):99–105, 2003. ISSN 1386-4564. 19

David Hawking, Ellen Voorhees, Nick Craswell, and Peter Bailey. Overview of the TREC-8 Web Track. In *Proceedings of the Eighth Text REtrieval Conference (TREC 2000)*, Maryland, U.S.A, 2000. 8, 36

Scott B. Huffman and Michael Hochster. How well does result relevance predict session satisfaction? In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 567–574, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. 5

Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Process. Manage.*, 36(2):207–227, 2000. ISSN 0306-4573. 5, 44, 115

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002. 21, 26, 43, 53, 54, 68, 81, 82

Kalervo Järvelin, Susan L. Price, Lois M. L. Delcambre, and Marianne Lykke Nielsen. Discounted cumulated gain based evaluation of multiple-query IR sessions. In *ECIR'08: Proceedings of the IR research, 30th European conference on Advances in information retrieval*, pages 4–15, Berlin, Heidelberg, 2008. Springer-Verlag. 26

Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM. 5

Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 206–215, New York, NY, USA, 2004. ACM. 72

Robert Krovetz and W. Bruce Croft. Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems*, 10:115–141, 1992. 35

Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul Vitányi. The similarity metric. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '03, pages 863–872, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics. 70, 71

Jimmy Lin. Is Question Answering Better Than Information Retrieval? A Task-Based Evaluation Framework for Question Series. In *Proceedings of the 2007 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2007)*, pages 212–219, Rochester, New York, 2007. 40, 42

Jimmy Lin and Boris Katz. Building a reusable test collection for Question Answering. *Journal of the American Society for Information Science and Technology*, 57(7):851–861, 2006. 68, 69

Hua Liu, Ruihua Song, Jian yun Nie, and Ji rong Wen. Building a test collection for evaluating search result diversity: A preliminary study. In *In Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 31–32, Boston, MA, U.S.A, 2009. 117

Alistair Moffat and Justin Zobel. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems*, 27(1):1–27, 2008. 25, 43, 53, 57, 58, 59

George Nemhauser, Laurence A. Wolsey, and Fisher M.L. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978. 56

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. The Pyramid Method: Incorporating human content selection variation in summarization evaluation . *ACM Trans. Speech Lang. Process.*, 4(2):4, 2007. ISSN 1550-4875. 37, 42

Paul Over. TREC-6 Interactive Track Report. In *Proceedings of the 6th Text Retrieval Conference*, pages 57–64, Maryland, U.S.A, 1997. 32

Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. 3

Filip Radlinski and Susan Dumais. Improving personalized web search using result diversification. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 691–692, New York, NY, USA, 2006. ACM. 31

Filip Radlinski, Martin Szummer, and Nick Craswell. Metrics for assessing sets of subtopics. In *SIGIR '10: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 853–854, New York, NY, USA, 2010a. ACM. ISBN 978-1-4503-0153-4. 83

Filip Radlinski, Martin Szummer, and Nick Craswell. Inferring query intent from reformulations and clicks. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1171–1172, New York, NY, USA, 2010b. ACM. 76

Stephen Robertson. The probabilistic character of relevance. *Information Processing & Management*, 13(4):(247 – 251), 1977a. 20

Stephen Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33: 294–304, 1977b. 48, 49

Stephen Robertson. On GMAP: and other transformations. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 78–83, New York, NY, USA, 2006. ACM. ISBN 1-59593-433-2. 28

Stephen Robertson. A new interpretation of average precision. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 689–690, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. 26, 98

Stephen Robertson and Nicholas Belkin. Ranking in principle. *Journal of Documentation*, 34(2): 93–100, 1978. 20

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. Simple BM25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 42–49, New York, NY, USA, 2004. ACM. 3

J.J. Rocchio. Relevance feedback in information retrieval. In *The Smart Retrieval System — Experiments in Automatic Document Processing*, pages 313–323, 1971. 4

Tetsuya Sakai. Ranking the NTCIR systems based on multigrade relevance. In *AIRS*, pages 251–262, 2004. 22, 86

Tetsuya Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 525–532, 2006. 80

Tetsuya Sakai. On the reliability of information retrieval metrics based on graded relevance. *Information Process & Management*, 43(2):531–548, 2007. 81, 82

Tetsuya Sakai, Nick Craswell, Ruihua Song, Stephen Robertson, Z. Dou, and C. Y. Lin. Simple evaluation metrics for diversified search results. In *3rd International Workshop on Evaluating Information Access(EVIA), 2010.*, 2010. 81, 83

Gerald Salton. *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971. 2

Mark Sanderson. Word sense disambiguation and information retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 142–151, New York, NY, USA, 1994. Springer-Verlag New York, Inc. ISBN 0-387-19889-X. 36

Mark Sanderson, Falk Scholer, and Andrew Turpin. Relatively relevant: Assessor shift in document judgements. In *Proceedings of the Australasian Document Computing Symposium*, pages 60–67, 2010. 118

Ian Soboroff and Donna Harman. Novelty detection: the TREC experience. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 105–112, Morristown, NJ, USA, 2005. Association for Computational Linguistics. 8, 31, 32

Ian Soboroff, Charles Nicholas, and Patrick Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 66–73, New York, NY, USA, 2001. ACM. 88

Ruihua Song, Zhenxiao Luo, Jian-Yun Nie, Yong Yu, and Hsiao-Wuen Hon. Identification of ambiguous queries in web search. *Information Processing & Management*, 45:216–229, March 2009. ISSN 0306-4573. 5

Ruihua Song, Dongjie Qi, Hua Liu, Tetsuya Sakai, Jian-Yun Nie, Hsiao-Wuen Hon, and Yong Yu. Constructing a test collection with multi-intent queries. In *EVIA 2010 Workshop*, pages 207–227, Tokyo, Japan, 2010. 117

Eero Sormunen. Liberal relevance criteria of TREC -: counting on negligible documents? In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 324–330, New York, NY, USA, 2002. ACM. 20, 48

Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Document retrieval systems*, pages 132–142, 1988. 2, 14

Karen Spärck Jones and C. Van Rijsbergen. Report on the need for and provision of an "ideal" information retrieval test collection. Technical report, British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge,England, 1975. 14, 16, 86

Karen Spärck Jones, Steven Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing & Management*, 36(6):779–808, 2000. ISSN 0306-4573. 3, 21

Karen Spärck Jones, Stephen E. Robertson, and Mark Sanderson. Ambiguous requests: Implications for retrieval tests. *SIGIR Forum*, 41(2):8–17, 2007. 8, 36

M.C. Tang and Y. Sun. Evaluation of web-based search engines using user-effort measures. *Library and Information Science Research Electronic Journal*, 13(2), 2003. 24

Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Beyond the commons: Investigating the value of personalizing web search. In *Proceedings of the Workshop on New Technologies for Personalized Information Access*, pages 84–92, 2005. 32

Stephen Tomlinson and Bruce Hedin. Measuring effectiveness in the trec legal track. In Mihai Lupu, Katja Mayer, John Tait, Anthony J. Trippe, and W. Bruce Croft, editors, *Current Challenges in Patent Information Retrieval*, volume 29 of *The Information Retrieval Series*, pages 167–180. Springer Berlin Heidelberg, 2011. 102

Ellen M. Voorhees. Using WordNet to disambiguate word senses for text retrieval. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–180, New York, NY, USA, 1993. ACM. ISBN 0-89791-605-0. 35, 36

Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 315–323, 1998. 48

Ellen M. Voorhees. Overview of the TREC 2004 Robust Track. In *proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, Maryland, U.S.A, 2004a. 27

Ellen M. Voorhees. Overview of the TREC 2004 Question Answering Track. In *Proceedings of the Thirteenth Text REtrieval Conference*, Gaithersburg, Maryland, 2004b. 39

Ellen M. Voorhees and Donna Harman. Overview of the seventh text retrieval conference TREC-7. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, pages 1–24, Maryland, U.S.A, 1998. 92

Ellen M. Voorhees and Donna Harman. Overview of the eighth text retrieval conference TREC-8. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, 2000. 92

William Webber, Alistair Moffat, Justin Zobel, and Tetsuya Sakai. Precision-at-ten considered redundant. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 695–696, New York, NY, USA, 2008. ACM. 19

Yunjie Xu and Hainan Yin. Novelty and topicality in interactive information retrieval. *Journal of the American Society for Information Science and Technology*, 59(2):201–215, 2008. 51

Emine Yilmaz and Javed A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 102–111, New York, NY, USA, 2006. ACM. ISBN 1-59593-433-2. 29, 86

ChengXiang Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–17, 2003. 32, 77, 82

Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. 25, 28, 86, 102