Exploiting User Signals and Stochastic Models to Improve Information Retrieval Systems and Evaluation

Maria Maistro

Department of Information Engineering

University of Padua

This dissertation is submitted for the degree of

Doctor of Philosophy

School on Information Engineering
Curriculum: Information and Communication Science and Technologies

Supervisor: Prof. Nicola Ferro

October 2017

Abstract

The leitmotiv throughout this thesis is represented by IR evaluation. We discuss different issues related to effectiveness measures and novel solutions that we propose to address these challenges. We start by providing a formal definition of utility-oriented measurement of retrieval effectiveness, based on the representational theory of measurement. The proposed theoretical framework contributes to a better understanding of the problem complexities, separating those due to the inherent problems in comparing systems, from those due to the expected numerical properties of measures. We then propose AWARE, a probabilistic framework for dealing with the noise and inconsistencies introduced when relevance labels are gathered with multiple crowd assessors. By modeling relevance judgements and crowd assessors as sources of uncertainty, we directly combine the performance measures computed on the ground-truth generated by each crowd assessor, instead of adopting a classification technique to merge the labels at pool level. Finally, we investigate evaluation measures able to account for user signals. We propose a new user model based on Markov chains, that allows the user to scan the result list with many degrees of freedom. We exploit this Markovian model in order to inject user models into precision, defining a new family of evaluation measures, and we embed this model as objective function of an LtR algorithm to improve system performances.

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

$a_k(t)$	Assessor Accuracy
D	Set of Documents
$\mathscr{D}$	Universe Set of Retrieved Documents
$D(n)$	Set of Retrieved Documents
$\mathbb{E}$	Expectation
GT	Ground-Truth
J	Set of Relevance Judgements
$\mathrm{m}(\cdot)$	Performance Measure
M_k	Assessor Measure
$(X_n)_{n\geq 0}$	Markov Chain
I	State Space
n	Length of the Run
$\mathbb{P}$	Probability
$P=(p_{i,j}\colon\, i,j\in I)$	Transition Matrix
r_t	Run
$\mathscr{R}$	Universe Set of Judged Documents
REL	Set of Relevance Degrees

$r_t[j]$	j-th Element of the Run
$\hat{r}_t$	Judged Run
$\hat{r}_t[j]$	j-th Element of the Judged Run
S	Set of Systems
T	Set of Topics

Greek Symbols

Λ	Set of Assessors
λ	Initial Distribution
$\mu(\cdot)$	Scoring Function

Acronyms / Abbreviations

AP	Average Precision
$AWARE$	Assessor-driven Weighted Averages for Retrieval Evaluation
$bpref$	Binary Preference
CG	Cumulated Gain
CTR	Click Through Rate
DCG	Discounted Cumulated Gain
EM	Expectation Maximization
ERR	Expected Reciprocal Rank
ESL	Expected Search Length
$i.i.d.$	independent and identically distributed
IR	Information Retrieval
IRS	Information Retrieval System
LtR	Learning to Rank
ML	Machine Learning

MP	Markov Precision
MV	Majority Vote
nCG	normalized Cumulated Gain
nDCG	normalized Discounted Cumulated Gain
nMCG	Normalized Markov Cumulated Gain
RBP	Rank-Biased Precision
SERP	Search Engine Result Page
SMART	System for the Mechanical Analysis and Retrieval of Text
TREC	Text REtrieval Conference

Chapter 1

Introduction

> But do you know that, although I have kept the diary [...] for months past, it never once struck me how I was going to find any particular part of it in case I wanted to look it up?
>
> Stoker [1897]

Information has always been a valuable and central resource, necessary to keep and transmit knowledge, and people have realized the importance of storing and maintaining information for thousands of years. At the same time, just storing information without providing any tool to search and find relevant items, makes the information itself pointless.

The origin of Information Retrieval (IR) is dated in 1950s, and it initially evolved in the context of libraries. Indeed, after the second World War there was an increase in the number of scientific publications and many researchers started to tackle the problem of manually searching a large collection of documents [Harman, 2011]. At that time it became evident the necessity of an automatic system whose "task [...] is to retrieve documents [...] with information content that is relevant to a user's information need" [Spärck Jones, 1997]. Therefore, an Information Retrieval System (IRS) takes as input a query, formulated by a user from her information need, and returns a run, which is a set or a list of documents relevant with respect to that query.

With the development of IR systems it became necessary to design a framework to evaluate and compare different retrieval strategies. Indeed, progress and innovation are driven by experiments, but experimentation is useless without an objective evaluation measure that allow researchers to detect the improvements and identify the successful strategies.

In IR, a fundamental question faces the definition of evaluation itself. Evaluation means to "ascertain the value or amount of something or to appraise it" [Kiewitt, 1979], however in the context of IR, it is not completely clear what is the quantity that describes the quality of a system. In general the purpose of evaluation is to determine whether a system is successful or not in performing a task, but what does it mean for an IRS to be successful? The focus is on two main quantities: efficiency and effectiveness [van Rijsbergen, 1979]. Efficiency can be expressed with physical measures, as for example the response time of the system to a query, the memory consumption and the utilization of computing resources. Effectiveness deals with something intangible and vague represented by user satisfaction.

Due to the experimental nature of IR, accurately interpreting the result of a system in terms of user satisfaction is fundamental to push the research in the correct direction. Therefore, measuring systems effectiveness continues to be an active area of research and discussion in the scientific community. It is also the case of this thesis, whose leitmotiv is an investigation of effectiveness measures exploited in different aspects of IR.

Our first aim was to provide a formal and theoretical definition of effectiveness measure. In literature, effectiveness is often considered proportional to the amount of relevance returned by a system, basically expressed as a function of the number of relevant documents retrieved, their positions in the ranking, and the total number of relevant documents in the collection. Several evaluation measures have been proposed since the beginning of IR, starting from simple ratios between relevant and retrieved documents to more complex functions discounting each rank positions and accounting for plausible user models [Sanderson, 2010].

However, even if much research was conducted, a prior question is still just partially fulfilled: what is a general definition of IR evaluation measure? This is a primary concern, since "a methodology for evaluation ultimately invokes a theory of evaluation" [Spärck Jones, 1997]. Chapter 3 encompasses this challenge and gives a formal definition of utility-oriented measurement of retrieval effectiveness [Ferrante et al., 2015], based on the representational theory of measurement [Krantz et al., 1971].

The main issue presented in Chapter 3 is the lack of a shared agreement on a possible total ordering among the outputs of different systems. In simpler words, given two systems and their corresponding output runs, it is not always clear which run should be selected as the best one. Therefore, we need to first distinguish between the problems arising from the absence of a total ordering among IR systems, and those arising from the operation of measuring. The deeper message conveyed by Chapter 3 is that when two runs are outside the partial ordering, i.e. they are not comparable, the measure is in charge to determine the best run, therefore different measures can produce different orderings. This might justify the complexity of this task and the reason why so many evaluation measures have been proposed so far.

A further complexity of evaluation in IR is represented by relevance. Evaluation measures are tightly related to the relevance of the results returned by a system, since relevance is also associated to the satisfaction of the user. Unfortunately relevance is subjective, the information need is unique, and the user is the only person able to provide a fair and reliable judgement of a document in terms of relevance. Since it is not possible to directly ask to the user to provide relevance judgements when she performs a search, IR evaluation relies on test collections, with documents that are judged for relevance by assessors, which may have different levels of training and expertise.

TREC was initially building test collections with relevance judgements provided by retired analysts, trained and qualified to perform this task [Harman, 2011]. However, the need for more and more large test collections called for the exploration of alternative ways to collect relevance assessments, as the use of crowdsourcing platforms, which allow to gather a larger number of relevance labels at lower cost [Alonso and Mizzaro, 2009; Alonso et al., 2008]. The drawbacks are less control on the quality of the assessments and the introduction of noise in the data, indeed a crowd assessor "[...] is not a device that reliably reports a gold standard judgment of relevance of a document to a query" [Manning et al., 2008].

The introduction of some noise in the test collection affects the evaluation of the systems, therefore the same query-document pair is assigned to more than one crowd assessor to prevent potential errors caused by wrong labels. This makes necessary to merge possibly discording labels generated by different workers. Most of the state of the art approaches work directly with the relevance labels, we called them downstream approaches. An example is Majority Vote (MV), which considers each assessor as a voter and assigns to a document the relevance grade which receives the majority of the votes.

In Chapter 4 we propose our upstream approach called Assessor-driven Weighted Averages for Retrieval Evaluation (AWARE) [Ferrante et al., 2017]. AWARE is defined as an upstream approach because it directly combines the scores of the evaluation measures computed from the relevance labels of each assessor, instead of merging the labels and then computing the measures. The focus is then shifted from the documents and the labels to the evaluation measures. This allows to account for the error introduced by incorrect labels and to develop a framework which estimates performance measures in a way more robust to crowd assessors.

Up to now we provide a formal definition of utility-oriented measurement of retrieval effectiveness and we developed an approach to estimate performance measures when there is some noise due to crowd assessors variability. Thus the effectiveness of a system is measured just in terms of the amount of relevance retrieved, however, what about the user perspective? Is it possible to account more for the user-system interactions? These questions suggest that the user and her interactions with the system should be included in the evaluation process [Robertson and Hancock-Beaulieu, 1992].

When it comes to user interactions, the straightforward solution is to exploit click log data recored from Web search engine. However, "estimating user preferences in real Web search settings is a challenging problem, since real user interactions tend to be more noisy than commonly assumed in the controlled settings" [Agichtein et al., 2006a]. A model of user behaviour is necessary in order to cope with the noise of user signals and extract the valuable information [Joachims et al., 2005].

In Chapter 5 we present a novel user model defined on top of a Markov process. Differently from many traditional models, which assume a user linearly scanning the result list, this model allows the user to follow complex paths when browsing the run, as moving backward and forward in the list, skipping some documents or considering already visited documents. Based on this model, we defined two different evaluation measures Markov Precision (MP) and Normalized Markov Cumulated Gain (nMCG), both involving the user in the evaluation process.

MP [Ferrante et al., 2014a] injects the user model into precision with the invariant distribution of a Markov chain, which is the probability of finding the user in a given rank position after a long time. MP stems from the idea that if a user does not see a document, even if the document is relevant, the evaluation measure should account less for it, while it should account more for documents that have been visited. Therefore, we defined MP as a weighted average of precision, where the weights are the invariant distribution computed on each rank position. The measure can be defined in a batch setting, with predefined reasonable transition models, or in an online setting, by calibrating the model directly with click log data. In this latter case, the model can account also for the time dimension, i.e. the time spent by each user in reading a document.

By exploiting the same model, we defined nMCG-MART [Ferro et al., 2017] a measure calibrated with real word click log data. nMCG-MART is an evaluation measure that accounts for the user dynamic on different types of queries. We observed that the invariant distribution depends on the query type, meaning that the amount of relevance, retrieved as a response to the query, affects the user dynamic. Indeed, with query retrieving just a few relevant documents, the users tend to focus at the beginning of the list, while for queries retrieving more relevant documents or no relevant documents, the users tend to explore the whole list of results.

nMCG-MART is exploited as objective function in a state of the art Learning to Rank (LtR) algorithm. LtR applies Machine Learning (ML) algorithms to the specific task of ordering documents, thus given a query they learn how to produce a ranked list of results [Liu, 2011]. By using nMCG-MART as objective function, we embed the user dynamic in the learning process and we push the algorithm to rank the results by considering the user experience.

To conclude, the reader might note how evaluation measures are pervasive in IR. Evaluation measures can not be relegated merely to the task of evaluating the performances of a system. They are central to LtR algorithms, which basically learn to rank the documents by optimizing an evaluation measure. This implies that the measures have to be aligned with the user preferences and behaviour and have to account not only for the number of relevant results, but also for user signals. Moreover, the evaluation framework depends on the notion of relevance, which is subjective and difficult to grasp. Therefore, it is essential for evaluation measures to be able to cope with the errors deriving from noise in the relevance assessments. All these reasons justify the importance of a formal framework encompassing the problem of evaluation and explaining the behaviour and properties of evaluation measures.

1.1 Organization of the Thesis

The thesis is organized as follows. Chapter 2 illustrates some core definitions and concepts of IR, a short historical summary and an overview of evaluation measures. The original contributions of the thesis start from Chapter 3, which presents a formal framework for IR evaluation measures; Chapter 4 describes a novel approach to robustly estimate evaluation measures when dealing with noise in the relevance labels; Chapter 5 discusses a Markovian model for the user behaviour and applies it to define a new family of evaluation measures and to improve LtR. Finally, Chapter 6 presents some general conclusions and future directions.

Introduction to Information Retrieval

In this chapter we present an overview of IR, with a special focus on evaluation. The chapter starts with the definition of IR and the description of an IRS, outlining the main challenges in this research area. Some of the most popular ranking models are illustrated, starting from the Boolean model and reaching the modern LtR framework.

Successively we present the historical evolution of evaluation in IR, the Cranfield paradigm, which represents the starting point of IR evaluation, the proposal of the ideal test collection and its implementation within the TREC conference. We explain the methodology adopted to build modern test collections, the pool creation and its limitations, the advantages and disadvantages of collecting relevance labels with crowdsourcing assessors, and the exploitation of click log data as implicit feedback.

Finally, we discuss the problem of defining proper evaluation measures and how it was approached in the last 60 years. We firstly present set based evaluation measures, as precision and recall, and we then proceed to discuss rank based evaluation measures and their complexities.

Towards a Formalism for IR Evaluation Measures

In this chapter we present a formal framework to define and study the properties of utility-oriented measurements of retrieval effectiveness [Ferrante et al., 2015], like Average Precision (AP), Rank-Biased Precision (RBP), Expected Reciprocal Rank (ERR) and many other popular IR evaluation measures. The proposed framework is laid in the wake of the representational theory of measurement, which provides the foundations of the modern theory of measurement in both physical and social sciences, thus contributing to explicitly link IR evaluation to a broader context. The proposed framework is minimal, in the sense that it relies on just one axiom, from which other properties are derived. Finally, it contributes to a better understanding and a clear separation of what issues are due to the inherent problems in comparing systems in terms of retrieval effectiveness and what others are due to the expected numerical properties of a measurement.

AWARE: Merging Relevance Judgements via Evaluation Measures

In this chapter we propose the Assessor-driven Weighted Averages for Retrieval Evaluation (AWARE) probabilistic framework [Ferrante et al., 2017], a novel methodology for dealing with multiple crowd assessors, who may be contradictory and/or noisy. By modeling relevance judgements and crowd assessors as sources of uncertainty, AWARE takes the expectation of a generic performance measure, like AP, composed with these random variables. In this way, it approaches the problem of aggregating different crowd assessors from a new perspective, i.e. directly combining the performance measures computed on the ground-truth generated by the crowd assessors instead of adopting some classification technique to merge the labels produced by them. We propose several unsupervised estimators that instantiate the AWARE framework and we compare them with state-of-the-art approaches, i.e. Majority Vote (MV) and Expectation Maximization (EM), on TREC collections. We found that AWARE approaches improve in terms of their capability of correctly ranking systems and predicting their actual performance scores.

User Model Based on Markov Chain and Applications

In this chapter we propose a new user model based on Markov chains. Each document in the ranked result list represents a state of the Markov chain and the transition matrix describes the users' paths in exploring the list of results. With the Markovian model we can describe the user behaviour with many degrees of freedom, i.e. the user can move forward and backward on the ranked list, skip documents and visit already visited documents. We exploit this model to define a new family of evaluation measures, called Markov Precision (MP), and we embed the model in a LtR algorithm.

MP [Ferrante et al., 2014a] exploits continuous-time and discrete-time Markov chains in order to inject user models into precision. Continuous-time MP behaves like time-calibrated measures, bringing the time spent by the user into the evaluation of a system; discrete-time MP behaves like traditional evaluation measures. We conduct a thorough experimental evaluation of MP on standard TREC collections in order to show that MP is as reliable as other measures and we provide an example of calibration of its time parameters based on click logs from Yandex.

Furthermore, with the same Markovian model we define the user dynamic and we calibrate it on a click log dataset containing real world user interactions. We explore the possibility of integrating the user dynamic directly into the LtR algorithms. Specifically, we propose nMCG-MART [Ferro et al., 2017], a new version of LambdaMart, a state-of-the-art LtR algorithm, where we exploit a new discount loss function calibrated on the proposed Markovian model of user dynamic. We evaluate the performance of the proposed approach on publicly available LtR datasets, finding that the improvements measured over the standard algorithm are statistically significant.

Chapter 2

Introduction to Information Retrieval

> The task of an IR system is to retrieve documents [...] with information content that is relevant to a user's information need.
>
> Spärck Jones [1997]

In 1950, the term Information Retrieval (IR) firstly appeared in [Mooers, 1950] and, one year later, the same author gave one of the earliest definition of Information Retrieval (IR) [Mooers, 1951]:

> The goal of a machine method of information retrieval is purely and simply that of being able to find and to recover at will information stored in a collection of documents. [...] It is oriented completely towards actual use of the information, and to the convenience of the user.

Early research in the subject was primarily conducted by librarians to carry out bibliographic searches. To search document collections, they were relying on manual tools such as the card catalogue and universal classification schemes for books or journal articles [Spärck Jones, 1997].

However, the costs of manual retrieval strategies started to grow after the Second World War since the volume of journal publications was increasing at an exceptional rate [Wilson, 1952]. This ever increasing volume of scientific literature calls for the design and development of automated search approaches.

The first rigorous experiments in IR were carried out by Cleverdon between 1950s and 1960s with the goal of defining a formal methodology to evaluate retrieval strategies. The experiments were performed manually and requested a lot of effort, in terms of time and human involvement. Nevertheless, the outcome of Cleverdon's experiments, called the Cranfield paradigm, formed the foundation of IR evaluation and is still considered as a standard.

The experimental methodology proposed by Cleverdon encouraged the exploration of new automatic retrieval strategies. For this purpose Salton started the System for the Mechanical Analysis and Retrieval of Text (SMART) project in 1961 [Salton, 1971]. The SMART system exploited the paradigm proposed by Cranfield to evaluate fully automated research strategies. Moreover, many of the ideas that are currently well established in the structure of Web search engines have their origins there, as for example the usage of purely automated approaches both for retrieval and evaluation, the scoring function to estimate the probability of a document to be relevant and the consequent ranking of documents instead of a simple set based retrieval.

It is nevertheless important to see that, despite all the advancements in developing automatic IRS, there were still some issues concerning the evaluation of IRS. The Cranfield paradigm assumes that each document in the test collection has to be manually judged to determine whether it is relevant or not to a given topic. Therefore, due to the cost of completely judging a collection of documents, it was not feasible to develop large test collections for IRS evaluation.

A solution came with the ideal test collection proposal [Spärck Jones and Van Rijsbergen, 1975], which was later implemented by TREC. The first TREC was organized in 1992 by Harman [Harman, 1992b] and it is still continuing to the present, founded by the National Institute of Standards and Technology (NIST), a US government agency. Its goal was to build a realistically-sized test collection and to improve IR research through the development of a common evaluation framework, performed on shared experimental data. The TREC initiative gained an extraordinary success, and it allowed researchers to validate their approaches, while also inspiring new work within this framework.

Especially with the rise of the World Wide Web, the amount of documents to be searched grew at an alarming rate, the ranking functions started to become more and more complex with many parameters to tune and variables to control. This called for new techniques able to cope with this huge volume of data, therefore, new approaches brought from other communities, particularly from the ML community, started to be applied to improve IRS [Fuhr, 1989]. These approaches were based on learning from examples, given a huge amount of queries and documents they aim at defining those features that are determining to infer the relevance of a document. Nowadays, these methods are pervasive and widely used in IR, where they are known as LtR algorithms [Liu, 2011].

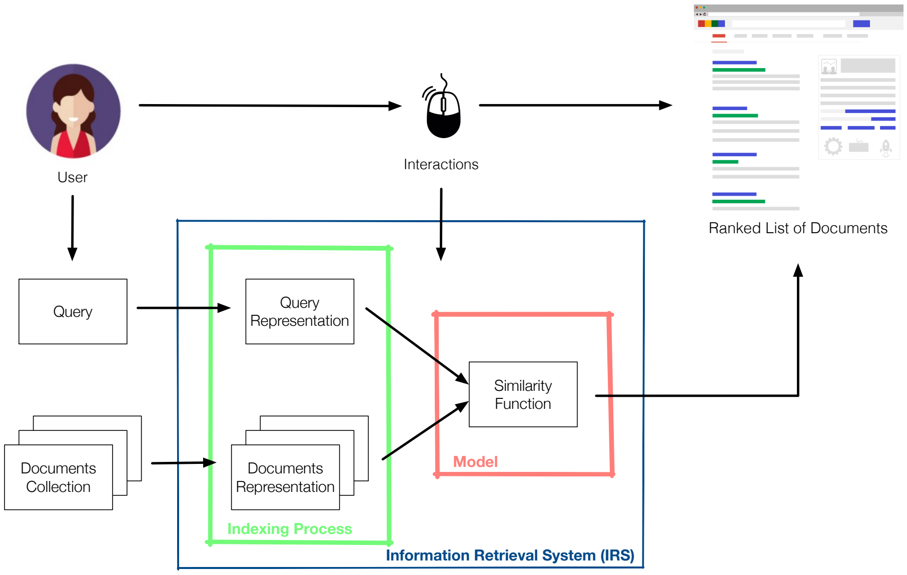


Fig. 2.1 Main components of an IRS and user interactions.

Furthermore, due to the success of Web search engines and their widespread use, query log datasets started to record not only queries and the corresponding returned documents, but even user interactions and behaviour. User signals such as clicks, dwell time, i.e. the time spent in viewing a document, and other query and browsing features, are nowadays exploited as feature to improve LtR [Agichtein et al., 2006b].

This chapter is organized as follows: Section 2.1 illustrates some key concepts in IR and the main components of an IRS; Section 2.2 will provide an overview on the history of IR evaluation, from the Cranfield experiments to TREC, highlighting the main challenges; finally Section 2.3 illustrates some of the most popular measures adopted in IR to evaluate set-based and rank-based retrieval.

2.1 Information Retrieval Core Concepts

A typical IR scenario starts with a user and a collection of documents, where the term document denotes any information conveying item [Spärck Jones, 1997]: text documents, images, video and audio. The task of an Information Retrieval System (IRS) is to retrieve documents whose content is relevant to the user's information need [Spärck Jones, 1997], i.e. the user lack of some information necessary to answer a question, solve a problem or performing a task [Taylor, 1962]. Notice that, in the following chapters, we may refer to the information need with the word topic, which is a surrogate representing the information need (see Section 2.2.1).

The main components of an IRS are illustrated in Figure 2.1. The system takes as input the collection of documents and the user's query, and gives as output a set or a ranked list of documents. The user formulates her information need in natural language and expresses it with a query, often vague and ambiguous. Usually, queries comprise a small number of terms, with two to three terms being typical for Web search [Büttcher et al., 2016].

Having the user's query on the input side, the purpose of the IRS is to retrieve all the relevant documents, and at the same time, retrieve as few as non relevant documents as possible [van Rijsbergen, 1979]. At this point, the main challenges for the IR community are two: (i) correctly interpreting the user query and information need and (ii) define a representative of the query and the documents that allow the IRS to return the set or list of documents which mostly adheres to the user interests.

The notion of relevance is central to IR, however, due to the complexity of this concept, interpreting and correctly representing relevance is still an open issue [Allegretti et al., 2015; Koopman and Zuccon, 2014; Mizzaro, 1997; Saracevic, 1975]. Indeed relevance is subjective, different users might have different opinions regarding the relevance of a document with respect to a specific topic. Moreover, even when only one user is considered, her level of understanding of the topic can change during the information seeking process and therefore modify her perception of relevance. Further details on the solutions adopted by the IR community to handle the relevance notion will be discussed in Section 2.2.2 and Chapter 4, which presents our approach to deal with relevance labels.

An IRS attempts to capture the user information need expressed with the query and to optimally match it with the information content of each document. During the indexing process, documents and queries are represented with the same set of features to allow a match between their representations by means of a similarity function. Documents representations are permanently stored in the index as a list of terms with their frequency and a link to the original document, and they are usually obtained by removing the high frequency words, stripping suffixes and detecting equivalent stems [van Rijsbergen, 1979]. Different models were developed to characterize documents, queries and similarity functions, Section 2.1.1 will discuss some of these models: the Boolean model, the vector space model [Salton and McGill, 1986], the probabilistic model [Maron and Kuhns, 1960; Robertson, 1977], the language model [Ponte and Croft, 1998] and the more recent LtR framework [Liu, 2011].

Each model defines a different similarity function, whose main purpose is to assign a score to each query document pair, which represents a system estimate of the amount of relevant content held by the document with respect to the query. Finally, documents are ranked in decreasing order of score and presented to the user as a ranked list, where top ranked documents are those considered most relevant. Note that the earliest and easiest retrieval models, as the Boolean model, return an unordered set of document instead of a ranked list. This set or ranked list of retrieved documents with respect to a given query is known as run, and will be formally defined in Chapter 3.

Up to this point, the user is considered only as a static component of the IRS, who provides the information need to trigger the search process. However, to let IR systems to better fulfill the user satisfaction, the user started to be actively involved in the retrieval process, as shown in Figure 2.1. In particular, IRS started to collect and store log data, i.e. the interactions between the user and the ranked list of documents. The most important signals, exploited by the majority of search engines are clicks on the Search Engine Result Page (SERP), and dwell time, i.e. an estimation of the time that the user is spending in visualizing a document. User signals are used as an implicit source of relevance feedback, both to represent documents and in the definition of the scoring function. More details will be provided in Section 2.2.6 and in Chapter 5, where we will present a novel approach to integrate the user behaviour in a LtR algorithm.

2.1.1 Information Retrieval Models

As previously mentioned, different models define different strategies to represent documents and queries and to build a scoring function, that estimates the relevance of a document with respect to a given query. An IR model is an abstraction of the retrieval task, which aims at predicting and explaining what a user will find relevant given the query [Büttcher et al., 2016]. Moreover, it can be considered as an explanatory model of the data that can be used as a blueprint to build and implement a well grounded IRS [Goker and Davies, 2009; Ponte and Croft, 1998]. In the following sections some of the most commonly used retrieval models will be briefly introduced and discussed, starting from the easiest models, which return an unordered set of documents, and proceeding to the more complex ones returning a ranked list of documents.

The Boolean Model

The Boolean model represents the first model designed for IR and one of the easiest ones. It views each document as a set of terms, and queries are expressed as terms combined with the Boolean operators AND, OR, and NOT. Intuitively, if the query is considered as a Boolean expression, the model retrieves all the documents that are true for the query.

As an example, consider a query expressed as `Information AND Retrieval`; the system will retrieve all the documents that contain both the words `Information` and `Retrieval` and it will discard those documents that contain only the word `Information` or `Retrieval` or none of them.

Notice that, since the model is exploiting sets of terms as documents representations and Boolean expressions as queries representations, the result output will be a set of documents without any ordering among them. This inability to support rank based retrieval was considered as one of the most significant disadvantages of the Boolean model.

On the other side the Boolean model is easy and simple to use for expert users, i.e. users that are familiar with the Boolean operators. Indeed, the model allows the user to control the resulting set of documents: if a document is not included in the resulting set the cause is directly linkable to the terms and operators used as query. However, if the users are not trained, it might be difficult for them to understand the usage of these operators and they might get frustrated by a system which is so strict in deciding which documents are included in the resulting set or not.

The Vector Space Model

The vector space model [Salton and McGill, 1986] is based on the statistical approach to IR proposed by Luhn [Luhn, Hans Peter, 1957]. This model overcomes the limitations of the Boolean model since it does not make any use of Boolean operators and returns as output a ranked list of documents.

The model defines queries and documents as vectors embedded in a high dimensional Euclidean space, where each component of the vector represents a term. The ranking of the documents is obtained by computing the similarity of each document with respect to the query and by ordering the documents with decreasing similarity. The similarity function is typically defined as the cosine of the angle between the document and the query vector, therefore the smaller the angle the more similar the vectorial representations of the document and the query.

An example of algorithm built on top of the vector space model is the relevance feedback algorithm [Rocchio, 1971]. Rocchio suggested an approach to account for the user feedback: by assuming that the user gives feedback on the retrieved documents, when a query is revised the similarity function moves the query vectors towards the centroid of the known relevant documents and away from the centroid of the known non relevant documents. Although its simplicity, this approach is one of the first examples of a IR system able to account for user interactions.

Furthermore, the vector space model allows to assign different weights to each vector component, i.e. to each term of the collection. One of the most successful term weighting strategies is TF-IDF proposed by Salton and Yang [Salton and Yang, 1973], where TF stands for term frequency, i.e. the number of times that a term occurs in a document, and IDF stands for inverse document frequency, i.e. a value inversely related to the the number of documents that contain the term. The weight assigned to each term is computed as the product of these two quantities. The idea underlying the TF-IDF weighting scheme is that the term frequency alone is not enough to estimate the relevance of a document, therefore the IDF component accounts for high frequency terms, indeed a term appearing in many documents is not a good representative of the document content and it should be assigned a lower weight than a term appearing in just a few documents [Büttcher et al., 2016].

The vector model was criticized because of its entirely heuristic nature and its too simplistic theoretical framework. However, this model, in particular with the TF-IDF weighting scheme, is still used to define useful features that are used by more advanced LtR algorithms.

The Probabilistic Model

The foundations of the probabilistic model are grounded on the Probability Ranking Principle (PRP):

> If a reference retrieval system's response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data [Robertson, 1977].

The probabilistic model introduces the notion of uncertainty in IR by trying to estimate how likely is that a document is relevant to an information need and then sorting the documents accordingly to this probability. Therefore, if $R_{d,q}$ is a random variable equal to 1 when the document d is relevant with respect to the query q, the model ranks the documents decreasingly with respect to $\mathbb{P}[R_{d,q} = 1|q,d]$ [Robertson and Spärck Jones, 1976].

One of the easiest way to estimate the probability $\mathbb{P}[R_{d,q} = 1|q,d]$ is by considering the Binary Independence Model (BIM), which represents the documents and the query as binary vectors, i.e. each component stands for a term and is equal to 1 if the term occurs in the document or query and 0 vice versa. The main assumption at the basis of the BIM states that each term appears in the documents independently from any other term.

Initially, this model was mainly tested on domains composed of short catalog records and abstracts of consistent length. When it was tested on different domains, where documents have variable length, the model performance where quite poor[Büttcher et al., 2016]. Therefore, for full text collections, the model was extended to account for terms frequency and documents length. The result was the development of the BM25 model [Spärck Jones et al., 2000], which still represents one of the best performing retrieval model within the field, often used as a baseline to compare the performances of new retrieval systems [Büttcher et al., 2016]. As well as for the TF-IDF weighting scheme, BM25 scores are widely used to define features to represent document-query pairs in LtR algorithms.

The Language Models

IR language models developed around 1990s from probabilistic models of language generation, originally implemented for speech recognition systems. The basic idea is that the user has a reasonable knowledge of the terms that will appear in a document relevant for her information need, and she will include these terms when formulating the query. This can be translated by assuming that a document is a good candidate, if the language model produced from the document can generate the query terms with high probability. Therefore, a language model M_d is built for each document d and the probability $\mathbb{P}[q|M_d]$ is estimated. Notice that, the starting point is similar to the probabilistic model, i.e. the model attempts to estimate the probability of a query given a document, instead of computing a similarity score as in the vector model.

The first proposed language model is presented in [Ponte and Croft, 1998], where the documents are ranked accordingly to the probability of their language model to generate the query. Similarly, [Hiemstra and Kraaij, 1998] defines a model that unifies approches from IR and Natural Language Processing (NLP) in a single statistical framework. Finally, [Miller et al., 1999] presents a new language model for IR that exploits an Hidden Markov Model (HMM) to incorporate multiple words generation.

Learning to Rank (LtR)

In recent years, with the advent of Web search engines it has become more and more necessary to develop algorithms able to efficiently and effectively find the desired information. Due to its success in many research area, ML has been applied to solve the problem of ranking, i.e. to estimate the correct order among a set of documents, generating a new branch of ML called Learning to Rank (LtR) [Liu, 2011]. In a LtR framework the input documents are represented by feature vectors, which can depend on the query-document pair, only on the document, or only on the query. The output space contains the learning target, i.e. given a query, the output produces a ranking of documents. The algorithm aims at learning a function that maps the input space, i.e. the vectors of features, into the output space, i.e. tries to predict the ground truth labels as accurately as possible. Finally, a loss function determines to which extent an algorithm is accurate or not in predicting the ranking, by comparing the predicted scores with the ground truth ones.

In IR, three different ML approaches are mostly used: pointwise, pairwise, and listwise algorithms. Pointwise approaches learn a function that given a query and a document, represented as a vector of features, predicts the relevance degree of the document. Therefore the loss function is defined as the accuracy of the prediction.

As suggested by the name, pairwise approaches learn a function that takes a pair of document and a query as input, and returns the pairwise preference between the documents as output. In this case the loss function can be defined as the number of inconsistencies between the predicted preferences and those obtained from the ground truth.

Finally, listwise approaches work with the full set of documents, therefore given the collection they learn a function that returns a permutation of the documents. The accuracy of these algorithms are often evaluated in terms of standard IR evaluation measures, as those presented in Section 2.3.

Chapter 5 will describe in detail one of the mostly used LtR algorithm and will present how we integrate the user behaviour, in particular the user dynamic, in this state of the art algorithm. We now proceed with a description on how to perform evaluation in IR and a list of the most popular IR evaluation measures.

2.2 Information Retrieval Evaluation

IR evaluation aims at measuring how well a system retrieves and ranks relevant documents, and how to develop tests that will enable researchers to better understand both what is happening inside a system and the ability of a system to satisfy the user [Spärck Jones, 1997]. Therefore, evaluation of IR systems is a broad topic, which can be approached from many different perspectives, including information-seeking behaviour, usability of the system's interface, search context, and computing efficiency, cost, and resources required from the IRS [Sanderson, 2010].

Basically, the development and the evaluation of IR systems focus on improving two main aspects: efficiency and effectiveness. Efficiency is measured in terms of the amount of computer resources used, such as core, backing store, and Central Processing Unit (CPU) time, while effectiveness deals with some sort of user satisfaction [van Rijsbergen, 1979]. Effectiveness is often measured as the ability of the system to retrieve relevant documents, and at the same time, to suppress the retrieval of non relevant documents.

While measuring efficiency is somehow a straightforward task, correctly measuring effectiveness can be rather challenging. This was, and still is, a complex problem since it is difficult to determine the costs and benefits of having or not having some information and the utility that derives from it. For example, given two different ranked lists generated from two different IR systems on the same collection as a response to the same query, how is it possible to correctly determine which system is the best performing one? Finding an answer to this question is quite complex and still represents an open issue for the IR community, which constantly develops new approaches and solutions to tackle the evaluation task [Carterette et al., 2012; Ferro et al., 2016b; Smucker and Clarke, 2012a].

The purpose of the following sections is to briefly illustrate different strategies to test and evaluate systems from the effectiveness point of view. The next section describes the Cranfield paradigm, which incorporates the basics steps of IR evaluation, later on consolidated by the TREC conferences, presented in Section 2.2.3.

2.2.1 The Cranfield Paradigm

Librarians and their strategies to manually indexing text documents lie at the origin of modern IR. Indeed, at that time, scientific papers were not available online as today, therefore scientists needed to rely on librarians to find the most updated research papers. However, after World War II there was an increase in the number of published papers and it became hard for librarians to deal with such a huge volume of documents. Moreover, manually indexing these collections was expensive and determining the best strategy to index a collection started to be of concern to librarians.

Two main experiments are fundamental for the development of IR evaluation, both of them run by Cyril Cleverdon between 1950s and 1960s. The first of the two experiments carried out by Cleverdon to evaluate different manual indexes, is represented by Cranfield I [Cleverdon, 1960, 1962]. The experiment was run between 1958 and 1962 to test four manual indexing methods on a collection of $18,000$ papers. Cleverdon estimated that $1,600$ search questions were enough to evaluate the indexes and to guarantee that the results would pass significance tests. Nevertheless, due to the huge number of questions and documents, it seemed impossible to retrieve all the documents that were relevant to each question. Therefore, Cleverdon decided to reduce the size of the result set by turning the search task to known item search, i.e. finding just one document that was guaranteed to be relevant for a given question. To assure that each question had a corresponding relevant document, Cleverdon directly asked to the authors of the documents in the collection to formulate some questions that could be exhaustively answered by one of their papers in the collection.

The evaluation process consisted in using each indexing method to search for the right document for each question, and to record the time needed to perform the task and the success or failure of the search. Unfortunately, the experimental results were inconclusive, with a failure rate of 35% and no notable difference depending on the indexing method. Moreover, all the failures could be attributed to human indexing errors, rather than the adopted index [Harman, 2011]. Despite the apparent failure of the first experiment, Cleverdon realized that the similarities in the performances of the different indexes, were not due to the index strategy, but to the features used to describe the content of each document, as for example the number of terms for each descriptor, the number of descriptors, the weighting scheme for each term and many others.

These considerations encouraged Cleverdon to perform a second series of experiments, named Cranfield II [Cleverdon and Keen, 1966]. From the first experiment, it was evident how hard was to handle a big collection, therefore the new collection was composed of fewer documents and questions, 1,400 research papers and 221 questions. Again particular attention was devoted to the choice and definition of the questions: the authors of the papers included in the collection were contacted, and this time not only they were asked to formulate a question which summarized the content of their paper, but they were given instructions to assign a relevance judgement. They had to label 10 references cited in their paper and the relevance was expressed with a number from one to five. Since the entire collection needed to be assessed for relevance, the complete relevance judgements were provided preliminarily by five graduate students, and successively validated by the authors.

Cleverdon was particularly meticulous in designing this experimental collection. He thought that it was crucial to develop the test collection, together with the set of questions and the relevance judgements, before performing the indexing and search processes. Furthermore, he believed that a lot of effort and care should have been put in defining the users' information needs: the questions should reflect real users' needs and the relevance judgements real users' assessments.

This experimental setup lies at the heart of the modern IR evaluation process. It is known as the Cranfield paradigm and constitutes a standard for IR evaluation [Harman, 2011], establishing the basis of all the following evaluation activities.

The central core of the Cranfield paradigm is represented by the definition of test collection:

$$\mathscr{C} = \{D, T, J\}$$

which is formed by:

Documents D is the set of documents, also called corpus, is the set of items that the system can access to satisfy the user's information need;

Topics T is the set of topics, surrogates of the users' information needs, from which queries can be generated according to different strategies;

Relevance Judgments J is the ground truth, also referred as qrels, is a list of topic-document pairs which specifies the relevance grade of a document with respect to the given topic.

Relevance judgments are assigned by an assessor, who might or might not be the actual holder of the information need, and they are typically expressed on an ordinal scale, they can be binary, i.e. relevant or not relevant, or multi-graded, for example not relevant, partially relevant, fairly relevant and highly relevant [Kekäläinen and Järvelin, 2002]. Some works explored different choices to represent relevance, as for example magnitude estimation [Maddalena et al., 2017; Turpin et al., 2015] and preference judgements [Carterette et al., 2008; Rorvig, 1990], however this goes beyond the scope of this thesis and we will restrict to binary and multi-graded representations. Further details on how relevance judgments are assigned will be described in Section 2.2.2

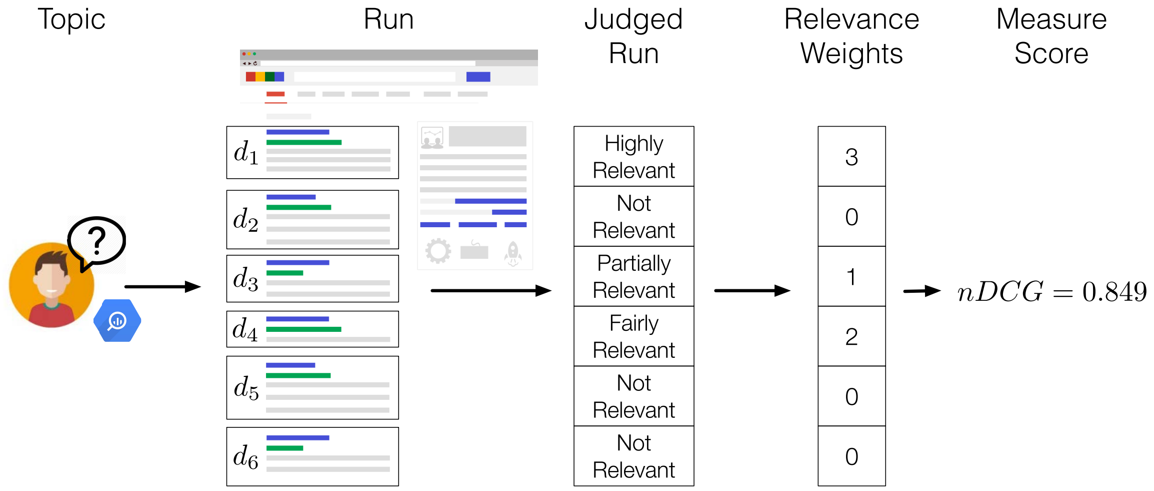


Fig. 2.2 IR collection based evaluation work flow.

Figure 2.2 graphically represents each passage of the evaluation process. First the IRS is provided with the set of document D to be searched and with a topic representing the user's information need. It will return as output a run, which originally was a set of documents, and later was replaced with a ranked list of documents with descending probability of being relevant with respect to the given topic. The run is compared with the set of relevance judgements to determine the relevance grade of each document, this forms the judged run. Then, each relevance grade is mapped to a relevance weight, often an integer number, which aims at quantifying the amount of relevance conveyed by the document. Finally, an evaluation measure is computed to attribute an effectiveness score to the run, generally returning a single

Collection	Year	#Documents	#Questions
Cran-2	1964	1398	225
IRE-3	1965	780	34
ADI	1965	82	35
Medlars	1970	1033	30
TIME	1970	425	83
NLP	1970	11429	93
INSPEC	1982	12684	84
CACM	1982	3204	52

Table 2.1 Test collections used during the SMART project, with approximate year of creation, number of documents and number of queries included in the collection.

number, belonging to the interval $[0,1]$. In addition to the simple run, evaluation measures can take as input the total number of relevant documents in the collections, to account for the completeness of the results set. The historical evolution of IR measures and the most popular evaluation measures are presented in Section 2.3, while their definitions, properties and further details are throughly discussed in Chapter 3 as part of our formal framework to model the evaluation process.

2.2.2 Early Evaluation Experiments

The introduction of the Cranfield paradigm inspired several new experiments and investigations. Of particular importance was the System for the Mechanical Analysis and Retrieval of Text (SMART), designed and led by Gerard Salton from early 1960s up to 1990s. SMART was a pioneering project that explored and designed a fully automated system for indexing and searching text documents. Many ideas were proposed and implemented during the project, for example some of the first fully automated methods to index and retrieve text documents, the scoring function to assign to each document a value representing its estimated relevance, and the presentation of the retrieved documents in a ranked list ordered by decreasing relevance instead of a set.

During the SMART project, the first instances of completely automatic evaluation experiments were carried out. This experimental setup had many advantages, it allowed to repeat the experiment, and to keep direct control of the experimental variables and the process itself. In addition to this framework to ease experimentation, a set of new collections based on the Cranfield paradigm were built to conduct experiments. Some example are IRE-1, a collection of abstracts from computer science literature, later extended as IRE-3 [Lesk and Salton, 1968], ADI was a collection of short academic papers, Medlars [Salton and Yu, 1973] consisted of abstracts of medical publications, TIME [Lesk et al., 1997], a collection composed of full text articles from Time magazines, and CACM, which included the Communications of the ACM (CACM) articles published between 1958 and 1979. Table 2.1 presents some of the test collections, used during the SMART project, with the year of first appearance or usage, and the number of documents and queries included, as reported in [Harman, 2011]. Not all the collections used in SMART were built inside the project, for example Cran-2 was the original Cranfield test collection, the NLP collection [Vaswani, 1970] came from Britain and contained titles and abstracts of journal papers, and the INSPEC collection was built at Syracuse University with scientific abstracts in electrical engineering.

From table 2.1 it is evident that the size of experimental collections was quite limited, especially if you think that the systems tested on these collections should scale to a much greater number of documents and queries. To understand the extension of the problem and its growth, consider that in early 1960s, systems were searching several tens of thousand of documents [Dennis et al., 1962], which grew to hundreds of thousands in mid 1970s [Bjørner and Ardito, 2003], were estimated to be more than 300 millions in 1998 [Lawrence and Giles, 1998], and, only for Web search, reaches almost 50 billions on the present day[1] [van den Bosch et al., 2016].

Existing test collections were not only small, but their quality was often variable, each collection was built to test a specific approach, and there was a lack of standardization. One of the first attempt to compensate the absence of a common experimental framework was made by Karen Spärck-Jones. She realized the necessity of a shared framework for evaluation experiments, after testing the same approach on different collections, and observing that the results were substantially different [Spärck Jones, 1973]. Therefore, in 1975, Karen Spärck-Jones and Cornelis Van Rijsbergen wrote a proposal for a large test collection [Spärck Jones and Van Rijsbergen, 1975], called the "ideal test collection".

The proposal was not dealing exclusively with the creation of a better and larger test collection, it was calling for a collection to consolidate research findings, that allows to repeat and reproduce the experiments and could be easily used for different purposes. However, a major stumbling block was represented by relevance judgements. It is extremely hard to collect the ground truth data required to fully asses systems' performances and it is not feasible for an assessor to read thousands, or even millions of documents, and assign them a rate.

The solution, to address the issue represented by relevance judgements, was also included in the proposal. Instead of proceeding with the complete assessment of the whole collection,

[1] http://www.worldwidewebsize.com/

the idea was to submit for relevance judgements only a small subset of documents, chosen in such a way to be a good representative of the entire set of relevant documents in the collection. This technique was called pooling, and the subset selected for relevance judgement was called pool. The proposal claimed that, if many independent and diversified approaches to retrieval are applied to the same collection, it is possible to pool their output and obtain a sample containing enough relevant documents to properly perform comparative evaluation tests. Indeed, the purpose of the ideal test collection was to allow for a comparison among systems run on the same collection, not an absolute evaluation of system performances.

However, mainly due to the insufficient financial support, the proposal did not have immediate impact on the research community. Moreover, a key challenge still remained unsolved: how to gather many independent and diversified approaches to search a test collection. The solution was proposed and carried out almost 20 years later, with the first TREC conference.

2.2.3 Text REtrieval Conference (TREC)

Between 1989 and 1990, DARPA, a US government agency, funded the National Institute of Standards and Technology (NIST) to build a large test collection for the evaluation of IR systems with text document. The collection resulting from NIST was named TIPSTER and contained $750,000$ text documents, much more documents than any other collection that had been built up to that time, as witnessed from table 2.1. The TIPSTER collection was not only large, but differently from previous collections, it was including mostly full articles, not just articles' abstracts, and the resulting collection size was around 2GB of text in total [Harman, 1992a]. Moreover the documents were coming from different sources, as newspapers, news wires, news releases, and technical abstracts.

In 1991, NIST decided to make the TIPSTER collection publicly available for researchers working in the field. Therefore, the government agency organized the first Text REtrieval Conference (TREC), an evaluation campaign held in 1992 with the purpose of promoting research and collaboration in IR. The conference was and is still organized as follows: each year NIST provides a test set, constituted by a set of documents and a set of questions. Each participant group runs their own IRS on the provided corpus and returns to NIST the top retrieved documents for each topic. Finally, NIST pools the result documents, perform the relevance judgements and returns a ranking with the systems score.

The format of the conference was highly influenced by the ideal test collection proposal and it was particularly committed to realize the pool generated with many independent and diversified approaches. The reasons behind making the TIPSTER collection available were twofold. First, by distributing the collection for no cost to the researchers in the community, each research group could test their retrieval system on the collection and give the run back to TREC. This provided a framework to obtain the independent and diversified runs to form the pool. Moreover, TREC hired expert assessors to judge the documents in the pool, which provided the performance evaluation of each system. During the conference, a final ranking of the systems was published to determine which strategy was the best performing one. The competitive challenge triggered by this format was guaranteeing that the runs submitted where the best performing and was encouraging the researcher to develop more and more advanced retrieval strategy.

Although the TREC conference started with the main purpose of developing a test collection for IR evaluation, it simultaneously achieved many more goals. It promotes the development of a standard framework which represents a necessary requirement to recognize and develop good performing systems. Moreover, the conference itself represents an occasion for researchers to meet, discuss and disseminate their work. Finally, by giving access to the relevance judgements and the runset, i.e. the runs submitted by each participant group, TREC had a significant impact on the definition and analysis of evaluation measures.

Due to all these compelling reasons, TREC has a considerable influence in IR, and it is now in its 26th edition. Many papers were written thanks to TREC collections, also this thesis based the experiments presented in Chapter 4, and Chapter 5 on some of those test collections. Moreover, the successful outcomes of the TREC conference encouraged the creation of other evaluation campaigns. For example the Conference and Labs of the Evaluation Forum (CLEF), focused on European languages, the NII Testbeds and Community for Information access Research (NTCIR) with an emphasis on Asian languages, the south Asian Forum for Information Retrieval Evaluation (FIRE) and, the INitiative for the Evaluation of XML Retrieval (INEX), to search semi-structured data.

2.2.4 Pool Construction and Reliability

Without any information about which documents are relevant or are not relevant to a user with an information need, there can not be any type of performance evaluation. However, it is extremely hard to collect the ground truth data required to fully asses systems' performances. Indeed, it is impossible to ask to an assessor to read thousands or even millions of documents and rate them.

The pooling methodology developed by TREC aims at defining an unbiased sample of relevant documents which will be judged for relevance. During the early years of TREC, this sample was created by merging all the top ranked documents returned as output of the different systems which were participating at the conference. Therefore, given the pool depth, said d, the union of the top d documents for each run and topic were selected and merged to form the pool, and then assessed for relevance. Usually, the standard pool depth is chosen equal to 100.

It is clear that with the pooling technique the majority of documents in the collection will not be included in the pool and therefore they will not be assessed for relevance. Furthermore, the standard approach assumes that those documents which are not included in the pool and are not be judged, can be considered as not relevant documents. This poses two main concerns questioning the validity of the pool: the first related to the completeness of relevance judgements and the reliability of system comparison when based on incomplete judgements; the second related to pool robustness, that is whether new evaluation approaches can be evaluated with the same collection, even if they did not participate to the pool creation.

In [Harman, 1995] Harman studied the effect of pool incompleteness. She examined TREC-2 and TREC-3 collections and assessed an additional pool formed with the documents in ranks from 101 to 200. The results show that less than one new relevant document per run was founded, which represent 11% additional relevant documents in TREC-2 and 21% relevant documents in TREC-3. Harman concluded that these levels of incompleteness are acceptable and it is reasonable to assume that relevance judgements are complete.

Also Zobel investigated the reliability of the pooling methodology [Zobel, 1998]: he reached a similar conclusion by studying the relationship between the number of relevant documents and the pool depth. Furthermore, he studied the robustness of the pool by removing the relevant documents unique to a particular system and by performing the ranking of all the systems, even the removed one, with the smaller pool. He compared the ranking obtained from the reduced and the original pool and found that the results are not biased towards a system: i.e. if a system does not contribute any document to the pool can still be evaluated fairly.

Finally, later works [Lipani et al., 2015; Soboroff and Robertson, 2003; Voorhees and Harman, 1999] highlight the importance of building the pool with many systems that implements highly diverse research strategy. If this hypothesis is satisfied, then the pool can be used to fairly evaluate not only contributing systems, but even new approaches which were not involved in the pool creations.

2.2.5 Crowdsourcing

Even if the TREC conference overcame the issues of collecting documents coming from independent and diversified approaches, the subsets of documents included in the pool still remains quite large, making relevance judgement an expensive activity. TREC was using retired intelligence analysts as assessors, who, although they were guaranteeing the high quality of judgements, were limited in the number of documents that they could assess.

Furthermore, TREC collections have another limitation: researchers who work with these collections are restricted by the IR tasks proposed during the conference. Therefore, if they want to investigate a new idea or if they want to test a new algorithm they might not be able to do it with a TREC collection.

One option to overcome these issues is to use a crowdsourcing platform, as Amazon Mechanical Turk[2] or Crowdflower[3] [Alonso and Mizzaro, 2009; Alonso et al., 2008]. These online services allow researchers to upload their tasks, which will be executed by an undefined and generally large group of people, called crowdworkers. The advantages of using crowd assessors are numerous: the costs to develop a test collection are drastically cut, since crowdworkes salary is much lower than expert assessors' salary; many crowdworkers will work simultaneously on the same task, therefore relevance judgements can be collected in less time and for more documents; finally crowdsourcing tasks are customizable and researchers are allowed to build their own test collection, designed to fit with their algorithms and their evaluation experiments.

Despite all these benefits, running an experiment on a crowdsourcing platform may entail some drawbacks. First, as mentioned in Section 2.1 relevance is subjective, therefore different users can disagree on the relevance of a document with respect to the same topic. It is not possible to ensure that the crowd assessor judgements reflect the actual user information need, since they are asked to put themselves in the position of a hypothetical user and figure out the information need that drove the formulation of the query.

Moreover, expert assessors from TREC where accurately trained to label documents and they were given some guidelines to be followed when performing the task. Even crowdworkers are provided with some guidelines, but they can not be controlled and their background is unknown. Indeed, crowdworkers may come from different places in the world with different experiences and culture, and they may not be qualified to perform some specific tasks. Since the quality of their work can not be throughly controlled, even the quality of their assessments can not be guaranteed.

To address these issues and reduce the noise, the simplest solutions consist in adding some control questions, which should be answered correctly by the workers, routing task between different crowd assessors, and collecting multiple judgements for the same query-document pair. Therefore, for a given topic, the same document can be judged by more than one assessor. This calls for a series of different approaches to merge the scores provided by different assessors. Some examples of state of the art approaches are MV, which considers each assessor as a voter and assigns the label with the higher number of votes, and EM [Bashir

[2] https://www.mturk.com/mturk/welcome

[3] https://www.crowdflower.com/

et al., 2013; Hosseini et al., 2012], which exploits a probabilistic model to infer the accuracy of each assessors and trusts more the assessors with higher accuracy. More details about these approaches will be described in Chapter 4, where the AWARE framework is presented.

2.2.6 Click Log and Query Data

As discussed in the previous section, even with crowdsourcing platforms, obtaining reliable relevance labels is not a straightforward task and many limitations have not been addressed yet. First, obtaining relevance labels still remains a costly activity. Indeed, crowdworkers need to be paid, furthermore due to the noise and the subjectivity related to relevance assessments, each document has to be judged by more than one worker.

Moreover, it is not easy to collect a large amount of labeled data. To the best of our knowledge, the largest labeled dataset, used in a published paper, contains around ten thousands queries and millions of documents, with only 684 assessed queries, each with 50 judged documents on average [Carterette, Ben and Pavlu, Virgiliu and Fang, Hui and Kanoulas, Evangelos, 2009]. This is far away from the necessities of commercial search engines, which are indexing almost 50 billions of Web pages on the present day[4] [van den Bosch et al., 2016].

Second, even by assuming that enough resources are available and allow the construction of a potentially large collection, how should these resources be spent? Is it better to label more queries and less documents for each query, or adopt the opposite strategy? Moreover, how can we select the documents to be labeled?

To overcome these limitations, a possible solution is to infer valuable and reliable information from click log data. Indeed, commercial search engines commonly record users interactions with their interface: as for example query keywords, clicked urls, queries and clicks timestamps and many others. These records can be considered as a triplets (q, r, c), where q is the query, r is the ranking presented to the user, and c the set of links clicked by the user. [Joachims, 2002]

Click log data offer several advantages [Joachims et al., 2017]: they are easy and inexpensive to collect, they are available in real time and they are user centered, i.e. they directly represent user preferences. However, even if click data has proven to be a valuable resource of implicit feedback, they are biased and noisy and they are intrinsically difficult to interpret [Joachims et al., 2005].

While it is still not feasible to completely replace relevance judgements with click data, they can be exploited as features for LtR. In [Agichtein et al., 2006b] a set of features, based

[4] http://www.worldwidewebsize.com/

on the user browsing behaviour, are defined as additional inputs to represent documents, while [Joachims, 2002] used click logs to optimize search algorithms. Furthermore, online LtR exploits click log data to infer user preferences between documents [Hofmann et al., 2013b]. Chapter 5 will present our novel strategy to embed the user behaviour in a LtR algorithm by exploiting click log data.

2.3 Information Retrieval Evaluation Measures

Since the beginning of IR evaluation much effort and research has been devoted to properly evaluate IRS. Early thinking on the subject focused on the motivations behind experimental evaluation: what exactly should be measured and how it can be measured [van Rijsbergen, 1979]. Regarding the motivations, experimental evaluation lay the foundations of the empirical method since its beginning: it allows the comparison between different systems and it provides reliable data, which let researchers to identify improvements in performance.

The problem of what should be measured, when an IRS is evaluated, raised a lot of questions and considerations. In 1966, Cleverdon listed six measurable quantities, which can be taken into account for IRS evaluation [Cleverdon and Keen, 1966]:

1. The collection coverage, i.e. whether the system indexed relevant documents;
2. The time lag, how much time the system needs to return a ranked list of documents in response to a user query;
3. The interface characteristics, as for example the representation of documents;
4. The effort requested to the user to satisfy her information need;
5. The recall of the system, which is the proportion of relevant documents that are retrieved;
6. The precision of the system, which is the proportion of retrieved documents that are relevant.

While quantities as time lag, collection coverage, CPU or memory consumptions are easier to measure and are related to the system efficiency, the challenging task is to measure the system effectiveness, i.e. determining to what extent an IRS is able to satisfy the user's information need by returning relevant documents. Cleverdon mainly related this task to Precision and Recall, which have been the most popular evaluation metrics from the beginning of IR to the present day.

However, the definition of effectiveness implies that an IRS is successful if it fulfills the user satisfaction. When stated in this way, it should be evident that the user has to be involved in the evaluation process. This lead to a dichotomy between different experimental methodologies: system centered experiments, solely based on the system output, and user centered experiments, based on the actual user experience [Robertson, 2008b].

System centered experiments have many advantages: they provide reliable data to allow formal comparisons between different systems, they ease replicability, and the experimental collections developed for testing systems can be reused to tests other systems. On the other side, system centered evaluation does not involve the user, and it makes some abstract assumptions to simplify the problem and reduce the noise. For example, the subjectivity of relevance is not taken into account and documents are judged by a third party assessor. Moreover, the search journey, i.e. the full process that leads the user to satisfy her information need with query reformulation and refinement, is not considered and the evaluation is only restricted to a specific query and a specific output. Finally, the system interface and the document representation on the result output page are not regarded.

On the contrary, user centered approaches overcome all the aforementioned disadvantages. Laboratory user studies allow researchers to directly observe their behaviour and to collect their impressions by asking them to fill a survey. Furthermore, when users' log data are available, researchers have the opportunity to exploit a large amount of data and to infer user preferences. However, user center experiments embodies many other drawbacks [Voorhees, 2008]. First, they depend on many factors, as the system interface or the subjects involved in the study, therefore they are not replicable and the experimental results that they provide can not be generalized. Second, they are expensive and time consuming; someone can argue that building a test collection is expensive too, however test collections can be reused, while if there is any change in the experimental setting, the user study has to be performed again. Finally, user data are often noisy, especially query log data, which are biased and difficult to interpret [Joachims et al., 2005].

In this thesis we will focus mainly on system based or offline evaluation, with the next section giving an insight about the most popular evaluation measures. Moreover, Chapter 3 will present a novel framework, based on the representational theory of measurement [Krantz et al., 1971], to formally define and study the properties of IR evaluation measures.

2.3.1 Evaluation of Set-based IRS

With the definition of the Cranfield paradigm and the creation of test collections, it became necessary to define a proper way to objectively measure system performances. Recall that, at the time of Cranfield experiments, IR systems were based on the Boolean model and

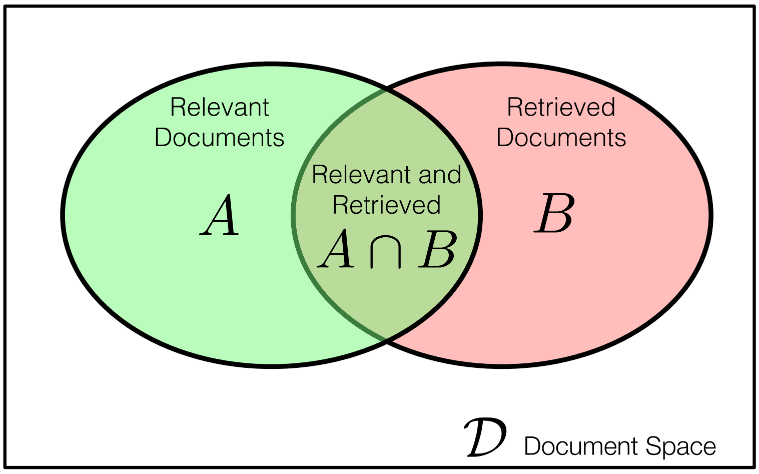


Fig. 2.3 Division of the document space in relevant documents, A, and retrieved documents, B.

	Relevant	Not Relevant
Retrieved	$A \cap B$	$\overline{A} \cap B$
Not Retrieved	$A \cap \overline{B}$	$\overline{A} \cap \overline{B}$

Table 2.2 Contingency Table

they were returning an unordered set of documents, rather than a ranked list of documents. Therefore, the first evaluation measures, adopted by Cleverdon in his experiments, were not taking into account the rank position and they were based on the quantities illustrated in Figure 2.3 and expressed by the contingency table 2.2, where A denotes the set of relevant documents, B the set of retrieved document, $\overline{A}$ is the complementary set and $|A|$ is the cardinality, i.e. the number of elements in the set.

In [Cleverdon and Keen, 1966], Cleverdon and Keen define three different IR evaluation measures, based on the afore mentioned quantities:

precision: the ratio between the number of relevant retrieved documents and the total number of retrieved documents

$$\text{Precision} = \frac{|A \cap B|}{|B|} ,$$

recall: the ratio between the number of relevant retrieved documents and the total number of relevant documents

$$\text{Recall} = \frac{|A \cap B|}{|A|} ,$$

and **fallout**: the ratio between the number of not relevant retrieved documents and the total number of not relevant documents

$$\text{Fallout} = \frac{|\overline{A} \cap B|}{|\overline{A}|} .$$

From the beginning of IR, precision and recall were dominating the scenes, while fallout was overlooked. However, neither precision nor recall could be used alone [Kent et al., 1955] and Cleverdon experimentally showed that there is an inverse relationship between them [Cleverdon, 1962]. The idea underlying this assumption is that, if an IRS focuses on relevant documents and restricts the space of retrieved documents, then precision will be high, but recall might be low; on the other side, if the result set is broadened to include more documents, recall will increase, but precision will decrease [Sanderson, 2010]. Informally, precision measures the accuracy of the result set, while recall measures the completeness.

Later on, to combine precision and recall into a single value, a measure called **f** was proposed in [van Rijsbergen, 1974]:

$$\text{f} = 2\frac{\text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}} ,$$

which is the harmonic mean of precision and recall.

2.3.2 Evaluation of Rank-based IRS

The development of more sophisticated retrieval systems brought new challenges into the evaluation landscape, especially when systems' output turned into an ordered list of documents instead of a simple set. Therefore, evaluation measures not only had to consider the relevance of retrieved documents, but also the rank position where those documents where displayed. Indeed, those documents that highly satisfy the user's information need should be ranked higher than those which do not match the user query well, and an evaluation measure should account for it.

First attempts to develop an evaluation framework, able to exploit the additional information given by the ranking, were proposed by Swets [Swets, 1963]. First of all Swets tried to list the desiderata for an evaluation measure:

1. It would be a measure of solely effectiveness, which means that the measure would only distinguish between relevant and not relevant documents and would not account for any cost related to efficiency;

2. It would express a trade-off between the fraction of retrieved documents that are relevant and the fraction of relevant documents retrieved, i.e. a compromise between precision and recall;

3. It is preferable a measure with a single number as output, rather than a pair of covarying number or a curve of points;

4. It would allow a complete ordering between different systems and it would score each system in an absolute way, i.e. it would have a minimum and a maximum value.

Then he proposed a new model based on the statistical decision theory. His formalism was based on the idea that relevant and not relevant documents are described by a different probability distributions, therefore, a good system would produce two distributions for relevant and not relevant documents which are easily distinguishable, while a badly performing system would conflate relevant and not relevant distributions. Even if his formalism was confirmed later by experimental results [Manmatha et al., 2001], it was not adopted by the scientific community, which continued to investigate the possibility to adapt precision and recall to rank based retrieval.

The easiest solution to extend precision and recall to rank retrieval was to define a cut-off level k and to consider all the documents ranked before the given cut-off as an unordered set. Therefore, **precision at cut-off k** [Salton, 1968; van Rijsbergen, 1979] is defined as:

$$\mathrm{P@k} = \frac{1}{k}\sum_{i=1}^{k} r_i \ ,$$

and **recall at cut-off k** is defined as:

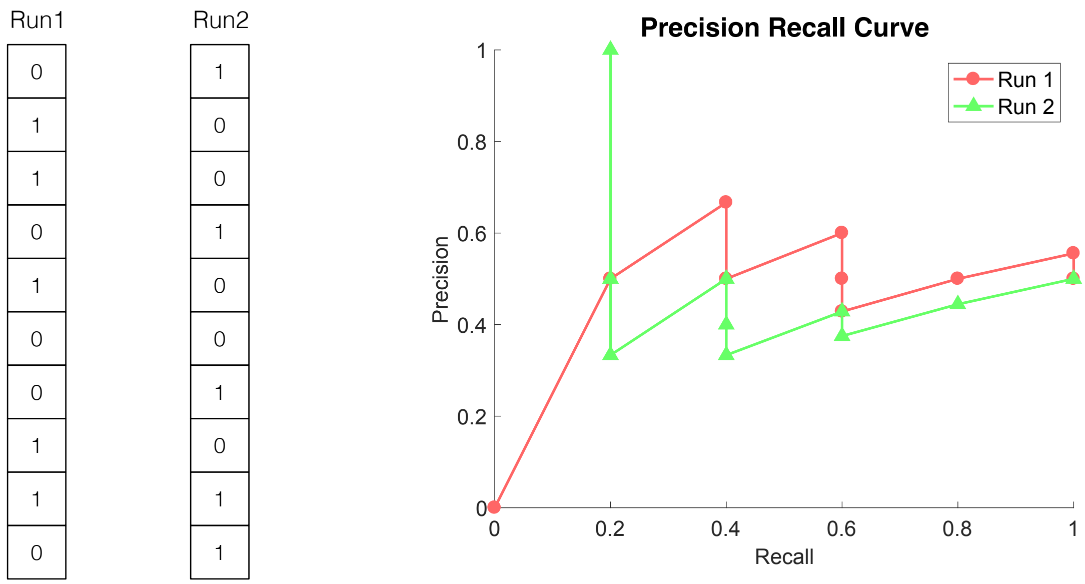
$$\mathrm{R@k} = \frac{1}{R}\sum_{i=1}^{k} r_i \ ,$$

where r_i is a binary variable, equal to 1 if the document at rank i is relevant or 0 otherwise, and R is the total number of relevant documents, i.e. $R = |A|$. While recall@k was not widely used, precision@k became more popular and it is still in common use, especially with $k = 10$, which corresponds to the number of documents displayed on the first page of many web search engines.

Despite its popularity, mainly due to its easy interpretation, precision@k raises some doubts. First, consider two runs that retrieve only one relevant document. With respect to precision@10, the score of the two runs will be 0.1, if the relevant document is either in position 1 or 10, therefore they are evaluated equally independently of the rank position of the relevant document. This means that precision@k is not informative with respect to the rank position and treats the run as an unordered set.

Second, assume that a topic has a great number of relevant documents, $R >> k$, then it would be easy for any system to retrieve a lot of relevant documents and to place them at the beginning of the run, this will result in precision@k$= 1$. Conversely, if $R < k$, then it is not possible to reach the perfect score equal to 1 and each system will be evaluated as low performing. Therefore, precision does not account for the total number of relevant documents and it can not be used without considering also recall.

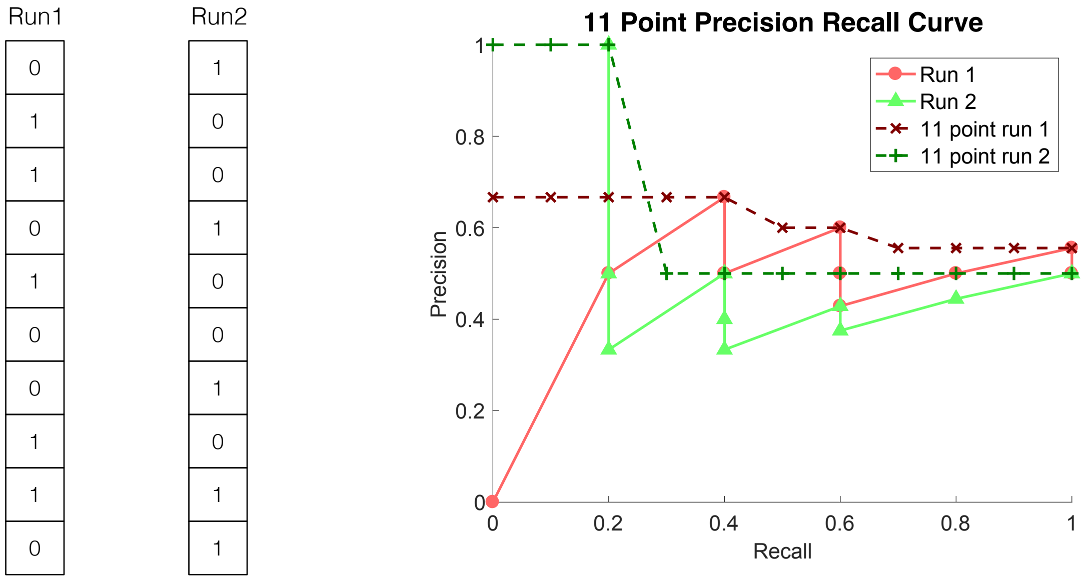
To overcome this second drawback, during TREC-2 [Harman, 1993] **RPrec** was defined as precision with k equal to the recall base R, i.e. the threshold before which a perfectly performing system would place all the relevant documents and after which it would place all the non relevant documents. Therefore, RPrec assumes a different cut-off for each topic, depending on the total number of relevant documents. However, this variant of precision still overlooks the rank of relevant documents and consider the documents ranked before the cutoff as an unordered set.



Run1	Run2
0	1
1	0
1	0
0	1
1	0
0	0
0	1
1	0
1	1
0	1

Fig. 2.4 Precision recall curves for two different runs

A popular strategy to combine precision and recall and adjust them to ranked retrieval was to consider both the measures and plot the precision recall curve, where precision is computed as a function of recall. Figure 2.4 shows precision and recall computed at each rank, with respect to two different runs. The precision recall curve has the classic saw-tooth shape [Manning et al., 2008], if the document at rank $k+1$ is not relevant, then recall will



Run1	Run2
0	1
1	0
1	0
0	1
1	0
0	0
0	1
1	0
1	1
0	1

Fig. 2.5 11 point interpolated precision recall curves for two different runs

remain constant, but precision will decrease. Conversely, if the document at rank $k+1$ is relevant, then both precision and recall will increase.

It is evident that obtaining a single score measure from the precision recall graph in Figure 2.4 requires extra techniques to average different curves. The solution was to consider an interpolated precision recall curve where precision is considered at each of the standard recall levels. In details, 11-point precision recall curve defines eleven recall levels, $r \in \{0.0, 0.1, 0.2, \ldots, 0.9, 1.0\}$ and for each level, the interpolated precision is defined as the highest precision obtained in any following level [Zhang and Zhang, 2009]:

$$p_{\text{iterp}}[r] = \max_{r' \geq r} p[r'] \; .$$

Figure 2.5 shows the precision recall curve together with the 11-point interpolated precision recall curve. Notice that the interpolated precision recall curve was used as a standard to evaluate systems both in the SMART project and the TREC conferences.

A solution to account both for precision and recall at different ranking positions was reached with one of the most widely used measure for retrieval effectiveness: **Average Precision (AP)** [Buckley and Voorhees, 2005; Harman, 1993], often referred as the gold standard. AP is the average of precision computed at each rank positions, where a relevant document is returned. The definition of AP is given by the following equation:

$$\text{AP} = \frac{1}{R}\sum_{i=1}^{n} r_i \cdot \text{P@}i = \frac{1}{R}\sum_{i=1}^{n} r_i \cdot \sum_{k=1}^{i} \frac{r_k}{k} \; , \tag{2.1}$$

where n is the length of the run, i.e. the total number of documents retrieved. As shown in [Aslam et al., 2005; Robertson et al., 2010] AP is an approximation of the area under the precision-recall curve. When the average per topic is computed, AP is often called **Mean Average Precision (MAP)**, which represents one of the primary evaluation measures used in IR literature [Sanderson, 2010].

AP introduces the notion of top heaviness, i.e. the higher a document is ranked, the more weight is assigned to it. This reflects the idea that if a relevant document is retrieved deep in the ranking, the system misplaced it, resulting in more effort requested to the user to find the relevant document. Moreover, the discount given to each document does not depend just on the rank position, but also on the recall base R.

Due to its dependency on the recall base, many criticisms has been raised against AP and all the other measures based on this quantity. First, exactly computing the total number of relevant documents is challenging especially for large collections, since the pooling method provides relevance assessments just for a fraction of the collection and considers all the documents outside the pool as non relevant. Second, the notion of recall base is not known by the user, therefore it can not affect the user satisfaction. However, since AP and RPrec rely on this quantity, they can not truly resemble the user behaviour. More details related to the user models that lie at the basis of evaluation measures will be presented in Chapter 5.

An early and complementary approach to measure retrieval effectiveness is Expected Search Length (ESL), proposed by Cooper in 1968 [Cooper, 1968]. ESL aims at including the user's need as a variable of the evaluation measure and it is based on the idea that the main purpose of an IRS is to help the user in satisfying her information need, by reducing the effort spent in searching relevant documents. Therefore, the measure is an estimation of the effort that the user will save when using an IRS instead of randomly search a collection of documents, and it also accounts for the difficulty of the query. ESL was not widely adopted at that time, however it inspired later works on IR evaluation, among those the family of cumulated gain metrics represents the most popular one [Järvelin and Kekäläinen, 2002].

Together with RBP, cumulated gain metrics can be framed in the broader family of rank weighted metrics. As shown in [Yilmaz et al., 2010; Zhang et al., 2010], effectiveness is often measured as the inner product of a relevance vector $\mathscr{R}$ and a discounting vector $\mathscr{W}$.

$$\text{Effectiveness} = \mathscr{R} \cdot \mathscr{W} = \sum_{i=1}^{n} \mathscr{R}_i \cdot \mathscr{W}_i$$

The elements $\mathscr{R}_i$ account for the benefit of ranking an high-quality document at the i-th position of the SERP, while $\mathscr{W}$ denotes such contribution for low-ranked documents, i.e. discounts the score assigned to a document accordingly to its rank position. The underlying assumption is that low-ranked documents receive less attention by the user and therefore they contribute less to the user-perceived quality of the SERP.

Discounted Cumulated Gain (DCG) [Järvelin and Kekäläinen, 2002] was the first measure that was explicitly proposed as a rank weighted measure. This measure is naturally multigraded, since $\mathscr{R}$ allows any possible weight to represent relevance grades. Each relevant document is considered as a gain for the user: $\mathscr{R}_i = l_i$, where l_i is the relevance label of the i-th ranked document and was originally defined as a natural number in $\{0,1,2,3\}$. However, this gain is discounted accordingly to the rank position where the document was displayed using a logarithmic factor:

$$\mathscr{W}_i = \begin{cases} \frac{1}{\log_b(i)} & \text{if } i > b \\ 1 & \text{otherwise} \end{cases} ,$$

where b is set equal to 2, when we are considering an impatient user, and equal to 10, when we are considering a persistent user. Finally, the score attributed to each document is cumulated over the rank positions:

$$\mathrm{DCG} = \sum_{i=1}^{n} \frac{l_i}{\max\{1, \log_b(i)\}} .$$

Another version of DCG was proposed in [Burges et al., 2005] and became quite popular, especially used as in combination with LtR. It is known with the name Microsoft DCG, and it is expressed by the following equation:

$$\mathrm{DCG} = \sum_{i=1}^{n} \frac{2^{l_i} - 1}{\log_2(i+1)} ,$$

where the weighting scheme for relevance is substituted with an exponential function of the relevance label, to give more importance to more relevant documents. Moreover, the discount function distinguishes between rank position 1 and 2, indeed the original definition of DCG, with $b = 2$, evaluates equally two runs which place a document with the same relevance weight either at rank position 1 or 2, as for example $r = (2,0,\ldots)$ and $r = (0,2,\ldots)$.

Cumulated Gain (CG), a simpler version of DCG was also proposed in [Järvelin and Kekäläinen, 2002]. CG is basically Discounted Cumulated Gain (DCG) without the discount function:

$$\mathrm{CG} = \sum_{i=1}^{n} l_i .$$

Therefore, it does not account for the rank position where a document was ranked, but just for the gain that it provides.

Despite their simplicity and popularity, both CG and DCG lack on one of the most important properties of a measure, listed as one of Swets' desiderata at the beginning of this section: a measure would have a minimum and a maximum values. The minimum value does not represent an issue, since a system that does not retrieve any relevant document will achieve a score equal to zero, which represents the minimum. However, the maximum achievable CG or DCG value is not clear, and it depends on the number of relevant documents for each grade, therefore it is different for each topic.

The easiest solution to address this problem and bound the measure in the standard interval $[0,1]$, is to divide the measure by the maximum score that it can achieve on a given topic. The maximum score is achieved with the ideal run, which is the run that ranks all the relevant documents at the top and arranges them decreasingly with respect to their relevance grade. Then we can define **Normalized Discounted Cumulated Gain (nDCG)** as

$$\text{nDCG} = \frac{\sum_{i=1}^{n} \frac{l_i}{\max\{1, \log_b(i)\}}}{\sum_{i=1}^{n} \frac{l_i^{id}}{\max\{1, \log_b(i)\}}}, \tag{2.2}$$

and **Normalized Cumulated Gain (nCG)** as

$$\text{nCG} = \frac{\sum_{i=1}^{n} l_i}{\sum_{i=1}^{n} l_i^{id}},$$

where l_i^{id} is the relevance weight of the document at rank position i of the ideal run. Notice that, the introduction of the normalization by mean of the ideal run implicitly assumes the knowledge of the number of relevant documents for each relevance grade, therefore the computation of the recall base.

Another popular rank weighted measure is **Rank-Biased Precision (RBP)** [Moffat and Zobel, 2008]. RBP is built on top of a simple user model: the user starts to examine the first documents returned in the run and then with probability p examines the next document in the ranking, or with probability $1-p$ she abandons the search. Therefore, the probability that the user examines the document at rank i is p^{i-1}, since before reaching i she has to examine all the previous documents. The discount vector is proportional to the probability of examining the document at rank i:

$$\mathscr{W}_i = (1-p)p^{i-1}$$

and RBP can be computed as

$$\text{RBP} = (1-p)\sum_{i=1}^{n} r_i \cdot p^{i-1} \ .$$

Since its definition depends on the geometric series, RBP takes values in the range $[0,1]$, therefore the measure does not require any further normalization with the recall base. However, in order to achieve a perfect score of 1, the system has to return a run with an infinite number of relevant documents and without any non relevant document, which is clearly not feasible. Therefore, if RBP does not require the estimation of the recall base to be computed, on the other side it suffers from limitations similar to those that Precision@k presents. For example, consider two topics, one with a high number of relevant documents and the other with just a few number of relevant documents. Every system will perform better on the topic with higher recall base simply because it will be more likely to place more relevant documents at the beginning of the ranking.

Finally, **Expected Reciprocal Rank (ERR)** [Chapelle et al., 2009] is another metric that discounts each rank position, but it does not belong to rank weighted metrics. ERR is based on a different user model named cascade user model and it is computed as follows:

$$\text{ERR} = \sum_{i=1}^{n} \frac{1}{i} \mathbb{P}[\text{user stops at position } i] \ .$$

The cascade model assumes that the user starts from the first document in the run and scans the ranked list of documents from each rank position to the following position. Let x_i be the probability that the user is satisfied by the document in position i, then $\mathbb{P}[\text{user stops at position } i]$ is the product of the probability that the user is satisfied by the document in position i and not satisfied by all the previous documents in the ranking. Therefore, ERR can be reformulated as:

$$\text{ERR} = \sum_{i=1}^{n} \frac{1}{i} \prod_{d=1}^{i-1} (1-x_d) x_i \ . \tag{2.3}$$

The satisfaction probability can be either calibrated on click log data or can be computed as a function of the relevance labels, with the same weighting scheme suggested in [Burges et al., 2005] for DCG:

$$x_i = \frac{2^{l_i}-1}{2^{l_{max}}} \tag{2.4}$$

where l_{max} is the maximum relevance weight, and the normalization is necessary since this quantity represents an estimation of a probability, so it should range in $[0,1]$.

Similarly to RBP, ERR does not make any use of the recall base and it is not even aware of the total number of relevant documents. However, this implies that ERR, as RBP, can not account for the differences between topics, in terms of amount of relevant information.

Although ERR is not a rank weighted measure, it is a top heavy measure, with the discount function that depends both on the rank position, with the factor $\frac{1}{i}$, and on the previously ranked documents, with the product $\prod_{d=1}^{i-1}(1-x_d)$. Therefore, even if AP, RBP, DCG and ERR are all top heavy mesures their discount functions are quite different from each other. A natural question might be, how different are these measures? Is AP more or less top heavy than the other measures? To which extent is AP more top heavy than RBP? Section 3.5 attempts to answer to this questions and discusses the *balancing index*, our approach to quantify the top heaviness of evaluation measures.

To conclude the list of evaluation measures, **Binary Preference (bpref)** [Buckley and Voorhees, 2004; Soboroff, 2006] is a measure designed to account for the unjudged documents in the run. It is based on binary preferences and it evaluates systems using only the judged documents. It can be thought as the inverse of the fraction of judged irrelevant documents that are retrieved before relevant ones:

$$bpref = \frac{1}{R} \sum_{i:\, r_i=1} \left(1 - \frac{|j \text{ ranked higher than } i|}{\min(R, NR)} \right) \tag{2.5}$$

where j is a member of the first R not relevant retrieved documents. bpref has proved to be quite robust in the case of incomplete and imperfect relevance judgements. In this thesis it represents a comparison point when evaluating measures with respect to reduced-size pools in Chapter 5.

It can be noted how heavily bpref depends on the recall base R. This is not only a scale factor as in the case of AP but it also determines the cardinality of the set from which the not relevant documents j are taken. Moreover, it makes use also of NR, the total number of judged not relevant documents, a kind of information which is hard to imagine available to any real user. So, in a sense, it seems much more a “pool-oriented” than a system-oriented measure since, for determining its score, it uses much more information about the pool than about the system under examination and this could be an explanation of its robustness to the pool reduction.

Finally, the reader may notice that this section presents several evaluation measures, but it does not give a general and formal definition of IR evaluation measure. Even if much effort and research has gone into finding proper evaluation strategies, most of them derive from experimental results and the general notion of evaluation measure is absent in the literature.

To this purpose Chapter 3 will detail our novel framework to formally define evaluation measures and describe their properties.

Chapter 3

Towards a Formalism for IR Evaluation Measures

> A methodology for evaluation ultimately invokes a theory of evaluation.
>
> Spärck Jones [1997]

Information Retrieval (IR) has been deeply rooted in experimentation since its inception and we often hear quotes like "*To measure is to know*" or "*If you cannot measure, you cannot improve it*", attributed to Sir William Thompson first baron of Kelvin, to remark the importance of experimental evaluation as a means to foster research and innovation in the field.

As pointed out at the end of Section 2.3.1, even if evaluation has greatly contributed to the advancement of IR, we still lack a deep comprehension about what the evaluation measures we daily employ are and this, somehow, hinders the "*to measure*" part in Lord Kelvin's quotes. This is witnessed by the fact that our understanding of evaluation measures is mostly tied to empirical evidence: for example, we use different kinds of correlation analysis [Kendall, 1948; Yilmaz et al., 2008] to see how close two evaluation measures are, we adopt different pool downsampling techniques to study the robustness of measures to incomplete information [Buckley and Voorhees, 2004; Yilmaz and Aslam, 2008], we analyse their sensitivity, stability and discriminative power [Buckley and Voorhees, 2000; Sakai, 2006], and so on.

We, as others [Amigó et al., 2013; Busin and Mizzaro, 2013; Fuhr, 2010; Moffat, 2013], think that a better comprehension of evaluation measures is needed and that the development of a formal theory to define what an evaluation measure is and to derive and study its properties can be the way to address this need.

In this chapter, we start to lay the foundations for a formal framework for utility-oriented measurements of retrieval effectiveness [Ferrante et al., 2015] and we present the notation that will be used throughout this thesis. In particular, we place our work in the broader framework of the *representational theory of measurement* [Krantz et al., 1971], which provides the foundations of the modern theory of measurement in both physical and social sciences.

Our work differs from previous attempts to formalize IR evaluation measures in three main aspects:

- for the first time, it explicitly puts IR measures in the wake of the measurement theory adopted in other branches of science;
- it provides a deeper understanding of what issues are due to the intrinsic difficulties in comparing runs rather than attributing them to the expected numerical properties of a measure;
- it is minimal, basically consisting of just one axiom (Definition 1), which makes the framework easy and intuitive to grasp and from which the other needed properties are (and will be) derived.

The chapter is organized as follows: Section 3.1 explains the basic concepts of the representational theory of measurement and how our framework will lay on it; Sections 3.2 to 3.5 introduce our framework; finally, Section 3.6 wraps up the discussion and outlooks some future work.

3.1 Background: Measurement and Measure

3.1.1 Representational Theory of Measurement

The act of measure pervades our daily routine: we use prices to measure the monetary quality of a product, height and width to determine the size of an item and weight to quantify the mass of an object. Moreover, a measure does not merely assigns a descriptor to a feature of an object that allows us to distinguish different items, but it eases the comparison among different entities. For example, we compare the price of similar products to decide which one we will buy, or we measure the height and length of different pieces of furniture to determine which one can fit better in our house.

It is evident that even though the set of assessments and comparisons that lead us to a decision is intuitive and natural, it is controlled by a precise set of rules. The definition of measurement proposed in [Fenton and Bieman, 2014] attempts to define this process:

> **Measurement** is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them accordingly to clearly defined rules.

where an entity denotes an item or object of the real world, and its attributes are the characteristics or features that describe the entity.

The *representational theory of measurement* [Krantz et al., 1971] aims at providing a formal basis to our intuition about the way we measure the attributes of the entities in the real world. According to the above definition of measurement, the numbers or symbols we collect as measures about the entities' attributes represent an abstraction of the way we perceive the real world. For example we can use numbers, as centimeters or inches for length, or symbols, as small, medium, and large for clothes' sizes. Regardless of the chosen representation, the numbers or symbols we collect as measures about the attributes of the entities, should be such that their processing and manipulation maintain the relationships among the actual entities under examination in the real world. Therefore, at the basis of measurement, there are the relationships among entities and how we empirically observe them [Finkelstein, 2003].

Consider, for example, the attribute "height" of a tree: in the real world, we are easily able to recognize that some trees are "taller than" others. "Taller than" is an **empirical relation** for height (of a tree) and we can think at it as a *mapping* from the real world to a formal mathematical one, namely from the set of trees to the set of real numbers, provided that, whenever a tree is "taller than" another one, any measure of height assigns a higher number to that tree.

This is the so called *representation condition*, which ensures that a measurement must map attributes of entities into numbers (symbols) and empirical relations into numerical (symbolic) ones, so that the empirical relations imply and are implied by the numerical (symbolic) ones.

More formally [Krantz et al., 1971; Mari, 2000], a **relational structure** is an ordered pair $\mathbf{X} = \langle X, R_X \rangle$ of a domain set X and a set of relations R_X on X, where the relations in R_X may have different arities, i.e. they can be unary, binary, ternary relations and so on. Given two relational structures $\mathbf{X}$ and $\mathbf{Y}$, a **homomorphism** $\mathbf{M} : \mathbf{X} \rightarrow \mathbf{Y}$ from $\mathbf{X}$ to $\mathbf{Y}$ is a mapping $\mathbf{M} = \langle \mathrm{M}, \mathrm{M}_R \rangle$ where:

- M is a function that maps X into $\mathrm{M}(X) \subseteq Y$, i.e. for each element of the domain set there exists one corresponding image element;

- M_R is a function that maps R_X into $\mathrm{M}_R(R_X) \subseteq R_Y$ such that $\forall r \in R_X$, r and $\mathrm{M}_R(r)$ have the same arity, i.e. for each relation on the domain set there exists one (and it is usually, and often implicitly, assumed: and only one) corresponding image relation;

with the condition that $\forall r \in R_X$, $\forall x_i \in X$,

$$\text{if} \quad r(x_1, \ldots, x_n) \quad \text{then} \quad \mathrm{M}_R(r)\Big(\mathrm{M}(x_1), \ldots, \mathrm{M}(x_n)\Big) \ ,$$

i.e. if a relation holds for some elements of the domain set then the image relation must hold for the image elements.

Note that we talk about a homomorphism rather than an isomorphism because M is generally not one-to-one; in general $M(a) = M(b)$ does not mean that two trees are identical but merely of equal height.

A relational structure **E** is called **empirical** if its domain set E spans over the entities under consideration in the real world, e.g. the set of trees; a relational structure **S** is called **symbolic** if its domain set S spans over a given set of symbols, e.g. the set of positive real numbers $\mathbb{R}_0^+ = \{x \in \mathbb{R} \mid x \geq 0\}$.

We can now provide a more precise definition of measurement on the basis of the just introduced concepts

> **measurement** is a homomorphism $\mathbf{M} = \langle \mathrm{M}, \mathrm{M}_R \rangle$ from the real world to a symbolic world. Consequently, a **measure** is the number or symbol assigned to an entity by this mapping in order to characterize an attribute [Fenton and Bieman, 2014].

As an example, consider a set of rods R [Krantz et al., 1971], where an order relation $\preceq$ and a concatenation operation $\circ$ among rods exist. Note that $\preceq$ is a binary relation on the set of rods R, while $\circ$ is a ternary one, which assigns to each pair of rods a third rod representing their concatenation. Then, the empirical relational structure, represented by the set of rods and their relation $\mathbf{E} = \langle A, \preceq, \circ \rangle$, can be mapped into the symbolic relational structure $\mathbf{S} = \langle \mathbb{R}_0^+, \leq, + \rangle$, using as mapping function $M(\cdot)$, the length of a rod, so that $a \preceq b \Leftrightarrow M(a) \leq M(b)$ and $M(a \circ b) = M(a) + M(b)$.

Note that this example covers also the basics of the classical measure theory [Billingsley, 1995; Folland, 1999], where the order relation among sets is given by $A \preceq B \Leftrightarrow A \subseteq B$ and the concatenation operation between two disjoint sets $A \cap B = \emptyset$ is given by $\circ = A \cup B$; a measure is then requested to be *monotonic* $A \subseteq B \Rightarrow M(A) \leq M(B)$ and *additive* $A \cup B \Rightarrow M(A) + M(B)$ when two sets are disjoint $A \cap B = \emptyset$.

In the IR context, finding an empirical relational structure is much more challenging, since it is not clear and straightforward how to determine an ordering between outputs of different systems. Moreover, some of the easiest notions, as the concatenation, does not have a plain translation in IR. For example, what does a concatenation of rankings mean? How is it possible to account for it in the symbolic relational structure? What is the meaning of summing the measure scores of different runs? The next section will give an insight about how we framework the problem, and Section 3.3 will present our solution and the limitations deriving from the lack of properties in the real world, i.e. the lack of a well defined ordering between runs and the lack of meaning of some sort of operations between runs, as for example the concatenation.

3.1.2 Our Framework

The core of our framework is to start individuating an empirical relational structure

$$\mathbf{E} = \langle IRS, \preceq \rangle$$

which allows us to compare and order different IR systems on the basis of the utility they provide to their users [Carterette, 2011; Cooper, 1973; Sakai, 2014a]. Clearly, being an empirical relational structure, it is assumed to exist in the real word, i.e. users have their own intuitive notion of when a system is better than another one. In Section 3.3 we will make this intuitive notion explicit, at least for the cases where it is possible to determine a common agreement about when a system is better than another one, thus leading to a partial ordering among systems.

We will then individuate a suitable symbolic relational structure $\mathbf{S} = \langle \overline{\mathbb{R}}_0^+, \leq \rangle$ with $\overline{\mathbb{R}}_0^+ = \mathbb{R}_0^+ \cup \{\infty\}$ and, in Section 3.4, we will provide a definition of IR utility-oriented measurement as a homomorphism between these two relational structures, i.e. we will provide a representation condition. We will also provide an equivalence theorem which allows us to easily verify the representation condition in terms of two simple properties, *swap* and *replacement*, i.e. to check in practice when an evaluation measure like AP or nDCG is actually a measurement in the previous sense.

Note that, according to the above definition, AP or nDCG should be called *measurement*, since they represent the homomorphism between the empirical and symbolic relational structures, while the actual numerical value computed by AP or nDCG for a given run and topic should be called *measure*. However, in the rest of this thesis we will use the term measure more frequently, both to refer to the measurement and to the actual measure value, since this is the common practice in IR.

Finally, we will also introduce the concept of *balancing* meant to explore the behaviour of a measurement when, in the empirical relational structure, the ordering between two systems is not a priori known. We will show that balancing accounts for the top heaviness of a measurement and we will conduct a preliminary experiment to validate the meaningfulness of its numerical value.

3.1.3 Related Work

The problem of grounding IR evaluation measures into a broader approach to measuring is a longstanding and crucial one [Fuhr, 2010]. C. J. van Rijsbergen was early pointing out the issues we encounter with IR evaluation measures [van Rijsbergen, 1981]:

> In the physical sciences there is usually an empirical ordering of the quantities we wish to measure [...] Such a situation does not hold for information retrieval. There is no empirical ordering for retrieval effectiveness and therefore any measure of retrieval effectiveness will by necessity be artificial.

We are not claiming to have fully addressed this hard problem in the present work, but rather to have started laying the foundations which can contribute to its solution. Moreover, to the best of our knowledge, this is the first attempt to systematically apply the representational theory of measurement in the context of IR evaluation.

A first approach to provide a formal framework for evaluation measures was proposed in [Amigó et al., 2009], which focuses on the evaluation of documents clustering rather than documents retrieval and ranking. Later, the same approach was extended in [Amigó et al., 2013], to include also IR measures.

In [Amigó et al., 2013], the authors frame the problem of documents retrieval and ranking in a broader context, called generic document organization problem, which in addition to document retrieval, embodies clustering and filtering. For each subclass of the document organization problem, a set of formal constraints is proposed. Each constraint is a verifiable property that any evaluation measures should satisfy to properly evaluate IR systems. Some examples of these constraints are the priority constraint and the closeness threshold constraint. The priority constraint states that moving a relevant document from a lower rank position to a higher rank position, or conversely moving a non relevant document from a higher rank position to a lower rank position, must increase the value of the measure. The closeness threshold constraint deals with the notion of top heaviness, claiming that there exists always a rank position n small enough, such that, retrieving a relevant document in the first position is worse than retrieving n non relevant documents followed by n relevant documents.

Similarly, [Moffat, 2013] characterizes some of the most commonly used IR evaluation measures according to seven numerical properties. For example, the converge property states that if a document, ranked after a threshold k, is swapped with a less relevant document, ranked before a threshold k, the measure score should increase strictly. Another property is called top-weightedness and says that if a document in the top k rank positions is swapped with a less relevant document at a higher rank position, the measure score should increase strictly.

Both [Amigó et al., 2013] and [Moffat, 2013] stated numerical properties and constraints that IR evaluation measures should comply on a case-by-case basis, e.g. when a system retrieves one more relevant document than another one, but they did not build up on an explicit relational structure among systems. Moreover, rather than proposing a framework that can embrace and define IR evaluation measures, they describe some properties to explain the differences between evaluation measures. Therefore, [Amigó et al., 2013] needed to build two new measures, reliability and sensitivity, able to verify all the proposed constraints. [Moffat, 2013] showed that the proposed properties are not compatible and it is not possible for an evaluation measure to satisfy all those properties simultaneously.

In [Busin and Mizzaro, 2013; Maddalena and Mizzaro, 2014] the notions of measure and measurement are used to propose a general definition of IR effectiveness measure and some axioms, that evaluation measures should verify, are presented. The proposed theoretical framework is based on two different quantities: the system relevance measurement, which is the automatic process that allows IR systems to assign a relevance score to each document, and the user relevance measurement, i.e. the relevance grade given by human assessors to each document. To gain good performances, IR systems should attempt to resemble human measurements as much as possible, i.e. maximize the similarity between system and user relevance measurements. Therefore, an effectiveness evaluation metrics is defined as a function that takes as input, in addition to the sets of documents and queries, the system and user measurements, and gives as output a numeric value.

Even if the authors used the definition of measure and measurement, they focus their framework on the notion of measurement scale [Fenton and Bieman, 2014; Stevens, 1946], which somehow comes after the definition of measurement. Indeed, they claim that system relevance measurements are on a ratio or absolute scale, while user relevance measurements are on an ordinal scale. This makes the definition of a similarity between measurements hard to be formulated. Here, we prefer to start from the definition of what IR utility-oriented measurements should be and we leave for future work a throughout study of the issues concerning the scales for such measurements.

Finally, [Bollman, 1984] sought for two axioms which allowed him to decide when an IR evaluation measure could be expressed as a linear combination of the number of relevant retrieved documents and the number of non relevant not retrieved documents, which is a different problem from the one of the present chapter.

3.2 Preliminary Definitions

We stem from [Angelini et al., 2014; Ferro et al., 2016b] for defining the basic concepts of topics, documents, ground-truth, run, and judged run. To the best of our knowledge, these basic concepts have not been explicitly defined in previous works [Amigó et al., 2013; Busin and Mizzaro, 2013; Maddalena and Mizzaro, 2014; Moffat, 2013].

Note that we need to define the same concepts for both set-based retrieval and rank-based retrieval and, to keep the notation compact and stress the similarities between these two cases, we will use the same symbols in both cases – e.g. r_t for run, $D(n)$ for set of documents retrieved by a run, $\mathscr{D}$ for universe set of documents and so on – being clear later on from the context whether we will refer to the set-based or rank-based version.

3.2.1 Topics, Documents, Ground-truth

Let us consider a set of **documents** D and a set of **topics** T; note that D and T are typically finite sets but we can account also for countable infinite ones.

Let $(REL, \preceq)$ be a totally ordered set of **relevance degrees**, i.e. they are defined on an ordinal scale [Stevens, 1946], where we assume the existence of a minimum, $\text{nr} = \min(REL)$, that we call the **non-relevant** relevance degree. Note that REL is typically a finite set but we can account also for an infinite one. In the former case, we can represent both binary relevance[1] $REL = \{\text{nr}, \text{r}\}$ (non relevant and relevant) and graded relevance [Kekäläinen and Järvelin, 2002], e.g. $REL = \{\text{nr}, \text{pr}, \text{hr}\}$ (non-relevant, partially relevant, highly relevant); in the latter case, we can represent both continuous relevance [Kekäläinen and Järvelin, 2002] and relevance assigned using unbounded scales, e.g. by using magnitude estimation [Maddalena et al., 2015, 2017]. Note that the definition of the REL set can accomplish both a notion of "immutable" relevance, as the one somehow adopted in evaluation campaigns, and a notion of relevance dependent on users and their context. In the latter case, we will have different REL sets corresponding to each user/context.

In the following, and without any loss of generality, we consider $REL \subseteq \mathbb{R}_0^+$ with the constraint that $0 \in REL$ and the order relation $\preceq$ becomes the usual ordering $\leq$ on real numbers, which ensures that a higher number corresponds to a higher relevance degree; the non-relevant degree is therefore given by $\min(REL) = 0$. Note that most of the algebraic operations we typically perform on numbers, like addition and multiplication, will be in general senseless on REL, since we take for granted only its order property. As above, this choice allows us to represent the most common cases, i.e. both binary relevance with

[1] Binary relevance is often thought to be on a categorical scale but, since the scale consists only of two categories one of which indicates the absence of relevance, we can safely consider it as an ordinal scale in fact.

$REL = \{0,1\}$ and graded relevance, either discrete with $REL \subseteq \mathbb{N}_0$ or continuous with $REL \subseteq \mathbb{R}_0^+$ in general.

For each pair $(t,d) \in T \times D$, the **ground-truth** GT is a map

$$GT : T \times D \to REL$$
$$(t,d) \mapsto rel$$

which assigns a relevance degree $rel \in REL$ to a document d with respect to a topic t. Note that, in the case of more complex situations like crowdsourcing for relevance assessment, we can define different GT maps, one for each crowd-worker. Moreover, we can extend the definition of ground truth and define a random variable that accounts for the source of randomness derived from the crowd assessors' judgements, as will be detailed in Chapter 4.

The **recall base** is the map RB from T into $\mathbb{N}$ defined as the total number of relevant documents for a given topic

$$t \mapsto RB_t = \big|\{d \in D : GT(t,d) > 0\}\big| \ .$$

The recall base is a quantity often hard to know in reality and, in some applications, it may be preferable to substitute it with a family of random variables $(t,\omega) \mapsto RB_t(\omega)$, which represents the unknown number of relevant documents present in the collection for every topic, that we will be able at most to estimate. For simplicity, in the sequel we will denote by RB_t the recall base in both the cases, omitting in the latter the dependence on ω.

3.2.2 Set-based Retrieval

Given a positive natural number N called the **length of the run**, we define the **set of retrieved documents** as

$$D(N) = \big\{\{d_1, \dots, d_N\} : d_i \in D\big\}$$

and the **universe set of retrieved documents** as $\mathscr{D} := \bigcup_{N=1}^{|D|} D(N) = 2^D$, which is the power set of D, i.e. the set of all the subsets of D.

A **run** r_t, retrieving a set of documents $D(N)$ in response to a topic $t \in T$, is a function from T into $\mathscr{D}$

$$t \mapsto r_t = \{d_1, \dots, d_N\} \ .$$

A multiset (or bag) is a set which may contain the same element several times and its multiplicity of occurrences is relevant [Knuth, 1981]. A **set of judged documents** is a (crisp) multiset $(REL,\ m) = \{rel_1, rel_2, rel_1, rel_2, rel_2, rel_4, \dots\}$, where m is a function from REL

into $\overline{\mathbb{N}}_0 = \mathbb{N}_0 \cup \{\infty\}$ representing the multiplicity of every relevance degree rel_j [Miyamoto, 2004]; if the multiplicity is 0, a given relevance degree is simply not present in the multiset, as in the case of rel_3 in the previous example. Suppose $\mathscr{M}$ is the infinite set of all the possible multiplicity functions m, then the **universe set of judged documents** is the set $\mathscr{R} := \bigcup_{m \in \mathscr{M}} (REL, m)$ of all the possible sets of judged documents (REL, m).

We call **judged run** the function $\hat{r}_t$ from $T \times \mathscr{D}$ into $\mathscr{R}$, which assigns a relevance degree to each retrieved document

$$(t, r_t) \mapsto \hat{r}_t = \big\{GT(t, d_1), \dots, GT(t, d_N)\big\} = \big\{\hat{r}_{t,1}, \dots, \hat{r}_{t,N}\big\}$$

Finally, we define the **ideal run** i_t for a given topic t as the run retrieving all the relevant documents, i.e. satisfying satisfying $N \geq RB_t$ and $\big|\{d \in i_t : GT(t,d) > 0\}\big| = RB_t$. In a similar way, we define the **worst run** w_t for a given topic t as the run not retrieving any relevant documents, i.e. satisfying $GT(t,d) = 0$ for any $d \in w_t$. Clearly, both of these runs are usually not unique.

3.2.3 Rank-based Retrieval

Given a positive natural number N called the **length of the run**, we define the **set of retrieved documents** as

$$D(n) = \{(d_1, \dots, d_N) : d_i \in D, d_i \neq d_j \text{ for any } i \neq j\} \ ,$$

i.e. the ranked list of retrieved documents without duplicates, and the **universe set of retrieved documents** as $\mathscr{D} := \bigcup_{N=1}^{|D|} D(N)$.

A **run** r_t, retrieving a ranked list of documents $D(N)$ in response to a topic $t \in T$, is a function from T into $\mathscr{D}$

$$t \mapsto r_t = (d_1, \dots, d_N)$$

We denote by $r_t[j]$ the j-th element of the vector r_t, i.e. $r_t[j] = d_j$. Note that, since the cardinality of D may be infinite, we can model also infinite rankings, as those assumed by [Moffat and Zobel, 2008; Webber et al., 2010]. We define the **universe set of judged documents** as $\mathscr{R} := \bigcup_{N=1}^{|D|} REL^N$.

We call **judged run** the function $\hat{r}_t$ from $T \times \mathscr{D}$ into $\mathscr{R}$, which assigns a relevance degree to each retrieved document in the ranked list

$$(t, r_t) \mapsto \hat{r}_t = \big(GT(t, d_1), \dots, GT(t, d_N)\big)$$

We denote by $\hat{r}_t[j]$ the j-th element of the vector $\hat{r}_t$, i.e. $\hat{r}_t[j] = GT(t, d_j)$.

We define the **ideal run** i_t for a given topic t as the run retrieving all the relevant documents in the top ranks and in decreasing order of relevance, i.e. satisfying $N \geq RB_t$ and $\hat{i}_t[j-1] \geq \hat{i}_t[j]$ for any $j \leq N$; note that it corresponds to the notion of perfect retrieval of [Egghe, 2008]. In a similar way, we define the **worst run** w_t for a given topic t as the run not retrieving any relevant documents, i.e. satisfying $\hat{w}_t[j] = 0$ for any j. Clearly both of these runs are not unique.

Finally, given a run r_t we define the set of the ranks of the relevant documents as $\mathscr{L} = \{j\colon\, j = 1, \ldots, N \text{ and } \hat{r}_t[j] > 0\}$, with cardinality $RR = |\mathscr{L}|$, which indicates the total number of relevant retrieved documents by the run for the given topic.

3.3 Empirical Relational Structure

As discussed in Section 3.1, a key point in defining a measurement is to start from a clear empirical relational structure among the attributes of the entities you would like to measure, in our case the effectiveness of IR systems in terms of the utility they provide to their users [Carterette, 2011; Cooper, 1973; Sakai, 2014a]. Therefore,

$$\mathbf{E} = \langle T \times \mathscr{D}, \preceq \rangle$$

is our empirical relational structure, i.e. the set of all the runs and an ordering relation between them, where the utility systems provide to their users is roughly expressed in terms of the "amount" of relevance: the more relevance is retrieved by a run, the greater it is.

This is an especially critical point since, as highlighted out by [van Rijsbergen, 1981], "there is no empirical ordering for retrieval effectiveness". The hardness of this problem clearly emerges also when you consider the actual properties of the set $\mathscr{D}$. Typically, when you define a measurement, you start from sets having very good properties. For example, in the case of the theory of measure [Billingsley, 1995; Folland, 1999], the standard setting is represented by σ-algebras.

Recall that a σ-algebra of a set X is a collection of subsets of X which satisfies the following conditions:

1. $\emptyset \in \mathscr{F}$;
2. if $A \in \mathscr{F}$ then $A^c \in \mathscr{F}$;
3. if $A_1, A_2, \ldots$ is a countable collection of sets in $\mathscr{F}$, then $\bigcup_{i=1}^{\infty} A_i$ is in $\mathscr{F}$;

Therefore, σ-algebras are closed under countable unions, intersections, and complements and the inclusion relation among sets leads to a natural partial ordering. All these nice properties are then reflected in measures and probabilities: since a σ-algebra is closed under countable union, a measure is then requested to be σ-*additive*, i.e. if $\{A_n\}_{n\in\mathbb{N}}$ is a family of disjoints subsets, then $M(\bigcup_{n\in\mathbb{N}} A_n) = \sum_{n\in\mathbb{N}} M(A_n)$ and from this property one obtains that is also *monotone* $A \subseteq B \Rightarrow M(A) \leq M(B)$, since $B = A \cup (A^C \cap B) \Rightarrow M(B) = M(A) + M\big(A \cup (A^C \cap B)\big) \geq M(A)$, which in turn reflects the ordering induced by the inclusion relation on the σ-algebra.

Unfortunately, the set $\mathscr{D}$ lacks many of these desirable properties. For example, union and inclusion on $\mathscr{D}$ would not be as intuitive and agreeable as they are in the case of σ-algebras and this hampers the possibility of requiring additivity or monotonicity as a properties of an IR utility-oriented measurement.

Let us consider inclusion: we could say that $r_t \subseteq s_t$ if s_t appends one more document to r_t. Differently from σ-algebras, inclusion would not induce an ordering on $\mathscr{D}$, since you may think that a run retrieving one more relevant document is greater than another one not retrieving it [Amigó et al., 2013; Moffat, 2013], but you may also think that a run retrieving one more not-relevant document is smaller than another one not retrieving it [Amigó et al., 2013], or it should stay equal [Moffat, 2013].

The above inclusion can be seen also as a form of union, i.e. as concatenating a run with another one constituted by just a single document, i.e. somehow $s_t = r_t \cup \{d_j\}$. Almost no one would require additivity, i.e. $M(s_t) = M(r_t) + M(d_j)$, and as discussed above there is neither agreement on monotonicity, i.e. when it should be $M(s_t) > M(r_t)$ and when $M(s_t) < M(r_t)$. This is even more evident if you think at data fusion, a kind of much more complicated union: no one would quest for additivity, even in the case of runs without any common document, and consider the performance of the fused run as the sum of the performances of the composing runs, nor they could a priori guarantee monotonicity, ensuring that the performance of the fused run is always greater than or equal to the the performances of the composing runs.

The above mentioned issues with inclusion and union of runs make it difficult also to deal with runs of different length, e.g. constraining the behaviour of a measurement in the symbolic relational structure **S** when runs of different length are somehow contrasted, as it is done in [Amigó et al., 2013; Busin and Mizzaro, 2013; Maddalena and Mizzaro, 2014; Moffat, 2013], since we basically do not know how to unite and compare them in the empirical relational structure **E**.

Therefore, in this chapter, we will focus on a partial ordering among runs of the same length in the empirical relational structure **E**, leading to monotonicity in the symbolic relational structure **S**, and we leave for future work a deeper investigation of inclusion, union, additivity and their implications. In particular, we will restrict ourselves only to those cases where the ordering is intuitive and it is possible to find a commonly shared agreement. Examples of very basic cases are: a run retrieving a relevant document in the first rank position is greater than another one retrieving it in the second position, or a run retrieving a more relevant document in a given rank position is greater than another one retrieving a less relevant document in the same position.

The above discussion points out one key contribution of this chapter, i.e. highlighting that the core problem in defining an IR measurement is not to constraint its numerical properties (symbolic world), but rather our quite limited understanding of the operations and relationships among runs (empirical world). Indeed, if we better clarify how runs behave in the empirical relational structure, a measurement, intended as a homomorphism between the empirical and symbolic worlds, has to comply with them by construction.

Note that this vision is somehow implicitly present in [Busin and Mizzaro, 2013; Maddalena and Mizzaro, 2014]. Their framework is based on the idea that there must be an agreement between two distinct "relevance measurements", one made by assessors and the other by systems, i.e. how assessors and systems rank documents on the basis of their relevance to a query. Then, they constrain what they call "metric" to the behaviour of the similarity between these two "relevance measurements", but without actually defining what this similarity is. In relation to our work, we could say that the assessor and system "relevance measurements" may somehow resemble the notion of relational structures in the empirical world and the "metric" may in some way approximate the notion of measurement as homomorphism between empirical and symbolic worlds. However, we think that framing the problem in the context of the representational theory of measurement provides more advantages than an ad-hoc approach: it streamlines the core concepts, helps to discuss and address issues at the proper level, either in the empirical or symbolic worlds, and better links IR evaluation to other sciences. Moreover, we provide an actual partial ordering among runs in the empirical world, from which we derive properties for a measurement, while the concept of similarity is not actually defined by [Busin and Mizzaro, 2013; Maddalena and Mizzaro, 2014].

3.3.1 Set-based Retrieval

Let us consider two runs r_t and s_t with the same length N. We introduce a **partial ordering among runs** as

$$r_t \preceq s_t \;\Leftrightarrow \left|\{j : \hat{r}_{t,j} \geq rel\}\right| \leq \left|\{j : \hat{s}_{t,j} \geq rel\}\right| \;\; \forall rel \in REL$$

	≥ 0	≥ 1	≥ 2	≥ 3
$\hat{r}_t = \{0, 1, 1, 2, 2\}$	5	4	2	0
	$\parallel$	$\parallel$	$\parallel$	$\wedge$
$\hat{s}_t = \{0, 1, 1, 2, 3\}$	5	4	2	1

(a) Example of comparable runs.

	≥ 0	≥ 1	≥ 2	≥ 3
$\hat{r}_t = \{0, 1, 1, 2, 2\}$	5	4	2	0
	$\parallel$	$\parallel$	$\vee$	$\wedge$
$\hat{w}_t = \{0, 1, 1, 1, 3\}$	5	4	1	1

(b) Example of not comparable runs.

Fig. 3.1 Example of comparison between set-based runs: the runs in Figure 3.1a are comparable, with $r_t \preceq s_t$, since the pointwise comparison of the vectors does not present any inversion, while in Figure 3.1b it is not clear whether r_t is better than w_t or vice versa, as shown by the inversions of the ordering relation.

which counts, for each relevance degree, how many items there are above that relevance degree and, if a run has higher counts for each relevance degree, it is considered greater than another one.

An easy way to compare two runs and determine if they are comparable, and in the positive case determine which run is the smaller or greater, is illustrated in Figure 3.1. For each relevance grade, we can count the number of documents with relevance greater or equal to the fixed grade and store the output in a vector. Then we can compare the two resulting vectors, if each entry of one vector is greater or equal (alternatively smaller or equal) than each entry of the other vector, then we can conclude that the two runs are comparable and that one run is greater (or smaller) than the other run. Whenever we find an inversion of the ordering between the vector entries, as for example in Figure 3.1b the third and fourth table entries, we can conclude that the two runs are not comparable. In section 3.3.2 we will see that this graphical comparison can be generalized to rank based retrieval with matrices instead of vectors.

For example, if we have four relevance degrees $REL = \{0, 1, 2, 3\}$, the run $\hat{r}_t = \{0, 1, 1, 2, 2\}$ is smaller than the run $\hat{s}_t = \{0, 1, 1, 2, 3\}$, as illustrated in Figure 3.1a. While if we consider the run $\hat{r}_t = \{0,1,1,2,2\}$ and the run $\hat{w}_t = \{0,1,1,1,3\}$ in Figure 3.1b, they are not comparable since, relying just on an ordinal scale for the relevance degrees, it is not a priori known whether the decrease from a document with relevance degree 2 to one with relevance degree 1 is compensated or not by the increase from a document with relevance degree 2 to one with relevance degree 3, actually we cannot even say if the two runs are equal.

If we have the relevance grades $REL = \{0,1,\cdots,q\}$, among all the runs with a fixed number of relevant documents, the run $\{1,\ldots,1,0,\ldots,0\}$ is the smallest, while $\{q,\ldots,q,0,\ldots,0\}$ is the greatest one.

In the case of binary relevance, i.e. $REL = \{0,1\}$, we obtain an intuitive total ordering

$$r_t \preceq s_t \;\Leftrightarrow \big|\{j : \hat{r}_{t,j} \geq 1\}\big| \leq \big|\{j : \hat{s}_{t,j} \geq 1\}\big|$$

where r_t is less than s_t if it retrieved less relevant documents than s_t.

If REL relies on a more powerful scale, e.g. a ratio scale where we can know, for example, that a highly relevant document is twice as relevant as a partially relevant one, the above definition becomes a total ordering also in the case of graded relevance, by basically summing up how many “relevance units” there are in each run.

For example, assume that a document with relevance degree 2 is twice more relevant than a document with relevance degree 1, and similarly a document with relevance degree 3 is three times more relevant then a document with relevance degree 1. Then, if you consider the examples in Figure 3.1b, r_t and w_t become comparable and the result is that they are equal, because the decrease from a document with relevance degree 2 to one with relevance degree 1 is perfectly balanced by the increase from a document with relevance degree 2 to one with relevance degree 3. Indeed both these transactions have value equal to 2 in terms of relevance unit.

3.3.2 Rank-based Retrieval

We now proceed to define an order relation when the output of an IRS is a ranking instead of a set. If for set-based retrieval the major problem in defining a total ordering relation is represented by the relevance grades, which are not on a ratio scale, for rank-based retrieval there is an additional source of complexity represented by the rank position. Therefore, when defining an ordering relation we can not just account for the amount of relevance, but we have also to consider the rank position where relevant document are placed.

Let us consider two runs r_t and s_t with the same length N. We introduce a **partial ordering among runs** as

$$r_t \preceq s_t \;\Leftrightarrow \big|\{j \leq k : \hat{r}_t[j] \geq rel\}\big| \leq \big|\{j \leq k : \hat{s}_t[k] \geq rel\}\big|$$
$$\forall rel \in REL \;\text{ and }\; k \in \{1,\ldots,N\}$$

which counts, for each relevance degree and rank position, how many items there are above that relevance degree and, if a run has higher counts for each relevance degree and rank position, it is considered greater than another one. You might notice that this formulation of partial ordering is much more complex than the formulation for set-based retrieval, which is due to the additional complexity introduced by the ranking.

Analogously to the set-based case, we can determine whether two runs are comparable or not by performing a comparison between matrices, as shown in Figure 3.2. Each run can be associated with a matrix, whose rows represent the rank positions and whose columns represent the relevance grades. Therefore, the (i, j)-entry of the matrix stores the number of documents with relevance grade $\geq j$ from rank position 1 to rank position i. Once the matrices are computed, we can compare them pointwise: if all the entries of one matrix are greater or equal (alternatively smaller or equal) than the entries of the other matrix, then we can conclude that its corresponding run is greater (or smaller) than the other run, as in Figure 3.2a where all the entries of the two matrices are equal except of $(5,4)$, which ensures that r_t is smaller than s_t. Whenever this condition is not satisfied and there is an inversion, in Figure 3.2b this happens for the entries $(4,3), (5,3)$ and $(5,4)$, then the two runs are not comparable.

For example, if we have four relevance degrees $REL = \{0,1,2,3\}$, the run $\hat{r}_t = (0,1,1,2,2)$ is smaller than the run $\hat{s}_t = (0,1,1,2,3)$, as illustrated in Figure 3.2a. While if we consider the the run $\hat{r}_t = (0,1,1,2,2)$ and the run $\hat{w}_t = (0,1,1,1,3)$ in Figure 3.2b they are not comparable since, relying just on an ordinal scale for the relevance degrees, it is not a priori known whether the decrease from a document with relevance degree 2 to one with relevance degree 1 at rank 4 is compensated or not by the increase from a document with relevance degree 2 to one with relevance degree 3 at rank 5, as it happens in the set-based retrieval case. If we overlook the ranking, in this example the complexity is given by the relevance degrees which are not on a ratio scale.

On the other hand, the run $\hat{r}_t = (0,1,1,2,2)$ is not comparable even with the run $\hat{v}_t = (2,0,1,2,1)$ because, even if the document with relevance degree 2 moves forward from rank 5 to rank 1, the backward movement of the document with relevance degree 1 from rank 2 to rank 5 may or may not compensate for it. This latter case points out the effect of ranking

$\hat{r}_t$
0
1
1
2
2

Relevance Grade

Rank	≥ 0	≥ 1	≥ 2	≥ 3
≤ 1	1	0	0	0
≤ 2	2	1	0	0
≤ 3	3	2	0	0
≤ 4	4	3	1	0
≤ 5	5	4	2	0

$\hat{s}_t$
0
1
1
2
3

Relevance Grade

Rank	≥ 0	≥ 1	≥ 2	≥ 3
≤ 1	1	0	0	0
≤ 2	2	1	0	0
≤ 3	3	2	0	0
≤ 4	4	3	1	0
≤ 5	5	4	2	1

Relevance Grade

Rank	≥ 0	≥ 1	≥ 2	≥ 3
≤ 1	1 1	0 0	0 0	0 0
≤ 2	2 2	1 1	0 0	0 0
≤ 3	3 3	2 2	0 0	0 0
≤ 4	4 4	3 3	1 1	0 0
≤ 5	5 5	4 4	2 2	0 < 1

(a) Example of comparable runs.

$\hat{r}_t$
0
1
1
2
2

Relevance Grade

Rank	≥ 0	≥ 1	≥ 2	≥ 3
≤ 1	1	0	0	0
≤ 2	2	1	0	0
≤ 3	3	2	0	0
≤ 4	4	3	1	0
≤ 5	5	4	2	0

$\hat{w}_t$
0
1
1
1
3

Relevance Grade

Rank	≥ 0	≥ 1	≥ 2	≥ 3
≤ 1	1	0	0	0
≤ 2	2	1	0	0
≤ 3	3	2	0	0
≤ 4	4	3	0	0
≤ 5	5	4	1	1

Relevance Grade

Rank	≥ 0	≥ 1	≥ 2	≥ 3
≤ 1	1 1	0 0	0 0	0 0
≤ 2	2 2	1 1	0 0	0 0
≤ 3	3 3	2 2	0 0	0 0
≤ 4	4 4	3 3	1 > 0	0 0
≤ 5	5 5	4 4	2 > 1	0 < 1

(b) Example of not comparable runs.

Fig. 3.2 Example of comparison between rank-based runs: the two runs in Figure 3.2a are comparable, with $r_t \preceq s_t$, because the pointwise comparison of the matrices does not present any inversion, while in Figure 3.2b it is not clear whether r_t is better than w_t or vice versa, as shown by the bottom right corner of the comparison matrix.

with respect to the previous case of set-based retrieval, which would have considered these two runs as equal.

Notice that, in rank-based retrieval, we cannot achieve a total ordering even when we assume that the relevance degrees are on a ratio scale, differently from the set-based retrieval. Indeed, we have to account for rank positions, and having different amounts of relevance units in different positions can not be directly compared as in the set-based retrieval. As before, assume that a document with relevance degree 2 is twice more relevant than a document with relevance degree 1, and similarly a document with relevance degree 3 is three times more relevant then a document with relevance degree 1. Then, the run $\hat{r}_t = (0,2,1,3,0)$ is not comparable to the run $\hat{s}_t = (3,0,1,0,2)$ because you cannot a priori say whether the forward movement of the 3 relevance units from rank 3 to rank 1 is compensated or not by the backward movement of 2 relevance units from rank 2 to rank 5.

Moreover, even if we remove the complexity related to the relevance degrees and we consider just the case of binary relevance, still we cannot achieve a total ordering. For example, the run $\hat{r}_t = (0,1,0,1,0)$ is not comparable to the run $\hat{s}_t = (1,0,0,0,1)$ because you cannot a priori say whether the forward movement of the relevant document from rank 2 to rank 1 is compensated or not by the backward movement of the relevant document from rank 4 to rank 5.

A possible segmentation of all the runs can be performed in terms of the total number of relevant documents, where a minimum and maximum run can be found. Taking for simplicity $REL = \{0,1,\dots,q\}$ and considering a run r_t, always retrieving just one relevant document, we have that it lays between the minimum and maximum below:

$$(0,\dots,0,1) \preceq \hat{r}_t \preceq (q,0,\dots,0)$$

More in general, for any run r_t retrieving k relevant documents, it holds:

$$(0,\dots,0,1,\dots,1) \preceq \hat{r}_t \preceq (q,\dots,q,0,\dots,0) \tag{3.1}$$

Summing up, differently from the case of set-based retrieval, this partial ordering cannot become a total order, neither in the case of binary relevance, nor in the case of relevance degrees on more powerful scales, e.g. ratio ones. Indeed, the presence of the ranking adds a further dimension which makes impossible to compare every run pair because it is not a priori known how much each rank position influences the ordering.

3.4 Utility-oriented Measurements of Retrieval Effectiveness

We define a **utility-oriented measurement of retrieval effectiveness** as a homomorphism between the empirical relational structure $\mathbf{E} = \langle T \times \mathscr{D}, \preceq \rangle$, discussed in the previous section, and the symbolic relational structure $\mathbf{S} = \langle \overline{\mathbb{R}}_0^+, \leq \rangle$, that is a mapping which assigns to any set or sequence of documents $D(N)$ retrieved by a system for a given topic t, a non negative number, i.e. a **utility-oriented measure of retrieval effectiveness**.

More in detail, a utility-oriented measurement of retrieval effectiveness is the composition of a judged run $\hat{r}_t$ with a **scoring function** μ from the universe set of judged documents $\mathscr{R}$ into $\overline{\mathbb{R}}_0^+$ which assigns to any set or sequence of judged documents a non negative number, ensuring that the ordering $\preceq$ among the runs is properly mapped in the ordering $\leq$ among real numbers.

Definition 1. A function

$$M : T \times \mathscr{D} \rightarrow \overline{\mathbb{R}}_0^+$$

defined as $M = \mu(\hat{r}_t)$, i.e. the composition of a judged run $\hat{r}_t$ with a scoring function $\mu : \mathscr{R} \rightarrow \overline{\mathbb{R}}_0^+$ is a **utility-oriented measurement of retrieval effectiveness** if and only if for any two runs r_t and s_t with the same length N such that $r_t \preceq s_t$, then $\mu(\hat{r}_t) \leq \mu(\hat{s}_t)$.

Any utility-oriented measurement of retrieval effectiveness is indeed the specification of the scoring function μ and the property which ensures a proper mapping between the empirical and symbolic relational structures is the *monotonicity* of μ. In this respect, a utility-oriented measurement of retrieval effectiveness is not a "measure" in the classical sense of the measure theory [Billingsley, 1995; Folland, 1999], since it lacks the additivity property, but shares with fuzzy measures [Wang and Klir, 1992] the fact of relying just on monotonicity.

Note that the monotonicity requested in the definition above differs from the notion of monotonicity in [Moffat, 2013], since this latter one applies to runs of different length, which is not our case for the motivations we discussed in the previous section. Similar considerations hold for the notion of document/query monotonicity in [Maddalena and Mizzaro, 2014] which applies to unions of documents/queries.

Even if the previous definition fits our purposes, it could be difficult to check it in practice. Therefore, we introduce two "monotonicity-like" properties, called **replacement** and **swap**, which we will prove to be equivalent to the required monotonicity, but easier to check.

Replacement If we replace a less relevant document with a more relevant one in the same rank position, a utility-oriented measurement of retrieval effectiveness should not decrease. More formally, if

$$r_t = (d_1, \dots, d_{i-1}, \mathbf{d_i}, d_{i+1}, \dots, d_N)$$

and

$$s_t = (d_1, \dots, d_{i-1}, \tilde{\mathbf{d}}_\mathbf{i}, d_{i+1}, \dots, d_N)$$

with $\mathbf{d_i} \neq \tilde{\mathbf{d}}_\mathbf{i}$ and $\hat{r}_t[i] \leq \hat{s}_t[i]$, then

$$M(r_t) \leq M(s_t)$$

For example, if we consider $\hat{r}_t = (3,1,1,0,0)$ by replacing the document at rank 2 with a more relevant document we can obtain $\hat{s}_t = (3,2,1,0,0)$. With the replacement property we claim that the measure score of r_t should be less or equal than the measure score of s_t.

Swap If we swap a less relevant document in a higher rank position with a more relevant one in a lower rank position, a utility-oriented measurement of retrieval effectiveness should not decrease. More formally, if

$$r_t = (d_1, \dots, d_{i-1}, \mathbf{d_i}, d_{i+1}, \dots, d_{j-1}, \mathbf{d_j}, d_{j+1}, \dots, d_N)$$

and

$$s_t = (d_1, \dots, d_{i-1}, \mathbf{d_j}, d_{i+1}, \dots, d_{j-1}, \mathbf{d_i}, d_{j+1}, \dots, d_N)$$

with $\hat{r}[i] \leq \hat{r}[j]$, then

$$M(r_t) \leq M(s_t)$$

Consider $\hat{r}_t = (0,1,1,0,3)$ as an example, and switch the document at rank 1 with a the more relevant document at rank 5. The new run that we obtain is $\hat{s}_t = (3,1,1,0,0)$, which has a measure score greater than or equal to r_t, as stated by the swap property.

The above definitions of replacement and swap are formulated in the case of rank-based retrieval; clearly, for set-based retrieval only replacement makes sense, while swap does not apply since there is no ranking among documents.

Note that the swap property somehow recalls the idea of priority constraint in [Amigó et al., 2013] and of convergence in [Moffat, 2013].

Theorem 1 (Equivalence)**.** *A scoring function μ defined from $\mathcal{R}$ into $\overline{\mathbb{R}}_0^+$ leads to a utility-oriented measurement of retrieval effectiveness M if and only if it satisfies the Replacement and the Swap properties.*

Proof. If μ leads to a utility-oriented measurement of retrieval effectiveness, the Replacement property is clearly a special case of the monotonicity of μ.

Let us now define

$$A(r_t,k,p) = |\{i \leq k : \hat{r}_t[i] \geq p\}|$$

and assume that

$$r_t = (d_1,\dots,d_{i-1},\mathbf{d_i},d_{i+1},\dots,d_{j-1},\mathbf{d_j},d_{j+1},\dots,d_N)$$

and

$$s_t = (d_1,\dots,d_{i-1},\mathbf{d_j},d_{i+1},\dots,d_{j-1},\mathbf{d_i},d_{j+1},\dots,d_N)$$

with $\hat{r}_t[i] \leq \hat{r}_t[j]$.

It is clear that $A(r_t,k,p) = A(s_t,k,p)$ for any $k \leq i-1$ and $p \in \mathbb{R}_0^+$. If $k = i, i+1,\dots,j-1$, we have $A(r_t,k,p) = A(s_t,k,p)$ for $p < \hat{r}_t[j]$ and $A(r_t,k,p) < A(s_t,k,p)$ for $p \geq \hat{r}_t[j]$, while for $k > j$ again $A(r_t,k,p) = A(s_t,k,p)$ for any $p \in \mathbb{R}_0^+$. This implies that $r_t \preceq s_t$: by the monotonicity we get that $\mu(\hat{r}_t) \leq \mu(\hat{s}_t)$ and the Swap property is proved.

Let us now assume that the Replacement and the Swap properties are satisfied by M. Taken $r_t \preceq s_t$, our aim is to prove that we are able to construct an increasing sequence of runs

$$r_t = r_t^0 \preceq r_t^1 \preceq r_t^2 \preceq \ldots \preceq r_t^h = s_t$$

such that $\mu(\hat{r}_t^j) \leq \mu(\hat{r}_t^{j+1})$ for any $j = 0,\dots,h-1$, which proves the monotonicity of μ. Let us start from the last term in both the collections of judged runs. If $\hat{r}_t[N] = \hat{s}_t[N]$, we define $r_t^1 = r_t$ and pass to the $N-1$-th element. If $\hat{r}_t[N] < \hat{s}_t[N]$, we replace the last document in r_t with a document of relevance degree $\hat{s}_t[N]$ and define this new run as r_t^1. We have that $r_t^0 = r_t \preceq r_t^1$, by the replacement that $\mu(\hat{r}_t^0) \leq \mu(\hat{r}_t^1)$ and we pass to consider the $N-1$-th element. If $\hat{r}_t[N] > \hat{s}_t[N]$, we swap the last document in r_t with the closest document of minimum relevance grade of the same run. For example, if

$$\hat{r}_t = (1,0,1,0,1,1) \text{ and } \hat{s}_t = (1,1,0,1,1,0)$$

we define $\hat{r}_t^1 = (1,0,1,1,1,0)$. It is immediate to see that the new last element of r_t has a relevance degree smaller than or equal to $\hat{s}_t[N]$. Indeed, if on the contrary we assume that $\hat{r}_t[k] > \hat{s}_t[N]$ for any $k < N$ and we define $p = \min\{\hat{r}_t[i], 0 \leq i \leq N\}$, we have that

$$A(r_t,N,p) > A(s_t,N,p)$$

which is in contradiction with the hypothesis that $r_t \preceq s_t$. We have that $r_t^0 = r_t \preceq r_t^1$ and by the swap property that $\mu(\hat{r}_t^0) \leq \mu(\hat{r}_t^1)$. Proceeding now as before in the case that $\hat{r}_t^1[N] = \hat{s}_t[N]$ or $\hat{r}_t^1[N] < \hat{s}_t[N]$, we (possibly) define a new run r_t^2 such that $r_t^1 \preceq r_t^2$ and we pass to consider the $N-1$-th element. Repeating this procedure to the $N-1$-th element, the $N-2$-th element and so on we construct the desired sequence of runs and the monotonicity is proved. □

The same theorem can be proved in the case of set-based retrieval by using just the Replacement property.

As a final remark, note that for any two runs r_t and s_t such that $r_t \preceq s_t$, Definition 1 ensures that any two utility-oriented measurements M_1 and M_2 will order r_t below s_t, i.e. $M_1(r_t) \leq M_1(s_t)$ and $M_2(r_t) \leq M_2(s_t)$. On the contrary, when two runs are not comparable, i.e. when they are outside the partial ordering $\preceq$ and we cannot say which one is greater, we can find two utility-oriented measurements M_1 and M_2 which order them differently.

Consider, for example the following runs

$$r_t = (1,0,0,1,0) \text{ and } s_t = (0,1,1,0,1),$$

We obtain that

$$Prec(r_t)[5] = \frac{2}{5} < Prec(s_t)[5] = \frac{3}{5}$$

while

$$AP(r_t) = \frac{1}{RB_t}\frac{3}{2} > AP(s_t) = \frac{1}{RB_t}\frac{53}{30}$$

Therefore, Precision judges preferable s_t, while AP r_t.

3.4.1 Examples of Application of the Equivalence Theorem

In this section, we use the equivalence Theorem 1 to show how to demonstrate that an existing IR evaluation measure is an utility-oriented measurements of retrieval effectiveness, i.e. we will show that all these measures satisfy the replacement and swap conditions.

The proof is trivial in the case of Average Precision (AP), Rank-Biased Precision (RBP) [Moffat and Zobel, 2008], and Normalized Discounted Cumulated Gain (nDCG) [Järvelin and Kekäläinen, 2002] and not reported here. Here, we present the case of Expected Reciprocal Rank (ERR) [Chapelle et al., 2009], which is more interesting.

Average Precision (AP)

Given a run r_t of length N, recall the definition of AP in Equation (2.1):

$$AP(r_t) = \frac{1}{RB} \sum_{k=1}^{N} \mathrm{Prec}(r_t)[k] \cdot \hat{r}_t[k] \ ,$$

where $\hat{r}_t[k] = 1$ if the document at rank k is relevant.

Let us consider the *replacement property* with two runs r_t and s_t, where s_t is generated from r_t by replacing the document at position i. To avoid trivial cases, assume that $\hat{r}_t[i] < \hat{s}_t[i]$, i.e. the document at rank i is replaced with a more relevant one, $\hat{r}_t[i] = 0$ and $\hat{s}_t[i] = 1$.

If we consider just precision, we have that:

$$\mathrm{Prec}(r_t)[k] = \mathrm{Prec}(s_t)[k] \quad \text{for } k < i \ , \tag{3.2}$$

$$\mathrm{Prec}(r_t)[k] < \mathrm{Prec}(s_t)[k] \quad \text{for } k \geq i \ . \tag{3.3}$$

Therefore, if we compute the difference $AP(r_t) - AP(s_t)$ we obtain:

$$\begin{aligned} AP(r_t) - AP(s_t) &= \frac{1}{RB} \sum_{k=1}^{N} \left(\mathrm{Prec}(r_t)[k] \cdot \hat{r}_t[k] - \mathrm{Prec}(s_t)[k] \cdot \hat{s}_t[k]\right) \\ &= \frac{1}{RB} \left(-\mathrm{Prec}(s_t)[i] + \sum_{k=i+1}^{N} \left(\mathrm{Prec}(r_t)[k] - \mathrm{Prec}(s_t)[k]\right) \hat{r}_t[k] \right) \\ &< 0 \end{aligned}$$

where $\hat{r}_t[k] = \hat{s}_t[k]$ for $k \neq i$ and the first passage is justified by Equation (3.2) and the last inequality holds, since all the terms are negative, as shown by Equation (3.3).

We proceed with the *swap property*, consider again two runs r_t and s_t, where s_t is obtained from r_t by swapping the document at position i with the document at position j, with $i < j$. To avoid trivial cases, assume that $\hat{r}_t[i] < \hat{r}_t[j]$, i.e. the document at rank i is swapped with a more relevant one in a lower rank position, $\hat{r}_t[i] = 0$ and $\hat{r}_t[j] = 1$.

As before, we first consider just precision:

$$\mathrm{Prec}(r_t)[k] = \mathrm{Prec}(s_t)[k] \quad \text{for } k < i \ , \tag{3.4}$$

$$\mathrm{Prec}(r_t)[k] < \mathrm{Prec}(s_t)[k] \quad \text{for } i \leq k < j \ , \tag{3.5}$$

$$\text{Prec}(r_t)[k] = \text{Prec}(s_t)[k] \quad \text{for } k \geq j \,, \tag{3.6}$$

and we compute the difference $AP(r_t) - AP(s_t)$:

$$\begin{aligned} AP(r_t) - AP(s_t) &= \frac{1}{RB} \sum_{k=1}^{N} (\text{Prec}(r_t)[k] \cdot \hat{r}_t[k] - \text{Prec}(s_t)[k] \cdot \hat{s}_t[k]) \\ &= \frac{1}{RB} \left(\text{Prec}(r_t)[i] - \text{Prec}(s_t)[j] + \sum_{k=i+1}^{j-1} \underbrace{(\text{Prec}(r_t)[k] - \text{Prec}(s_t)[k])}_{<0} \hat{r}_t[k] \right) \end{aligned}$$

where the first equality is given by $\hat{r}_t[k] = \hat{s}_t[k]$ for $k \neq i, j$ and Equations (3.4) and (3.6). In the second equality, all the term of the sum are negative, thanks to Equations (3.5), therefore we just need to prove that $\text{Prec}(r_t)[i] - \text{Prec}(s_t)[j] < 0$. By the definition of precision, if we move the cutoff rank to a lower position the score of the measure should not decrease, $\text{Prec}(r_t)[i] \leq \text{Prec}(r_t)[j]$ thus

$$\text{Prec}(r_t)[i] - \text{Prec}(s_t)[j] \leq \text{Prec}(r_t)[j] - \text{Prec}(s_t)[j] = 0 \,.$$

Rank-Biased Precision (RBP)

Given a run r_t of length N, recall the definition of RBP in Equation (2.3.2):

$$RBP(r_t) = \sum_{k=1}^{N} p^{k-1} \hat{r}_t[k] \,,$$

where $\hat{r}_t[k] = 1$ if the document at rank k is relevant and $p \in [0,1]$ represents the persistence of the user.

Let us consider the *replacement property* with two runs r_t and s_t, where s_t is obtained with the replacement from r_t as in the previous case. We can rewrite RBP as follows:

$$RBP(r_t) = (1-p) \underbrace{\sum_{k=1,\, k \neq i}^{N} p^{k-1} \hat{r}_t[k]}_{A(p)} + p^{i-1} \hat{r}_t[i] \,.$$

Therefore, if we compute the difference $RBP(r_t) - RBP(s_t)$ we have

$$\begin{aligned} RBP(r_t) - RBP(s_t) &= (1-p)\,\big(A(p) + p^{i-1}\hat{r}_t[i] - \big(A(p) + p^{i-1}\hat{s}_t[i]\big)\big) \\ &= (1-p)\,\big(p^{i-1}(\hat{r}_t[i] - \hat{s}_t[i])\big) \qquad\qquad < 0 \end{aligned}$$

where the last inequality holds because $\hat{r}_t[i] < \hat{s}_t[i]$. To prove the *replacement property* we can proceed similarly and reformulate RBP as:

$$RBP(r_t) = (1-p)\Big(\underbrace{\sum_{k=1,\ k\neq\{i,j\}}^{N} p^{k-1}\hat{r}_t[k]}_{D(p)} + p^{i-1}\hat{r}_t[i] + p^{j-1}\hat{r}_t[j]\Big)\ ,$$

therefore when we consider the difference $RBP(r_t) - RBP(s_t)$, where s_t is generated by r_t by swapping the document at rank position i with the document at rank position j. Then we obtain

$$\begin{aligned} RBP(r_t) - RBP(s_t) &= (1-p)\Big(p^{i-1}(\hat{r}_t[i] - \hat{s}_t[i]) + p^{j-1}(\hat{r}_t[j] - \hat{s}_t[j])\Big) \\ &= (1-p)\Big(p^{i-1}(\hat{r}_t[i] - \hat{r}_t[j]) - p^{j-1}(\hat{r}_t[i] - \hat{r}_t[j])\Big) \\ &= (1-p)\Big((p^{i-1} - p^{j-1})(\hat{r}_t[i] - \hat{r}_t[j])\Big) \\ &< 0 \end{aligned}$$

where the last inequality is verified since $p^{i-1} - p^{j-1} > 0$ and $\hat{r}_t[i] < \hat{r}_t[j]$.

Normalized Discounted Cumulated Gain (nDCG)

Given a run r_t of length N, recall the definition of nDCG in Equation (2.2):

$$\text{nDCG}(r_t) = \frac{\text{DCG}(r_t)}{\text{DCG}(i_t)}\ , \qquad \text{and} \qquad \text{DCG}(r_t) = \sum_{k=1}^{N} \frac{\hat{r}_t[k]}{dsc(k,b)}$$

where i_t is the ideal run, $\hat{r}_t[k]$ is the relevance weight and $disc(k,b)$ is the discount function for the position k, specifically $dsc(k,b) = \max\{1, \log_b(k)\}$ in the original definition of nDCG [Järvelin and Kekäläinen, 2002], while $dsc(k,2) = \log_2(k+1)$ for Microsoft nDCG [Burges et al., 2005].

Note that the ideal run depends on the topic and not on the run, therefore to prove that nDCG satisfies both the replacement and swap properties we can consider DCG instead of nDCG. Indeed, if we consider two runs r_t and s_t and we want to compare nDCG(r_t) and

$\text{nDCG}(s_t)$ we have

$$\text{nDCG}(r_t) < \text{nDCG}(s_t) \iff \frac{\text{DCG}(r_t)}{\text{DCG}(i_t)} < \frac{\text{DCG}(s_t)}{\text{DCG}(i_t)} \iff \text{DCG}(r_t) < \text{DCG}(s_t)$$

since $\text{DCG}(i_t)$ is a positive quantity.

We can now proceed with the *replacement property*, assume that s_t is generated by the replacement of document at rank i of r_t, with $\hat{r}_t[i] < \hat{s}_t[i]$. We can rewrite DCG as follows:

$$\text{DCG}(r_t) = \underbrace{\sum_{k=1,\ k\neq i}^{N} \frac{\hat{r}_t[k]}{dsc(k,b)}}_{A(b)} + \frac{\hat{r}_t[i]}{dsc(i,b)} \ .$$

Since we need to prove that $\text{nDCG}(r_t) < \text{nDCG}(s_t)$ we can consider the difference between these two quantities:

$$\begin{aligned} \text{DCG}(r_t) - \text{DCG}(s_t) &= A(b) + \frac{\hat{r}_t[i]}{dsc(i,b)} - A(b) - \frac{\hat{s}_t[i]}{dsc(i,b)} \\ &= \frac{1}{dsc(i,b)}\left(\hat{r}_t[i] - \hat{s}_t[i]\right) \\ &< 0 \end{aligned}$$

where the last inequality holds because $dsc(i,b)$ is positive and $\hat{r}_t[i] < \hat{s}_t[i]$.

We can prove the *swap property* by following a similar reasoning. Assume that s_t is equal to r_t after swapping the document at rank i with the document at rank j, where to avoid trivial cases $\hat{r}_t[i] < \hat{r}_t[j]$. Consider the following reformulation of DCG:

$$\text{DCG}(r_t) = \underbrace{\sum_{k=1,\ k\neq\{i,j\}}^{N} \frac{\hat{r}_t[k]}{dsc(k,b)}}_{D(b)} + \frac{\hat{r}_t[i]}{dsc(i,b)} + \frac{\hat{r}_t[j]}{dsc(j,b)} \ .$$

therefore if we consider the difference $\text{nDCG}(r_t) - \text{nDCG}(s_t)$ we have

$$\begin{aligned}
\text{DCG}(r_t) - \text{DCG}(s_t) &= D(b) + \frac{\hat{r}_t[i]}{dsc(i,b)} + \frac{\hat{r}_t[j]}{dsc(j,b)} - D(b) - \frac{\hat{s}_t[i]}{dsc(i,b)} - \frac{\hat{s}_t[j]}{dsc(j,b)} \\
&= \frac{1}{dsc(i,b)}(\hat{r}_t[i] - \hat{r}_t[j]) + \frac{1}{dsc(j,b)}(\hat{r}_t[j] - \hat{r}_t[i]) \\
&= \left(\frac{dsc(j,b) - dsc(i,b)}{dsc(i,b) \cdot dsc(j,b)}\right)(\hat{r}_t[i] - \hat{r}_t[j]) \\
&< 0
\end{aligned}$$

where the last inequality holds whenever $dsc(j,b) > dsc(i,b)$, i.e. the discount for the rank position j is greater than the discount for the rank position i, with $j > i$. This is always true for Microsoft nDCG and for the original DCG, when $i \leq b < j$. If $i < j \leq b$, then the measure behaves like a set-based measure and $\text{DCG}(r_t) - \text{DCG}(s_t) = 0$, this is not in contrast with the swap property which claims that the measure should not decrease. Finally, notice that the discount function and the relevance function can be replaced with different functions, therefore this proof can be easily extended to all the rank weighted measures.

Expected Reciprocal Rank (ERR)

Given a run r_t of length N, recall the definition of ERR in Equation (2.3):

$$ERR(x_1,\dots,x_N) = \sum_{i=1}^{N} \frac{1}{i} \prod_{k=1}^{i-1} (1 - x_k) x_i$$

with the convention that $\prod_{i=1}^{0} = 1$ and x_i represents the probability that a user leaves his search after considering the document at position i. An additional assumption is that the map $\hat{r}_t[i] \mapsto x_i(\hat{r}_t[i])$ is increasing and $x_i(0) = 0$.

Let us consider the **Replacement property** and to avoid trivial cases, take $\hat{r}_t[i] < \hat{s}_t[i]$. The property is satisfied if the function $(x_1,\dots,x_N) \mapsto ERR(x_1,\dots,x_N)$ is non-decreasing in any variable. With this aim, we will prove that the partial derivatives $\frac{\partial}{\partial x_k} ERR > 0$ for any $k \leq N$ and $(x_1,\dots,x_N) \in [0,1]^N$. It is immediate that $\frac{\partial}{\partial x_N} ERR = \frac{1}{N} \prod_{k=1}^{N-1} (1 - x_k) > 0$.

Let us now consider $\frac{\partial}{\partial x_{N-1}} ERR$. Denoting $A(x_i,\dots,x_j) = \prod_{k=i}^{j} (1 - x_k)$, we get

$$\frac{\partial}{\partial x_{N-1}} ERR = A(x_1,\dots,x_{N-2}) \Big(\frac{1}{N-1} - \frac{x_N}{N} \Big) > 0$$

since $\frac{1}{N-1} - \frac{x_N}{N} > \frac{1}{N-1} - \frac{1}{N} > \frac{1}{(N-1)N} > 0$.

The general case follows similarly: take $k < N-1$ and consider $\frac{\partial}{\partial x_k} ERR$. This partial derivative will be positive if and only if

$$S(x_{k+1},\dots,x_N) = \frac{1}{k} - \frac{1}{k+1}x_{k+1}$$

$$-\frac{1}{k+2}A(x_{k+1})x_{k+2} - \dots - \frac{1}{N}A(x_{k+1},\dots,x_{N-1})x_N > 0.$$

Considering the last two terms, we get

$$\frac{1}{N-1}A(x_{k+1},\dots,x_{N-2})x_{N-1} + \frac{1}{N}A(x_{k+1},\dots,x_{N-1})x_N$$

$$\leq A(x_{k+1},\dots,x_{N-2})\frac{1}{N-1} .$$

This implies that

$$S(x_{k+1},\dots,x_N) > \frac{1}{k} - \dots - \frac{1}{N-2}A(x_{k+1},\dots,x_{N-3})x_{N-2}$$

$$-\frac{1}{N-1}A(x_{k+1},\dots,x_{N-2})$$

Applying the previous computation with the new last two terms and repeating this procedure on and on, at the end we obtain that

$$S(x_{k+1},\dots,x_N) > \frac{1}{k} - \frac{1}{k+1} > 0$$

and the replacement is proved for ERR.

The **Swap property** is a little more challenging. We have

$$ERR = F(x_1,\dots,x_{i-1}) + \frac{1}{i}\prod_{k=1}^{i-1}(1-x_k)\mathbf{x_i}$$

$$+\frac{1}{i+1}\prod_{k=1}^{i-1}(1-x_k)(1-\mathbf{x_i})x_{i+1} + \dots$$

$$\dots + \frac{1}{j-1}\prod_{k=1}^{i-1}(1-x_k)(1-\mathbf{x_i})(1-x_{i+1})\cdots(1-x_{j-2})x_{j-1}+$$

$$+\frac{1}{j}\prod_{k=1}^{i-1}(1-x_k)(1-\mathbf{x_i})\cdots(1-x_{j-1})\mathbf{x_j} + G(x_1,\dots,x_N) ,$$

where F and G are suitable functions, while $ERR(s)$ has the same expression with the $\mathbf{x_i}$'s and $\mathbf{x_j}$'s interchanged. It is immediate that $ERR(r_t) \leq ERR(s_t)$ if $j = i+1$. Indeed, we have that the previous inequality holds if and only if

$$\frac{1}{i}\mathbf{x_i} + \frac{1}{i+1}(1-\mathbf{x_i})\mathbf{x_{i+1}} \leq \frac{1}{i}\mathbf{x_{i+1}} + \frac{1}{i+1}(1-\mathbf{x_{i+1}})\mathbf{x_i}$$

which is equivalent to $\frac{1}{i(i+1)}\mathbf{x_i} \leq \frac{1}{i(i+1)}\mathbf{x_{i+1}}$. If $|i-j|>1$, $ERR(r_t) \leq ERR(s_t)$ if and only if

$$\mathbf{x_i}D(x_{i+1},\ldots,x_{j-1}) \leq \mathbf{x_j}D(x_{i+1},\ldots,x_{j-1})$$

where

$$\begin{aligned} D(x_{i+1},\ldots,x_{j-1}) = & \frac{1}{i} - \frac{1}{i+1}x_{i+1} - \frac{1}{i+2}(1-x_{i+1})x_{i+2} - \ldots \\ & \ldots - \frac{1}{j-1}(1-x_{i+1})\cdots(1-x_{j-2})x_{j-1} \\ & -\frac{1}{j}(1-x_{i+1})\cdots(1-x_{j-2})(1-x_{j-1}) \end{aligned}$$

It will be therefore sufficient to prove that $D(x_1,\ldots,x_k) > 0$ for any $(x_1,\ldots,x_k) \in [0,1]^k$, where $k = j-i-1 > 0$. Let us prove this by induction on k: if $k=1$ we get

$$D(x_1) = \frac{1}{i} - \frac{x_1}{i+1} - \frac{(1-x_1)}{i+2} \geq \frac{1}{i(i+1)}$$

for any $x_1 \in [0,1]$. Let us now assume that $D(x_1,\ldots,x_i) > 0$ for any $i \leq k-1$ and $(x_1,\ldots,x_i) \in [0,1]^i$. It holds

$$D(x_1,\ldots,x_k) = D(x_1,\ldots,x_{k-1}) + \frac{1}{(i+k-1)(i+k)}(1-x_1)\cdots(1-x_{k-1}) > 0$$

for any $(x_1,\ldots,x_k) \in [0,1]^k$ and the property is proved.

3.5 Balancing Index

In this section, we explore the behaviour of utility-oriented measurements when two runs r_t and s_t are not comparable according to the the partial ordering $\preceq$.

Let N be the length of a run, let r_t and s_t be two runs, $q_{min} = \min\{rel \in REL : rel > 0\}$ be the minimum relevance degree above not relevant and $q_{max} = \max\{rel \in REL\}$ be the maximum relevance degree, and $M(\cdot)$ a utility-oriented measurement. We assume here that $0 < q_{min} \leq q_{max} < \infty$.

We define the **Balancing Index** as

$$B(N) = \max\Big\{b \in \mathbb{N} \colon M\big(r_t : \hat{r}_t[1] = q_{max}, \hat{r}_t[j] = 0,\ 1 < j \leq N\big) \\ \leq M\big(s_t : \hat{s}_t[i] = 0,\ 1 \leq i < b, \hat{s}_t[j] = q_{min},\ b \leq j \leq N\big)\Big\}$$

As an example, let us consider the case of four relevance degrees $REL = \{0,1,2,3\}$ and runs of length 5. The balancing index seeks the maximum rank position b for which $M\big((3,0,0,0,0)\big)$ is balanced by $M\big((0,0,0,0,1)\big)$ or $M\big((0,0,0,1,1)\big)$ or $M\big((0,0,1,1,1)\big)$ or $M\big((0,1,1,1,1)\big)$, i.e. it determines when the greatest run possible with just one maximally relevant document (3 in this case) is scored "the same" as the smallest run possible with an increasing number of minimally relevant documents (1 in this case). If we choose DCG as M, computed with Equation (2.3.2), we obtain:

$$DCG\big((3,0,0,0,0)\big) = 3 \qquad \begin{aligned} DCG\big((0,0,0,0,1)\big) &\simeq 0.4307 \\ DCG\big((0,0,0,1,1)\big) &\simeq 0.9307 \\ DCG\big((0,0,1,1,1)\big) &\simeq 1.5616 \\ DCG\big((0,1,1,1,1)\big) &\simeq 2.5616 \\ DCG\big((1,1,1,1,1)\big) &\simeq 3.5616 \end{aligned}$$

therefore $B(5) = 1$, which means that to compensate a highly relevant document at the beginning of a run of length 5, you need fill the run with partially relevant documents.

The balancing index is not always defined and there are cases when if a system fails to retrieve a highly relevant document, it can not regain the lost utility. For example, if we choose ERR as M and we follow the weighting scheme of Equation (2.4), we obtain:

$$ERR\big((3,0,0,0,0)\big) = 0.875 \qquad \begin{aligned} ERR\big((0,0,0,0,1)\big) &= 0.025 \\ ERR\big((0,0,0,1,1)\big) &\simeq 0.0531 \\ ERR\big((0,0,1,1,1)\big) &\simeq 0.0882 \\ ERR\big((0,1,1,1,1)\big) &\simeq 0.1396 \\ ERR\big((1,1,1,1,1)\big) &\simeq 0.2472 \end{aligned}$$

therefore, even with a run full of relevant documents with relevance weight equal to 1, it is not possible to compensate a highly relevant document in the first position. In this case the balancing index will not be defined: $B(5) = \max\{\emptyset\}$ and we adopt the convention that $\max\{\emptyset\} = -\infty$.

The balancing index exploits the Replacement and Swap properties in a way, different from the one used in the equivalence theorem, that allows us to move among runs not comparable for the empirical ordering $\preceq$. In the above example, we have that

$$\begin{aligned}(3,0,0,0,0) &\xrightarrow[\succeq]{\text{Swap}} (0,0,0,0,3) \xrightarrow[\succeq]{\text{Replacement}} (0,0,0,0,1)\\ &\xrightarrow[\preceq]{\text{Replacement}} (0,0,0,1,1) \xrightarrow[\preceq]{\text{Replacement}} (0,0,1,1,1)\\ &\xrightarrow[\preceq]{\text{Replacement}} (0,1,1,1,1)\end{aligned}$$

where every two adjacent run pairs in the chain are comparable according to the empirical ordering $\preceq$, but not the first run with the last ones, e.g. $(3,0,0,0,0)$ is not a priori comparable to $(0,0,0,1,1)$ because neither you know whether the loss of a document with relevance degree 3 is compensated or not by two documents with relevance degree 1 nor you know the effect of ranking.

The balancing index allows us to explore cases that fall outside the empirical ordering $\preceq$ and to characterize the behaviour of the measurements in those circumstances where Definition 1 cannot ensure they will a priori act in a homogeneous way.

In particular, a measurement with $B(N) \to N$ behaves like a set-based measure, being extremely sensitive to the presence of additional relevant documents in the lowest ranks. On the contrary, a measurement with $B(N) \to 1$ is not sensitive to the presence of additional relevant documents after a relevant one in the top rank.

The balancing index models the concept of *top heaviness*, an important and somehow desired characteristic of a measurement, as highlighted also in previous works. The closeness threshold constraint [Amigó et al., 2013] resembles it, even if it is formulated as a constraint stating that relevant documents in top ranks should count more rather than an index that you can actually compute to characterize a measurement; similar considerations hold for the notion of top-weightedness [Moffat, 2013]. However, it should be noted that, instead of requesting top heaviness to be an a-priori propriety as in [Amigó et al., 2013; Moffat, 2013], the balancing index explicitly points out that top heaviness is a property of the measurements that concerns the area where runs are not a priori comparable, i.e. outside the empirical ordering $\preceq$, and this, in turn, causes measurements to possibly behave differently one from another, being more or less top heavy.

With respect to other empirical indexes for quantifying top heaviness, the balancing index has the advantage that it can be derived analytically. Below, some example of balancing indexes for some popular measurements are reported:

AP

$$B(N) = \max\left\{ b \in \mathbb{N} \colon \sum_{k=1}^{N-b+1} \frac{k}{k+b-1} \geq 1 \right\}$$

RBP

$$B(N) = \max\left\{ b \in \mathbb{N} \colon b \geq \log_p(1-p+p^N)+1 \right\}$$

where p is the persistence parameter of RBP.

ERR

$$B(N) = \max\left\{ b \in \mathbb{N} \colon x_{min} \sum_{k=b}^{N} \frac{(1-x_{min})^{k-1}}{k} \geq x_{max} \right\}$$

where x_{min} represents the probability that a user leaves his search after considering a document of relevance q_{min} and x_{max} represents the probability that a user leaves his search after considering a document of relevance q_{max}.

nDCG

$$B(N) = \max\{b_1, b_2\}$$

where

$$b_1 = \max\left\{ b > a \in \mathbb{N} \colon \sum_{k=0}^{N-b} \frac{q_{min}}{log_a(k+b)} \geq q_{max} \right\},$$

$$b_2 = \max\left\{ b \leq a \in \mathbb{N} \colon (a-b+1)q_{min} + c \geq q_{max} \right\},$$

$$c = \sum_{k=0}^{N-a-1} \frac{q_{min}}{log_a(k+a+1)}$$

and a is the base of the logarithm in nDCG. Recall that $\max\{\emptyset\} = -\infty$.

It can be noted that some of the above formulas depend explicitly on the length of the run under consideration, as in the case of RBP, while others have an implicit dependence on it and might be more complex to be computed.

We report Algorithm 1, which allows us to compute the balancing index numerically. The complexity of the algorithm is $O(N)$, since, assuming that the computation of the measurement M requires a constant number of operations, the while loop carries out at most $N-1$ iterations and at any iterations it performs a constant number of operations.

Note that, even if we compute the balancing index in a numerical way, it is not an empirical indicator, as for example the discriminative power [Sakai, 2006] is, whose computation depends on a given experimental collection and a set of runs and whose value may change from dataset to dataset.

ALGORITHM 1: Algorithm to compute the balancing index.

```
Data: $N$, the length of the run; $q_{min}$ and $q_{max}$ the minimal and maximal relevance degrees
Result: $b$, the balancing index for a run of length $N$
$refValue \leftarrow M(r : \hat{r}_t[1] = q_{max},$ and $\hat{r}_t[j] = 0$ if $1 < j \leq N)$;
$cmpValue \leftarrow M(r : \hat{r}_t[j] = 0$ if $1 \leq j < N,$ and $\hat{r}_t[N] = q_{min})$;
$b \leftarrow N$;
while $refValue > cmpValue$ do
    $b--$;
    $cmpValue \leftarrow M(r : \hat{r}_t[j] = 0$ if $1 \leq j < b,$ and $\hat{r}_t[j] = q_{min}$ if $b \leq j \leq N)$;
end
return $b$;
```

Figure 3.3 reports the balancing index for several evaluation measurements at different run lengths. We computed the balancing index in the binary case, this means that $q_{min} = q_{max}$, in this case equal to 1. Note that this assumption prevent us to fall in the case of a not defined balancing index, as in the previous example with ERR. Indeed, even if the measure is strongly top heavy, a run filled with relevant documents will be greater than a run with just one relevant document at the beginning. This is ensured by the replacement property: $(1,1,\dots,1)$ can be obtained from $(1,0,\dots,0)$ by adding a relevant document for each rank position from 2 to N and after each replacement we will obtain a greater run.

In Figure 3.3 it can be noted that for AP and nDCG we have $B(N) \rightarrow N$ since it is close to the bisector, indicating that they are not strongly top-heavy measurements and that they are sensitive to relevant documents in the lower ranks. On the other hand, ERR is the most top-heavy measurement since its balancing index is $b = 1$ for any run length, meaning that missing a relevant document in the first rank position can not be compensated even by a run filled in with relevant documents from the second rank position to the end. RBP falls somehow in-between, still being a quite top-heavy measurement; it can be noted as for $p = 0.8$ the balancing index saturates to $b = 8$ for run lengths greater than 20 while, as p increases, it tends to be less top-heavy with almost $b = 60$ for $p = 0.95$.

In order to assess the meaningfulness of the balancing index, we conducted the following experiment with RBP and $p = 0.8$. We simulated two runs of length $N = 1000$ consisting of 50 topics each, generated as shown in Figure 3.4.

In the top ranks up to *rnk* they have the same proportion (20%) of relevant documents; in the ranks from *rnk* to 20 they have different proportions of relevant documents 70% for r_t and 30% for s_t; in the ranks from 21 to $N = 1000$ they have still different proportions of relevant documents 10% for r_t and 70% for s_t. Then, we increased *rnk* from 0 to 20: when $rnk = 0$, r_t contains more than twice relevant documents in the top ranks than s_t and much less

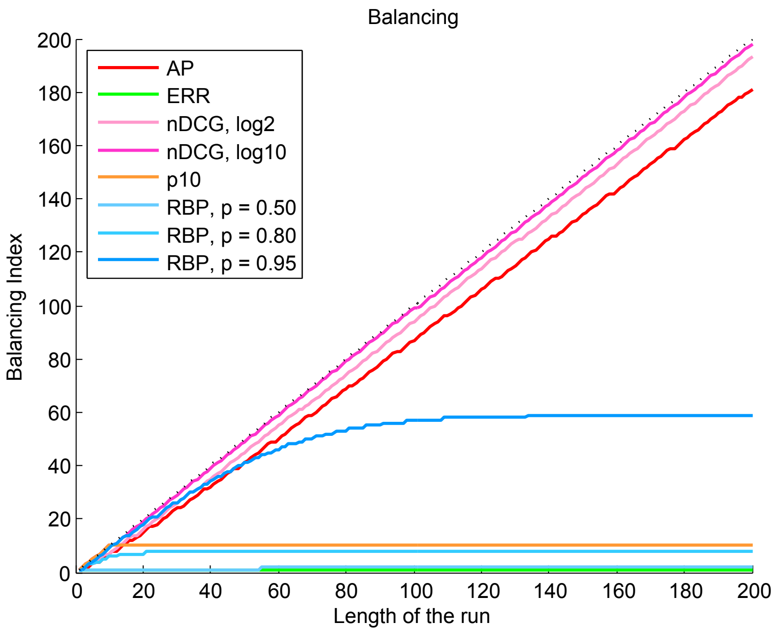


Fig. 3.3 Balancing index for AP, RBP, ERR, P@10, nDCG for different run lengths.

relevant documents in the very long tail; when $rnk = 20$, r_t and s_t have the same proportion of relevant documents in the top ranks, but r_t has much less relevant documents than s_t in all the other rank positions. For each increasing value of rnk, we performed a Student's t test with $\alpha = 0.05$ to assess whether r_t and s_t were significantly different. We repeated this experiment $10,000$ times and, for each value of rnk, we computed the probability that the two runs are considered significantly different as the ratio among the number of times the Student's t test rejects the null hypothesis and $10,000$, the total number of trials.

Figure 3.5 shows the results of this experiment. It can be noted that, as far as rnk grows up the balancing index $b = 8$, the fact that r_t contains a bigger proportion of relevant documents

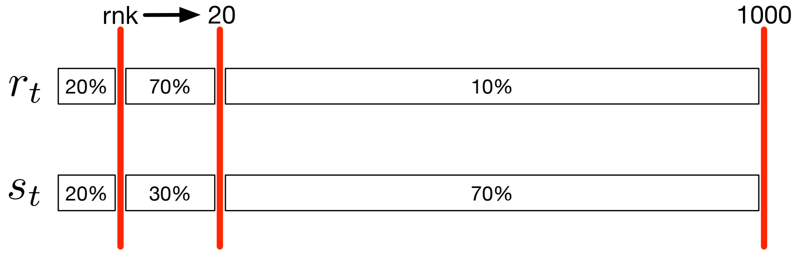


Fig. 3.4 Creation of the simulated runs for assessing the meaningfulness of the balancing index in the case of RBP. Note that the percentages are not referred to the whole run but to each segment separately. Therefore, they do not need to sum up to 100%, but to be between 0% and 100% within each segment.

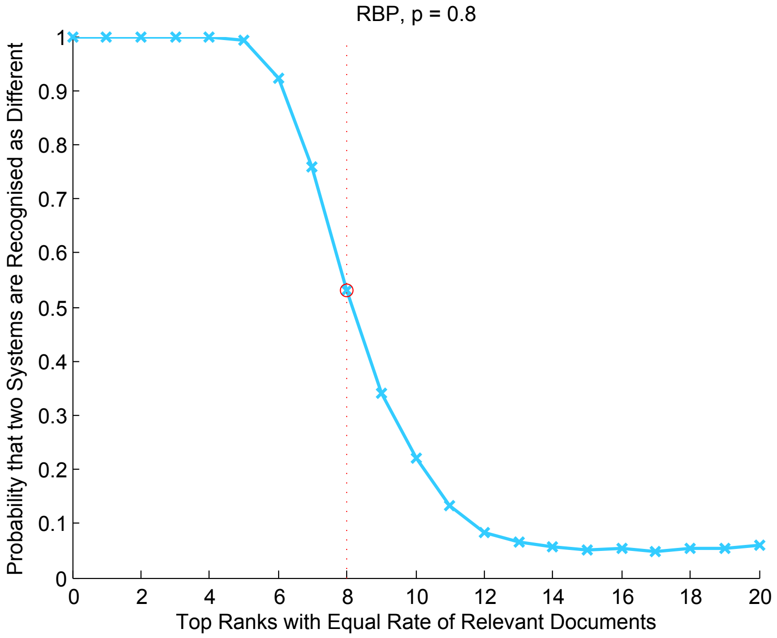


Fig. 3.5 Test of the meaningfulness of the balancing index for RBP with $p = 0.8$.

than s_t in the top ranks almost always leads to consider the two runs as significantly different. On the other hand, as soon as *rnk* passes the balancing index $b = 8$ and the proportion of relevant documents in the top ranks of r_t and s_t starts to get more and more similar, the probability of considering the two runs significantly different gets lower and lower, completely ignoring the long tail where they are actually quite different. This is a clear indicator of top-heaviness, well reflected by the balancing index.

3.6 Summary

In this chapter we have laid the foundations of a formal framework for defining what a utility-oriented measurement of retrieval effectiveness is, on the basis of the representational theory of measurement, putting IR evaluation in the wake of other physical and social sciences as far as measuring is concerned. A core contribution of the chapter is to address the problem by clearly separating what are the issues in dealing with comparable/not comparable runs in the empirical world, from what are the expected properties of a measurement in the symbolic world.

We proposed a minimal definition of measurement, based on just one axiom (Definition 1), and provided an equivalence theorem (Theorem 1) to check it in practice, as well as examples of its application.

Finally, we proposed the balancing index as an indicator of the top-heaviness of a measurement, providing both formulas and an algorithm to compute it. We have also conducted a preliminary experiment to show that its numerical value is a meaningful indicator of top-heaviness.

Future work will concern a deeper exploration of the core problems such measurements have, as for example additivity. We will also exploit the theory of scales of measurement in order to study the scales actually adopted by common measurements like AP, RBP, ERR, nDCG and others.

Furthermore, we will consider the application of the proposed framework to other cases, such as measures based on diversity. This will lead to a different definition of the partial ordering $\preceq$ in the empirical relational structure $\mathbf{E}$ to capture the notion of diversity, but Definition 1 of IR measurement of retrieval effectiveness will remain the same. Moreover, this may also require to individuate properties different from Swap and Replacement to provide an equivalence theorem in the vein of Theorem 1 suitable for this case.

Chapter 4

AWARE: Merging Relevance Judgements via Evaluation Measures

> A human is not a device that reliably reports a gold standard judgment of relevance of a document to a query.
>
> Manning et al. [2008]

Ground-truth is central to IR evaluation since it enables the scoring and comparison of algorithms and systems with respect to human judgments, determining whether documents are relevant, or not, to user information needs.

Creating a dataset and, in particular, gathering relevance assessments is an extremely demanding activity: it involves sizable costs for hiring assessors and a fairly large amount of time to judge a pool of documents. Therefore, there is an increasing interest for more effective and affordable ways of gathering assessments [Halvey et al., 2014], especially to face the ever increasing number of new search tasks that need an appropriate dataset to be evaluated.

Crowdsourcing [Alonso and Mizzaro, 2012; King et al., 2016; Lease and Yilmaz, 2013; Marcus and Parameswaran, 2015] has emerged as a viable option for ground-truth creation since it allows to cheaply collect multiple assessments for each document. However, it raises many questions regarding the quality of the collected assessments. Therefore, in order to obtain a ground-truth good enough to be used for evaluation purposes, the possibility of discarding the low quality assessors and/or combining them with more or less sophisticated algorithms has been considered.

The problem of merging multiple crowd assessors has been addressed mostly from a classification point of view, i.e. choosing among the set of possible judgements (labels) those best supported by the evidence provided by the crowd assessors. In detail, traditional approaches focus mainly on how to select assessors and/or discard low quality assessors, how to merge judgments from multiple assessors into a single assessor, and how to route tasks to assessors. They typically determine the "best" relevance judgements, combining those produced by multiple crowd assessors according to some criteria, and use them to compute a performance measure, like AP, and score systems. We can consider this as a kind of "upstream" approach, because the aggregated ground-truth is created before systems are evaluated and performance scores are computed.

In this chapter, we address the problem of ground-truth creation in a crowdsourcing context from a new angle, i.e. we investigate how to estimate performance measures in a way more robust to crowd assessors. To the best of our knowledge, what happens when you aggregate the different performance scores directly computed on the judgements produced by multiple assessors is yet to be explored. In particular, we seek a better estimation of the true expected value of a performance measure, by leveraging its multiple observations, generated separately by the relevance judgements of each crowd assessor. We can consider this as a kind of "downstream" approach with respect to the classification ones, since the aggregation happens after performance measures have been computed.

The main intuition behind our approch is based on the idea that the choice of the "best" relevance judgments, operated ahead at the pool level, may have a diverse impact on different systems and on various performance measures. Indeed, systems rank the same documents differently and therefore the same correctly labelled or mis-labelled documents impact the performances of different systems in different ways. Moreover, performance measures embed different user models, weighting differently even the same system ranking; therefore, the same correctly labelled or mis-labelled documents have a different impact on different performance measures. As a consequence, even a small error over a whole pool of documents may affect systems and performance measures in quite different ways.

To make an intuitive yet extreme toy example, suppose that out of 10 relevant documents in a pool, just 1 document has been wrongly labelled as not relevant, thus there is a 10% error with respect to the whole pool. Now consider a run which retrieves that mis-labelled document, represented as a blue *R* in italics, somewhere in the ranks from 1 to 5 and it also retrieves a few other relevant documents in the ranks from 6 to 10, marked as a plain R.

Rank	1	2	3	4	5	6	7	8	9	10	P@5	AP
Run_1						R			R	R	0.0000	0.0765
Run_2					*R*	R			R	R	0.2000	0.1407
Run_3				*R*		R			R	R	0.2000	0.1463
Run_4			*R*			R			R	R	0.2000	0.1556
Run_5		*R*				R			R	R	0.2000	0.1741
Run_6	*R*					R			R	R	0.2000	0.2296

Run_1 represents the case where the mis-labelled document is not detected in any ranks from 1 to 5, while the other runs show what could have happened if it had been correctly labelled. You can see how for P@5, i.e. precision at 5 retrieved documents, wherever this document is in the ranks from 1 to 5, it makes the difference between P@5 = 0 and P@5 = 20%, which represents a 100% error; for AP, it changes from AP = 7.65% to AP between 14.07% and 22.96%, i.e. an error ranging between 45.61% and 66.67%. In all these cases, the effect of a single mis-labelled document has a different impact on different runs and for different performance measures and, in the extreme example at hand, it is much greater than the error on the pool itself.

We propose the Assessor-driven Weighted Averages for Retrieval Evaluation (AWARE) probabilistic framework [Ferrante et al., 2017], which allows us to combine multiple versions of a performance measure, computed from the ground-truth created by each crowd assessor, into a single composite measure, which we call the AWARE version of it. The AWARE framework specifies how performance measures have to be merged on the basis of the estimated crowd assessor accuracies and we propose several unsupervised estimators of such accuracies. Intuitively, these unsupervised estimators compute some kind of "distance" between the selected performance measure computed on the ground-truth produced by the crowd assessor and the same performance measure computed on the ground-truth produced by different types of random assessors: the greater this "distance", the better the accuracy of the crowd assessor.

We conduct a thorough experimental evaluation, using the ground-truth created by the crowd assessors of the TREC 21, 2012, Crowdsourcing track [Smucker et al., 2013] with respect to the systems submitted to the TREC 08, 1999, Ad-hoc [Voorhees and Harman, 1999] and the TREC 13, 2004, Robust [Voorhees, 2015] tracks. We experiment with the following performance measures: Average Precision (AP) [Buckley and Voorhees, 2005], Normalized Discounted Cumulated Gain (nDCG) [Järvelin and Kekäläinen, 2002], and Expected Reciprocal Rank (ERR) [Chapelle et al., 2009]. The experimentation shows that AWARE approaches improve in terms of capability of correctly ranking systems and predicting their actual performance scores.

This Chapter is organized as follows: Section 4.1 introduces related works and provides a description of state-of-the-art algorithms for combining multiple assessors which will be used for comparison with the AWARE approach; Section 4.2 introduces the AWARE framework; Section 4.3 proposes several unsupervised estimators for determining the assessors accuracies to be used for combining AWARE measures; Section 4.4 describes the experimental setup; Section 4.5 and Section 4.6 carry out a thorough evaluation using TREC collections; finally, Section 4.7 draws some conclusions and presents an outlook for future work.

4.1 Background

4.1.1 Crowdsourcing for Ground-truth Creation

One of the first investigated issues, assuming the quality of the assessors for granted, concerned the impact of the inter-assessor disagreement. What happens if we assign the same set of topics and documents to another assessor? Will the ranking of the systems remain stable? Several studies [Burgin, 1992; Lesk and Salton, 1968; Voorhees, 1998, 2000] have shown that even a not negligible amount of inter-assessor disagreement does not severely impact the ability of ranking systems and, more recently, [Webber et al., 2012] has provided evidence that the rank of a document is a factor influencing the probability of disagreement among assessors. Other issues concern the expertise of the assessors on the domain of the topics they are judging: [Bailey et al., 2008; Kinney et al., 2008] noted that this factor has some impact on the evaluation.

Moreover, regarding the comparison of different types of assessors, a lot of work was done to investigate the relation between domain experts and crowd assessors [Clough et al., 2013], authoritative and alternative assessors [Webber and Pickens, 2013], primary and secondary assessors [Wakeling et al., 2016], NIST assessors and user studies participants [Smucker and Jethani, 2011a], crowd assessors and university laboratory participants [Smucker and Jethani, 2011b]. Finally, [Ruthven, 2014] studies the assessors' characteristics that lead to different relevance assessments and [Sanderson and Zobel, 2005] investigates how to build test collections in order to optimize the assessor effort.

Research in crowdsourcing has focused on several different issues: aggregating labels from multiple assessors to improve the quality of the gathered assessments, by using unsupervised [Bashir et al., 2013; Hosseini et al., 2012], supervised [Pillai et al., 2013; Raykar and Yu, 2012; Raykar et al., 2010], and hybrid [Harris and Srinivasan, 2013] approaches; behavioural aspects [Kazai et al., 2012b]; proper and careful design of Human Intelligent Tasks (HITs) [Alonso, 2013; Grady and Lease, 2010; Ipeirotis and Gabrilovich, 2014; Kazai et al., 2011], also using gamification to improve quality [Eickhoff et al., 2012] and game theory to increase user engagement [Moshfeghi et al., 2016]; human-machine collaborative methods for training crowdsource workers [Abad, 2017; Abad et al., 2017]; and, routing tasks to proper assessors [Jung and Lease, 2015; Law et al., 2011].

There is a growing concern about the quality of the gathered assessments [Kazai, 2011; Kazai et al., 2013a; Vuurens and de Vries, 2012], how assessor quality and errors impact evaluation [Carterette and Soboroff, 2010; Kazai et al., 2012a], how much tolerant evaluation measures are to these errors [Li and Smucker, 2014], and how crowd and editorial assessors agreement relates to user intent and click-based measures [Kazai et al., 2013b].

In recent years, several evaluation activities have focused on crowdsourcing for ground-truth creation, as witnessed by the TREC Crowdsourcing track series[1] from 2011 to 2013 [Smucker et al., 2013, 2014], the MediaEval Crowdsourcing tracks[2] in 2013 and 2014 [Loni et al., 2013; Yadati et al., 2014], or the CrowdScale 2013 Shared Task Challenge[3] [Josephy et al., 2014]. There is also a growing interest and attention about how crowdsourcing affects the repeatability and reproducibility of IR experiments [Blanco et al., 2011; Ferro, 2017; Ferro et al., 2016a].

In this Chapter we are interested in aggregating labels from multiple assessors and, in the experimental part in Sections 4.5 and 4.6 we will compare our proposed approach, AWARE, with two state-of-the-art approaches for label aggregation, namely Majority Vote (MV) and Expectation Maximization (EM) [Bashir et al., 2013; Hosseini et al., 2012], which are briefly summarized in the following sections.

4.1.2 Majority Vote

Hereafter we use the definitions of set of documents D, set of topics T, set of relevance degrees $(REL, \preceq)$, and ground-truth GT, as they are introduced in Chapter 3. We restrict ourselves to the case of binary relevance and we assume $REL = \{0,1\}$. Moreover, let $\Lambda = \{W_1, \dots, W_l\}$ be a finite **set of assessors**, we define as $GT_k(t,d)$ the discrete variable with values in $\{0,1\}$, which represents the label given by the assessor k to the document d with respect to the topic t. Note that this is the only information that we are provided with, indeed we assume that the relevance judgments, $GT(t,d)$, are not known. We further suppose that each document receives at least one relevance label. Finally, let $\mathbb{1}_{\{GT_k(t,\cdot)=g\}}$ be a binary variable that is equal to 1 if the assessor k assigns the label g to the document d and zero otherwise.

[1] https://sites.google.com/site/treccrowd/

[2] http://www.multimediaeval.org/

[3] http://www.crowdscale.org/shared-task

The simplest way of estimating the true relevance labels is the Majority Vote (MV) algorithm, which views each worker as a voter. If the number of voters which consider a given document as relevant is greater than the number of voters that consider it as not relevant, that document will be classified as relevant. Hence, if $n_t[d,g] = \sum_{k=1}^{l} \mathbb{1}_{\{GT_k(t,d)=g\}}$ is the number of times that the document d is labeled as g for the topic t, we will assign to d the relevance g that maximizes $n_t[d,g]$, that is g such that $n_t[d,g] = argmax_g\{n_t[d,0], n_t[d,1]\}$. In the case of tie, i.e. $n_t[d,0] = n_t[d,1]$, a coin is tossed to determine whether the document is relevant or not.

4.1.3 Expectation Maximization

The Expectation Maximization (EM) algorithm is an alternative to MV for defining the relevance of the documents. We follow the same approach described in [Hosseini et al., 2012] to implement the EM algorithm.

Suppose that a latent confusion matrix, $\pi_t[\cdot,\cdot](k)$, $k \in \{1,\dots,l\}$, is assigned to each assessor, this matrix has as many rows and columns as the number of relevance grades, i.e. two in the binary case. Each row represents the true relevance grade and each column the label given by the worker. We define $\pi_t[g,h](k) = \mathbb{P}\big[GT_k(t,\cdot) = h | GT(t,\cdot) = g\big]$, i.e. the probability that the assessor k assigns to a document the relevance grade h, given that the true relevance label of the document is g. For instance, $\pi_t[1,0](k)$ is the probability that the worker k labels a document as not relevant, given that this document is relevant. The matrix $\pi_t[g,h](k)$ could be estimated by:

$$\frac{\text{number of times the worker } k \text{ provides label } h \text{ while the true label is } g}{\text{number of labels provided by worker } k \text{ for documents of relevance } g} \; .$$

Note that, in the binary case:

$$\pi_t[g,0](k) + \pi_t[g,1](k) = 1 \quad \forall\, k \in \{1,\dots,l\} \text{ and } g \in \{0,1\} \; .$$

Moreover, we define $p_t[g] = \mathbb{P}\big[GT(t,\cdot) = g\big]$, the probability that a randomly chosen document has relevance grade g, i.e. $p_t[0]$ is the probability that a document drawn at random is not relevant and $p_t[1]$ is the probability that it is relevant.

The EM algorithm consists of five main steps that we will describe in the following, and we will indicate with the symbol $\tilde{}$ a possible estimate of the parameter or the variable under the $\tilde{}$.

Step 1: Initialization Firstly we initialize the parameters of our model, we adopt two different strategies that we will illustrate later in detail.

Step 2: Estimate the maximum likelihood Then we compute the maximum likelihood estimates of $\pi_t[\cdot,\cdot](\cdot)$ and $p_t[\cdot]$ as follows:

$$\tilde{\pi}_t[g,h](k) = \frac{\sum_{d=1}^{|D|} \mathbb{1}_{\{GT(t,d)=g\}} \mathbb{1}_{\{GT_k(t,d)=h\}}}{\sum_{h \in REL} \sum_{d=1}^{|D|} \mathbb{1}_{\{GT(t,d)=g\}} \mathbb{1}_{\{GT_k(t,d)=h\}}} \,,$$

$$\tilde{p}_t[g] = \frac{\sum_{d=1}^{|D|} \mathbb{1}_{\{GT(t,d)=g\}}}{|D|} \,.$$

Step 3: Estimate the probability of relevance We compute the new estimate of the relevance judgments based on $\tilde{\pi}_t[\cdot,\cdot](\cdot)$ and $\tilde{p}_t[g]$:

$$\mathbb{P}\big[GT(t,d) = g | GT_{\cdot}(t,\cdot), \pi_t[\cdot,\cdot](\cdot)\big] = \frac{\tilde{p}_t[g] \prod_{k=1}^{l} \prod_{h \in REL} (\tilde{\pi}_t[g,h](k))^{\mathbb{1}_{\{GT_k(t,d)=h\}}}}{\sum_{g \in REL} \tilde{p}_t[g] \prod_{k=1}^{l} \prod_{h \in REL} (\tilde{\pi}_t[g,h](k))^{\mathbb{1}_{\{GT_k(t,d)=h\}}}} \,.$$

Step 4: Iterate We repeat the steps 2 and 3 until the results converge.

Step 5: Define the relevance labels Finally, for each document d, we assign the label g to the documents with the maximal probability of having relevance grade g; i.e. we compute $argmax_{g \in REL}\{\mathbb{P}\big[GT(t,d) = g | GT_{\cdot}(t,\cdot), \pi_t[\cdot,\cdot](\cdot)\big]\}$, then we set $GT(t,d) = g$. Notice that in the binary case all the documents with probability of relevance greater than 0.5 are considered as relevant, and documents with probability equal or lower than 0.5 are considered as not relevant.

The convergence of the EM algorithm strongly depends on many assumptions that, if not satisfied, could compromise the convergence of the algorithm [Dawid and Skene, 1979; Wu, 1983]. In particular, the starting point of the EM algorithm represents a criticality that has to be treated properly. Therefore, we define two different instantiations of the EM algorithm, by interpreting the initialization step in two different ways:

EM-MV We use the algorithm of [Hosseini et al., 2012] and we set the initial relevance labels as the result of the MV algorithm, as done in [Raykar and Yu, 2012; Raykar et al., 2010];

EM-NEU We initialize each worker confusion matrix and the probability p_t as done in [Bashir et al., 2013]:

$$\tilde{\pi}_t[\cdot,\cdot](k) = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix} \quad , \quad \tilde{p}_t = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \quad .$$

Hence, we make the hypothesis that each worker honestly assigns the relevance labels. Then, we initialize the relevance labels by computing the probability of relevance as in the third step of the EM algorithm.

4.2 The AWARE Framework

In order to cope with and leverage crowd assessors, we need to extend the definitions of Chapter 3 and frame them in a probabilistic context. In particular, we assume that the relevance of a document is not deterministically known, but it is described by a probability distribution: instead of specifying a single value from *REL* as results of the relevance assessment, we model the uncertainty entailed in the assessment process as a whole distribution of possible values associated to each (t,d) pair. Furthermore, we assume that the ability of the crowd assessors themselves is stochastically determined by a probability assigned to them, that we call their accuracy.

More precisely, we assume that there exists a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, which provides the source of randomness and encompasses the judgements done by all the possible crowd assessors, on all the possible documents for any possible topic. Considering this space, we can extend the definition of the ground-truth as follows:

$$GT : \Omega \times T \times D \to REL$$

In this way, to any pair (t,d) we associate a random variable $GT(\cdot,t,d)$ with value on REL, whose distribution describes the relevance of the document d with respect to the topic t. This distribution can be modeled by means of various parameters, for example, the expected relevance obtained by all the possible crowd assessors who judge that pair.

All the definitions of Chapter 3 (judged run, performance measure and so on) remain unchanged, provided that it is understood that all the objects are now random variables. For example, a **(random) judged run** will be the random variable $\hat{r}_t$ from $\Omega \times T \times \mathscr{D}$ into $\mathscr{R}$, which assigns a (random) relevance degree to each retrieved document in the ranked list

$$(\omega, t, r_t) \mapsto \hat{r}_t = \big(GT(\omega,t,d_1), \ldots, GT(\omega,t,d_n)\big)$$

In the sequel, as it usually done in probabilistic frameworks, we omit to explicitly write the dependence of the random variables on ω.

Let $\Lambda = \{W_1, \ldots, W_l\}$ be a finite set of crowd assessors and let us assume that there exists a random variable, $W : \Omega \times T \to \Lambda$, whose distribution identifies the ability of a single crowd assessor with respect to any given topic. In practice, we can assume to be able, from the judgments of all the documents and with respect to a given topic t, to weight the average ability of any single crowd assessor with a positive number; the distribution on Λ can be then obtained from these numbers once normalized to 1. We call $a_k(t) = \mathbb{P}[T = t, W = W_k]$ the **accuracy** of crowd assessor W_k in assessing topic t and we assume that $a_k(t)$ is determined by the expected ability she/he demonstrates in assessing all the possible documents for that topic.

The easiest way to jointly cope with these random objects, i.e. ground-truth and crowd assessors, is to consider their expectations. The expected ground-truth of a pair (t,d), i.e. the expected relevance of document d for topic t, by the law of total expectation, is given by

$$\mathbb{E}\big[GT(t,d)\big] = \mathbb{E}\Big[\mathbb{E}\big[GT(t,d)\big|W\big]\Big] = \sum_{k=1}^{l} \mathbb{E}[GT(t,d)|W = W_k]\ a_k(t) \tag{4.1}$$

The conditional expectation $\mathbb{E}\big[GT(t,d)\big|W = W_k\big]$ in (4.1) represents the "best" possible approximation of $GT(t,d)$ given that the assessment has been provided by the crowd assessor W_k, where "best" refers to the minimal distance in mean square between them. This is, for example, the approach adopted by MV, under some strong assumptions: the crowd assessors W_k are independent and identically distributed (i.i.d.) and the accuracies $a_k(t)$ are uniformly distributed.

For a performance measure m$(\cdot)$, we can proceed in a similar way and define its AWARE version as its expectation with respect to $\mathbb{P}$:

$$\text{aware-m}\big(t, r_t\big) = \mathbb{E}\Big[\mu\big(\hat{r}_t\big)\Big] = \sum_{k=1}^{l} \mathbb{E}\big[\mu\big(\hat{r}_t\big)\big|W = W_k\big]\ a_k(t) \tag{4.2}$$

To make this approach feasible, we need to have a simple but yet reasonable way to estimate $\mathbb{E}\big[\mu\big(\hat{r}_t\big)\big|W = W_k\big]$ and $a_k(t)$.

For the first term, we estimate $\mathbb{E}\big[\mu\big(\hat{r}_t\big)\big|W = W_k\big]$ by $\mu\big(\hat{r}_t^k\big)$, where $\hat{r}_t^k$ represents the judged run under the assessments done by the crowd assessor W_k. Indeed, we typically have available just one judgement for each (t,d) pair by each crowd assessor and therefore the expectation collapses into that single observation.

The estimation of the accuracies $a_k(t) = \mathbb{P}[T = t, W = W_k]$ is somehow more problematic. Indeed, the estimation of the probability $\mathbb{P}$ calls for multiple observations and this is addressed by state-of-the-art approaches like MV and EM by assuming that crowd assessors are somehow i.i.d.. However, this is quite a strong assumption since crowd assessors are very different from each other and even the same crowd assessor may have a quite different behavior across different topics.

Therefore, we remove the i.i.d. assumption about the crowd assessors and we look for something to compare our not-i.i.d. crowd assessors against, something that can be truly i.i.d. and allows us to perform inferential statistics. We therefore take a **random assessor** as a truly i.i.d. comparison point. In the case of binary relevance, i.e. when $REL = \{0, 1\}$, an assessor W_k is a **random assessor of parameter** $p \in [0,1]$, if for any pair (t,d) the conditional random variables $GT(t,d)|W = W_k \sim Bin(1, p)$, where $Bin(1, p)$ denotes a Binomial random variable with parameter p, and are mutually independent.

A random assessor, of any possible parameter p, is the prototype of a “bad” or at least a “shallow” assessor, since p is the same for any possible pair (t,d). As the definition of the random assessor is purely theoretic, we can assume that we are able to produce a sample of i.i.d. random assessors with the same parameter p. This fact allows us to provide classical inferential constructions of the estimates of the accuracy $a_k(t)$, as will be described in detail in the next section. The basic idea that we will apply in the next section is that the farther a crowd assessor is from the random ones, the better she is and the higher her accuracy will be.

Thanks to these considerations, we define the estimated version of AWARE as follows

$$\widetilde{\text{aware-m}}(t, r_t) = \sum_{k=1}^{l} \mu\big(\hat{r}_t^k\big) a_t^k \tag{4.3}$$

where a_t^k represents an estimate of the unknown accuracies $a_k(t)$.

Let us discuss how equation (4.3) works and the potential benefits of the AWARE approach by means of a toy example. Let us consider AP as performance measure, a pool containing just 3 relevant documents, and a run of length 5 where the first and the third documents are relevant, while the second, fourth and fifth are not relevant:

$$\hat{r}_t = (1, 0, 1, 0, 0) \Rightarrow \text{AP}\big(\hat{r}_t\big) = 0.5556$$

Suppose that we have three crowd assessors, judging that documents as follows:

$$\begin{aligned}
\hat{r}_t^1 &= (1, 1, 0, 0, 0) \Rightarrow \text{AP}\big(\hat{r}_t^1\big) = 0.6667 \\
\hat{r}_t^2 &= (1, 1, 1, 0, 0) \Rightarrow \text{AP}\big(\hat{r}_t^2\big) = 1.0000 \\
\hat{r}_t^3 &= (0, 1, 1, 0, 1) \Rightarrow \text{AP}\big(\hat{r}_t^3\big) = 0.5889
\end{aligned}$$

By using the MV and EM approaches we can compute a merged ground-truth, which in this case is the same for both approaches, and thus we obtain:

$$\hat{r}_t^{\text{MV}} = \hat{r}_t^{\text{EM}} = (1, 1, 1, 0, 0) \Rightarrow \text{AP}\big(\hat{r}_t^{\text{MV}}\big) = \text{AP}\big(\hat{r}_t^{\text{EM}}\big) = 1.0000$$

which represents a 20% error in terms of relevance labels but an 80% error in terms of AP. If in equation (4.3) we take the simplest estimator possible of $a_k(t)$, i.e. a uniform distribution $a_t^k = \frac{1}{3}$, $k = 1,2,3$, which basically is the same underlying uniform approach used by MV, we obtain

$$\widetilde{\text{aware}}\text{-AP}(\hat{r}_t) = \frac{1}{3}\text{AP}(\hat{r}_t^1) + \frac{1}{3}\text{AP}(\hat{r}_t^2) + \frac{1}{3}\text{AP}(\hat{r}_t^3) = 0.7518$$

which represents a 35% error in terms of AP.

4.3 Estimating Crowd Assessor Accuracy

This sections aims at providing several unsupervised estimators of the accuracy $a_k(t)$ of a crowd assessor. We introduce some notation and an intuitive overview of the proposed estimators and then we go into their details.

4.3.1 Notation

Let S be the **set of systems** under experimentation and $s \in S$ be a generic system.

We call **assessor measure** the $|T| \times |S|$ matrix M_k containing the scores of each system for each topic, computed using a performance measure $\text{m}(\cdot)$, according to the ground-truth generated by the crowd assessor W_k.

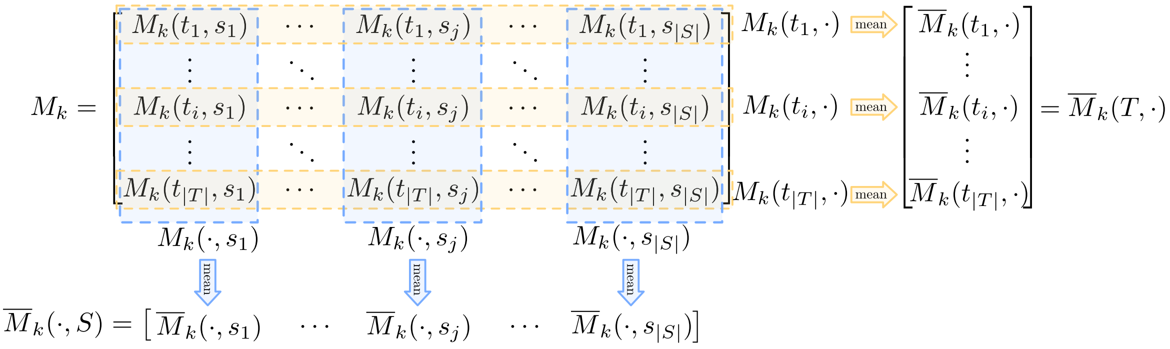


Fig. 4.1 Matrix notation for the assessor measure M_k.

The notation $M_k(\cdot, s)$ indicates a column vector containing all the performance figures for a given system s; the $\overline{M}_k(\cdot, s)$ indicates the average of the previous column vector; $\overline{M}_k(\cdot, S)$ indicates the average across the rows for all the systems; similarly, $M_k(t, \cdot)$, $\overline{M}_k(t, \cdot)$, and $\overline{M}_k(T, \cdot)$ indicate a row vector containing all the performance figures for a given topic t, its average, and the average across the columns for all the topics. Finally, the notation $M_k(:)$ indicates the linearization of the matrix, i.e. the row-wise concatenation of all its elements. A visualization of this matrix notation is reported in Figure 4.1.

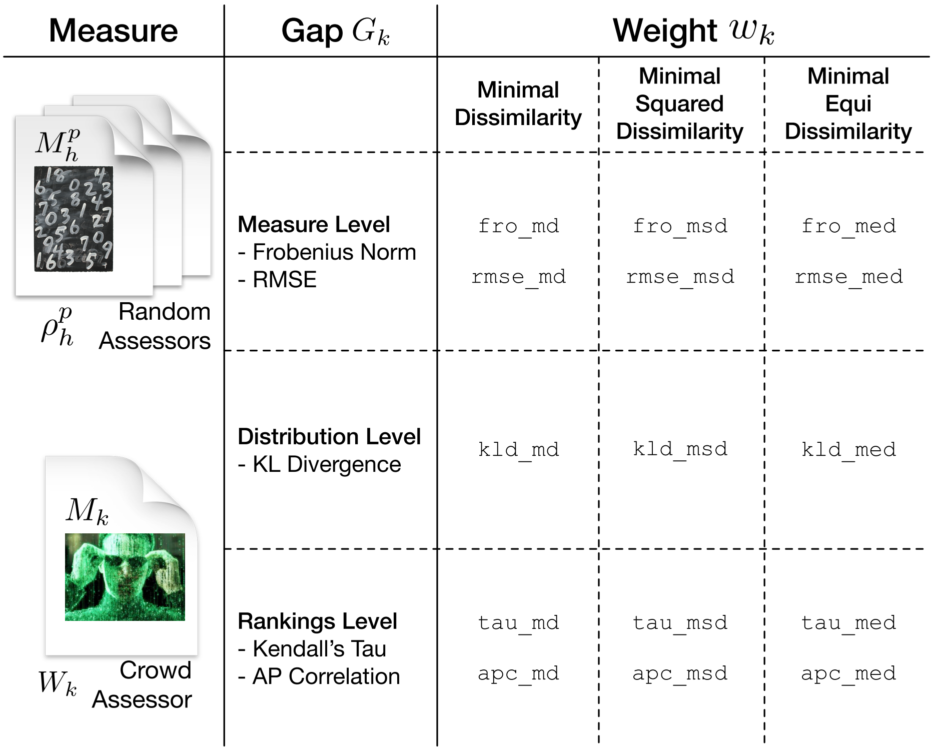


Fig. 4.2 Approach to determine the accuracy of a crowd assessor W_k with respect to a random assessors ρ_h^p.

For example, in the case of AP, each cell of AP_k contains the values of AP for system s on topic t according to assessor W_k; if we average over the topics $\overline{M}_k(\cdot, S)$, we obtain the MAP for all the systems $s \in S$ according to assessor W_k.

4.3.2 Intuitive Overview

Figure 4.2 shows the main steps (granularity, gap and weight) we use to estimate the accuracy of a crowd assessor and the different estimators we can obtain by combining the various alternatives at each step. The basic idea is to compare the crowd assessor against a set of random assessors and how “different” this crowd assessor is from the random ones, i.e. how much better she is.

For each pool we generate, ρ_h^p, $h = 1, 2, \ldots, H$, a set of random assessors of level p, i.e. which randomly evaluate as relevant the p per cent of the documents in the pool. As above, each of these random assessors gives origin to an assessor measure M_h^p for a given performance measure $\mathrm{m}(\cdot)$. We consider three different classes of random assessors, each of which contains a set of H random replicates:

- **uniform random assessor** $\rho_h^{\mathtt{uni}}$: this tosses a coin to judge a document, i.e. $p = 0.5$;

- **underestimating random assessor** $\rho_h^{\texttt{und}}$: this tends to judge documents as non relevant, e.g. $p = 0.05$;
- **overestimating random assessor** $\rho_h^{\texttt{ovr}}$: this tends to judge documents as relevant, e.g. $p = 0.95$.

Note that the idea of generating random assessors resembles [Soboroff et al., 2001] when they investigated the impact of random assessors compared to real assessors. However, to generate the random assessors [Soboroff et al., 2001] used a normal distribution with a proportion of relevant/not relevant documents derived by the same proportion in the case of real assessors. In our case, being a fully unsupervised approach, we do not have the real proportion of relevant documents available; when it comes to the distribution to be used, we chose the uniform distribution to avoid any assumption on assessor behavior, but a normal distribution or others could be an interesting future exploration.

Similarly, the approaches proposed by [Carterette and Soboroff, 2010; Li and Smucker, 2014] to simulate different types of assessors and different types of assessor errors cannot be applied in this unsupervised context, since they both start from a gold standard ground-truth and modify the assigned labels according to some desired distribution of truly/falsely relevant/not relevant documents. Even in [Moshfeghi et al., 2016] the authors present a way of simulating assessors based on a probabilistic approach, however they are interested in simulating the time that each assessor spends in completing a task.

Therefore, the intuitive idea described above boils down to determining some sort of "difference" between the measure M_k of a crowd assessor W_k and those M_h^p of the three random assessors ρ_h^p and turning this "difference" into an estimated accuracy $a_t(k)$ assigned to the crowd assessor W_k to compute the AWARE version of the performance measure $\mathrm{m}(\cdot)$. This is achieved in two main steps:

- **gap** G_k: this quantifies what "different" means. We consider three alternatives:
  - *measure level*: this operates directly on the assessor measures by computing either the Frobenius norm[4] of their difference (labelled `fro`, see Section 4.3.3) or their Root Mean Square Error (RMSE) (labelled `rmse`, see Section 4.3.3);
  - *distribution level*: this works on the performance distributions estimated from the assessor measures by using Kernel Density Estimation (KDE) and computes the Kullback-Leibler Divergence (KLD) between them (labelled `kld`, see Section 4.3.3);

[4] We used the Frobenius norm because it is the Euclidean norm in the space $\mathbb{R}^{n \times m}$ and it has many desirable properties, such as invariance under rotations, which makes it robust for our purposes.

  - *rankings level*: this considers the system rankings induced by the assessor measures and compares them by using either the Kendall's tau correlation (labelled `tau`, see Section 4.3.3) or the AP correlation (labelled `apc`, see Section 4.3.3);

- **weight** w_t^k: this turns the gap computed in the previous step into an estimated accuracy to be assigned to a crowd assessor. In particular, we reason in terms of *dissimilarity* from random assessors since, for a crowd assessor W_k, being close to a random one ρ_h^p can be considered as an indicator of her/his poor quality. We have three alternatives:

  - *minimal dissimilarity* (labelled `md`, see Section 4.3.4): this computes a weight which is proportional to the minimum gap from one of the random assessors (uniform, underestimating, and overestimating), i.e. the closer to one of the random assessors, the smaller the weight;
  - *minimal squared dissimilarity* (labelled `msd`, see Section 4.3.4): this is similar to the previous case but uses the minimum squared gap;
  - *minimal equi-dissimilarity* (labelled `med`, see Section 4.3.4): this computes a weight which is proportional to the crowd assessor being equally distant from all three random assessors (uniform, underestimating, and overestimating).

For each of the three random assessor classes, we generate a set of H replicates to cope with the uncertainty of the random generation process and to obtain better estimates. Therefore, for each crowd assessor W_k, we obtain a set of H estimates and we need to aggregate them into a single one; we compute a mean gap $\bar{G}_k$, averaging over the set of H gaps computed with respect to each random assessor ρ_h^p.

Finally, the described procedure produces an estimated accuracy $a_t(k)$ to be assigned to a crowd assessor W_k for each topic $t \in T$; this is what we call **topic-by-topic score granularity**, labelled `tpc`. However, we are also interested in the case when a single accuracy score is assigned to a crowd assessor W_k, i.e. when the $a_t(k)$ are the same for all the topics; this is what we call **single score granularity**, labelled `sgl`.

4.3.3 Gap

Frobenius Norm

Given an $m \times n$ matrix A, its Frobenius norm [Golub and Van Loan, 2012] is:

$$||A||_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} \tag{4.4}$$

which is also equal to the square root of the matrix trace $||A||_F = \sqrt{\mathrm{Tr}(AA^H)}$, where A^H is the transpose conjugate of A.

Single Score Granularity This is given by the Frobenius norm of the matrices of the crowd and random assessor measures, as defined below:

$$G_k^p = ||M_k - M_h^p||_F \tag{4.5}$$

Topic Score Granularity For each topic $t \in T$, this is given by the Frobenius norm of row vectors of the crowd and random assessor measures for that topic, as defined below:

$$G_k^p(t) = ||M_k(t,\cdot) - M_h^p(t,\cdot)||_F \tag{4.6}$$

Root Mean Square Error

Given two m elements vectors X and Y, their Root Mean Square Error (RMSE) [Kenney and Keeping, 1954] is:

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{m} \frac{(X_i - Y_i)^2}{m}} \tag{4.7}$$

Note that $\mathrm{RMSE} = \frac{1}{\sqrt{m}}||X - Y||_F$.

Single Score Granularity This is given by the RMSE of the vectors of the crowd and random assessor measures averaged by topic, as defined below:

$$G_k^p = \mathrm{RMSE}\big(\overline{M}_k(\cdot,S) - \overline{M}_h^p(\cdot,S)\big) \tag{4.8}$$

Topic Score Granularity For each topic $t \in T$, this is given by the RMSE of row vectors of the crowd and random assessor measures for that topic, as defined below:

$$G_k^p(t) = \mathrm{RMSE}\big(\overline{M}_k(t,\cdot) - \overline{M}_h^p(t,\cdot)\big) \tag{4.9}$$

KL Divergence

To compute the Kullback-Leibler Divergence (KLD) [Kullback and Leibler, 1951], we need the Probability Density Function (PDF) of the performance measures, which we estimate by using a Kernel Density Estimation (KDE) [Wand and Jones, 1995] approach.

Given a vector X of m elements, the KDE estimation of its PDF is given by

$$\hat{f}_X(x) = \frac{1}{m\bar{b}} \sum_{i=1}^{m} K\left(\frac{x - X_i}{\bar{b}}\right) \tag{4.10}$$

where $\bar{b}$ is a positive number called bandwidth or window width; $K(\cdot)$ is the kernel satisfying $\int_{-\infty}^{+\infty} K(x)dx = 1$.

Given two m elements vectors X and Y, the KLD between their PDFs is given by

$$D_{KL}\big(X\big|\big|Y\big) = \sum_{x} \ln\left(\frac{\hat{f}_X(x)}{\hat{f}_Y(x)}\right) \hat{f}_X(x) \tag{4.11}$$

$D_{KL} \in [0, +\infty)$ denotes the information lost when Y is used to approximate X [Burnham and Anderson, 2002]; therefore, 0 means that there is no loss of information and, in our settings, it will mean that two assessors are considered the same; $+\infty$ means that there is full loss of information and, in our settings, it will mean that two assessors are considered completely different. Note that D_{KL} is not symmetric and so, in general, $D_{KL}\big(X\big|\big|Y\big) \neq D_{KL}\big(Y\big|\big|X\big)$.

Single Score Granularity This is given by the KLD of the vectors of the crowd and random assessor linearize measures, as defined below:

$$G_k^p = D_{KL}\big(M_k(:)\big|\big|M_h^p(:)\big) \tag{4.12}$$

Topic Score Granularity For each topic $t \in T$, this is given by the KLD of row vectors of the crowd and random assessor measures for that topic, as defined below:

$$G_k^p(t) = D_{KL}\big(M_k(t,\cdot)\big|\big|M_h^p(t,\cdot)\big) \tag{4.13}$$

Kendall's Tau Correlation

Given two m elements vectors X and Y, their Kendall's τ correlation [Kendall, 1948] is given by

$$\tau\big(X,Y\big) = \frac{C - D}{m(m-1)/2} \tag{4.14}$$

where C is the total number of concordant pairs (pairs that are ranked in the same order in both vectors) and D the total number of discordant pairs (pairs that are ranked in opposite order in the two vectors).

Single Score Granularity This is given by the τ correlation of the vectors of the crowd and random assessor measures averaged by topic, as defined below:

$$G_k^p = \tau\big(\overline{M}_k(\cdot,S) - \overline{M}_h^p(\cdot,S)\big) \tag{4.15}$$

Topic Score Granularity For each topic $t \in T$, this is given by the τ correlation of row vectors of the crowd and random assessor measures for that topic, as defined below:

$$G_k^p(t) = \tau\big(M_k(t,\cdot), M_h^p(t,\cdot)\big) \tag{4.16}$$

AP Correlation

AP correlation τ_{ap} [Yilmaz et al., 2008] is a correlation coefficient inspired by the Kendall's τ correlation, but it puts more emphasis on the order of the top ranked systems.

Given two m elements vectors X and Y, their AP correlation is given by

$$\tau_{ap}\big(Y,X\big) = \frac{2}{m-1}\sum_{i=2}^{m}\frac{C(i)}{i-1} - 1 \tag{4.17}$$

where $C(i)$ is the number of items above rank i in X and correctly ranked with respect to the item at rank i in Y, which acts as a reference. Note that τ_{ap} is not symmetric and so, in general, $\tau_{ap}\big(Y,X\big) \neq \tau_{ap}\big(X,Y\big)$.

Note that τ_{ap} does not handle tied values in the two vectors, so we adopt the same approach suggested in the TREC 2013 Crowdsourcing track [Smucker et al., 2014] where, in case of ties, they sample over possible orders and average the obtained τ_{ap} coefficients.

Single Score Granularity This is given by the τ_{ap} correlation of the vectors of the crowd and random assessor measures averaged by topic, as defined below:

$$G_k^p = \tau_{ap}\big(\overline{M}_k(\cdot,S), \overline{M}_h^p(\cdot,S)\big) \tag{4.18}$$

Topic Score Granularity For each topic $t \in T$, this is given by the τ_{ap} correlation of row vectors of the crowd and random assessor measures for that topic, as defined below:

$$G_k^p(t) = \tau_{ap}\big(M_k(t,\cdot), M_h^p(t,\cdot)\big) \tag{4.19}$$

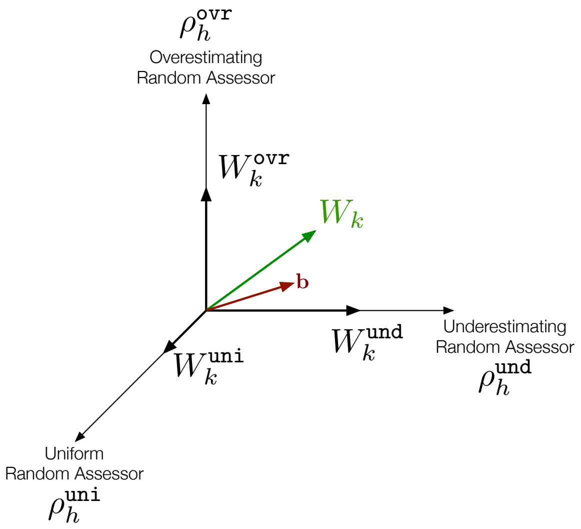


Fig. 4.3 Vector space representation of the crowd assessor W_k and the random assessors ρ_h^p.

4.3.4 Weight

As anticipated above, the basic idea is to understand how close a crowd assessor W_k is to a random one ρ_h^p and consider this as an indicator of being a poor quality assessor. Therefore, we are interested in reasoning in terms of dissimilarity from random assessors: the farther away from a random assessor the higher the accuracy assigned to a crowd assessor.

As shown in Figure 4.3, we can create a vector space whose base is given by the three random assessors ρ_h^p, represent each crowd assessor W_k in this space, and project the crowd assessor on the random assessors (indicated by $W_k^{\texttt{uni}}$, $W_k^{\texttt{ovr}}$, and $W_k^{\texttt{und}}$ respectively); **b** is the bisector of the first quadrant. Note that the projections of the crowd assessor on the random assessors are given by the gaps described above and properly normalized as discussed in the following section.

Normalization

When you reason in terms of similarity between vectors, if two vectors **v** and **w** are equal, then the norm of $\mathbf{v} - \mathbf{w}$ will be equal to 0, i.e. 0 means equal. However, in the vector space of Figure 4.3, we reason in terms of dissimilarity between vectors: 0 means different from random assessor and 1 means equal to random assessor. Therefore, in the following section, first we normalize all the gaps to the range $[0,1]$; then, when needed, we also transform them, e.g. by reversing the $[0,1]$ range, to ensure that these normalized gaps have the expected meaning of 0 “different from random assessor” and 1 “equal to random assessor”.

Frobenius Norm The Frobenius norm is in the $\big[0, \sqrt{|T|\cdot|S|}\big]$ range, where 0 means equal to a random assessor. So we need to divide it by its maximum and reverse it so that 0 means different from a random assessor:

$$G' = 1 - \frac{G}{\sqrt{|T|\cdot|S|}} \tag{4.20}$$

Note that when we consider the single score G_k, the equation holds as above; if we consider the topic score $G_k(t)$ we have to set $|T| = 1$ in the above equation.

Root Mean Square Error The RMSE is in the $\big[0, 1\big]$ range, where 0 means equal to a random assessor. So we need to reverse it so that 0 means different from a random assessor:

$$G' = 1 - G \tag{4.21}$$

KL Divergence The KLD is in the $\big[0, \infty\big)$ range, where 0 means equal to a random assessor. So we map it to the $\big(0, 1\big]$ range by the negative exponential so that 0 means different from a random assessor

$$G' = \mathrm{e}^{-\beta G} \tag{4.22}$$

where $\beta > 0$ is a positive real number.

Kendall's Tau Correlation The Kendall's τ correlation is in the $\big[-1, 1\big]$ range, where 0 means different from a random assessor, 1 means equal to a random assessor and -1 completely opposite to a random assessor[5]. We consider -1 as 1:

$$G' = \big|G\big| \tag{4.23}$$

AP Correlation The τ_{ap} correlation is in the $\big[-1, 1\big]$ range, where 0 means different from a random assessor, 1 means equal to a random assessor and -1 completely opposite to a random assessor[6]. We consider -1 as 1:

$$G' = \big|G\big| \tag{4.24}$$

[5] Consider an assessor that has correlation equal to -1 with one of the random assessors. This means that the assessor gives the exact opposite relevance judgement for each document. Therefore, this assessor can be considered a random assessor as well, and it is correct to give him a weight equal to 1.

[6] Same considerations as in the case of Kendall's τ hold here as well.

Minimal Dissimilarity

If we take the minimum between the dissimilarities of the assessor W_k from the random assessors, the assessor W_k cannot be closer to any of the random assessors more than this minimum. Therefore, we compute the minimum of the scalar products of the dissimilarity vector with the axes of the vector space shown in Figure 4.3:

$$w_k = \min\left(\left(G_k^{\texttt{und}}\right)', \left(G_k^{\texttt{uni}}\right)', \left(G_k^{\texttt{ovr}}\right)'\right) \tag{4.25}$$

Minimal Squared Dissimilarity

We reason as in the previous case, but we consider the square of the gaps to have steeper behaviour:

$$w_k = \min\left(\left(\left(G_k^{\texttt{und}}\right)'\right)^2, \left(\left(G_k^{\texttt{uni}}\right)'\right)^2, \left(\left(G_k^{\texttt{ovr}}\right)'\right)^2\right) \tag{4.26}$$

Minimal Equi-Dissimilarity

The bisector vector **b** represents the direction with the greatest equal dissimilarity from all the random assessors at the same time. Therefore, the closer the crowd assessor W_k is to the bisector **b**, the farther away she/he is from all the random assessors at the same time. The scalar product between the crowd assessor vector and the bisector represents this quantity:

$$w_k = \left(G_k^{\texttt{und}}\right)' + \left(G_k^{\texttt{uni}}\right)' + \left(G_k^{\texttt{ovr}}\right)' \tag{4.27}$$

4.3.5 Summary

Algorithm 2 shows the pseudo-code for computing the estimated accuracy of a crowd assessor W_k in the case of the single score granularity, while Algorithm 3 describes the case of the topic-by-topic score granularity. The inputs of the algorithms are the ground-truth produced by the crowd assessor W_k, i.e. the relevance judgments assigned by crowd assessor W_k, the ground-truths generated by each replicate of the random assessors with level p equals to 0.5 (`uni`), 0.05 (`und`) and 0.95 (`ovr`) and the performance measure to be computed. As output the algorithm will give the accuracy a_k for the crowd assessor W_k, which will be a single number for the single score granularity and a vector of length $|T|$ for the topic score granularity.

Firstly the performance measure is computed on the ground-truth provided by the crowd assessor W_k and by the H replicates of the three types of random assessors, obtaining

ALGORITHM 2: How to estimate assessor accuracy a_k for the single score granularity.

```
Data: $r_t^k$ ground-truth generated by the $k$-th assessor; $r_h^p$ ground-truth generated by the $h$-th random assessor of
      level $p$, where $h \in \{1,\dots,H\}$ and $p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$; $\mathrm{m}(\cdot)$ performance measure
Result: $a_k$ single score granularity accuracy for the $k$-th assessor;
/* Compute the performance measure $M_k$ for the $k$-th assessor and $M_h^p$ for each random
   assessors                                                                              */
$M_k \leftarrow$ compute $\mathrm{m}(\cdot)$ on $r_t^k$;
$M_h^p \leftarrow$ compute $\mathrm{m}(\cdot)$ on $r_h^p$, $\forall\, h \in \{1,\dots,H\}$ and $\forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
/* Compute the Gap $G_{k,h}^p$ with respect to each random assessor:  $h \in \{1,\dots,H\}$ and
   $p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$                                     */
for $h \in \{1,\dots,H\}$ do
    if measure level then
        if Frobenius norm then
            $G_{k,h}^p = \left|\left|M_k - M_h^p\right|\right|_F \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
            $(G_{k,h}^p)' = 1 - \frac{G_{k,h}^p}{\sqrt{|S|}} \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$
        else if RMSE then
            $G_{k,h}^p = \mathrm{RMSE}\big(\overline{M}_k(\cdot,S) - \overline{M}_h^p(\cdot,S)\big) \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
            $(G_{k,h}^p)' = 1 - G_{k,h}^p \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
        end
    else if distribution level then
        $G_{k,h}^p = D_{KL}\big(M_k(:)\big|\big|M_h^p(:)\big) \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
        $(G_{k,h}^p)' = \mathrm{e}^{-\beta G_{k,h}^p} \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
    else if ranking level then
        if Kendall's Tau then
            $G_{k,h}^p = \tau\big(\overline{M}_k(\cdot,S) - \overline{M}_h^p(\cdot,S)\big) \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
            $(G_{k,h}^p)' = \big|G_{k,h}^p\big| \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
        else if AP Correlation then
            $G_{k,h}^p = \tau_{ap}\big(\overline{M}_k(\cdot,S), \overline{M}_h^p(\cdot,S)\big) \quad \forall\, r \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
            $(G_{k,h}^p)' = \big|G_{k,h}^p\big| \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
        end
    end
end
/* Aggregate the Gap with respect to the random assessor replicates                       */
$(G_k^p)' \leftarrow \mathrm{mean}\big((G_{k,h}^p)'\big) \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
/* Compute the weight $w_k$                                                               */
if minimal dissimilarity then
    $w_k = \min\Big(\big(G_k^{\mathtt{und}}\big)', \big(G_k^{\mathtt{uni}}\big)', \big(G_k^{\mathtt{ovr}}\big)'\Big)$;
else if minimal squared dissimilarity then
    $w_k = \min\left(\Big(\big(G_k^{\mathtt{und}}\big)'\Big)^2, \Big(\big(G_k^{\mathtt{uni}}\big)'\Big)^2, \Big(\big(G_k^{\mathtt{ovr}}\big)'\Big)^2\right)$;
else if minimal equi-dissimilarity then
    $w_k = \big(G_k^{\mathtt{und}}\big)' + \big(G_k^{\mathtt{uni}}\big)' + \big(G_k^{\mathtt{ovr}}\big)'$;
end
```

ALGORITHM 3: How to estimate assessor accuracy a_k for the topic-by-topic score granularity.

```
Data: $r_t^k$ ground-truth generated by the $k$-th assessor; $r_h^p$ ground-truth generated by the $h$-th random assessor of
    level $p$, where $h \in \{1,\dots,H\}$ and $p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$; $\mathrm{m}(\cdot)$ performance measure
Result: $a_k$ vector of length $|T|$ containing the topic score granularity accuracy for the $k$-th assessor;
/* Compute the performance measure $M_k$ for the $k$-th assessor and $M_h^p$ for each random
   assessors                                                                               */
$M_k \leftarrow$ compute $\mathrm{m}(\cdot)$ on $r_t^k$;
$M_h^p \leftarrow$ compute $\mathrm{m}(\cdot)$ on $r_h^p$, $\forall\, h \in \{1,\dots,H\}$ and $\forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
/* Compute the Gap $G_{k,h}^p(t)$ with respect to each random assessor:  $h \in \{1,\dots,H\}$ and
   $p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$                                       */
for $t \in \{1,\dots,|T|\}$ do
    for $h \in \{1,\dots,H\}$ do
        if measure level then
            if Frobenius norm then
                $G_{k,h}^p(t) = \big|\big|M_k(t,\cdot) - M_h^p(t,\cdot)\big|\big|_F \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
                $(G_{k,h}^p(t))' = 1 - \frac{G_{k,h}^p(t)}{\sqrt{|T|\cdot|S|}} \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$
            else if RMSE then
                $G_{k,h}^p(t) = \mathrm{RMSE}\big(\overline{M}_k(t,\cdot) - \overline{M}_h^p(t,\cdot)\big) \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
                $(G_{k,h}^p(t))' = 1 - G_{k,h}^p(t) \quad \forall\, r \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
            end
        else if distribution level then
            $G_{k,h}^p(t) = D_{KL}\big(M_k(t,\cdot)\big|\big|M_h^p(t,\cdot)\big) \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
            $(G_{k,h}^p(t))' = \mathrm{e}^{-\beta G_{k,h}^p(t)} \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
        else if ranking level then
            if Kendall's Tau then
                $G_{k,h}^p(t) = \tau\big(M_k(t,\cdot), M_h^p(t,\cdot)\big) \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
                $(G_{k,h}^p(t))' = \big|G_{k,h}^p(t)\big| \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
            else if AP Correlation then
                $G_{k,h}^p(t) = \tau_{ap}\big(M_k(t,\cdot), M_h^p(t,\cdot)\big) \quad \forall\, r \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
                $(G_{k,h}^p(t))' = \big|G_{k,h}^p(t)\big| \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$;
            end
        end
    end
end
/* Aggregate the Gap with respect to the random assessor replicates                       */
$(G_k^p(t))' \leftarrow \mathrm{mean}\big((G_{k,h}^p(t))'\big) \quad \forall\, p \in \{\mathtt{uni},\mathtt{und},\mathtt{ovr}\}$ and $\forall\, t \in \{1,\dots,|T|\}$;
/* Compute the weight $w_k$                                                                */
for $t \in \{1,\dots,|T|\}$ do
    if minimal dissimilarity then
        $w_k(t) = \min\Big(\big(G_k^{\mathtt{und}}(t)\big)', \big(G_k^{\mathtt{uni}}(t)\big)', \big(G_k^{\mathtt{ovr}}(t)\big)'\Big)$;
    else if minimal squared dissimilarity then
        $w_k(t) = \min\left(\Big(\big(G_k^{\mathtt{und}}(t)\big)'\Big)^2, \Big(G_k^{\mathtt{uni}}(t)\big)'\Big)^2, \Big(G_k^{\mathtt{ovr}}(t)\big)'\Big)^2\right)$;
    else if minimal equi-dissimilarity then
        $w_k(t) = \big(G_k^{\mathtt{und}}(t)\big)' + \big(G_k^{\mathtt{uni}}(t)\big)' + \big(G_k^{\mathtt{ovr}}(t)\big)'$;
    end
end
```

respectively the $|T| \times |S|$ matrices M_k and M_h^p. Then the gap between the crowd assessor W_k and the random assessors is computed with respect to the strategies previously described:

- *Measure Level*:
  - Frobenius Norm: Equation (4.5) for single score granularity and Equation (4.6) for topic score granularity, Equation (4.20) to normalize the accuracy;
  - Root Mean Square Error: Equation (4.8) for single score granularity and Equation (4.9) for topic score granularity, Equation (4.21) to normalize the accuracy;
- *Distribution Level*:
  - KL Divergence: Equation (4.12) for single score granularity and Equation (4.13) for topic score granularity, Equation (4.22) to normalize the accuracy;
- *Ranking Level*:
  - Kendall's τ: Equation (4.15) for single score granularity and Equation (4.16) for topic score granularity, Equation (4.23) to normalize the accuracy;
  - AP Correlation: Equation (4.18) for single score granularity and Equation (4.19) for topic score granularity, Equation (4.24) to normalize the accuracy;

Finally, the normalized Gap is averaged over the H replicates of each random assessors class and the weight of the crowd assessor W_k is computed with respect to one of the following methods:

- *Minimal Dissimilarity*: Equation (4.25);
- *Minimal Squared Dissimilarity*: Equation (4.26);
- *Minimal Equi-Dissimilarity*: Equation (4.27).

4.4 Experimental Setup

4.4.1 Crowd Assessors Collection

We use the TREC 21, 2012, Crowdsourcing [Smucker et al., 2013] data sets developed in the Text Relevance Assessing Task (TRAT). The TRAT required participating groups to simulate the relevance assessing role of the NIST for 10 of the TREC 08, 1999, Ad-hoc topics [Voorhees and Harman, 1999], using binary relevance. Participating groups had to submit a binary relevance judgment for every document in the judging pools of the ten topics. The 10 topics selected were: 411, 416, 417, 420, 427, 432, 438, 445, 446, and 447. In total 33 pools were submitted to TRAT; we excluded two of them (`INFLB2012` and `Orc2Stage`) because, for some topics, they did not assess any document as relevant; indeed, this prevents the computation of some evaluation measures because you lack the information about the recall base. Therefore, we actually used 31 out the 33 submitted pools for TRAT.

In TRAT, the majority vote of the submitted pools was compared to the NIST relevance judgments; when the majority vote differed from the NIST judgment, TRAT organizers adjudicated the final relevance judgment for a document. The TRAT adjudicated pool constitutes the gold standard for our experimentation.

4.4.2 Evaluation Measures

When it comes to measures for evaluating the effectiveness of the different approaches, we adopt two criteria used in the TREC 22, 2013, Crowdsourcing track [Smucker et al., 2014]:

- *rank correlation*: we use AP correlation [Yilmaz et al., 2008] to compare the ranking of the systems produced for a given performance measure $\mathrm{m}(\cdot)$ computed over the gold standard with respect to the ranking produced for the same performance measure computed over the ground-truth generated by one of the approaches under examination;

- *score accuracy*: in addition to correctly ranking systems, it is important that the performance scores are as accurate as possible. To this end, for a given performance measure $\mathrm{m}(\cdot)$, we use the RMSE between the performance measure computed over the gold standard and the one computed over the ground-truth created by one of the approaches under examination.

Note that the above use of AP correlation and RMSE is not related to their use as gaps between assessors, explained in Section 4.3; here they are used as evaluation measures for comparing the different algorithms and methods under examination. Moreover, we do not adopt some of the evaluation measures used in the TREC Crowdsourcing tracks, such as the Logistic Average Misclassification (LAM) rate [Cormack and Lynam, 2005] and the Area Under the ROC Curve (AUC) [Fawcett, 2006], because these measures specifically deal with classification tasks and basically compare the assigned relevance labels, but this does not apply to our case because AWARE does not generate relevance labels.

4.4.3 Performance Measures

When it comes to the assessor measures M_k and M_h^p, we consider the following performance measures presented in Chapter 2:

- Average Precision (AP) [Buckley and Voorhees, 2005], computed as in Equation (2.1), represents the "gold standard" measure in IR, known to be stable and informative, with a natural top-heavy bias and an underlying theoretical basis as approximation of the area under the precision/recall curve [Robertson et al., 2010];

- Normalized Discounted Cumulated Gain (nDCG) [Järvelin and Kekäläinen, 2002] discounts the gain provided by each relevant retrieved document proportionally to the rank at which it is retrieved as in Equation (2.2). We use nDCG@20, which is calculated up to rank position 20.

- Expected Reciprocal Rank (ERR) [Chapelle et al., 2009], computed as in Equation (2.3), is a particularly top-heavy measure since it highly penalizes systems placing not-relevant documents in high positions, as shown in Section 3.5. We use ERR@20.

4.4.4 Systems

Two TREC Adhoc tracks used these 10 topics over the years: the TREC 08, 1999, Ad-hoc track [Voorhees and Harman, 1999] (labeled `T08`), which contains 129 runs and from which these topics were selected; and, the TREC 13, 2004, Robust track [Voorhees, 2015] (labeled `T13`), which contains 110 runs and whose goal was to specifically experiment against hard topics.

Both `T08` and `T13` adopt a corpus of about 528K news documents, i.e. disk 4 and 5 of the TIPSTER collection minus the Congressional Record.

4.4.5 Parameters Setup

For nDCG we use a log base $b = 2$ and gains 0 and 5 for not relevant and relevant documents, respectively. For ERR we use gains 0 and 5 for not relevant and relevant documents, respectively.

We generate $H = 1,000$ replicates of the random assessors in each class – uniform, underestimating and overestimating assessors.

Let $l = 31$ be the total number of available crowd assessors and $k < l$ the number of assessors we are merging using the AWARE framework or other approaches. For each of the above evaluation measures, we experimented all the $k = 2, 3, \dots, 30$. For each value of

k, there are $\binom{31}{k} = \frac{31!}{k!(31-k)!}$ possible ways of choosing the k assessors to be merged; we randomly sampled 1,000 k-tuples out of the $\binom{31}{k}$ possible ones. The evaluation measures we report – AP correlation and RMSE – are averaged over these 1,000 samples.

For the computation of AP correlation in the case of ties, we sample and average over 100 randomly generated orderings.

For the KDE of a performance measure in equation (4.10), we use 100 equally spaced values x in the range $[0,1]$, a Gaussian kernel $K(\cdot)$, and a bandwidth $b = 0.015$.

For the normalization of the KLD in equation (4.22), we set $\beta = 1$.

For the EM algorithms we set a limit of 1,000 iterations and a tolerance of 10^{-3}.

All the experiments were developed using the MATlab Toolkit for Evaluation of information Retrieval Systems (MATTERS) library[7] and their source code is publicly available[8] to favour reproducibility.

4.4.6 Experiments

We experiment all the combinations of factors for the estimation of a crowd assessor accuracy, as described in Section 4.3:

- **granularity**: whether, for a crowd assessor, we compute a single accuracy (`sgl`) or a separate accuracy for each topic (`tpc`);
- **gap**: how we compute the "difference" between a crowd and a random assessor (`fro`, `rmse`, `kld`, `tau`, or `apc`);
- **weight**: how we turn a "difference" between a crowd and a random assessor into a final accuracy estimation (`md`, `msd`, or `med`).

The combination of these three factors gives raise to 30 different approaches for estimating a crowd assessor accuracy. We introduce the following notation to facilitate the comprehension of the main characteristics of an estimator from its name:

```
<granularity>_<gap>_<weight>
```

So, for example, the tag `sgl_apc_med` indicates a single crowd assessor accuracy a_k for all the topics using AP correlation as "difference" between crowd and random assessors and the minimal equi-dissimilarity weighting criterion.

[7] http://matters.dei.unipd.it/

[8] https://bitbucket.org/frrncl/tois-aware

We consider three baselines, representing the state of the art: the MV algorithm, labeled `mv`, and two variants of the EM algorithm: `emmv`, i.e. EM seeded by the pool generated by the MV algorithm, and `emneu`, i.e. EM initialized using the worker confusion matrix, as explained in Section 4.1.

Finally, we experiment also a fourth baseline labeled `uni`, representing AWARE in absence of any information, i.e. using uniform accuracies for all the merged crowd assessors, as done in the toy example of Section 4.2.

We conduct the following experiments:

- a factorial analysis to isolate the contributions of different factors – k-tuple size, the performance measure under consideration, and the considered systems (Section 4.5). This analysis allows us to understand: (i) which approaches perform best across a wide range of influencing factors, net their effects; (ii) how these factors interact with each other;
- a break-down of the contribution of the different components of the AWARE estimators – namely granularity, gap, and weight (Section 4.6). This analysis allows us to dig into the AWARE estimators themselves and better understand how they work.

4.5 Factorial Analysis of Ktuple, Approach, Measure and System Effects

4.5.1 Methodology

The goal of this section is to conduct a deep analysis to investigate how the AWARE approaches and the state-of-the-art baselines behave with respect to different factors, namely the k-tuple size, the performance measure under consideration, and the considered systems. To this end, we adopt the following General Linear Mixed Model (GLMM) model for the three-way ANalysis Of VAriance (ANOVA) with repeated measures [Maxwell and Delaney, 2004; Rutherford, 2011]:

$$Y_{ijkl} = \underbrace{\mu_{....} + \kappa_i + \alpha_j + \beta_k + \gamma_l}_{\text{Main Effects}} + \underbrace{\alpha\beta_{jk} + \alpha\gamma_{jl} + \beta\gamma_{kl}}_{\text{Interaction Effects}} + \underbrace{\varepsilon_{ijkl}}_{\text{Error}} \tag{4.28}$$

where: Y_{ijkl} is the score of the i-th subject in the j-th, k-th, and l-th factors; $\mu_{....}$ is the grand mean; κ_i is the effect of the i-th subject, i.e. the k-tuple size $k = 2, \dots, 30$; α_j is the effect of the j-th factor, i.e. both the AWARE and the state-of-the-art approaches; β_k is the effect of the k-th factor, i.e. the performance measures under consideration, namely AP, nDCG@20, and ERR@20; and, γ_l is the effect of the l-th factor, i.e. the systems submitted to the T08 and T13 tracks. We consider also the interaction effects among approaches and performance measures ($\alpha\beta_{jk}$), approaches and systems ($\alpha\gamma_{jl}$), and performance measures and systems ($\beta\gamma_{kl}$). Finally, ε_{ijkl} is the error committed by the model in predicting the score of the i-th subject in the three factors j,k,l.

For each model, we report the ANOVA table which summarizes the outcomes of the ANOVA test on the above model indicating, for each factor, the Sum of Squares (SS), the Degrees of Freedom (DF), the Mean Squares (MS), the F statistics, and the p-value of that factor. In the following, we consider a confidence level $\alpha = 0.05$ to determine if a factor is statistically significant.

We are not only interested in determining whether a factor effect is significant, i.e. its p-value in the ANOVA table is less than 0.05, but also which proportion of the variance is due to it. Therefore, we need to estimate its *effect-size measure* or Strength of Association (SOA). The SOA is a "standardized index and estimates a parameter that is independent of sample size and quantifies the magnitude of the difference between populations or the relationship between explanatory and response variables" [Olejnik and Algina, 2003; Sakai, 2014b]. We use the $\hat{\omega}^2_{\langle fact \rangle}$ SOA:

$$\hat{\omega}^2_{\langle fact \rangle} = \frac{df_{fact}(F_{fact} - 1)}{df_{fact}(F_{fact} - 1) + N} \tag{4.29}$$

which is an unbiased estimator of the variance components associated with the sources of variation in the design, where N is the total number of elements under analysis.

The common rule of thumb [Murphy et al., 2014] when classifying $\hat{\omega}^2_{\langle fact \rangle}$ effect size is: 0.14 and above is a large effect, 0.06–0.14 is a medium effect, and 0.01–0.06 is a small effect. $\hat{\omega}^2_{\langle fact \rangle}$ values could happen to be negative and in such cases they are considered as zero.

In addition to the ANOVA table, we also show both the main effects and the interaction effects plots in order to get a better appreciation of the behaviour of the different levels of each factor. In particular, the main effects plot graphs the response mean for each factor level connected by a line. An interaction effects plot displays the levels of one factor on the X axis and has a separate line for the means of each level of the other factor on the Y axis; it allows us to understand whether the effect of one factor depends on the level of the other factor.

A *Type I* error occurs when a true null hypothesis is rejected and the significance level α is the probability of committing a Type I error. When performing multiple comparisons, the probability of committing a Type I error increases with the number of comparisons and we keep it controlled by applying the Tukey Honestly Significant Difference (HSD) test [Hochberg and Tamhane, 1987] with a significance level $\alpha = 0.05$. Tukey's method is used in ANOVA to create confidence intervals for all pairwise differences between factor

Table 4.1 ANOVA table for AP Correlation considering the k-tuple size, approach, measure and systems effects.

Source	SS	DF	MS	F	p-value	$\hat{\omega}^2_{\langle fact \rangle}$
K-tuple Size	3.5161	28	0.1256	580.7705	< 0.0001	
Approach	1.2264	33	0.0372	171.8716	< 0.0001	0.4880
Measure	13.0727	2	6.5364	30,230.1290	< 0.0001	0.9109
Systems	1.9857	1	1.9857	9,183.9134	< 0.0001	0.6082
Approach*Measure	2.0701	66	0.0314	145.0584	< 0.0001	0.6164
Approach*Systems	0.3008	33	0.0091	42.1620	< 0.0001	0.1867
Measure*Systems	5.3240	2	2.6620	12,311.4096	< 0.0001	0.8063
Error	1.2433	5,750	0.0002			
Total	28.7391	5,915				

levels, while controlling the family error rate. Two levels u and v of a factor are considered significantly different when

$$|t| = \frac{|\hat{\mu}_u - \hat{\mu}_v|}{\sqrt{MS_{error}\left(\frac{1}{n_u} + \frac{1}{n_v}\right)}} > \frac{1}{\sqrt{2}} q_{\alpha,k,N-k} \quad (4.30)$$

where $\hat{\mu}_u$ and $\hat{\mu}_v$ are the marginal means, i.e. the main effects, of the two factors; n_u and n_v are the sizes of levels u and v; $q_{\alpha,k,N-k}$ is the upper $100 * (1 - \alpha)$th percentile of the studentized range distribution with parameter k and $N - k$ degrees of freedom; k is the number of levels in the factor and N is the total number of observations.

In the following, we have a section dedicated to each evaluation measure, i.e. AP correlation and RMSE.

Note that when we analyse AP correlation, we can use the data as they are, since all the scores are in the same range $[0, 1]$ and they are measured in the same way. On the other hand, when we analyse RMSE, even if all the measures are in the range $[0, 1]$ and so also RMSE is, AP $= 0.20$ is not exactly the same as ERR@20 $= 0.20$ because of their different user models and they typically assume different values in the range $[0, 1]$. As a consequence, an RMSE 0.15 for AP is not directly comparable with an RMSE 0.15 for ERR@20. Therefore, we need to apply some kind of normalization first to make the scores comparable and we normalize them by the maximum value achieved on the dataset, thus reasoning in term of ratios.

4.5.2 AP Correlation

Table 4.1 shows that all the main and interaction effects are statistically significant. As far as main effects are concerned, we can see that Measure is a large size effect and it explains the largest share of variance; Systems is a large size effect as well and it is the second largest main effect; finally, also Approach is a large size effect but about 2 times smaller than Measure effect and 1.25 times smaller than Systems effect. Overall, this supports the intuition that led to the development of the AWARE framework: performance Measures and Systems effects do matter a lot when merging assessors and they should be taken into the play, instead of optimizing upstream, as also illustrated in the toy example at the beginning of this Chapter.

When it comes to the interaction effects, Approach*Measure is a large size effect, about 1.27 times greater than the Approach effect alone, while Approach*Systems is a large size effect but less than half the Approach effect alone. These two facts further strengthen the intuition behind AWARE: not only do Measures and Systems effects play an important role alone, they also influence and interact a lot with the Approaches for merging assessors, where Measures have a greater impact on Approaches than Systems.

Finally, there is also a large size interaction effect between Measure and Systems, indicating that different measures score systems differently, but this is less interesting for the purposes of the present discussion because it is an intrinsic phenomenon of the relationship between performance measures and systems.

The main effects plot in Figure 4.4 shows the marginal mean contributions of each effect together with their confidence interval (shaded). Figure 4.4(a) shows the contributions of the different approaches across all the conditions and net of their effects, thus allowing us to appreciate the best and most stable approaches in many operational settings. We can see that the AWARE approaches lie in a somehow stable range of performances, with the only exception of sgl_rmse_msd which is the worst performing one but still better than emmv and emneu.

As expected, we can observe from Figure 4.4(b) that increasing the number of merged assessors improves the performances; you can also note how the confidence interval slightly increases as the k-tuple size increases, denoting a higher variability due to the larger number of (potentially heterogenous) assessors merged.

Figure 4.4(c) shows how the different performance measures lead to quite different performances when it comes to merging assessors and, in particular, nDCG@20 and ERR@20 are more challenging than AP. Finally, Figure 4.4(d) highlights how the targeted systems affect the performances as well, with the T13 ones somehow being more difficult.

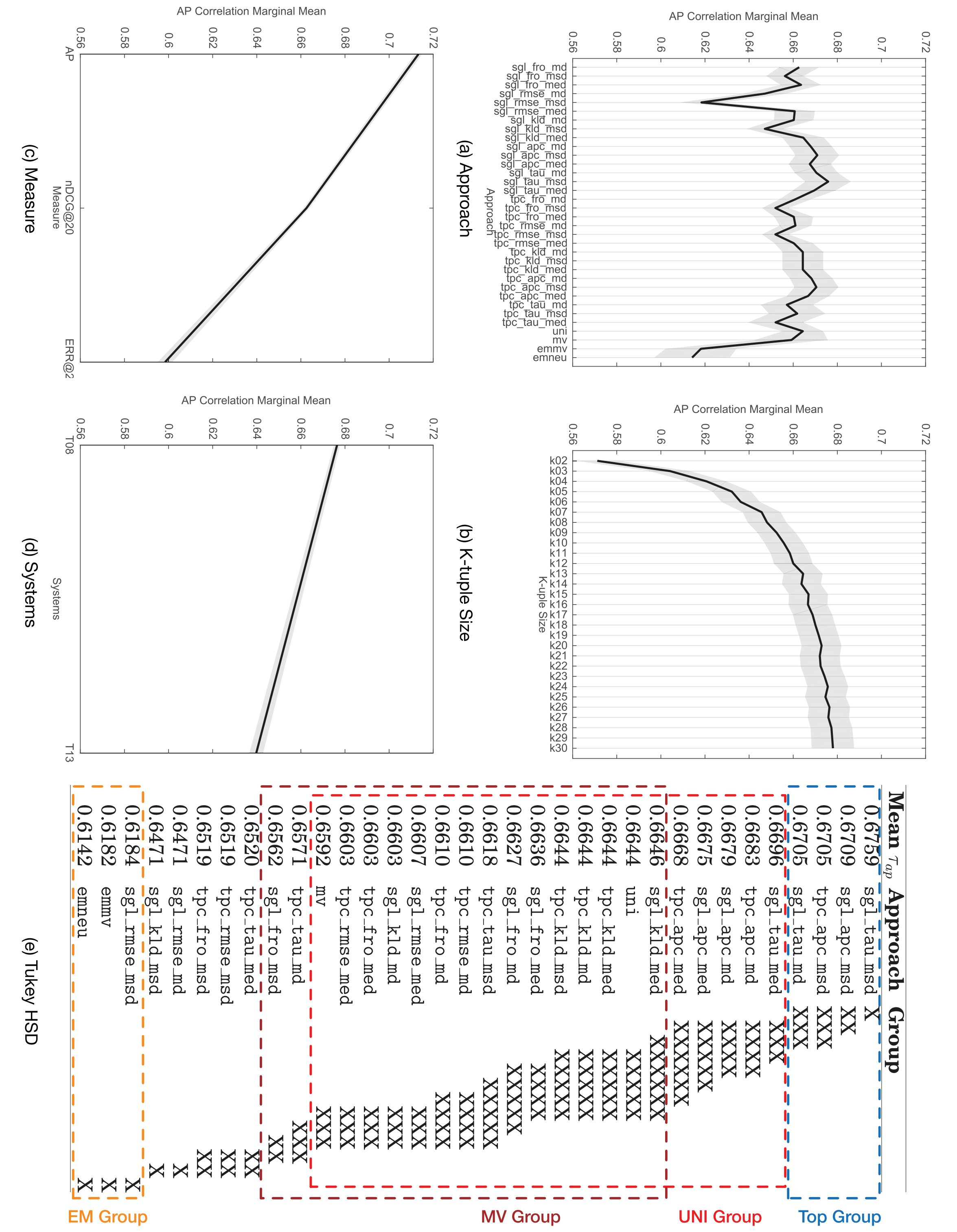


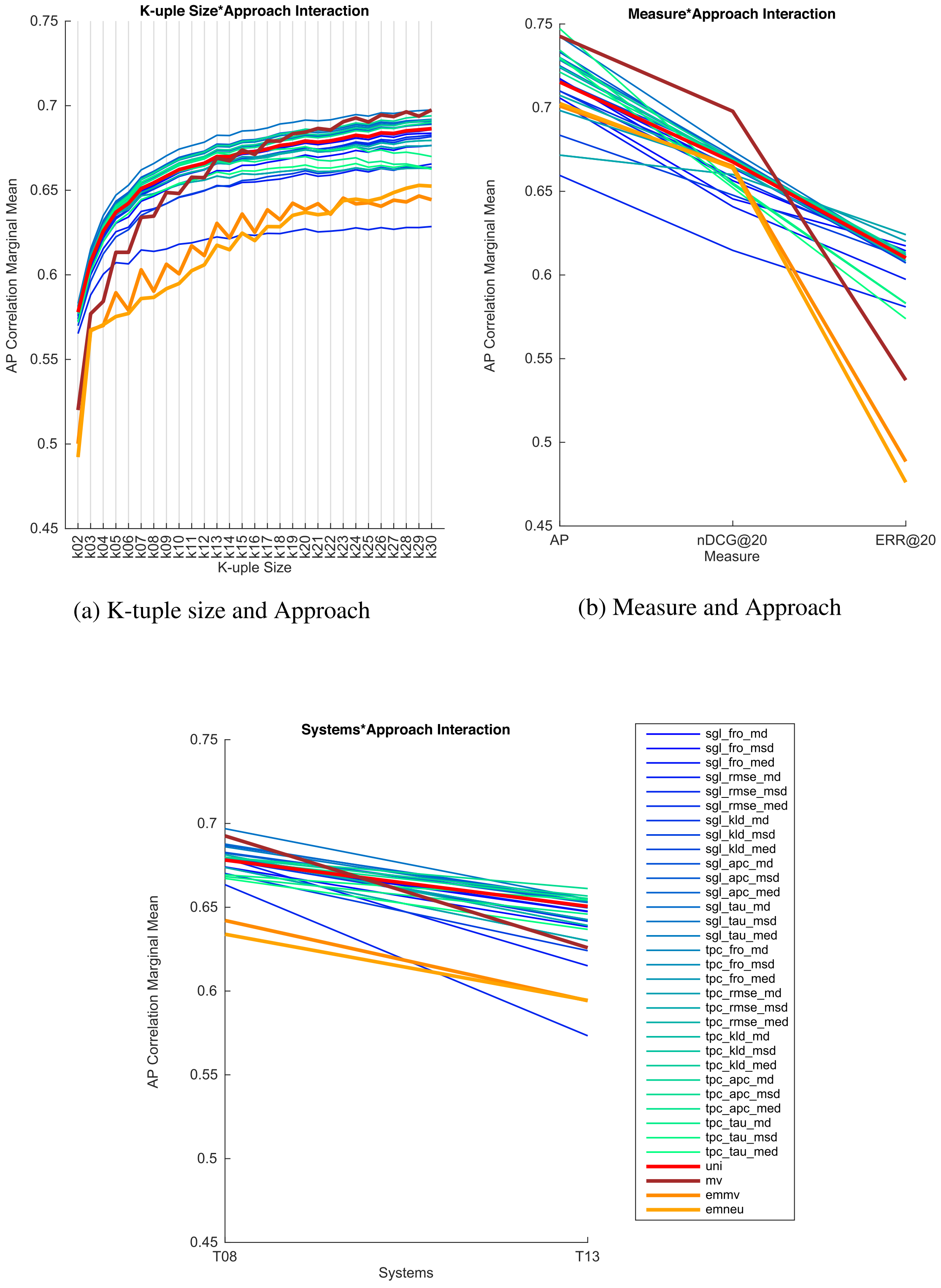
Fig. 4.4 AP correlation: main effects plots for `Approach` (a), `K-tuple Size` (b), `Measure` (c), `Systems` (d), and Tukey HSD multiple comparison test for the `Approach` factor (e).

The Tukey HSD multiple comparison analysis reported in Figure 4.4(e) highlights the top group (dashed blue line), the group of approaches not significantly different from the uni baseline (dashed bright red line), the group of approaches not significantly different from mv (dashed dark red line), and the group of approaches not significanty different from emmv and emneu (dashed orange line). We can note how the top group is separated from the others while the uni and mv groups partially overlaps. In particular, we can see that the approaches significantly better than all the others are sgl_tau_msd (the top one), sgl_apc_msd, tpc_apc_msd, and sgl_tau_md, suggesting that the single score granularity is preferable to the topic-by-topic one and that the tau and apc gaps help to rank systems better. State-of-the-art approaches, namely mv (the best one in this group), emmv, and emneu are clearly separated from the top group. Finally, the AWARE uni baseline exhibits better performances than mv, even though it is not significantly different from it. As also shown in the toy example of Section 4.2, among the AWARE approaches, uni is the closest to mv, in that they both merge assessors attributing the same weight to all of them; yet performing this operation on the measures rather than on the relevance judgments proves to be slightly more effective.

Figure 4.5 shows the interaction plots. We used the following color convention: we selected cool colors for the proposed models, based on the AWARE framework, and warm colors for state-of-the-art models, i.e. mv, emmv, emneu and the AWARE uni baseline.

As shown in Figure 4.5a, we can see that K-tuple Size has a positive effect for all the Approaches. Figure 4.5a also allows us to understand which approaches perform best for a given number of crowd assessors, i.e. for a given k-tuple size. AWARE approaches start higher for low k-tuple sizes while state-of-the-art ones grow faster as the k-tuple size increases. In particular, mv reaches uni at $k = 13$ merged assessors and surpasses it from $k = 17$ onwards, attaining an interaction level as positive as sgl_tau_msd just from $k = 25$ merged assessors. On the other hand, the emneu and emmv methods start to behave better at higher numbers of merged assessors and this is consistent with previous findings in the literature [Raykar et al., 2009, 2010].

Being effective already at low numbers of merged assessors is a clear advantage of the AWARE approaches, since this helps in containing the costs and effort for creating a pool. Moreover, when considering the increasingly better performances of the mv method with high numbers of merged assessors, we have also to remember how the gold standard has been created: TREC 2012 Crowdsourcing organizers took the majority vote of the submitted pools and then adjudicated it with respect to the NIST pool. Therefore, it is somehow natural that when you use almost all the crowd assessors, i.e. all the submitted pools, the performances



(a) K-tuple size and Approach

(b) Measure and Approach

(c) System and Approach

Fig. 4.5 AP correlation: interaction effects plots for K-tuple size (a), measure (b) and systems (c).

Table 4.2 ANOVA table for normalised RMSE considering the k-tuple size, approach, measure and systems effects.

Source	SS	DF	MS	F	p-value	$\hat{\omega}^2_{\langle fact \rangle}$
K-tuple Size	20.6579	28	0.7378	272.9961	< 0.0001	
Approach	32.2530	33	0.9774	361.6465	< 0.0001	0.6680
Measure	56.7010	2	28.3505	10,490.3151	< 0.0001	0.7800
Systems	3.7700	1	3.7700	1,394.9723	< 0.0001	0.1907
Approach*Measure	45.4675	66	0.6889	254.9091	< 0.0001	0.7391
Approach*Systems	2.4886	33	0.0754	27.9039	< 0.0001	0.1305
Measure*System	0.6374	2	0.3187	117.9227	< 0.0001	0.0380
Error	15.5396	5,750	0.0027			
Total	177.5149	5,915				

of the majority vote tend to become the best ones, since you start converging towards what has been used as the gold standard.

When it comes to the interaction between Measures and Approaches (Figure 4.5b), AWARE approaches react more proportionally to the increasing difficulty of the different performance measures; indeed, while mv is among the best interacting approaches for AP and the best one for nDCG@20, it suffers from a very consistent drop in the case of ERR@20 (and similarly for emmv and emneu). Finally, in the case of the interaction between Systems and Approaches (Figure 4.5c), AWARE approaches behave similarly while mv loses more when it comes to the T13 systems. Again, all of this supports the intuition behind the AWARE approaches about taking into account performance measures and systems in the merging process.

4.5.3 RMSE

Table 4.2 shows how all the main effects as well as all the interaction effects are statistically significant. The Measure factor is a large size effect with the greatest impact; Approach is a large size effect but, unlike the case of AP correlation, it is almost as important as Measure; finally, Systems is a large size effect but much smaller than the previous two. Overall, this further supports the intuition behind AWARE, but it also suggests that Approaches are much more prominent for the accurate estimation of the actual value of a performance measure, (i.e. what is assessed by the RMSE) than for ranking systems correctly (i.e. what is assessed by AP correlation).

When it comes to the interaction effects, we can see that Approach*Measure and Approach*Systems are both large size effects and that the Approach*Measure is the second largest effect, a bit bigger than Approach alone; again, these two facts strengthen the motivations behind AWARE. Finally, the Measure*Systems factor is a small size effect but this is less relevant for our discussion, as explained in the previous section.

The main effects plots in Figure 4.6 show: (i) that increasing the number of merged assessors has the expected positive impact, with a greater variability when merging a higher number of (possibly heterogenous) assessors, see Figure 4.6(b); (ii) how the different performance measures influence the effectiveness, with AP being the most challenging one while nDCG@20 and ERR@20 display a somewhat similar behavior, see Figure 4.6(c); (iii) that the targeted systems affect the performances as well, with T08 being somehow more difficult, see Figure 4.6(d).

Figure 4.6(a) shows the main effects of the Approach factor: we can see that the AWARE approaches are quite good, but with a few more exceptions than in the case of AP correlation, namely sgl_rmse_msd, tpc_fro_msd, and tpc_rmse_msd. The top group, reported in Figure 4.6(e), consists of sgl_rmse_med, tpc_rmse_med, tpc_fro_med (the top ones with extremely close performances), sgl_fro_med, and sgl_kld_md; this suggests that there is more balance between single and topic-by-topic score granularities and that the gaps operating closer to the assessors measures (fro, rmse, kld) are more effective. State-of-the-art approaches are clearly distinct from the top group and, in this case, AWARE uni is significantly better than mv and the rest of them, see Figure 4.6(e).

If we look at the interaction effects plots in Figure 4.7, we can see that K-tuple size has a positive effect for all the Approaches, apart from emmv and emneu, see Figure 4.7a. As in the case of AP correlation, AWARE approaches quickly gain at lower numbers of merged assessors, becoming more stable as the k-tuple size increases. Unlike the case of AP correlation, mv behaves like AWARE approaches up to $k = 16$ merged assessors whereas, afterwards, adding more assessors becomes even harmful.

When it comes to the Measure*Approach interaction effect in Figure 4.7b, we can see that emmv and emneu react badly to it, while mv behaves similarly to the AWARE approaches, even though many of them benefit from the Measure effect more than mv, which is one of the worst interacting approach in the case of ERR@20. Finally, for the Systems*Approach interaction effect in Figure 4.7c, emmv and emneu are almost insensitive to it and perform badly, while mv behaves better than most of the AWARE approaches for T08, but worse than most of them in the case of T13. Overall, these facts are a further confirmation of the intuition which led to the development of AWARE.

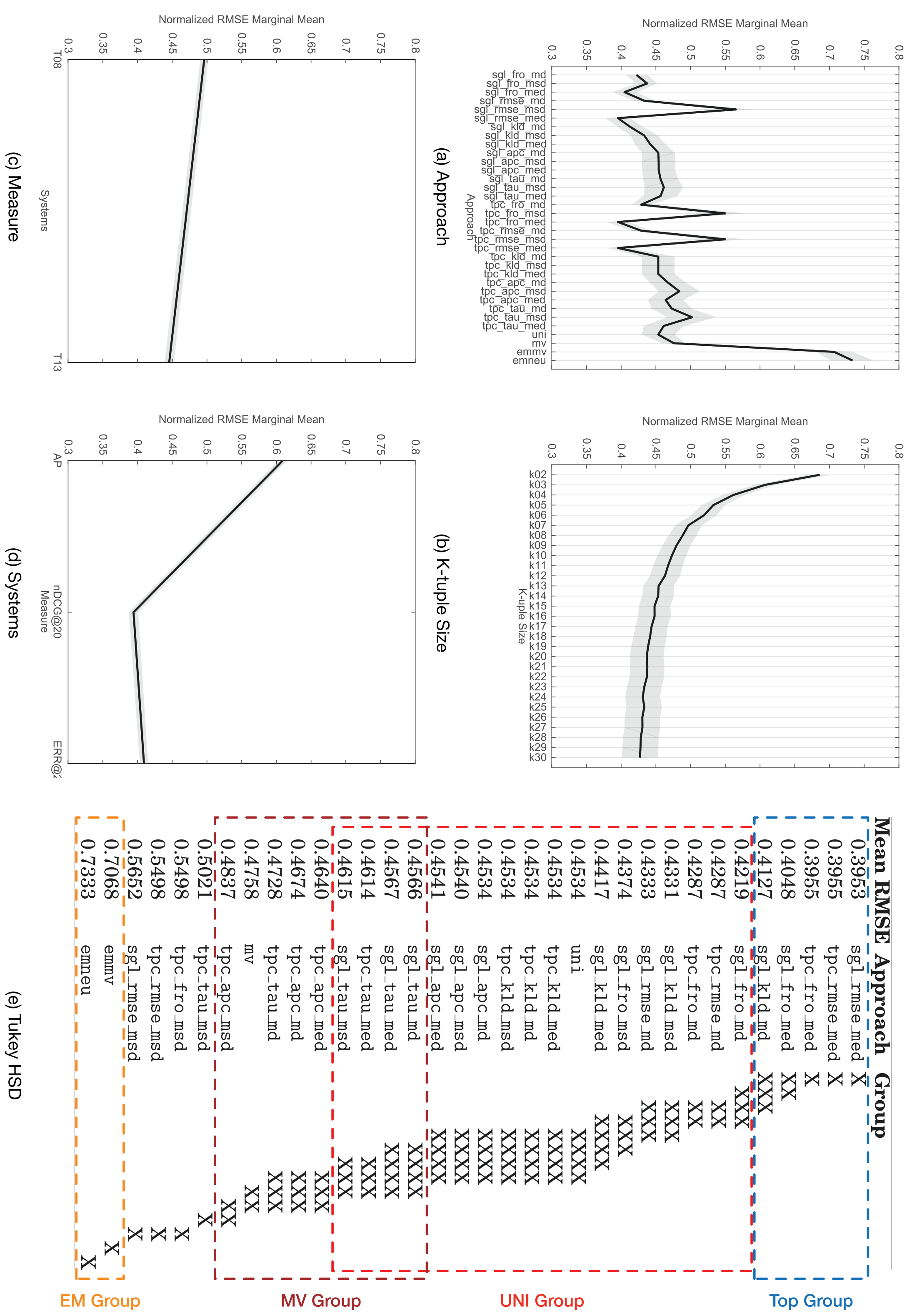
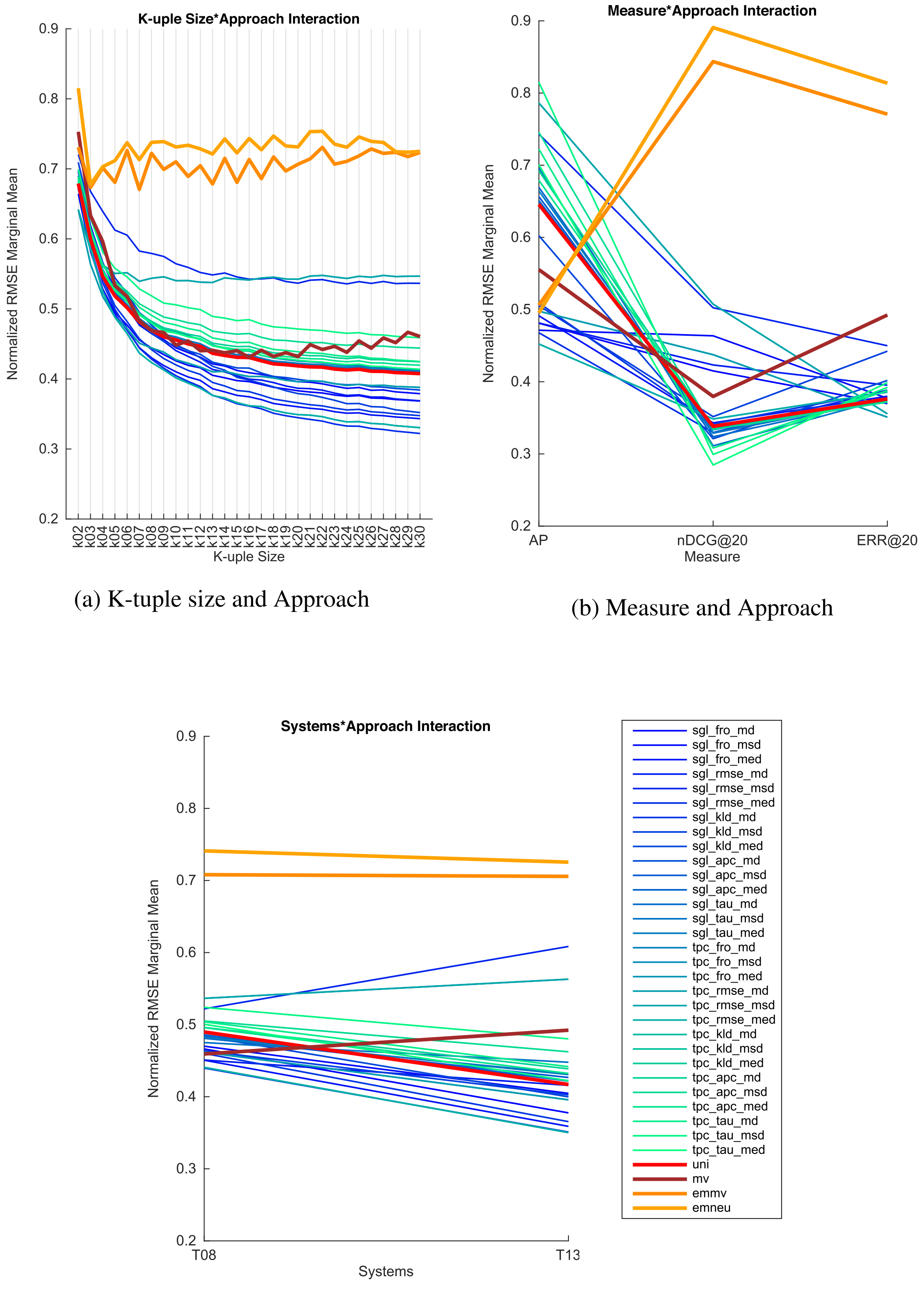


Fig. 4.6 RMSE: main effects plots for Approach (a), K-tuple Size (b), Measure (c), Systems (d), and Tukey HSD multiple comparison test for the Approach factor (e).



(a) K-tuple size and Approach

(b) Measure and Approach

(c) System and Approach

Fig. 4.7 RMSE: interaction effects plots considering k-tuple size (a), measure (b) and systems (c).

4.6 Factorial Analysis of AWARE Components

4.6.1 Methodology

The goal of this section is to conduct a break-down analysis to investigate how the different components of the AWARE accuracy estimators, namely the granularity, gap, and weight, behave at the net of the other factors, namely the k-tuple size, the performance measure under consideration, and the considered systems. To this end, we adopt the following GLMM model for the three-way ANOVA with repeated measures:

$$Y_{ijklmn} = \underbrace{\mu_{......} + \kappa_i + \alpha_j + \beta_k + \gamma_l + \delta_m + \zeta_n}_{\text{Main Effects}} + \underbrace{\alpha\beta_{jk} + \alpha\gamma_{jl} + \beta\gamma kl}_{\text{Interaction Effects}} + \underbrace{\varepsilon_{ijklmn}}_{\text{Error}} \qquad (4.31)$$

where: Y_{ijkl} is the score of the i-th subject in the j-th, k-th, l-th, m-th, and n-th factors; $\mu_{.....}$ is the grand mean; κ_i is the effect of the i-th subject, i.e. the ktuple size $k = 2, \ldots, 30$; α_j is the effect of the j-th factor, i.e. the granularity either `sgl` or `tpc`; β_k is the effect of the k-th factor, i.e. the adopted gap, namely `fro`, `rmse`, `kld`, `apc`, or `tau`; γ_l is the effect of the k-th factor, i.e. the adopted weight, namely `md`, `msd`, or `med`; δ_m is the effect of the m-th factor, i.e. the the performance measures under consideration, namely AP, nDCG@20, and ERR@20; and, ζ_n is the effect of the n-th factor, i.e. the systems submitted to the `T08` and `T13` tracks. We consider also the interaction effects among granularity and gap ($\alpha\beta_{jk}$), granularity and weight ($\alpha\gamma_{jl}$), and gap and weight ($\beta\gamma_{kl}$). Finally, ε_{ijklmn} is the error committed by the model in predicting the score of the i-th subject in the five factors j, k, l, m, n.

As in the previous section, also in this case we normalize the RMSE score by its maximum value for each performance measure before proceeding with the analyses.

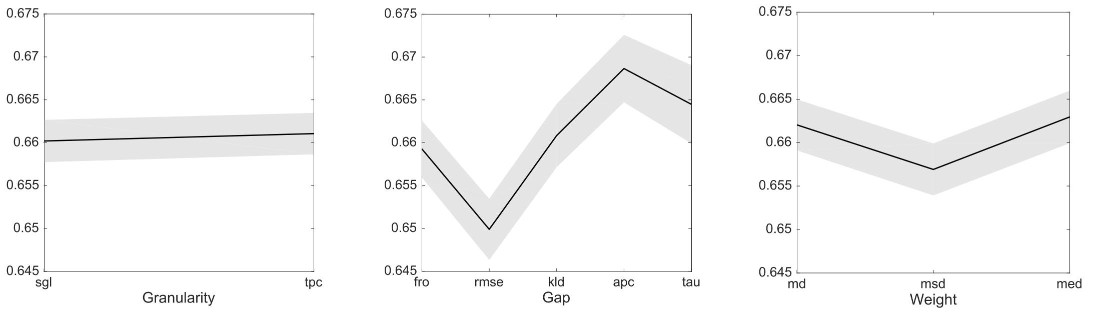
4.6.2 AP Correlation

Table 4.3 confirms that `K-tuple Size`, `Measure` and `Systems` are significant and large size factors that affect the performances as already observed in the previous section, with `Measure` and `Systems` being the most prominent effects. All the interaction effects are small size effects with `Granularity*Gap` and `Gap*Weight` quite similar in terms of size and `Granularity*Weight` about 6 times smaller.

When it comes to the break-down of the AWARE components, we can observe that `Granularity` is not a significant factor. This can also be noted in: (i) the main effects plot in Figure 4.8a, where `sgl` and `tpc` are connected by an almost straight line; (ii) the Tukey

Table 4.3 ANOVA table for AP correlation providing the break-down of AWARE components effects.

Source	SS	DF	MS	F	p-value	$\hat{\omega}^2_{\langle fact \rangle}$
K-tuple Size	2.8154	28	0.1005	80.7404	< 0.0001	
Granularity	0.0009	1	0.0009	0.7746	0.3788	0
Gap	0.2049	4	0.0512	41.1420	< 0.0001	0.0298
Weight	0.0369	2	0.0185	14.8185	< 0.0001	0.0053
Measure	9.6331	2	4.8166	3,867.6402	< 0.0001	0.5970
Systems	1.6418	1	1.6418	1,318.3279	< 0.0001	0.2015
Granularity*Gap	0.1373	4	0.0343	27.5633	< 0.0001	0.0199
Granularity*Weight	0.0256	2	0.0128	10.3056	< 0.0001	0.0036
Gap*Weight	0.1263	8	0.0158	12.6764	< 0.0001	0.0176
Error	6.4347	5,167	0.0012			
Total	21.0570	5,219				



(a) Granularity

(b) Gap

(c) Weight

Mean τ_{ap}	**Granularity**	**Group**
0.6611	`tpc`	X
0.6602	`sgl`	X

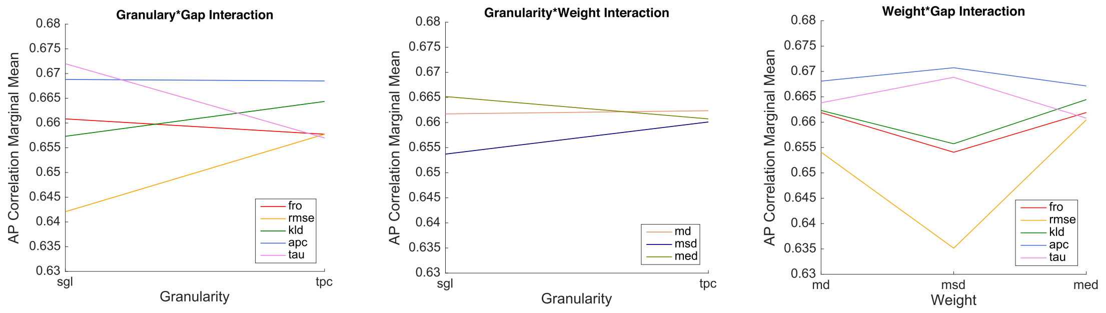
(d) Tukey HSD Granularity

Mean τ_{ap}	**Gap**	**Group**
0.6687	`apc`	X
0.6645	`tau`	XX
0.6608	`kld`	XX
0.6593	`fro`	X
0.6499	`rmse`	X

(e) Tukey HSD Gap

Mean τ_{ap}	**Weight**	**Group**
0.6630	`med`	X
0.6620	`md`	X
0.6569	`msd`	X

(f) Tukey HSD Weight



(g) Granularity and Gap

(h) Granularity and Weight

(i) Weight and Gap

Fig. 4.8 AP correlation: main effects plots (a), (b), (c), Tukey HSD multiple comparison tests (d), (e), (f), and interaction plots (g), (h), (i), considering granularity, gap, and weight.

HSD multiple comparison analysis in Figure 4.8d, which shows that sgl and tpc are not significantly different since their ranges overlap.

Both the Gap and the Weight factors are significant but small size effects, see Figures 4.8b and 4.8c, even though Gap is about 5.8 times Weight in terms of explained variance. In particular, the top gaps are apc and tau, see Figure 4.8e, while med and md are the top weights, see Figure 4.8f. Overall, this suggests that, in terms of AP correlation, the key ingredient of the AWARE approaches is the Gap component and this is corroborated also by the top approaches emerging from Figure 4.4(e), i.e. sgl_tau_msd (the top one), sgl_apc_msd, sgl_tau_md, and tpc_apc_msd, which are a combination of the top Gaps and Weights.

When it comes to the interaction between the different AWARE components in Figure 4.8g, it turns out that the apc and fro gaps are almost insensitive to either the sgl or the tpc granularities and that the tau gap works better with the sgl granularity while the opposite is true for the kld and rmse gaps. Overall, this suggests that gaps closer to the assessor measures, i.e. rmse and fro, benefit from a pinpoint granularity more than progressively less close ones, as the kld, tau, and apc gaps are.

As far as Granularity*Weight interaction is concerned in Figure 4.8h, it is interesting to note the difference in behavior between the kinds of weighting schemes: the minimal (squared) dissimilarity ones, i.e. md and msd, benefit more from tpc than sgl (especially msd) while the opposite is true for the other weighting scheme, i.e. med.

Finally, the Weight*Gap interaction in Figure 4.8i reveals that all the Gaps are almost insensitive to the md and med weights while they either gain a lot (apc and tau) or lose a lot (kld, fro, rmse) with the msd weight. This suggests that the sharpness of the weighting scheme, i.e. minimal squared dissimilarity, affects the gaps more than the difference in the kind of weighting schemes, i.e. minimal dissimilarity vs minimal equi-dissimilarity, and this becomes more and more detrimental as you choose a gap closer and closer to the assessor measures.

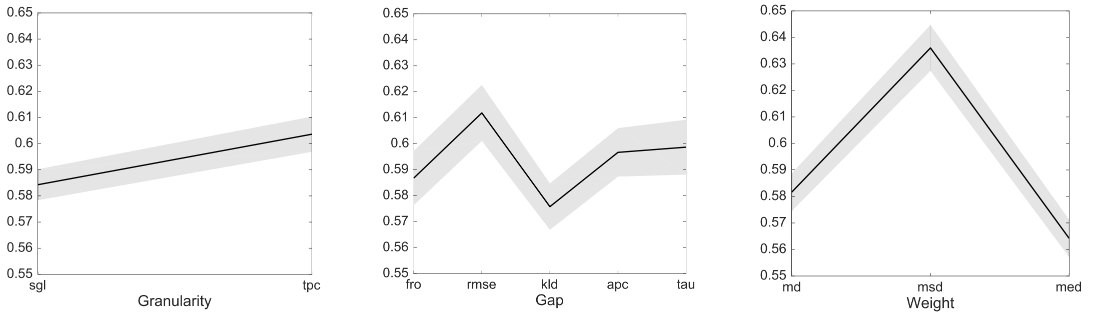
4.6.3 RMSE

As in the case of AP correlation, Table 4.4 confirms that K-tuple Size, Measure and Systems are significant and large size factors, with the Measures and Systems being quite close in terms of size.

Unlike the case of AP correlation, for RMSE all the AWARE components factors are statistically significant and, while Granluarity and Gap are small size effects, Weight is a medium size effect. The interaction effects Granularity*Gap and Granularity*Weight are small size effects, while Gap*Weight is a medium size effect, greater than Weight alone.

Table 4.4 ANOVA table for RMSE providing the break-down of AWARE components effects.

Source	SS	DF	MS	F	p-value	$\hat{\omega}^2_{\langle fact \rangle}$
K-tuple Size	36.9657	28	1.3202	136.8070	< 0.0001	
Granularity	0.4871	1	0.4871	50.4805	< 0.0001	0.0094
Gap	0.7642	4	0.1910	19.7969	< 0.0001	0.0142
Weight	4.8828	2	2.4414	252.9934	< 0.0001	0.0880
Measure	25.0089	2	12.5044	1,295.7805	< 0.0001	0.3316
Systems	18.7670	1	18.7670	1,944.7464	< 0.0001	0.2713
Granularity*Gap	0.3381	4	0.0845	8.7584	< 0.0001	0.0059
Granularity*Weight	0.2805	2	0.1403	14.5350	< 0.0001	0.0052
Gap*Weight	5.8342	8	0.7293	75.5719	< 0.0001	0.1026
Error	49.8622	5,167	0.0097			
Total	143.1908	5,219				



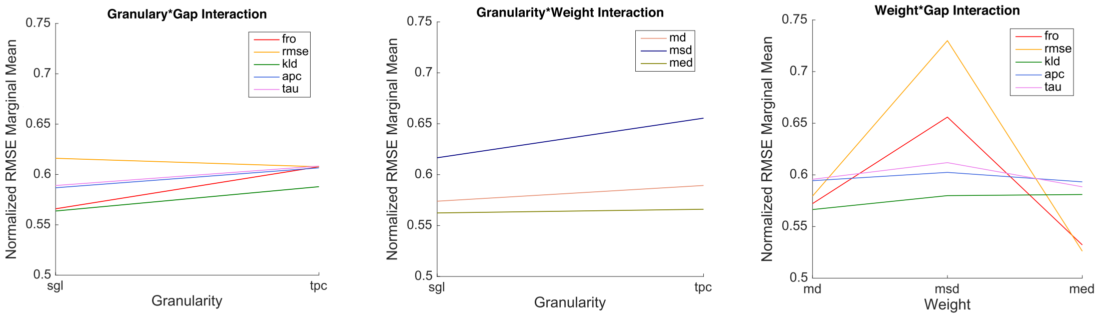
(a) Granularity (b) Gap (c) Weight

Mean RMSE	Granularity	Group
0.5843	sgl	X
0.6036	tpc	X

Mean RMSE	Gap	Group
0.5758	kld	X
0.5868	fro	XX
0.5967	apc	XX
0.5987	tau	X
0.6119	rmse	X

Mean RMSE	Weight	Group
0.5642	med	X
0.5816	md	X
0.6360	msd	X

(d) Tukey HSD Granularity (e) Tukey HSD Gap (f) Tukey HSD Weight



(g) Granularity and Gap (h) Granularity and Weight (i) Weight and Gap

Fig. 4.9 RMSE: main effects plots (a), (b), (c), Tukey HSD multiple comparison tests (d), (e), (f), and interaction plots (g), (h), (i), considering granularity, gap, and weight.

Looking at the main effects and Tukey HSD multiple comparison analyses in Figure 4.9, we can see that: sgl granularity is the best, see Figure 4.9d; the kld and fro gaps are the top ones, see Figure 4.9e, suggesting that gaps moderately close to assessor measures are preferable to better predict a performance score; and the med weight is better than both md and msd, see Figure 4.9f, indicating that its balanced distance from all the random assessors works best in predicting performance scores.

As suggested also by Table 4.4, the Weight*Gap interaction is the most prominent one: in Figure 4.9i the rmse and fro gaps lose most with the msd weight while they have a consistent gain with the med weight; the other gaps are almost insensitive to the weight, apart from a small drop with msd. This suggests that the closer the gap to the assessor measure, the stronger the interaction with the weights: a very negative one in the case of the msd weight, which is the sharpest one; a very positive one in the case of the med weight, which is the most balanced one.

When it comes to the Granularity*Gap interaction in Figure 4.9g gaps tend to improve passing from the tpc to the sgl granularity, especially fro, although rmse is an exception as slightly gains with the tpc granularity.

Finally, for the Granularity*Weight interaction in Figure 4.9h med and md are mostly insensitive to granularity, while msd improves using sgl.

4.7 Summary

In this Chapter, we presented the AWARE framework for robustly combining performance measures coming from multiple crowd assessors. The idea of AWARE stemmed from the observation of the potential impact of both performance measures and systems when it comes to correctly labeled/mis-labeled relevance judgements. Therefore, we proposed a probabilistic framework to take systems and performance measures into account during the estimation of the crowd assessors accuracies used to combine them.

We then exemplified how to instantiate the proposed stochastic framework by introducing many unsupervised estimators of the accuracy of crowd assessors.

Finally, we conducted a thorough evaluation on TREC collections, comparing AWARE against state-of-the-art approaches and studying their influencing factors, namely performance measures and systems. We also investigated the contributions and interactions of the different components of the AWARE estimators.

The experimentation has provided multiple evidence supporting the intuition behind the AWARE framework. Moreover, it has shown that AWARE approaches perform better than state-of-the-art ones in terms of both ranking systems and correctly predicting their

Table 4.5 Result Summary for AP Correlation and RMSE.

	AP Correlation	RMSE
Approach	sgl_tau_msd sgl_apc_msd tpc_apc_msd sgl_tau_md	sgl_rmse_med tpc_rmse_med tpc_fro_med sgl_fro_med sgl_kld_md
Granularity	sgl tpc	sgl
Gap	apc tau	kld fro
Weight	med md	med

performance scores. Finally, it has provided insights about which estimators work best in which context.

Table 4.5 summarizes the top AWARE approaches, analyzed in detail in Section 4.5, as well as the best AWARE components, namely granularities, gaps and weights, analyzed in Section 4.6; the table shows these analyses for both AP correlation, i.e. as far as ranking systems is concerned, and RMSE, i.e. as far as predicting system performances is concerned.

sgl_tau_msd is the best approach in terms of AP correlation while sgl_rmse_med is the best approach for RMSE. In general, AWARE approaches outperform the state-of-the-art ones which are never part of the top group. Moreover, for both AP correlation and RMSE, we can observe that increasing the number of crowd assessors improves the performances – see Figures 4.4(b) and 4.6(b) – but the AWARE approaches are more effective than the state-of-the-art ones for low numbers of assessors, as shown in Figures 4.5a and 4.7a. Therefore, besides better performance, AWARE provides the additional benefit of requiring less resources for ground-truth creation.

When it comes to components, in terms of AP correlation, the sgl and tpc granularities are not significantly different, even if the sgl granularity is predominant among top approaches. This is due to the interaction among components, analyzed in Figure 4.8, which boost the performances for some combinations of components, e.g. the sgl granularity performs best than all the others when it is combined with the tau gap, as shown in Figure 4.8g. As far as gaps are concerned, the top group is represented by apc and tau while med and md are the weights in the top group. As before, the fact that top approaches mostly use the msd weight is due to the interaction between components; indeed, as shown in Figure 4.8i, the performance of msd is boosted by the apc and tau gaps which, at the same time, lower the performance of md and med. The importance of the interaction effects is supported also by the effect sizes reported in Table 4.3, which shows that the Granularity*Gap and Gap*Weight interactions have size one order of magnitude greater than the Granularity*Weight interaction.

Respectively, for RMSE, the best granularity is `sgl` which is also the most frequent in the top group of approaches. The best gaps are `kld` and `fro` while `med` is the top weight. As discussed above, interaction plays an important role also in this case: indeed, the top approaches are `sgl_rmse_med` and `tpc_rmse_med` because of the strong positive interaction between `med` and `rmse`, shown in Figure 4.9i and supported by the medium effect size of the Gap*Weight interaction, which is two order of magnitude greater than all the other interactions effects, as reported in Table 4.4.

The proposed unsupervised estimators are, in a sense, mono-feature, since they operate on each performance measure separately. However, the experimentation has shown that the performance of the proposed estimators varies from measure to measure, e.g. ERR is more challenging than AP in terms of AP correlation. Therefore, as part of future work, we will investigate multi-feature estimators, i.e. estimators that take into account multiple performance measures at the same time to determine the accuracy of a crowd assessor; in this way, we plan to exploit the differences among various evaluation measures to obtain more robust estimators.

Another direction for future work will concern the development of supervised estimators, i.e. estimators that leverage a gold standard instead of random assessors for determining the accuracy of a crowd assessor. Also in this case, we can envision both mono-feature and multi-feature estimators, in the sense explained above.

Finally, it would be interesting to experiment what happens in the case of graded-relevance judgments. Not only is this a natural setting for nDCG and ERR, it also opens up to other evaluation measures such as Graded Average Precision (GAP) and its extensions [Ferrante et al., 2014b; Robertson et al., 2010] or effort-based measures such as Twist [Ferro et al., 2016b].

Chapter 5

User Model Based on Markov Chain and Applications

> Estimating user preferences in real web search settings is a challenging problem, since real user interactions tend to be more "noisy" than commonly assumed in the controlled settings.
>
> Agichtein et al. [2006a]

Nowadays IR systems are challenged with increasingly complex search tasks, where information about how users interact with IR systems plays a central role to adapt them to user needs and interests [Lucchese et al., 2013; Silvestri, 2009]. A lot of IR research focused on improving effectiveness, by exploiting information about user-system interactions recorded in the query logs of Web search engines. The number of clicks on a given query-result pair, the click-through rate, and the dwell time, are examples of actionable information to improve various aspects of IR systems.

Click logs have many advantages: first, the users' actual behaviour is recorded, and not reported by laboratory user study participants as subjective impressions [Harman, 2011]. Therefore, click logs represent a natural source of user feedback available in real time and which reflects the actual user preferences [Radlinski et al., 2008]. Second, click logs are easy to collect by search engines and available in large quantities, this allows researchers to perform experiments with different levels of resolution and to evaluate IR systems at a larger scale and lower costs than the Cranfield approach.

On the other hand, click data are noisy, not controlled or annotated, and they are biased with respect to the position, named presentation bias, and the quality of the Search Engine

Result Page (SERP), named quality bias [Joachims et al., 2005]. The presentation bias is related to the trust that users place on the IRS, they tend to click on the documents at the beginning of the ranking, because it is supposed that the system will display the most relevant results on the top rank positions. The quality bias encompasses the tendency of the users to click less when the quality of the SERP is poor, i.e. when the documents returned by the system are not relevant.

Even if log data are noisy and biased, they still represent an informative source of user feedback [Joachims et al., 2005]. Therefore, correctly interpreting them and understanding the reason behind users' actions is a crucial concern for researchers. This can be achieved with a proper user model, that can explain the bias and predict the user behaviour, by controlling or removing the noisy component.

User models can be applied to many aspects of IR which involve the interactions with a user. For example, a user model can improve the scoring function of an IRS, by accounting for user preferences. [Agichtein et al., 2006b] describes one of the first attempts to predict clicks and customize search results by modeling the bias derived from the rank position. Moreover, user models can be advantageous even for evaluation purposes: fitting a model to the user behaviour can provide a tool to estimate the level of satisfaction of a user with a particular system.

[Robertson, 2008a] proposed a simple, but moderately plausible user model for AP, which allows for a mix of different behaviors in the population of users. The author assumes that a user will stop her search at a given document in the ranked list, called satisfaction point, according to a common probability law, denoted as $p_s(n)$, i.e. the probability that the user will be satisfied by the document at rank position n.

We proposed a novel model of user behaviour which stems from the final considerations of [Robertson, 2008a], at page 690:

> this argument could provide the basis for a more elaborate model, by for example basing the set of $p_s(n)$ on some more sophisticated view of stopping behaviour

Our new user model exploits Markov chains [Norris, 1998] to describe different user patterns in exploring the SERP. We represent each position in a ranked result list with a state in a Markov chain and the different topologies and transition probabilities among the states of the Markov chain allow us to model the different and perhaps complex user behaviors and paths in scanning the ranked result list. The invariant distribution of the Markov chain provides us with the probability of the user being in a given state/rank position in stationary conditions. We apply this model to two different scenarios, first we propose a family of evaluation measures able to account for the user behaviour, second, through the same model we defined the user dynamic and we integrate it into a Learning to Rank (LtR) algorithm.

We called Markov Precision (MP), the family of measures of retrieval effectiveness based on our markovian model. MP injects different user models into precision and does not depend on the recall base. We use the invariant distribution of the Markov chain to compute a weighted average of precision, aiming at discounting the relevance of a document by the probability of the user visiting the given document.

The framework we propose is actually more general and it is based on continuous-time Markov chains in order to take into account also the time a user spends in visiting a single document. It is then possible to extract a discrete-time Markov chain, when considering only the transitions among rank positions and not the time spent in each document. This gives us a two-fold opportunity: when we consider the discrete-time Markov chain, we are basically reasoning as traditional evaluation measures which assess the utility for the user in scanning the ranked result list; when we consider the continuous-time Markov chain, we also embed the information about the time spent by the user in visiting a document and we have a single measure including both aspects. This represents a valuable contribution of the chapter since, up to now, rank and time have been two separate variables according to which retrieval effectiveness is evaluated [Smucker and Clarke, 2012a].

We then propose some basic models for the transition matrix of the Markov chain. We will also show how some of these models are extremely highly correlated to AP, thus suggesting how AP can be considered a very good approximation of more complex user strategies. This helps in shedding some light on why AP is the de-facto "gold standard" in IR, even though it has been so often criticized.

Finally, we conduct a thorough experimental evaluation of the MP measure both using standard TREC collections and click-logs with assessed queries made available by Yandex [Serdyukov et al., 2012]. The results show that MP is comparable to other measures for some desirable properties like robustness to pool downsampling, while the Yandex click-logs allow us to estimate the time spent by the users on the documents and apply the continous-time Markov chain.

Successively, we will explore the embedding of users' interactions into LambdaMart, a state-of-the-art LtR algorithm. In the context of LtR, user actions recorded in query logs are usually used to extract several important features [Agichtein et al., 2006a,b; Liu, 2011; Yu et al., 2015] as for example the number of time that a document was clicked or the time spent by the user in visiting the document. As an empirical evidence of the importance of user interaction features, we trained a LambdaMart [Burges, 2010; Wu et al., 2010] model on the MSLR-WEB10K LtR dataset[1] with and without user-interaction features: the nDCG measured on the test set without such features drops from 0.4636 to 0.4410.

[1] https://www.microsoft.com/en-us/research/project/mslr/

However, our approach is different from those approaches which aim at accounting for the user behaviour by defining a new set of features and then training the LtR model on this extended set of features, indeed we adopt a complementary approach and we integrate the user dynamic directly in an LtR algorithm. Therefore, we model the user dynamic in scanning a ranked result list with the proposed Markovian model trained on query log data and we modify the LAMBDAMART loss function to embed this trained Markov chain. To the best of our knowledge, the integration of the user dynamic in an LtR algorithm is novel and has not been addressed yet.

Our work stemmed from the observation that the user behaviour differs based on the query types. We defined two different types of queries based on the number of relevant documents retrieved: navigational queries are those which contain just one relevant document, while informational queries present no relevant results or more than one relevant results. From the analysis of the invariant distribution, we noticed that the users tend to focus their attention on the top rank positions for navigational queries, while they tend to explore the SERP and visit even low rank positions, for informational queries. We define the user dynamic as a mixture of these two macroscopic behaviours: navigational and informational, and based on the query log dataset provided by Yandex, we calibrate the user dynamic on real world user interactions.

To embed the user dynamic in LAMBDAMART, we replace its objective function, based on nDCG, with a new evaluation measure, called nMCG. nMCG is an extension of nDCG, where the discount function is represented by the user dynamic, therefore two different effectiveness scores are computed accordingly to the query type. This novel approach, called nMCG-MART is finally compared against LAMBDAMART, achieving better effectiveness scores, both in terms of the new nMCG and the standard nDCG.

This chapter is organized as follows: Section 5.1 presents the related works; Section 5.2 fully introduces the Markovian model and Section 5.3 describes its application to define MP, with Section 5.4 reporting the conducted experimental evaluation of MP. Then Section 5.5 explains how we integrate the Markovian model in LAMBDAMART by defining nMCG; Section 5.6 reports the experimental comparison between LAMBDAMART and nMCG-MART; and Section 5.7 draws some conclusion and provides an outlook for future work.

5.1 Related Works

5.1.1 Markov Chains

A Markov chain is a random process with the property that it has no memory regarding what happened in the past. This means that only the current state of the process has an influence on the next state that it will assume. Given a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and a random variable X defined on Ω, $X\colon \Omega \to I$, the following basic ingredients are needed to define a Markov chain:

- A countable set I called the state space, where each $i \in I$ is called state;
- An initial distribution λ_i, where $i \in I$ and $\mathbb{P}[X = i] = \lambda_i$.

Definition 2. A matrix $P = (p_{i,j}\colon i, j \in I)$ is called **stochastic** if every row $P_i = (p_{i,j}\colon j \in I)$ is a distribution, that is if

- Each entry is positive: $0 \leq p_{i,j} < \infty \quad \forall j \in I$ and
- The sum of each row is equal to 1: $\sum_{j \in I} p_{i,j} = 1$.

In the following, measures and distributions represented by λ are considered as row vectors whose components are indexed by I. Similarly, the transition matrix P is a $|I| \times |I|$ matrix and its entries are labeled as $p_{i,j}$. When I is finite, N denotes its cardinality, i.e. $|I| = N$, therefore λ will be a N-vector and P a $N \times N$-matrix.

Definition 3. A discrete time random process $(X_n)_{n \geq 0}$ is a **discrete time Markov Chain** with initial distribution λ and transition matrix P if

1. X_0 has initial distribution λ;
2. For $n \geq 0$, conditional on $X_n = i$, X_{n+1} has distribution $(p_{i,j}\colon j \in I)$ and is independent of $X_0, X_1, \ldots, X_{n-1}$.

In details, the previous conditions state that for $n \geq 0$ and $i_0, \ldots, i_{n+1} \in I$:

1. $\mathbb{P}[X_0 = i_0] = \lambda_{i_0}$ and
2. $\mathbb{P}[X_{n+1} = i_{n+1} | X_n = i_n, \ldots, X_0 = i_0] = \mathbb{P}[X_{n+1} = i_{n+1} | X_n = i_n] = p_{i_{n+1}, i_n}$.

In the following, the statement $(X_n)_{n \geq 0}$ is Markov(λ, P) means that $(X_n)_{n \geq 0}$ is a Markov chain with initial distribution λ and transition matrix P.

$$P = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$$

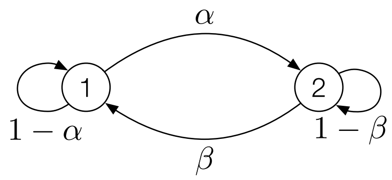


Fig. 5.1 Transition matrix with two states, $I = \{1,2\}$ and its corresponding graph

Given a Markov chain, its stochastic matrix represents the conditional distribution of the process, i.e. given two states i and j, the entry $p_{i,j}$ is the probability that after the next step the process will be on the state j, knowing that the current state is i, that is the probability that the process will go from i to j. In the following, $\mathbb{P}_i$ will denote the probability measure conditioned on $X_0 = i$ when $\lambda_i > 0$, i.e. $\mathbb{P}_i[A]$ stands for $\mathbb{P}[A|X_0 = i]$.

There is a one-to-one correspondence between stochastic matrices P and direct graphs called state transition diagrams. Indeed, direct graphs can be used to visually describe Markov chains and aid to the comprehension of the chain structure. The state space I represents the set of vertexes and given $i, j \in I$, if $p_{i,j}$ is strictly positive, then there is an edge between i and j with weight $p_{i,j}$. Therefore, the graph contains as many edges as the number of positive entries in the matrix P.

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

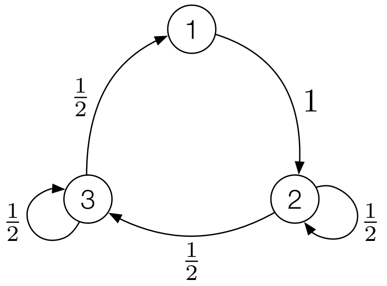


Fig. 5.2 Transition matrix with three states, $I = \{1,2,3\}$ and its corresponding graph

Figure 5.1 and Figure 5.2 report two examples of the correspondence between stochastic matrices and state transition diagrams. In Figure 5.1 the cardinality of the state space is equal to two and the graph contains two vertexes. The weight of the edge from 1 to 2 is equal to $p_{1,2} = 1 - \alpha$ and the weight of the edge from 2 to 1 is equal to $p_{2,1} = 1 - \beta$. Moreover, there are two self loops with weights $p_{1,1} = \alpha$ for the state 1 and $p_{2,2} = \beta$ for the state 2. Similarly, in Figure 5.2 the graph has three vertexes, equal to the number of states, and five edges, i.e. the number of positive entries of P. Diagonal entries represent the weight of self loops, while non diagonal entries represent the weight of edges between two different vertexes.

Before introducing further theorems and properties of Markov chains, it is worth to spend some words on how to compute the probability that after n steps the Markov chain is in a given state. This reduces to calculate the n power of the transition matrix P.

Matrix multiplication is calculated as follows:

$$(\lambda P)_j = \sum_{i \in I} \lambda_i p_{i,j}, \qquad (P^2)_{i,k} = \sum_{j \in I} p_{i,j} p_{j,k},$$

and it is extended to define P^n for each $n \geq 0$. Therefore, P^0 is the identity matrix $\mathbb{I}$, where $\mathbb{I}_{i,j} = \delta_{i,j}$, with

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Finally, P^n is defined as the product of P with itself n times and its entries are denoted as $p_{i,j}^{(n)} = (P^n)_{i,j}$. The following theorem gives an insight to understand the benefit of defining P^n and how it can be used to compute the probability to observe the process in a given state after n steps.

Theorem 2. *Let $(X_n)_{n \geq 0}$ be Markov-(λ, P). Then for all $n, m \geq 0$,*

1. $\mathbb{P}[X_n = j] = (\lambda P^n)_j$;

2. $\mathbb{P}_i[X_n = j] = \mathbb{P}[X_{n+m} = j | X_m = i] = p_{i,j}^n$.

As a consequence of Theorem 2, $p_{i,j}^n$ is called n-step transition probability from i to j.

When it comes to the investigation of long term properties of Markov chain, the definition of invariant distribution needs to be introduced.

Definition 4. A probability measure $\lambda = (\lambda_i \colon i \in I)$ is called **invariant** if

$$\lambda P = P$$

Definition 4 is equivalent to claim that λ is a left eigenvector for P, with the constraint that $\lambda_i > 0$, $\forall i \in I$. Alternatively, the invariant distribution is called stationary or equilibrium. Theorem 3 explains the meaning of the name stationary and Theorem 4 is related to the name equilibrium.

Definition 5. A discrete-time process $(X_n)_{n \geq 0}$ is called **stationary** if for any time points $i_1, \dots, i_n$ and any $m \geq 0$ the random vectors $(X_{i_1}, \dots, X_{i_n})$ and $(X_{i_1+m}, \dots, X_{i_n+m})$ have the same joint distribution.

Therefore, stationary refers to stationary in time: the distribution of X_n is the same for each n. This means that if a process is stationary and the distribution of each X_n is known, then even the long proportion of time that the Markov chain spends on each state is known.

Theorem 3. *Let $(X_n)_{n\geq 0}$ be Markov(λ, P) and suppose that λ is invariant to P. Then $(X_{m+n})_{n\geq 0}$ is also Markov(λ, P).*

Theorem 3 claims that when the initial distribution λ is invariant, then the Markov chain is a stationary process. Therefore, if X_0 has distribution λ, then also the distribution of X_n is λ for each $n \geq 0$. The following theorem, suggests that the invariant distribution π_j can be interpreted as the long run proportion of time that the Markov chain spends in the state j.

Theorem 4. *Let I be finite. Suppose that for some $i \in I$*

$$p_{i,j}^{(n)} = \pi_j \quad as \quad n \to \infty \quad for\ all\ j \in I.$$

Then $\pi = (\pi_j \colon j \in I)$ is an invariant distribution.

The previous theorem proves that if the state space I is finite and if for some i the limit exists for all $j \in I$, then the limit must be an invariant distribution. However, the limit does not always exists, as shown by the following example.

Example. Consider a two states Markov chain with transition matrix

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Since $P^{2n} = \mathbb{I}$ and $P^{2n+1} = P$ for all n, the n-step transition matrix does not converge to any limit. However, the matrix P admits an invariant distribution $\lambda = \left(\frac{1}{2}, \frac{1}{2}\right)$, therefore it is not always true that the invariant distribution can be considered as the limit of the n-step transition matrix.

Theorem 4 shows the relationship between invariant distributions and n-step transition distributions, it states that if the n-step transition distribution has a limit, then the limit is invariant. Certainly, the converse would be more useful, i.e. determine whether a matrix P admits an invariant distribution, whether it is unique and whether it is the limit of the n-step transition distribution. The following two theorems aim at proving a sort of converse, under the assumptions of irreducibility and recurrence each stochastic matrix P has a unique positive invariant distribution.

Definition 6. A Markov chain or transition matrix P is called **irreducible** if for any state $i, j \in I$, exists n such that $p_{i,j}^{(n)} > 0$.

Definition 6 states that a transition matrix is defined irreducible if given two states i and j, it exists a finite number of steps n such that the probability to go from i to j in n steps is strictly positive.

Definition 7. A state $i \in I$ is called **aperiodic** if exists $M > 0$ such that $p_{i,i}^{(n)} > 0$ for each $n \geq M$.

Definition 7 requires the existence of $M > 0$, such that for every $n \geq M$ it is possible to find a closed loop of n steps from i to i, which has positive probability.

Theorem 5. *Suppose that P is irreducible and has an aperiodic state i. Then, for all states j and k, $p_{j,k}^{n} > 0$ for all sufficiently large n. This mean that all states are aperiodic.*

Theorem 6 (Convergence to Equilibrium)**.** *Let P be irreducible and aperiodic, and suppose that P has an invariant distribution π. Let λ be any distribution and suppose that $(X_n)_{n\geq 0}$ is Markov-(λ, P). Then*

$$\mathbb{P}[X_n = j] \rightarrow \pi_j \quad as \quad n \rightarrow \infty \textit{ for all } j. \tag{5.1}$$

In particular,

$$p_{i,j}^{(n)} \rightarrow \pi_j \quad as \quad n \rightarrow \infty \textit{ for all } i, j. \tag{5.2}$$

These two latter theorems provide us with the tool to understand when the invariant distribution can be considered as the limit of the n-step transition probability. Therefore, if P is irreducible, has an aperiodic state and an invariant distribution, we can safely conclude that the invariant distribution represent the long term behaviour of the process, i.e. the probability to see the process in a given position after n steps, with n going to ∞.

5.1.2 Markovian Approaches for Information Retrieval

Markov-based approaches have been previously exploited in IR, for example: Markov chains have been used to generate query models [Lafferty and Zhai, 2001], for query expansion [Collins-Thompson and Callan, 2005; Maxwell and Croft, 2013], for document ranking [Daniłowicz and Baliński, 2001], and in language models [Miller et al., 1999; Wei and Croft, 2006]. However, to the best of our knowledge, Markov chains have not been applied to the definition of a fully-fledged measure for retrieval effectiveness or integrated in a LtR framework to describe the user behaviour.

[Wang et al., 2010] exploits a Partially Observable Markov Model (POM), i.e. a model based on hidden markov processes, to analyse a large query log dataset and infer the browsing patterns of users on the SERP. Their aim is to discover the unobservable aspects of the user behaviors that can not be recoreded in query log data, as for example the examination and browse of the snippets, i.e. how the users read the representations of the documents in the SERP. Moreover, they analyse the user behaviour at session level and, in addition to clicks, they use other signals, including hovering events, page loading and unloading, and query reformulation. The model was extended in [He and Wang, 2011] to account for the time dimension, i.e. the time that the users spend in performing every action, especially in reading the documents. However, this model differs from our since it is designed as a complementary method to the eye tracking experiments to uncover unobservable search events, it makes use of more signals and it does not define any evaluation measure or methodology to integrate it in a ranking algorithm.

Furthermore, [Yang et al., 2016] presents dynamic IR, i.e. a framework for IR systems as dynamic processes, able to respond and adapt to the changes in documents and users. The authors explore techniques based on a Partially Observable Markov Decision Processes (POMDP), a stochastic decision process with the Markov property, to make IR systems responsive to changes. Our model differs from this model since we are not considering hidden or partially observable states and we are not tackling the task of dynamic IR.

Finally, [Chierichetti et al., 2011] uses Markov chains to address the placement problem in the case of two-dimensional results presentation: they have to allocate images on a grid to maximize the expected total utility of the user, according to some evaluation measure, and the Markov chain models how the user moves in the grid. Their approach differs from ours since they are not defining a measure of effectiveness or a ranking strategy which embeds a Markov chain, but they rather solve an optimization problem via a Markov chain; moreover, they only use discrete-time Markov chains and limit transitions only to adjacent states. What we share is the idea that a Markov chain can be used to model how a user scans a result list, mono dimensional in our case, two-dimensional in their case.

5.1.3 IR Evaluation Measures and User Models

When it comes to other evaluation measures, the focus of the chapter is on lab-style evaluation with binary relevance. So, for example, measures for novelty and diversity are out of the scope of the present chapter [Clarke et al., 2011] as are measures for graded relevance like ERR [Chapelle et al., 2009], or Q-measure [Sakai, 2005].

A popular measure based on a user model is Rank-Biased Precision (RBP) [Moffat and Zobel, 2008], defined in Equation (2.3.2). RBP equation revolves around the definition of user's persistence, modelled through a parameter p. Specifically, the user starts from the top ranked document and with probability p, goes to the next document or with probability $1-p$

stops. Since the user starts always from the first ranked document, p^{j-1} is the probability that the user will reach rank j. By computing the sum for each rank position, we obtain a geometric sequence which converges to $1/(1-p)$ when n tends to ∞. Finally, the sum is multiplied by $(1-p)$ making the measure score to range in $[0,1]$.

As shown with the balancing index in Section 3.5, the persistence parameter allows to adjust the top heaviness of the metric. The lower the value of p, the less persistent the user, meaning that she does not go deep in the ranked list of documents and causing the measure to be highly top heavy. Conversely, the choice of higher values for p makes the measure discount less steep.

It can be noted that, despite its name, RBP does not depend on the notion of precision. Nevertheless, it represents a measure for binary relevance which does not depend on the recall base, and thus gives a comparison point for MP.

Furthermore, even if the user model which describes RBP is plausible, it still assumes a user that scans the run from the first ranked document, and proceeds to the next ranked document without the possibility to skip any document, to revisit an already visited document and to come back to a higher rank position. In Section 5.2 we will describe our novel user model, which allows the user to be completely free to assume any complex path when examining the run.

With regard to the time dimension brought in by the continuous-time Markov chain, the most relevant work is Time-Biased Gain (TBG) [Smucker and Clarke, 2012a,b]. We share the idea of getting time into evaluation measures but we adopted a different approach. While TBG substitutes traditional evaluation measures, MP provides a single framework for keeping both aspects depending on which Markov chain you use. With respect to the user model adopted in TBG, there are some relevant differences: first, we use full Markov models while [Smucker and Clarke, 2012b] at page 2014 points out that “our model can be viewed as a semi-Markov model”; then, TBG assumes a sequential scanning of the result lists where MP allows the user to move and jump backward and forward in the results list. What TBG addressed and is not in the scope of the present work is how to calibrate the measure with respect to time: [Smucker and Clarke, 2012a] proposed a procedure to calibrate time with respect to document length and [Smucker and Clarke, 2012b] extended it to stochastic simulation. In this chapter, we provide a basic example of calibration based on the estimation of average time spent per document from click logs, just to show how the parameters of the framework could be tuned. However, in the future, nothing prevents us (or others) from investigating more advanced calibration strategies or applying those proposed by [Smucker and Clarke, 2012a,b].

When it comes to other ways to integrate the user behaviour into evaluation measures, [Carterette, 2011] proposes to rely on three components: a browsing model, a model of document utility, and a utility accumulation model. Even if we took up from [Robertson, 2008a], MP can also be framed in the light of the work of [Carterette, 2011]. Indeed, the Markovian model provides us with the browsing model, precision accounts for the model of document utility, and the weighted average of precision by the invariant distribution of the Markov chain supplies the utility accumulation model. Similarly, nMCG, which is our extension of nDCG for LtR, accounts for the browsing model with the user dynamic and for the document utility and the cumulated utility by summing the exponential of the relevance weights for each rank position.

5.1.4 Learning to Rank and User Behaviour

Clickthrough data has became an essential source of information to improve various aspects of IR. For example user interactions are exploited to infer document relevance [Speicher et al., 2013], to learn user preferences and personalize search results [Qiu and Cho, 2006], to create user profiles that support personalized search [Speretta and Gauch, 2005], and to develop personalized recommendation systems [Rendle et al., 2009].

Previous work on click logs [Joachims et al., 2005] has reported that, on average, users scan ranked list in a forward linear fashion, while our Markovian model allow users to move forward and backward in a ranked list. As reported in Section 5.4.5, from Yandex logs, we found that 22.6% of the transitions in the ranked list are backward, thus supporting our assumption, even if more exploration on this is left for future work. This is also supported in [Wang et al., 2015], where the clicks and eye-tracking analysis shows that only 34% of the users follow a linear path when scanning a ranked list of documents.

In the context of LtR, the standard way to account for users' actions, recorded in query logs, is to extract some user dependent features as for example the number of time that a document was clicked or the time spent by the user in visiting the document. [Agichtein et al., 2006b] proposes one of the first attempts to account for the user behaviour in a LtR algorithm. The paper analyses different alternatives for ranking Web search results by exploiting real user behaviour signals. From a large scale analysis, conducted with a commercial search engine, the authors concluded that incorporating user features into the search process leads to significant improvements. However, we will propose a complementary approach, where instead of proposing a new set of features, we will try to model the user dynamic and embed it in a LtR algorithm.

Reinforcement Learning (RL) [Sutton and Barto, 1998] is specifically designed to account for user signals by developing ML algorithms, able to learn from the interactions between the users and the systems to maximize a reward function. In this context, [Joachims, 2002] developed an algorithm to use clickthrough data as relative preferences to train an online ranking algorithm. Later, [Hofmann et al., 2013a, 2014] proposed two approaches for reusing historical data in online learning with the purpose of making the learning process faster. Our work differs from these approaches since they use click log to infer preferences between rankers and their aim is to speed up online learning to rank, while we want to improve LtR algorithms effectiveness by taking into account the user dynamic.

Moreover, our algorithm will differ from Lerot [Schuth et al., 2013], which is an online LtR algorithm based on interleaving methods. In fact, we design an offline LtR algorithm which integrates the user dynamic in the learning process instead of using clicks as feedback for interleaving methods in an online algorithm.

5.2 A Markovian User Model

We assume that each user starts from a chosen document in the ranked list, not necessarily the first one, and considers this document for a random time, that is distributed according to a known positive random variable. Then she decides, according to a probability law, that we will specify in the sequel and independent from the random time spent in the first document, to move to another document in the list. She considers this new document for a random time and she successively moves, independently, to a third relevant document and so on.

We model the user behavior in the framework of the Markovian processes [Norris, 1998]. To fix the notation, we will denote by $X_0, X_1, X_2, \dots$ the (random) sequence of document ranks visited by the user and by T_0, T_1, T_2 the random times spent, respectively, visiting the first document considered, the second one and so on. Therefore, $X_0 = i$ means that the user starts from the first document at rank i and $T_0 = t_0$ means that she spends t_0 units of time visiting this first document, then $X_1 = j$ means that she visits the document at rank j as the second one, and so on.

First of all, we will assume that X_0 is a random variable on $\mathscr{I} = \{1, 2, \dots, N\}$ with a given distribution $\lambda = (\lambda_1, \dots, \lambda_N)$; so for any $i \in \mathscr{I}$, $\mathbb{P}[X_0 = i] = \lambda_i$. Then, we will assume that the probability to pass from the document at rank i to the document at rank j will only depend on the starting rank i and not on the whole list of documents visited before.

This can be formalized as follows:

$$\mathbb{P}[X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0] = \mathbb{P}[X_{n+1} = j | X_n = i] = p_{i,j} \tag{5.3}$$

for any $n \in \mathbb{N}$ and $i, j, i_0, \dots, i_{n-1} \in \mathscr{I}$.

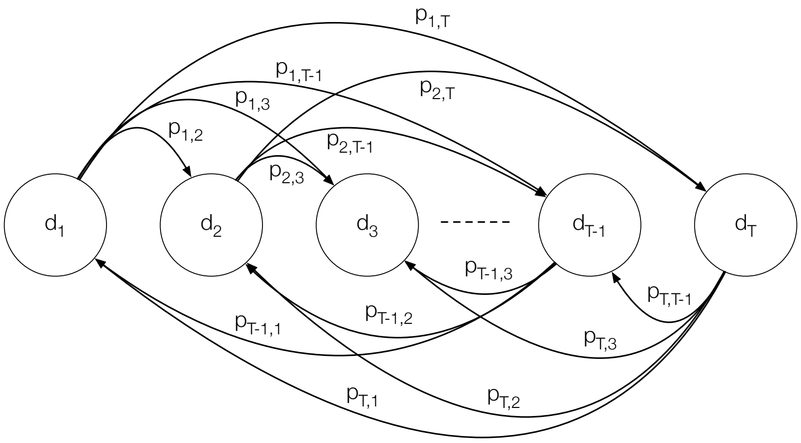


Fig. 5.3 Structure of the Markov chain $(X_n)_{n\in\mathbb{N}}$.

Thanks to the condition (5.3) and fixing a starting distribution λ, the random variables $(X_n)_{n\in\mathbb{N}}$ define a time homogenous discrete time Markov Chain, shown in Figure 5.3, with state space $\mathscr{I}$, initial distribution λ and transition matrix $P = (p_{i,j})_{i,j\in\mathscr{I}}$ (Markov(λ,P) in the sequel).

To obtain a continuous-time Markov Chain, we have to assume that the holding times T_n have all exponential distribution, i.e.

$$\mathbb{P}[T_n \leq t] = \begin{cases} 0 & t < 0 \\ \\ 1 - \exp(-\mu t) & t \geq 0 \end{cases}$$

Furthermore, conditioned on the fact that $X_n = i$, the law of T_n will be exponential with parameter μ_i, where μ_i is a positive real number that may depend on the specific state i of the chain the user is visiting at that time.

When our interest is only on the jump chain $(X_n)_{n\in\mathbb{N}}$, i.e. when we are interested in extracting the corresponding discrete-time Markov chain, we simply assume that all these variables are exponential with parameter $\mu = 1$. When we are also interested in the time dimension, we have to provide a calibration for these exponential variables. We report a simple example in Section 5.4 using click logs from Yandex.

Notice that the Markov chain approach relies on some assumptions – e.g. no long-term memory and exponentially distributed holding times – which may seem oversimplifications of the reality, e.g. a user who considers the whole history of visited documents to decide whether to stop or not. However, there are other models, as those based on POMs [He and Wang, 2011; Wang et al., 2010; Yang et al., 2016] which explicitly assume the Markovian memoryless property, and there are measures, such as RBP [Moffat and Zobel, 2008], where transitioning to the next document or stopping is a step-by-step decision based just on the persistence parameter, in this sense they can be considered memory-less. Moreover, a Markovian model is simple enough to be easily dealt with, while still being quite powerful, and this work intends to be a first step towards a richer world of models that we will explore in the future.

5.3 The Markovian Model as Evaluation Measure

Let us recall the notation introduced in Chapter 3. A run of length N is denoted by $r = (d_1, \dots, d_N)$, where $r_t[j]$ is the j-th element of the vector r, i.e. $r_t[j] = d_j$. GT is the ground-truth and $\hat{r}_t = \big(GT(t, d_1), \dots, GT(t, d_N)\big)$ is the judged run, where $\hat{r}_t[j]$ is the j-th element of the vector $\hat{r}_t$, i.e. $\hat{r}_t[j] = GT(t, d_j)$. In the following we restrict ourself to the case of binary relevance and we assume $REL = \{0, 1\}$. Finally, given a run r_t, the set of the ranks of the relevant documents is $\mathscr{L} = \{j\colon\ j = 1, \dots, N \text{ and } \hat{r}_t[j] = 1\}$, with cardinality $RR = |\mathscr{L}|$, which indicates the total number of relevant retrieved documents by the run for the given topic.

Among all the system centered evaluation measures presented in Section 2.3.1, AP [Buckley and Voorhees, 2005] represents the “gold standard” measure in IR [Yilmaz and Aslam, 2006], known to be stable [Buckley and Voorhees, 2000] and informative [Aslam et al., 2005], with a natural top-heavy bias and an underlying theoretical basis as approximation of the area under the precision/recall curve. Nevertheless, due to its dependence on the recall base, it assumes a perfect knowledge of the relevance of each document in the collection, which is an approximation, when pooling is adopted and not assessed documents are assumed to be not relevant [Harman, 1994], and is even more exacerbated in the case of large scale or dynamic collections [Buckley and Voorhees, 2004; Yilmaz and Aslam, 2006].

However, the strongest criticism to AP comes from the absence of a convincing user model for it, a feature which is deemed extremely important in order to make the interpretation of a measure meaningful and to bridge the gap between system-oriented and user-oriented studies [Carterette, 2011; Moffat et al., 2013; Smucker and Clarke, 2012a]. In this respect, [Moffat and Zobel, 2008] argued that the model behind AP is abstract, complex, and far from the real behavior of users interacting with an IR system, especially when it comes to its dependence on the recall base which is something actually unknown to real users. As a consequence, [Robertson, 2008a] proposed a simple, but moderately plausible user model for AP, which allows for a mix of different behaviors in the population of users.

[Robertson, 2008a] proposed a probabilistic user model measure of effectiveness called Normalized Cumulative Precision (NCP), which includes AP as a particular case. The author assumes that any given user will stop her search at a given document in the ranked list, that we call its satisfaction point, according to a common probability law.

Furthermore, he considers that a user will stop her search only at relevant documents and that the probability that she stops at any given relevant document is fixed and independent from the specific run she is considering, while it is 0 at any non relevant document. So, he defines a probability distribution p_s on the set of all the documents available for a given topic and $p_s(j)$ represents the probability that the user's satisfaction point is the relevant document at rank j.

Given a specific run and the set of its retrieved documents, the definition of the NCP is then the expectation (average) of the precision at the ranks of the retrieved, relevant documents, accordingly to the distribution $p_s(\cdot)$, i.e.

$$NCP(p_s) = \mathbb{E}_{p_s}[\mathrm{Prec}(j)] = \sum_{j=1}^{+\infty} p_s(d_j)\mathrm{Prec}(j) \ .$$

It is easy to see that the definition of AP in Equation 2.1 is in this context equal to the NCP measure when we choose the uniform law p_U over all the relevant documents for the topic

$$p_U(d_j) = \begin{cases} \dfrac{1}{RB} & \text{if } d_j \text{ is relevant, i.e. } \hat{r}_t[j] = 1 \\ \\ 0 & \text{otherwise} \end{cases}$$

The previous user model is simple and it can be considered as a starting point for more sophisticated models, as also suggested by Robertson [2008a] himself. As in the case of AP, the assumption that the user knows the recall base of a given topic is a weakness of this model. Furthermore, the probability that a user stops her search at a given document on a specific run depends on a probability distribution defined on the whole set of relevant documents available for a given topic.

The choice of the uniform distribution to determine the stopping point in a given search is itself of difficult interpretation, since this means that any relevant document in a ranked list of retrieved documents has the same probability.

We will see in the next section how, stepping from the intuition behind NCP, we can define a new evaluation measure based on the Markovian user model. Moreover, the same model can be adapted to define a more realistic user model for AP, and to generalize AP to a whole new class of Markovian models.

5.3.1 Markov Precision (MP)

Consider a user that examines a ranked list of results, after a random number of forward and backward movements along the ranked list, she will end her search and we will evaluate the total utility provided by the system to her by taking the average of the precision of the judged relevant documents she has considered during her search. According to this construction when we compute this average, the precision of a document visited k times will contribute to the mean with a k/n weight.

Let us assume hereafter that the matrix P will be irreducible. This means that we can move in a finite number of steps from any document to any other document with positive probability. Thanks to (5.3) and the multiplication rule, the probability to pass in n steps from the document i to the document j is equal to $p_{i,j}^{(n)}$, the (i,j) entry of the matrix P^n and the irreducibility means that given any pair (i,j) there exists $n > 0$ such that $p_{i,j}^{(n)} > 0$. Furthermore, the probability distribution of any random variable X_n, which denotes the rank of the document visited after n movements, is completely determined by λ and P, since

$$\mathbb{P}[X_n = j] = (\lambda P^n)_j \, .$$

Given such a model, we assume that a user will visit a number n of documents in the list and then they will stop their search. In order to measure their satisfaction, we will evaluate the average of the precision of the ranks of the judged relevant documents visited by the user during their search as

$$\frac{1}{n}\sum_{m=0}^{n-1} \mathrm{Prec}(Y_m) \, .$$

where $(Y_n)_{n\in\mathbb{N}}$ denotes the sub-chain of $(X_n)_{n\in\mathbb{N}}$ that considers just the visits to the judged relevant documents at ranks $\mathscr{L}$, and shown in Figure 5.4.

Note that this sub-chain has in general a transition matrix different form P. The new transition matrix $\widetilde{P}$ can be computed easily from P by solving a linear system as detailed in [Norris, 1998] and discussed in Section 5.3.3. Note that $\widetilde{P}$ computed in this way somehow "absorbs" and takes into account also the probabilities of passing through not relevant documents (which are basically redistributed over the relevant ones) and makes it different from the transition matrix that you would have obtained by using only the relevant documents since the beginning.

Clearly the previous quantity is of little use if evaluated at an unknown finite step n. However, the Ergodic Theorem of the theory of the Markov processes is perfect for approximating this quantity:

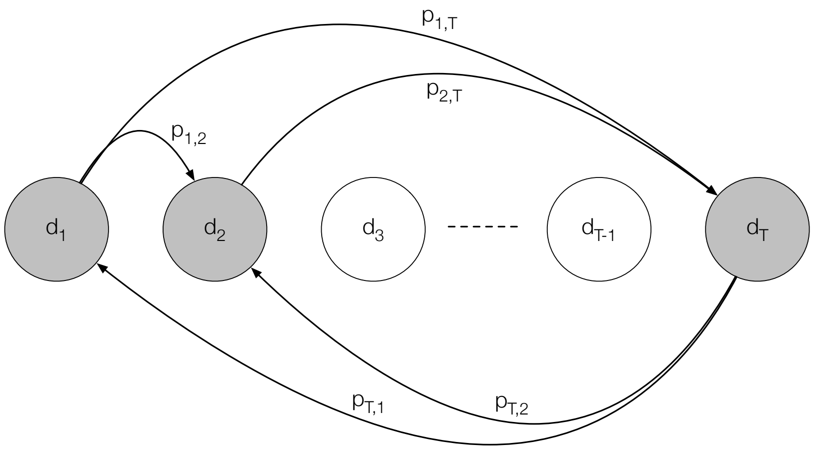


Fig. 5.4 Structure of the sub-Markov chain $(Y_n)_{n\in\mathbb{N}}$ (relevant documents are shown in grey; not relevant ones in white).

Theorem 7. *Let $\widetilde{P}$ be irreducible, λ be any distribution and $\mathscr{L}$ be the finite set containing the ranks of the relevant retrieved documents. If $(Y_n)_{n\geq 0}$ is Markov $(\lambda,\widetilde{P})$, then for any function $f : \mathscr{L} \to \mathbb{R}$ we have*

$$\mathbb{P}\left[\frac{1}{n}\sum_{m=0}^{n-1} f(Y_m) \to \overline{f} \text{ as } n \to \infty\right] = 1$$

where $\overline{f} = \sum_{j\in\mathscr{L}} \pi_j f(j)$ and π is the invariant distribution of $\widetilde{P}$.

The importance of this class of theorems is clear: almost surely and independently of the initial distribution λ, we can approximate, for n large, the average over the time by the (much simpler) average over the states of the Markov chain. Indeed, under the previous assumptions it is possible to prove that the matrix $\widetilde{P}$ admits a unique invariant distribution, i.e a probability distribution π such that if $(Y_n)_{n\geq 0}$ is Markov$(\pi,\widetilde{P})$, then for any n

$$\mathbb{P}[Y_n = j] = \pi_j \ .$$

Moreover, the invariant distribution in this case is the unique left eigenvector of the eigenvalue 1 of the matrix $\widetilde{P}$, i.e. the unique solution of the linear equation

$$\pi = \pi\widetilde{P} \ .$$

Remark. As presented in Section 5.1.1, under additional hypotheses, it can be proved that the invariant distribution itself is the limit of any row of the matrix $\widetilde{P}^n$, as $n \to \infty$, useful result in order to evaluate in practice the invariant distribution. The convergence is generally very fast and for $n = 10$ we already have a reasonable approximation of the true value of π.

This justifies the use of MP to approximate the mean precision of the usually few documents visited by a user.

We can now define a new family of user oriented retrieval effectiveness measures, called Markov Precision (MP), which depends on the specific user model and the invariant distribution derived.

Definition 8. Given a ranked list of retrieved documents, defined by $\mathscr{L}$ the ranks of its judged relevant documents and defined a Markov (λ,P) user model, the **Markov Precision** measure will be defined as

$$MP = \sum_{j \in \mathscr{L}} \pi_j \mathrm{Prec}(j).$$

where $\mathrm{Prec}(j)$ represent the Precision at j and π the (unique) invariant distribution of the Markov chain $(Y_n)_{n\in\mathbb{N}}$.

MP is defined without knowing the recall base *RB* of a given topic, but just the ranks of the judged relevant documents in a given run for this topic. As pointed out, for example in [Moffat and Zobel, 2008], the need to know the value of *RB* represents a weakness in AP that is overcome here.

In order to include the time dimension and thanks to the Ergodic Theorem for the continuous time Markov chains, we can replicate the previous computations and define a new measure

$$MPcont = \sum_{j \in \mathscr{L}} \widetilde{\pi}_j \mathrm{Prec}(j).$$

where

$$\widetilde{\pi}_j = \frac{\pi_j (\mu_j)^{-1}}{\sum_{i \in \mathscr{L}} \pi_i (\mu_i)^{-1}} \ ,$$

π denotes again the (unique) distribution of the Markov chain $(Y_n)_{n\in\mathbb{N}}$, and μ_j is the parameter of the holding time in state j. To use this alternative measure, we have to provide a calibration for the coefficients μ_j and we will compare MP with MPcont in a very simple example in Section 5.4 using click logs from Yandex.

5.3.2 Average Precision and the Markovian Model

In Section 2.3.1 the original definition of Average Precision (AP) [Buckley and Voorhees, 2005; Harman, 1993] is presented as the average over all *RB* judged relevant documents of the precision at their ranks, considering zero the precision at the not retrieved relevant documents:

$$AP = \frac{1}{RB} \sum_{i \in \mathscr{L}} \mathrm{Prec}(j) = \frac{RR}{RB} \cdot \frac{1}{RR} \sum_{j \in \mathscr{L}} \mathrm{Prec}(j) \tag{5.4}$$

where, in the last equation, the first operand is the recall and the second one is the arithmetic mean of the precisions at each relevant retrieved document. This formulation further highlights the dependence of AP on the recall base and the recall itself.

In order to define a simple Markovian user model, whose MP value will be AP, let us consider the following transition probabilities among the documents in a given ranked list:

$$\mathbb{P}[X_{n+1} = j | X_n = i] = \tfrac{1}{N-1} \tag{5.5}$$

for any $i, j \in \mathscr{I}$, $i \neq j$, and where, again, N denotes the cardinality of the set $\mathscr{I}$.

In this model we assume that a user moves from a document to another document with a fixed, constant probability, the value of which depends on the total number of relevant documents present in the specific run.

Since the invariant distribution is $\left(\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}\right)$ we obtain that

$$MP = \frac{1}{N} \sum_{j \in \mathscr{L}} Prec(j)$$

which is equal to AP once multiplied by $\frac{N}{RB}$. Note that if we create the Markov chain starting directly from the relevant documents $\mathscr{L}$ we have to multiply MP by $Rec(RR)$ as in equation 5.4. In this way, we explain AP with a slightly richer user model, where the user can move forward and backward among any document and is not forced to visit only the relevant ones. It is also clear from the equation above that MP is not AP unless you provide it with the same amount of information AP knows about the recall base, namely rescaling MP by the recall base.

Looking at this the other way around, this instantiation of MP (without the rescaling) can be considered a kind of AP where the artificial knowledge of the recall base has been removed and so, it tells us how AP might look like if you remove the dependency on the recall base and insert an explicit user model. This consideration will turn out to be useful in the experimental part when we will find other user models, highly correlated to AP, which may give a richer explanation of it.

Moreover, the previous constant invariant distribution is common to many others user models. For example, if the transition matrix is irreducible and symmetric or even just bistochastic, meaning that the sum of the entries on each column is equal to 1, the invariant distribution is again the above constant vector. In this sense, if the validity of the present Markovian user model is accepted, it shows once more why AP has become a reference point, since it represents a good approximation for a wide class of models that we can define.

5.3.3 Other Models

In this section we propose some basic models for the transition matrix of the Markov chain. Clearly, this is not intended to be an exhaustive list of all the possible models, but more of an exemplification of how it is possible to plug different user models into the framework. In Section 5.4.2 we will show how some of them are extremely highly correlated to AP, thus suggesting how AP can be considered a very good approximation of more complex user strategies.

We will analyze three possible choices:

- *state space choice*: the Markov chain $(X_n)_{n\in\mathbb{N}}$ is on the whole set $\mathscr{I}$, indicated with `AD` (all documents model), or on the set $\mathscr{L}$, indicated with `OR` (only relevant documents model);
- *connectedness*: the nonzero transition probabilities are among all the documents, indicated with `GL` (global model), or only among adjacent documents, indicated with `LO` (local model);
- *transition probabilities*: the transition probabilities are proportional to the inverse of the distance, indicated with `ID` (inverse distance model), or to the inverse of the logarithm of the distance, indicated with `LID` (logarithmic inverse distance model).

We will obtain eight models that we will call after the possible three choices. So, for example, MP `GL_AD_ID` is an effectiveness measure with transition probabilities among all the retrieved documents, based on a model on the whole set $\mathscr{I}$, and with transition probabilities proportional to the inverse of the distance of the documents in the ranked list and so on for the other combinations of the parameters.

State space choice

In the `AD` case, we consider the whole Markov chain $(X_n)_{n\in\mathbb{N}}$ on the whole set $\mathscr{I}$ with a given initial distribution λ and a transition matrix $P = (p_{i,j})_{i,j\in\mathscr{T}}$ and then we derive the subchain $(Y_n)_{n\in\mathbb{N}}$ on the set $\mathscr{L}$. In order to obtain the invariant distribution of the subchain, we will have to derive its transition matrix $\widetilde{P}$. It can be proved (see [Norris, 1998]) that this matrix can be defined as follows

$$\widetilde{p}_{i,j} = h_i^j \quad \text{for} \;\; i,j \in \mathscr{L}$$

where the vector $(h_i^j, i \in \mathscr{I})$ is the minimal non-negative solution to the linear system

$$h_i^j = p_{i,j} + \sum_{k \neq \mathscr{L}} p_{ik} h_k^j \, . \tag{5.6}$$

So, once this linear system is solved, we obtain the transition matrix $\widetilde{P}$ needed to compute the Markov Precision for the given model.

In the OR model, we create the Markov Chain $(X_n)_{n \in \mathbb{N}}$ directly on the set $\mathscr{L}$.

Connectedness

In the GL model, we assume that the transition probabilities $p_{i,j} > 0$ for any choice of $i \neq j$. In this case we will assume that there will be a positive, even if very small, probability to pass from any document in the ranked list to any other. For example, the previous model for Average precision is a GL model.

By contrast, in LO we will assume that there exist transition probabilities only among adjacent nodes. This is the same kind of logic behind RBP, even though RBP allows only for forward transitions, and is similar to the strategy of [Chierichetti et al., 2011] for the two-dimensional placement problem.

Transition probabilities

In the ID model, we assume that the probability to pass from one document to another one in the ranked list is proportional to the inverse of the relative distance of these two documents:

$$\alpha(i,j) = \begin{cases} \frac{1}{|i-j|+1} & \text{if } i \neq j \\ \\ 0 & \text{if } i = j \end{cases} \tag{5.7}$$

Denoting by $(s_1, \ldots, s_m)$ the states of the Markov chain, we thus have the following transition probabilities:

$$p_{s_i,s_j} = \frac{\alpha(s_i, s_j)}{\sum\limits_k \alpha(s_i, s_k)} \tag{5.8}$$

Table 5.1 Main features of the adopted data sets.

	Topics	Runs	Min. Rel	Avg. Rel	Max. Rel
TREC 7	50	103	7	93.48	361
TREC 8	50	129	6	94.56	347
TREC 10	50	97	2	67.26	372
TREC 14	50	74	9	131.22	376

It is immediately clear that the probabilities (5.8) define an irreducible transition matrix P of a discrete time Markov Chain on the state space and therefore we can define Markov precision for this model.

In the `LID` model, we smooth the distance by using the base 10 logarithm so that the transition probabilities do not decrease too fast. The choice of the base 10 for the logarithm is due to a typical Web scenario focused on the page of the first 10 results.

5.4 Experimental Evaluation of MP

5.4.1 Experimental Setup

Evaluation measures of direct comparison, which are used in this section, are those built around the concept of precision, namely AP, P@10, and Rprec [Buckley and Voorhees, 2005]. RBP [Moffat and Zobel, 2008] comes into play as a binary evaluation measure not dependent on the recall base, even though it is not built around the concept of precision despite its name. Finally, we are also interested in bpref [Buckley and Voorhees, 2004], just to have a comparison point when testing MP with respect to reduced-size pools. In this last respect, we are not interested in infAP [Yilmaz and Aslam, 2006], since we are neither looking for an estimator of AP nor investigating alternative strategies for pool downsampling. For the same reason, we are not interested here in experimenting with respect to condensed-list measures [Sakai, 2007].

In order to assess MP and compare it to the afore mentioned evaluation measures, we conducted a correlation analysis and we studied its robustness to pool downsampling. As far as RBP is concerned, we set $p = 0.8$, which indicates a medium persistence of the user.

We used the following data sets: TREC 7 Ad Hoc, TREC 8 Ad Hoc, TREC 10 Web, and TREC 14 Robust, whose features are summarized in Table 5.1. We used all the topics and all the runs that retrieved at least one document per topic. In the case of collections with graded relevance assessment (TREC 10 and 14), we mapped them to binary relevance with a lenient strategy, i.e. both relevant and highly relevant documents have been mapped to relevant ones.

Table 5.2 Kendall τ correlation between AP and the other comparison measures using complete judgments (high correlations marked with *).

	AP	P@10	Rprec	bpref	RBP
TREC 7	1.000	0.8018	0.9261*	0.9275	0.7886
TREC 8	1.000	0.8264	0.9219*	0.9361*	0.8090
TREC 10	1.000	0.7551	0.8730	0.8896	0.7401
TREC 14	1.000	0.7295	0.9377	0.8394	0.7229

As far as pool downsampling is concerned, we used the same strategy of [Buckley and Voorhees, 2004]: it basically creates separate random lists of relevant/not relevant documents and select a given fraction $R\%$ of them, ensuring that at least 1 relevant and 10 not relevant documents are in the pool. We used $R\% = [90, 70, 50, 30, 10]$.

As far as the calibration of time is concerned, we used click logs made available by Yandex [Serdyukov et al., 2012] in the context of the Relevance Prediction Challenge[2]. The logs consist of 340,796,067 records with 30,717,251 unique queries, retrieving 10 URLs each. We used the training set where there are 5,191 assessed queries which correspond to 30,741,907 records and we selected those queries which appear at least in 100 sessions each to calibrate the time.

The full source code of the software used to conduct the experiments is available for download[3] in order to ease comparison and verification of the results.

5.4.2 Correlation Analysis

Table 5.2 reports the Kendall τ correlation [Kendall, 1945] between AP and the other comparison measures, using complete judgements, for all the collections. Previous work [Voorhees, 2000, 2001] considered correlations greater than 0.9 as equivalent rankings and correlations less than 0.8 as rankings containing noticeable differences. Table 5.2 is consistent with previous findings, with a high correlation between AP, Rprec, and bpref and lower correlation values for P@10 and RBP.

Table 5.3 reports the Kendall τ correlation between the different models for MP, discussed in Section 5.3.3 and whose notation (GL/LO, AD/OR, ID/LID) is used here as well, and the performance measures of direct comparison, for all the considered collections[4]. For each variant of MP, the table reports its actual value and also a second row labelled with the

[2] http://imat-relpred.yandex.ru/en/

[3] http://matters.dei.unipd.it/

[4] The fact that the values for the LO_AD_ID and LO_AD_LID models are the same is not due to a copy&paste error but to the fact that the two chains, in the local model, are the same apart from a constant and so they produce equal rankings.

Table 5.3 Kendall τ correlation between different instantiations of MP and the other comparison measures using complete judgments (high correlations marked with *; extremely high correlations marked with **).

	TREC 7					TREC 8				
	AP	P@10	Rprec	bpref	RBP	AP	P@10	Rprec	bpref	RBP
MP GL_AD_ID	0.7381	0.7522	0.7703	0.7827	0.7490	0.8997	0.8510	0.9074*	0.9222*	0.8382
MP GL_AD_ID@$Rec(T)$	0.9823**	0.7916	0.9243*	0.9322*	0.7799	0.9815**	0.8128	0.9217*	0.9299*	0.7938
MP GL_AD_LID	0.7378	0.7638	0.7712	0.7802	0.7632	0.8912	0.8641	0.9033*	0.9173*	0.8551
MP GL_AD_LID@$Rec(T)$	0.9954**	0.7994	0.9252*	0.9277*	0.7858	0.9953**	0.8221	0.9209*	0.9337*	0.8041
MP GL_OR_ID	0.7322	0.8311	0.7797	0.7689	0.7689	0.8162	0.9081*	0.8349	0.8402	0.9152*
MP GL_OR_ID@$Rec(T)$	0.9117*	0.8316	0.8937	0.8848	0.8243	0.9208*	0.8756	0.9024*	0.9145*	0.8637
MP GL_OR_LID	0.7379	0.7853	0.7782	0.7788	0.7858	0.8664	0.8884	0.8853	0.8947	0.8858
MP GL_OR_LID@$Rec(T)$	0.9726**	0.8158	0.9238*	0.9232*	0.8029	0.9722**	0.8477	0.9281*	0.9390*	0.8324
MP LO_AD_ID	0.7435	0.7706	0.7706	0.7874	0.7685	0.8931	0.8642	0.9011*	0.9174*	0.8537
MP LO_AD_ID@$Rec(T)$	0.9946**	0.7994	0.9225*	0.9265*	0.7858	0.9953**	0.8248	0.9219*	0.9343*	0.8066
MP LO_AD_ID	0.7435	0.7706	0.7706	0.7874	0.7685	0.8931	0.8642	0.9011*	0.9174*	0.8537
MP LO_AD_ID@$Rec(T)$	0.9946**	0.7994	0.9225*	0.9265*	0.7858	0.9953**	0.8248	0.9219*	0.9343*	0.8066
MP LO_OR_ID	0.7271	0.8229	0.7754	0.7634	0.8393	0.8138	0.9013*	0.8305	0.8354	0.9176*
MP LO_OR_ID@$Rec(T)$	0.9130*	0.8283	0.8958	0.8853	0.8211	0.9195*	0.9195*	0.8714	0.8987	0.9127*
MP LO_OR_LID	0.7386	0.8065	0.7826	0.7787	0.8058	0.8534	0.8982	0.8708	0.8810	0.8995
MP LO_OR_LID@$Rec(T)$	0.9552*	0.8278	0.9166*	0.9142*	0.8164	0.9506*	0.8623	0.9186*	0.9319*	0.8466

	TREC 10					TREC 14				
	AP	P@10	Rprec	bpref	RBP	AP	P@10	Rprec	bpref	RBP
MP GL_AD_ID	0.7264	0.7832	0.7727	0.7611	0.8013	0.8351	0.8078	0.8566	0.7778	0.7980
MP GL_AD_ID@$Rec(T)$	0.9726**	0.7340	0.8631	0.8771	0.8771	0.9896**	0.7221	0.9333*	0.8360	0.7140
MP GL_AD_LID	0.7125	0.7971	0.7633	0.7494	0.8187	0.8294	0.8185	0.8501	0.7751	0.8071
MP GL_AD_LID@$Rec(T)$	0.9941**	0.7512	0.8707	0.8878	0.7360	0.9977**	0.7303	0.9385	0.8397	0.8397
MP GL_OR_ID	0.7034	0.8269	0.7663	0.7470	0.8590	0.7968	0.8461	0.8206	0.7677	0.8302
MP GL_OR_ID@$Rec(T)$	0.9117*	0.8316	0.8937	0.8848	0.8243	0.9601*	0.7526	0.9327*	0.8650	0.7444
MP GL_OR_LID	0.7052	0.8077	0.7672	0.7466	0.8396	0.8140	0.8291	0.8348	0.7716	0.8155
MP GL_OR_LID@$Rec(T)$	0.9738**	0.7575	0.8740	0.8916	0.7448	0.9924**	0.7375	0.9398*	0.8432	0.7293
MP LO_AD_ID	0.7240	0.7969	0.7703	0.7614	0.8159	0.8297	0.8180	0.8504	0.7783	0.8089
MP LO_AD_ID@$Rec(T)$	0.9742**	0.7376	0.8654	0.8802	0.7218	0.9970**	0.7295	0.9363*	0.8405	0.7214
MP LO_AD_ID	0.7240	0.7969	0.7703	0.7614	0.8159	0.8297	0.8180	0.8504	0.7783	0.8089
MP LO_AD_ID@$Rec(T)$	0.9742**	0.7376	0.8654	0.8802	0.7218	0.9970**	0.7295	0.9363*	0.8405	0.7214
MP LO_OR_ID	0.7035	0.8300	0.7646	0.7449	0.8618	0.7997	0.8348	0.8234	0.7714	0.8220
MP LO_OR_ID@$Rec(T)$	0.9326*	0.7726	0.8767	0.8960	0.7618	0.9674*	0.7429*	0.9348*	0.8597	0.7377
MP LO_OR_LID	0.7114	0.8172	0.7676	0.7533	0.8472	0.8084	0.8324	0.8306	0.7689	0.8180
MP LO_OR_LID@$Rec(T)$	0.9579*	0.7601	0.8747	0.8949	0.7477	0.9877**	0.7372	0.9381*	0.8489	0.7306

suffix @$Rec(T)$ to indicate a rescaled version of MP by recall. Indeed, this is the same operation needed to make MP equal to AP in the case of the model with constant transition probabilities discussed in Section 5.3.2 and corresponds to providing MP with the same level of information about the recall base that also AP uses. This has a twofold purpose: (i) to determine if there are other models beyond the ones of Section 5.3.2 which can give us an additional interpretation of AP; (ii) to get a general feeling of what is the impact of injecting information about the recall into an evaluation measure. In the table, we have marked high correlations, those above 0.90, with a star and we have marked extremely high correlations, those above 0.97, with two stars.

As a general trend MP tends not to have high correlations with the other evaluation measures, indicating that it takes a different angle from them. This can be accounted for by the effect of the user model explicitly embedded in MP which, for example, allows the user to move forward and backward in the result list while other measures allow only for sequential scans. On the other hand, the proposed models keep it not too far away from the other measures, especially those around precision (AP, P@10, Rprec), since the correlation never drops below 0.70. This is coherent with the fact that both MP and the other measures (AP, P@10, Rprec) are all around the concept of precision and so they have a common denominator.

Moreover, it can be noted that MP tends to be more correlated with P@10 and then with Rprec and AP. This is consistent with the fact that MP does not depend on the recall base, as P@10 does, while Rprec implicitly and AP explicitly depend on it.

Finally, the results show a moderate correlation with bpref and a slightly lower one with RBP, whose only common denominator is to not depend on the recall base.

With regard to @$Rec(T)$, we can note how they greatly boost the correlation with AP in almost all cases, often moving MP from low to high correlations, and, in turn, increase the correlation with Rprec and bpref (more correlated by themselves to AP) with respect to the one with RBP which tends to decrease.

In particular, there are some cases, like MP `GL_AD_LID` or MP `LO_AD_ID`, where it jumps between 0.97 and 1.00. We consider this a case in which MP is providing us with an alternative interpretation of AP, in the sense discussed in Section 5.3.2. For example, MP `GL_AD_LID` provided with information about recall tells us that we can look at AP as a measure that also models a user who can move backward and forward among all the documents in the list and who prefers smaller jumps to bigger ones. The fact that we have found a few models so highly correlated with AP suggests that AP has become a gold standard also because it represents some articulated user models.

5.4.3 Effect of Incompleteness on Absolute Performances

Figure 5.5 shows the effect of reducing the pool size on the absolute average performances, over all the topics and runs. We do not report figures for all the possible combinations reported in Table 5.3, but just some to give the reader an idea of the behavior of MP; the considerations made here are however valid also for the not reported figures.

It can be noted how MP shows consistent behavior over all the collections and for various models: its absolute average values decrease as the pool reduction rate increases in a manner similar to AP and Rprec. Consistently with previous results, P@10 and RBP exhibit a more marked decrease while bpref tends to stay constant. This positive property of bpref is an indicator that it is not very sensible or it does not fully exploit the additional information which is provided when the pool increases.

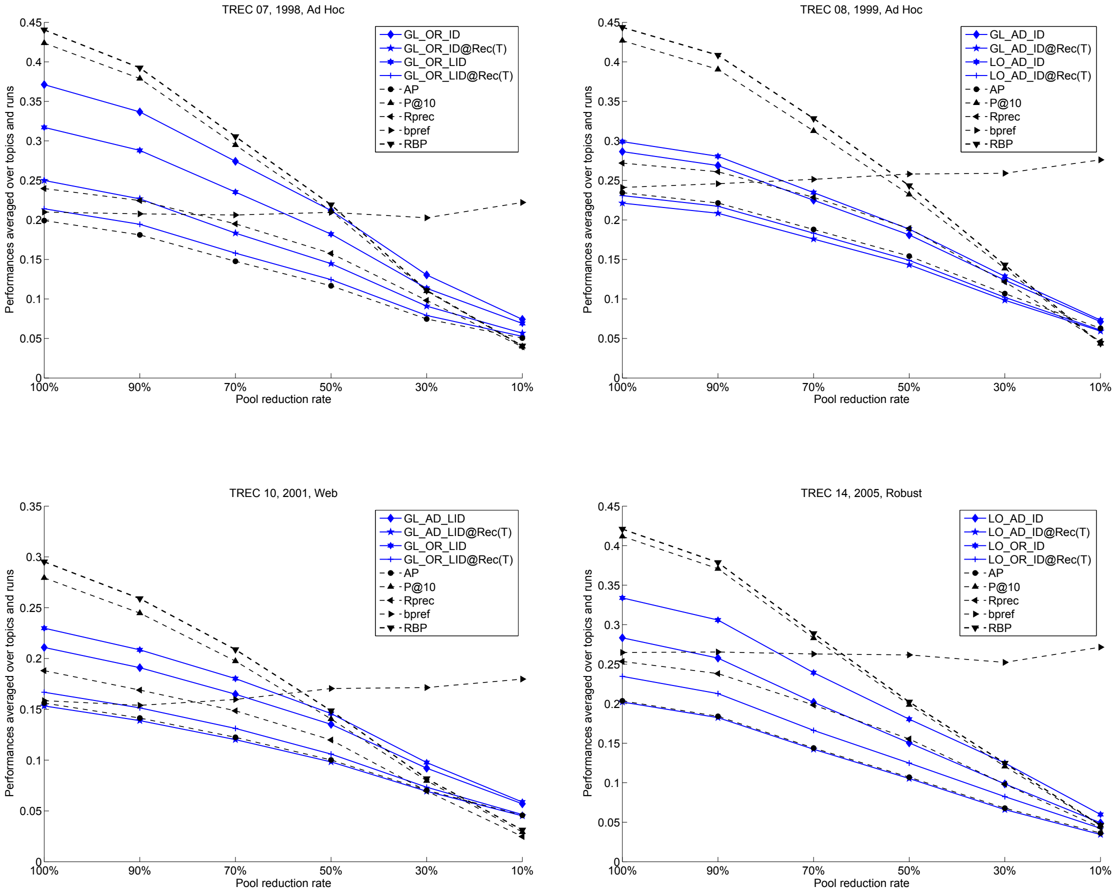


Fig. 5.5 Pool reduction rate (*x* axis) vs. performance averaged over topics and runs (*y* axis)

5.4.4 Effect of Incompleteness on Rank Correlation

Figure 5.6 shows the effect of reducing the pool size on the Kendall τ correlation between each measure on the full pool and the pool at a given reduction rate. The results shown are consistent with previous findings as far as the measures of direct comparison are concerned, showing that bpref is almost always the more robust measure to pool reduction. It is indeed plausible that, keeping bpref the absolute average performances almost constant, also the ranking of the systems does not change much.

As far as MP is concerned, we can note that global models [GL], shown in the case of TREC 7, 8 and 10, tend to perform comparably to AP and, when provided with the same information about the recall base, which both AP and bpref exploit, they consistently improve their performances and, in the case of TREC 8, they outperform AP and perform closely to bpref. This is an interesting result since, unlike bpref, the absolute average performances of MP vary at different pool reduction rates, indicating that MP is able to exploit the variable amount of information available at different pool reduction rates, still not affecting too much the overall ranking of the systems.

The global models [GL] on only relevant documents [OR] behave consistently with the global ones on all documents [AD], shown in the case of TREC 7 and TREC 10, even if they are a little bit more resilient to the pool reduction. This is consistent with the fact that they use less information than the [AD] ones and so they are less sensitive to the pool size. The TREC 7 also shows the effect of using the inverse of the distance [ID] or the log of the inverse of the distance [LID], which provides more robustness to pool reduction.

When it comes to local models [LO], these tend to behave comparably to the global ones in the case of all documents [AD], as can be noted in the case of TREC 8, while they are more affected by the pool reduction in the case of only relevant documents [OR], as can be noted in the case of TREC 14.

5.4.5 Time Calibration

On the basis of the click logs, 22.6% of the observed transitions are backward, a fact that validates our assumption that a user moves forward and backward along the ranked list.

To compare the discrete-time version of MP with the continuous-time one, we have considered 3 runs with 5 relevant documents and estimated the parameters of the exponential holding times by the inverse of the sample mean of the time spent by the users visiting these states, multiplied by $(N-1)/N$. We used the GL_AD_ID model and the values of discrete-time MP and continuous-time MP are reported in Table 5.4.

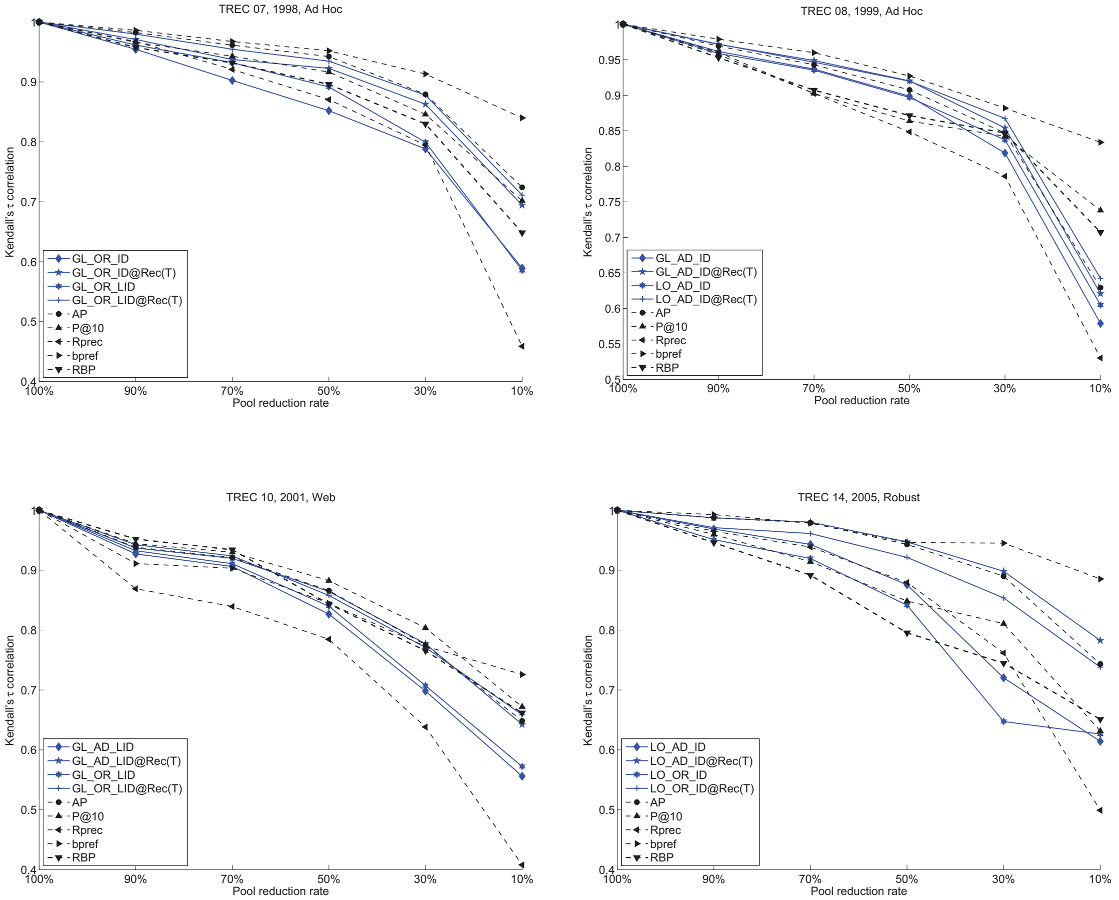


Fig. 5.6 Pool reduction rate (*x* axis) vs. Kendall's rank correlation (*y* axis)

Note that the precisions at each fixed rank *n* of the first, second and third runs are decreasing and as one expects MP of the three runs is decreasing. However, since the (estimated) holding times of the first documents in the first run are very low, continuos-time MP is smaller for the first run. This clearly shows that the use of continuous-time MP depends heavily on the calibration of the holding times.

Table 5.4 Estimated parameters of the exponential holding times for three runs and values of the discrete-time and continuous-time MP.

Run	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}	disc MP	cont MP
(1,1,1,1,0,0,0,1,0,0)	0.2000	0.0357	0.2000	0.0400	0.0056	0.0005	0.0035	0.0017	0.0034	0.0024	0.9205	0.6603
(1,1,1,0,1,0,0,0,1,0)	0.0177	0.0047	0.0037	0.0015	0.0041	0.0031	0.0057	0.0022	0.0061	0.0045	0.8668	0.8710
(1,1,0,1,1,0,0,0,0,1)	0.0056	0.0051	0.0062	0.0031	0.0046	0.0025	0.005	0.0022	0.007	0.005	0.8120	0.8001

5.5 nMCG: Embedding the Markovian Model into LtR

We now proceed to illustrate the second main result derived from the Markovian user model presented in Section 5.2. In the following we will define the user dynamic function and we will describe how we can integrate it in LAMBDAMART.

As shown in Section 2.3.1, effectiveness is often measured as the inner product of a relevance vector and a discounting vector, assuming that low ranked documents receive less attention, therefore they should contribute less to the system score. Defining a proper quality metric is crucial both for evaluating retrieval systems and for learning effective ranking models, as such metrics are used to drive the training process.

Most metrics assume that the user analyzes a SERP from top to bottom, and therefore define a decreasing discount vector, however some user studies suggest that the probability of observing a result depends on the quality of the documents ranked higher: if the user finds a relevant document at position i it is less likely that he will inspects the document at position $i+1$ [Zhang et al., 2010]. Furthermore, the user behavior is more complex, as she can move forward and backward, can jump from one document to any other and visit already visited documents, as suggested by [Sakai and Dou, 2013] and the results presented in Section 5.4.5.

Our work stems from the simple observation that the user behavior in visiting a SERP differs depending on the query type and the number of relevant results. For example, it is likely that on a SERP with a single highly relevant result in the first position the user assumes a **navigational** behavior, while a SERP with several relevant results may likely correspond to an **informational** query, where a more complex SERP visiting behavior can be observed [Broder, 2002]. Since at training time a list-wise LtR algorithm such as LAMBDAMART is aware of the number and distribution of relevance labels associated with the training samples for each query, we suppose that it can profit from the knowledge of the user dynamic associated with the specific kind of query. In the following we discuss our model of user dynamic and the methodology followed to integrate it into LAMBDAMART.

5.5.1 Modeling the User Dynamic

We model the user dynamic with the Markovian process [Norris, 1998] presented in Section 5.2, where the user scans the ranked documents in the SERP according to possibly complex paths. Recall that under the assumption of irreducibility and aperiodicity, the transition matrix P admits a unique stationary distribution $\pi = \pi P$, which is the limit of the n-step transition probabilities $p_{ij}^{(n)} \to \pi_j$ as $n \to \infty$ for all i, j [Norris, 1998]. When extending this analysis to a long-term query log, we can consider the behavior recorded for each user as a different observation of the same stochastic process, and the resulting stationary distribution

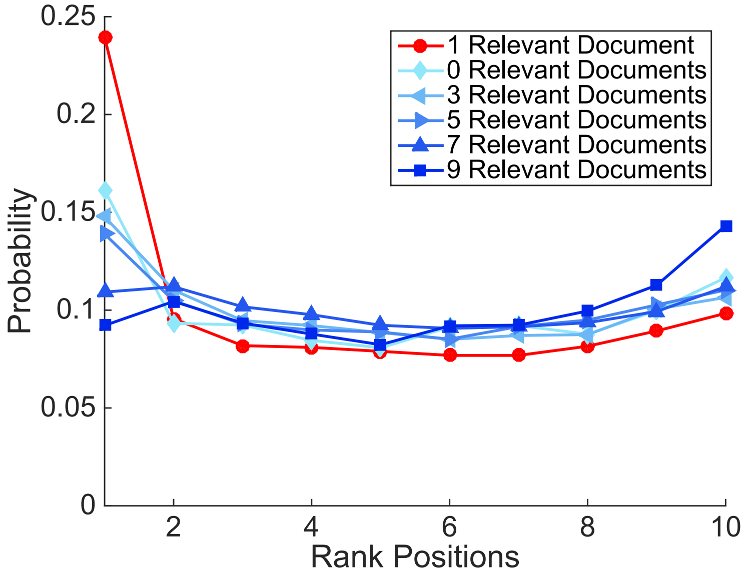


Fig. 5.7 Stationary distributions for queries retrieving different number of relevant documents.

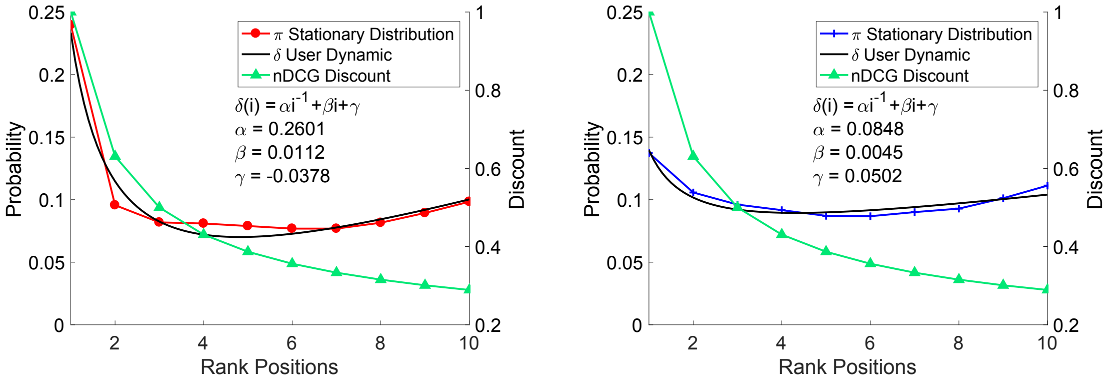
can be considered as an aggregated representation of user dynamics. In addition, since we observe that the behavior of users change depending on the number of relevant documents in the SERP, we can classify queries on the basis of the number of relevant documents returned and estimate different transition matrices $\hat{P}$ for different classes of queries. Specifically, we first aggregate the dynamics of different users on the basis of the typology of query, then we adopt the maximum likelihood estimator approach [Teodorescu, 2009] on the aggregated data:

1. for each $i \in \mathscr{I}$ let v_i be the number of times that the users visited the document at rank i given the query;

2. if $v_i = 0$, then $\hat{p}_{ij} = 0$ for all $j \neq i$ and $\hat{p}_{ii} = 1$;

3. if $v_i > 0$, let v_{ij} be the number of transitions from document at rank i to document at rank j, then $\hat{p}_{ij} = \frac{v_{ij}}{v_i}$.

Figure 5.7 plots the stationary distributions obtained from the Yandex query log detailed in Section 5.6.1. When considering queries with just one relevant retrieved document, i.e. the red line with circle markers in Figure 5.7, the user dynamic exhibits a spike with respect to the first rank position, while for queries without any relevant documents or with more than one relevant document, i.e. the blue lines, the probability tends to be distributed more uniformly, meaning that the user is exploring the whole SERP.



(a) Navigational Queries (b) Informational Queries

Fig. 5.8 Stationary distribution with its fitted curve and DCG discount for navigational (a), and informational queries (b).

We focus on these two distinct macroscopic behaviors, and, for the sake of simplicity, we call navigational the queries where users concentrated on just the first item, and we consider all the other queries as informational since users tend to visit more documents.

To embed user dynamics in the LAMBDAMART cost function, as detailed in the next section, we abstract these two observed behaviors (navigational and informational) by fitting a curve to the corresponding stationary distributions. In particular, the user dynamic is described as a mixture of the navigational and informational behavior. The navigational component is represented by the inverse of the rank position $\frac{1}{i}$, while the informational component is linear with respect to the rank position i. Therefore, we model the **user dynamic** as

$$\delta(i) = \alpha i^{-1} + \beta i + \gamma$$

where the parameters α, β and γ are calibrated in order to fit the estimated stationary distributions computed on the Yandex dataset.

Figures 5.8a and 5.8b show the stationary distributions together with the fitted curves for the navigational and informational cases, respectively. In Figure 5.8a the stationary distribution is the same reported in the red line of Figure 5.7, while to compute the stationary distribution reported in Figure 5.8b we aggregate all the user dynamics corresponding to the other queries, i.e. queries without relevant documents or with more than one relevant document.

The user dynamic defined above can actually be considered as a discounting vector to be exploited in any given quality metric. Differently from other approaches, the user dynamic is defined on the basis of two different query classes which exhibit a different user behavior.

Figures 5.8a and 5.8b show how different is the derived user dynamic with respect to the DCG discounting component: in both cases DCG discounts more the documents at rank position greater than 4, while the invariant distribution tends to consider them uniformly. Furthermore, at the top of the ranking, DCG discount almost coincides with the invariant distribution in the navigational case, however for the informational queries, there is a considerable difference between the two functions. Below we discuss how the user dynamic δ can be exploited in a state-of-the-art LtR algorithm.

5.5.2 Integrating the User Dynamic into LtR

A LtR algorithm exploits a ground-truth set of training examples in order to learn a document scoring function σ [Liu, 2011]. Such training set is composed of a collection of queries $\mathcal{Q}$, where each query $q \in \mathcal{Q}$ is associated with a set of assessed documents $D = \{d_0, d_1, \ldots\}$. Each document d_i is labeled by a relevance judgment $GT(q, d_i)$ according to its relevance to the query q. These labels induce a partial ordering over the assessed documents, thus defining an ideal ranking which the LtR algorithm aims at approximating. Each query-document pair (q, d_i) is represented by a vector of features x, able to describe the query (e.g., its length), the document (e.g., the in-link count) and their relationship (e.g., the number of query terms in the document).

Since IR measures are not differentiable, their optimization is very challenging. To address this issue, the state-of-the-art solution is the LAMBDARANK gradient approximation [Burges et al., 2007]. LAMBDARANK is a pairwise LtR algorithm, which learns the pairwise preferences between documents by measuring the cost variation after swapping any two documents in a given result list. As discussed in [Donmez et al., 2009], this approach can be applied to several IR measures and it is capable of accurately discovering local optima.

LAMBDARANK can be summarized as follows. Consider a ranking of documents generated for a query q after a training iteration of the model. Assume that d_i and d_j are two candidate documents for the same query q, with relevance labels $\hat{r}[i]$ and $\hat{r}[j]$ respectively, s_i and s_j are the currently predicted document scores. The lambda gradient of any given IR quality function Q is:

$$\lambda_{i,j} = S_{i,j} \left| \Delta Q_{ij} \cdot \frac{\partial C_{i,j}}{\partial o_{i,j}} \right|$$

where $S_{i,j}$ is equal to 1 if d_i is more relevant than d_j and -1 otherwise, therefore we have $S_{i,j} = \operatorname{sgn}(\hat{r}[i] - \hat{r}[j])$. ΔQ is the quality variation when swapping the documents d_i and d_j, i.e. the difference in the measure value when performing a swap operation in the sense of Section 3.4. Notice that when the documents have the same relevance degree $\hat{r}[i] = \hat{r}[j]$, then ΔQ is equal to 0. $C_{i,j}$ is the cross entropy score and is a function of $o_{i,j}$, where $o_{i,j}$ is the difference of the document scores $o_{i,j} = s_i - s_j$. $C_{i,j}$ is defined as:

$$C_{i,j}(o_{i,j}) = s_j - s_i + \log\left(1 + e^{(s_i - s_j)}\right)$$

therefore the derivative with respect to $o_{i,j}$ is

$$\frac{\partial C_{i,j}(o_{i,j})}{\partial o_{i,j}} = -\frac{1}{1 + e^{(s_i - s_j)}}$$

with the sigmoid function that accounts for the difference in the documents' scores. Therefore, if s_i is much greater than s_j, the derivative of the cost function tends to 0, if s_i is much lower than s_j, the derivative of the cost function tends to 1, and if s_i is close to s_j, then the cost function will take values close to $1/2$.

To conclude the lambda gradient are computed as:

$$\lambda_{ij} = \ \operatorname{sgn}(\hat{r}[i] - \hat{r}[j]) \left| \Delta Q_{ij} \cdot \frac{1}{1 + e^{s_i - s_j}} \right|$$

where, the sign is determined by the document labels only, the first factor ΔQ is the quality variation when swapping documents d_i and d_j, and the second factor is the derivative of the RankNet cost [Burges et al., 2005], which minimizes the number of disordered pairs.

When $\hat{r}[i] \geq \hat{r}[j]$, the $\lambda_{i,j}$ score is positive, meaning that the quality Q increases with the score of document d_i. The larger the quality variation ΔQ, the higher the document d_i should be scored. Note that the RankNet multiplier fades ΔQ if documents are scored correctly, i.e. if $s_i \geq s_j$ and $\hat{r}[i] \geq \hat{r}[j]$, indeed the derivative of the cost function will be close to zero. Conversely, if $s_i \leq s_j$ and $\hat{r}[i] \geq \hat{r}[j]$, which means that the more relevant document is placed lower in the ranking, then the document d_j needs to be pushed towards the top and the derivative boosts ΔQ accordingly to the difference between the scores s_i and s_j, since $\partial C_{i,j} / \partial o_{i,j}$ will tend to 1.

The lambda gradient for a document d_i is computed by marginalizing over all possible pairs in the result list: $\lambda_i = \sum_j \lambda_{ij}$. LambdaRank uses nDCG as Q and so ΔQ is the variation in nDCG caused by the swap of two documents. Finally, LambdaMart is a combination of LambdaRank and MART, a boosted tree algorithm [Friedman, Jerome H, 2001]. Since IR measures are not continuous, therefore it is not possible to compute the gradient, LambdaRank is exploited to approximate the gradient of the objective function, while MART is used to perform efficiently gradient descent.

We enhance the existing LambdaMart algorithm by replacing the above Q with a new quality measure which integrates the proposed user dynamic δ. This new measure is called

Normalized Markov Cumulated Gain (nMCG) and it is defined as follows:

$$\text{nMCG}@k = \frac{\sum_{i \le k} \left(2^{\hat{r}[i]} - 1\right) \cdot \delta^c(i)}{\sum_{h \le k, \text{sorted by } \hat{\text{id}}[\text{h}]} \left(2^{\hat{id}[h]} - 1\right) \cdot \delta^c(h)}$$

where $\hat{r}[i]$ is the relevance label of the i-th ranked document and $\delta^c(i)$ is the user dynamic function at rank i relative to the query class c, either navigational or informational. Basically, nMCG can be seen as an extension of nDCG where the discount function is defined by the user dynamic and depends on the query class. Recall that, the discount function of nDCG and the invariant distribution are quite different, being similar just for the navigational case, when considering the top rank positions. Moreover, since δ^c depends on the query class, i.e. depends on the query q, we are optimizing two different variants of the same quality measure nMCG across the training dataset. Finally, ΔnMCG_{ij} can be computed efficiently as follows:

$$\Delta\text{nMCG}_{ij} = \frac{-\left(2^{\hat{r}[i]} - 2^{\hat{r}[j]}\right)\left(\delta^c(i) - \delta^c(j)\right)}{\sum_{h \le k, \text{sorted by } \hat{\text{id}}[\text{h}]} \left(2^{\hat{id}[h]} - 1\right) \cdot \delta^c(h)}.$$

Hereinafter, we use **nMCG-MART** [Ferro et al., 2017] to refer to the described variant of LAMBDAMART aimed at maximizing nMCG.

Note that the query class is known at training time, and therefore the algorithm can optimize the proper user dynamic δ^c. However, neither the document relevance, nor the query class information are available at test time, therefore the algorithm should, at the same time, learn how to classify queries and how to rank documents according to the different class-based dynamics δ^c.

5.6 Experimental Evaluation of nMCG-MART

5.6.1 Experimental Setup

We remark that there is no publicly available dataset providing user session data, document relevance and query-document pairs features at the same time. Therefore, we have to use two different datasets: the first for the user dynamic derivation and the second for the LtR analysis.

We calibrate the proposed user model on the basis of the click log dataset provided by Yandex [Serdyukov et al., 2012] [5]. The dataset is composed of 340,796,067 records with

[5] http://imat-relpred.yandex.ru/en/

Table 5.5 nDCG@10 and nMCG@10 across test datasets for different model sizes (results are averaged across the 5 folds for Microsoft datasets.). Statistically significant differences at $p = 0.05$ and at $p = 0.01$ w.r.t. λ-MART marked resp. with * and **.

	MSLR-WEB30K			MSLR-WEB10K			Istella-S		
	100	500	Full	100	500	Full	100	500	Full
Algorithm	nDCG@10								
λ-MART	0.4564	0.4759	0.4793	0.4479	0.4637	0.4634	0.7031	0.7451	0.7536
nMCG-MART	0.4598**	0.4778**	0.4808**	0.4499**	0.4646	0.4648*	0.7070**	0.7466*	0.7549
Algorithm	nMCG@10								
λ-MART	0.4684	0.4878	0.4914	0.4609	0.4767	0.4768	0.7551	0.7970	0.8059
nMCG-MART	0.4718**	0.4898**	0.4933**	0.4626*	0.4782	0.4790**	0.7595**	0.8000**	0.8090**

30,717,251 unique queries, retrieving 10 URLs each. We used the training set, which consists of 5191 assessed queries with binary judgments, corresponding to 30,741,907 records. Notice that 9% of the sessions corresponds to navigational queries while the remaining 91% corresponds to informational ones.

The accuracy of the proposed algorithm is evaluated on three public LtR datasets, MSLR-WEB30K and MSLR-WEB10K, provided by Microsoft [Qin and Liu, 2013] and Istella provided by Tiscali Istella Web search engine [Dato et al., 2016]. Dataset MSLR-WEB30K encompasses 31,531 queries from the Microsoft Bing search engine for a total of 3,771,125 query-document pairs represented by 136 features. The dataset is provided as a 5-fold split. The MSLR-WEB10K dataset contains 10,000 queries samples at random from the previous. Dataset Istella provides 33,018 queries for a total of 3,408,630 query-document pairs represented by 220 features. The dataset is provided as a 60/20/20 train/validation/test split.

Both the Microsoft and Istella datasets use integer relevance labels in the range $[0,4]$. In order to classify queries as navigational or informational we adopt the following criterion. A query is considered as navigational if it contains only one result with relevance label ≥ 3. Approximatively 15% of the queries in the Microsoft datasets are classified according to this heuristic as navigational queries, which is quite similar to the value measured on the Yandex dataset. The Istella dataset instead contains a smaller set of navigational queries, covering about 3% of the dataset.

5.6.2 Experimental Results

We compare the effectiveness of state-of-the-art LtR algorithm λ-MART and nMCG-MART both in terms of nDCG@10 and nMCG@10 metrics. Recall that, at training time, λ-MART optimizes nDCG while nMCG-MART exploits the proposed nMCG metric. The algorithms' hyper-parameters were set after parameter sweeping, similarly to [Capannini et al., 2016], to a learning rate of 0.05, maximum number of leaves of 64, and a maximum number of trees of 1500. The actual number of trees is tuned on the validation set. We also evaluate smaller models with 100 and 500 trees. In Table 5.5 we report the effectiveness scores of the proposed algorithm computed in terms of nDCG@10 and nMCG@10.

We first observe that nMCG-MART is more effective in optimizing nMCG in every dataset and with every model size. This was expected as the proposed algorithm is the only one aimed at optimizing the proposed nMCG. At the same time, this confirms the soundness of the integration of nMCG into LAMBDAMART.

An interesting result is that nMCG-MART always provides higher nDCG@10 than LAMBDAMART. Recall that even relative improvements in nDCG below 1% are significant in terms of user satisfaction [Chapelle et al., 2012]. According to randomization test, the improvement is statistically significant at $p = 0.01$ on the larger MSLR-WEB30K dataset and on the other datasets limited to the small models with 100 trees. The proposed nMCG seems to provide more stable results, as optimizing nMCG also helps in optimizing nDCG. We believe that nMCG@k is somehow a simpler function to maximize: for informational queries it mainly discriminates between documents inside and outside the top-k results, and for navigational queries an additional boost is given if the relevant document is ranked first. This possibly drives the learning algorithm along a smoother cost function. The benefit is larger at the initial training iterations as suggested by the statistically significant improvements on small models with 100 trees, where difference is at $p = 0.01$ on every dataset. Larger models reach a plateau of effectiveness where it is anyway difficult to improve further. These hypotheses needs a detailed investigation as part of our future work.

We conclude that the proposed nMCG may provide a better modeling of the user behavior and perceived quality of a SERP, and that it may also provide high quality rankings according to other quality metrics of interest.

5.7 Summary

We proposed a user model based on Markov chain that allows the user to follow non linear patterns in scanning the ranked list of results. Based on this model we obtain two main results, we propose a family of new evaluation measures, called MP and we model the user dynamic that we integrate in LAMBDAMART.

MP exploits Markov chains in order to inject different user models and time into precision and is not dependent on the recall base. This allowed us to overcome some of the traditional criticisms of AP (lack of a clear user model, dependence on the recall base) while still offering a measure which is AP when provided with the same amount of information about the recall base that AP exploits. Moreover, MP goes beyond almost all the evaluation measures allowing for non sequential scanning of the result lists.

We have proposed some basic user interaction models and validated their properties, in terms of correlation to other measures and robustness to pool reduction, thus showing it is as reliable as them. We have also found that some of these models have an extremely high correlation with AP and this can help in providing alternative interpretations of AP in the light of more complex user models and in explaining why AP is a "gold standard" in IR.

MP also bridges the gap between "rank-oriented" and "time-oriented" measures, providing a single unified framework where both viewpoints can co-exist and allowing for direct comparison among the values of the "rank-oriented" (discrete-time Markov chain) and "time-oriented" (continuous-time Markov chain) versions. We have also provided an example of how time can be calibrated using click logs from Yandex.

Future works concern the investigation of alternative user models able to account also for the number of relevant/not relevant documents visited so far – a kind of information which is actually available to a real user – by employing a multidimensional Markov chains to not violate the memory-less assumption. A further interesting option would also be to investigate whether click model-based IR measures [Chuklin et al., 2013] can be represented via the Markov chain and thus embedded in MP, i.e. whether the transition probabilities of the Markov chain can be learned directly from click-logs, thus leveraging models fully induced by user behaviour.

Another area of interest concerns how to calibrate time into MP: work on click model-based measures can shed some light in this respect and the techniques proposed by [Smucker and Clarke, 2012a,b] for calibrating time with respect to document length can link MP not only to click logs but also to document collections.

Finally, the robustness of MP could be further investigated, for example evaluating how it performs on condensed-lists [Sakai, 2007].

The second main contribution is represented by a way to describe the user dynamic with the same model based on Markov chains. We calibrated the Markovian model on different query types, i.e. queries retrieving a different number of relevant documents, and we noticed that the invariant distribution exhibits different shapes depending on the query. Therefore, we defined two macroscopic behaviours: the navigational behaviour is characterized by users focusing mainly on the first ranked document, the informational behaviour is described by users that explore the results and browse through the ranked list, without focusing on any particular rank position. The user dynamic is defined as a mixture of these two behaviours and it is calibrated on real world query log.

Then, we integrated this dynamic in LambdaMart by defining a new quality measure called nMCG. nMCG is an extended version of DCG, where the discount function is defined as the user dynamic. Since nMCG depends on the query type, different discount functions are applied, depending on the query type. Therefore, for navigational queries the discount places more weight on the top rank positions and then steeply decreases, while for informational queries the discount decreases gradually.

nMCG is integrated in LambdaMart by replacing the objective function, this implies that nMCG-MART optimizes two different versions of the same quality measure. Experiments conducted on publicly available datasets showed that nMCG-MART improves over the state-of-the-art with respect to both nDCG and nMCG.

As future work we aim at analyzing the properties of nMCG, as for example the top heaviness expressed with the balancing index or the robustness to pool downsampling. We will also analyze the correlation of nMCG with other standard evaluation measures and with MP to understand the difference between these measures calibrated on real data and the predefined models of MP.

Moreover, we will conduct a user study in order to investigate whether the metric correlates with the quality of a ranking perceived by a user. The measure can also be extended to define a different dynamic based on the single user instead of the query type, this will allow to experiment a different version of nMCG-MART able to customize the ranking based on the user.

Finally, we believe that the user behavior on the first page of results can be extended similarly to the following pages, therefore we plan to validate the study of the user dynamic beyond the first page of results.

Chapter 6

Conclusion and Future Work

In this thesis we presented different approaches to address some challenges related to evaluation of IR systems. We started with a detailed analysis of the evaluation problem and how it can be theoretically and formally framed. We laid our work on the representational theory of measurement, which provides the foundations of the modern theory of measurement in both physical and social sciences. This allowed us to consider the problem from a higher abstract level and to better understand the intrinsic complexity of this task. We tried to define a total ordering among the runs produced as output of IR systems and we realized that there is no common agreement on how to order them. Therefore, we could only define a partial ordering relation and this put a lot of constraints on the possible properties and operations that could be defined. Nevertheless, based on this partial ordering we stated a formal definition for utility-oriented measurements of retrieval effectiveness.

Successively we faced the challenge of coping with noise and inconsistencies in relevance labels. With the use of crowdsourcing platforms to collect relevance assessments it became necessary to find reliable strategies to merge relevance labels assigned from different assessors to the same query-document pair. State of the art approaches work at pool level, so they directly define a relevance label by processing the multiple labels assigned by different assessors. We instead propose AWARE, a different methodology that aims at robustly estimating the true value of an evaluation measure when dealing with errors and noise. AWARE does not work at pool level, but at measure level: it combines the different values of the evaluation measures computed on the pools generated by each assessor. By assigning a different weight to account for assessors accuracy, AWARE approaches improve in terms of their capability of correctly ranking systems and predicting their actual performance scores.

Finally, we shifted our investigation on the user behaviour and how to account for the interactions between the user and the system in an evaluation task. We defined a user model based on Markov chains: each document in the result list is represented as a state and the transitions of the users among the documents define the transition matrix. Based on this model, we proposed a new family of evaluation measures called MP. MP injects the user behaviour into precision by means of the invariant distribution, i.e. each relevant document is weighted accordingly to the probability to see the user in that rank position. Moreover, we embedded this Markovian model in an LtR algorithm. We observed that the invariant distribution differs depending on the query type, therefore we extended nDCG with the discount component defined as the invariant distribution and exploited this new measure, nMCG, as objective function for LAMBDAMART.

Future work may deal with further explorations on our formal framework to describe evaluation measures, what if we consider online evaluation measures instead of batch evaluation measures? Probably our framework will not be valid and it would need to be extended to account for the user behaviour. Consider as an example MP, this family of measures gives more weight to the documents where it is more probable to see the user, therefore different users might lead to different evaluation scores. This is in contrast with system centered evaluation measures and with the partial ordering that we defined on the space of all possible runs. Therefore, to frame online evaluation measures in a broader formal context we will need to include the users, or at least their interactions, in the definition of the ordering among runs.

Moreover, we can study and propose new formal properties of evaluation measures. This might provide a tool for a better comprehension of evaluation measures and might have many beneficial effects and implications. For example, a better understanding of how measures behave can give an insight on how to choose objective functions for LtR and how a different measure can improve or not the ranking.

Furthermore, in the context of AWARE a better knowledge of measures properties can help in understanding to which extent a measure can be robustly estimated. We can also investigate new unsupervised approaches to define assessors accuracy by using multiple measures. Finally, we might develop some supervised approaches based on the comparison with the gold standard instead of the random assessors.

User models have proven to be valuable and advantageous in many aspects of IR. Thus the Markovian model can be adapted and extended to accomplish different tasks. For example, we might think to a possible extension to click models, i.e. the task of predicting clicks on a SERP. Moreover, similarly to what we did for nMCG-MART we can extend the model for online LtR and infer the user preferences by exploiting a proper Markovian model.

Even MP, our new family of evaluation measures, can be extended for the evaluation of new tasks, as for example the evaluation of system sessions instead of single runs. Furthermore, we can conduct a user study and investigate the correlation between MP and user satisfaction. Finally, we can further explore the effects of the time dimension and the calibration of the model with real log data.

References

Abad, A. (2017). *Controlling the Effect of Noisy Annotations in Crowdsourcing NLP Tasks*. PhD thesis, University of Trento. Cited on page 81.

Abad, A., Nabi, M., and Moschitti, A. (2017). Autonomous Crowdsourcing Through Human-Machine Collaborative Learning. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 873–876, New York, NY, USA. ACM. Cited on page 81.

Agichtein, E., Brill, E., and Dumais, S. (2006a). Improving Web Search Ranking by Incorporating User Behavior Information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 19–26, New York, NY, USA. ACM. Cited on pages 4, 121, and 123.

Agichtein, E., Brill, E., Dumais, S., and Ragno, R. (2006b). Learning User Interaction Models for Predicting Web Search Result Preferences. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 3–10, New York, NY, USA. ACM. Cited on pages 11, 27, 122, 123, and 132.

Allegretti, M., Moshfeghi, Y., Hadjigeorgieva, M., Pollick, F. E., Jose, J. M., and Pasi, G. (2015). When Relevance Judgement is Happening?: An EEG-based Study. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 719–722, New York, NY, USA. ACM. Cited on page 12.

Alonso, O. (2013). Implementing Crowdsourcing-based Relevance Experimentation: an Industrial Perspective. *Information Retrieval*, 16(2):101–120. Cited on page 80.

Alonso, O. and Mizzaro, S. (2009). Can we Get Rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, volume 15, page 16. Cited on pages 3 and 26.

Alonso, O. and Mizzaro, S. (2012). Using Crowdsourcing for TREC Relevance Assessment. *Information Processing & Management*, 48(6):1053–1066. Cited on page 77.

Alonso, O., Rose, D. E., and Stewart, B. (2008). Crowdsourcing for Relevance Evaluation. *SIGIR Forum*, 42(2):9–15. Cited on pages 3 and 26.

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, M. F. (2009). A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints. *Information Retrieval*, 12(4):461–486. Cited on page 46.

Amigó, E., Gonzalo, J., and Verdejo, F. (2013). A General Evaluation Measure for Document Organization Tasks. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 643–652, New York, NY, USA. ACM. Cited on pages 41, 46, 47, 48, 52, 60, and 71.

Angelini, M., Ferro, N., Santucci, G., and Silvello, G. (2014). VIRTUE: A Visual Tool for Information Retrieval Performance Evaluation and Failure Analysis. *Journal of Visual Languages & Computing (JVLC)*, 25(4):394–413. Cited on page 48.

Aslam, J. A., Yilmaz, E., and Pavlu, V. (2005). A Geometric Interpretation of R-precision and its Correlation with Average Precision. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 573–574, New York, NY, USA. ACM. Cited on pages 35 and 135.

Bailey, P., Craswell, N., Soboroff, I., Thomas, P., de Vries, A. P., and Yilmaz, E. (2008). Relevance Assessment: Are Judges Exchangeable and Does It Matter. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 667–674, New York, NY, USA. ACM. Cited on page 80.

Bashir, M., Anderton, J., Wu, J., Ekstrand-Abueg, M., Golbus, P. B., Pavlu, V., and Aslam, J. A. (2013). Northeastern University Runs at the TREC12 Crowdsourcing Track. In *The Twenty-First Text REtrieval Conference Proceedings*, TREC '12. National Institute of Standards and Technology (NIST), Special Publication 500-298, Washington, USA. Cited on pages 26, 80, 81, and 83.

Billingsley, P. (1995). *Probability and Measure*. John Wiley & Sons, New York, USA, 3rd edition. Cited on pages 44, 51, and 59.

Bjørner, S. and Ardito, S. C. (2003). Online Before the Internet: Early Pioneers Tell their Stories. *Searcher: The Magazine for Database Professionals*, 11(6). Cited on page 22.

Blanco, R., Halpin, H., Herzig, D. M., Mika, P., Pound, J., Thompson, H. S., and Tran Duc, T. (2011). Repeatable and Reliable Search System Evaluation Using Crowdsourcing. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 923–932, New York, NY, USA. ACM. Cited on page 81.

Bollman, P. (1984). Two Axioms for Evaluation Measures in Information Retrieval. In van Rijsbergen, C. J., editor, *Proceedings of the Third Joint BCS and ACM Symposium on Research and Development in Information Retrieval*, pages 233–245. Cambridge University Press, UK. Cited on page 47.

Broder, A. (2002). A Taxonomy of Web Search. *SIGIR Forum*, 36(2):3–10. Cited on page 150.

Buckley, C. and Voorhees, E. M. (2000). Evaluating Evaluation Measure Stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pages 33–40, New York, NY, USA. ACM. Cited on pages 41 and 135.

Buckley, C. and Voorhees, E. M. (2004). Retrieval Evaluation with Incomplete Information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 25–32, New York, NY, USA. ACM. Cited on pages 39, 41, 135, 143, and 144.

Buckley, C. and Voorhees, E. M. (2005). Retrieval System Evaluation. In *TREC. Experiment and Evaluation in Information Retrieval*, pages 53–78. MIT Press, Cambridge (MA), USA. Cited on pages 34, 79, 101, 135, 139, and 143.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. (2005). Learning to Rank Using Gradient Descent. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 89–96, New York, NY, USA. ACM. Cited on pages 36, 38, 65, and 154.

Burges, C. J. (2010). From RankNet to LambdaRank to LambdaMART: An Overview. Technical report. Cited on page 123.

Burges, C. J., Ragno, R., and Le, Q. V. (2007). Learning to Rank with Nonsmooth Cost Functions. In *Advances in Neural Information Processing Systems*, pages 193–200. MIT Press. Cited on page 153.

Burgin, R. (1992). Variations in Relevance Judgments and the Evaluation of Retrieval Performance. *Information Processing & Management*, 28(5):619–627. Cited on page 80.

Burnham, K. P. and Anderson, D. R. (2002). *Model Selection and Multimodel Inference. A Practical Information-Theoretic Approach*. Springer-Verlag, Heidelberg, Germany, 2nd edition. Cited on page 92.

Busin, L. and Mizzaro, S. (2013). Axiometrics: An Axiomatic Approach to Information Retrieval Effectiveness Metrics. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, ICTIR '13, pages 8:22–8:29, New York, NY, USA. ACM. Cited on pages 41, 47, 48, 52, and 53.

Büttcher, S., Clarke, C. L. A., and Cormack, G. V. (2016). *Information Retrieval: Implementing and Evaluating Search Engines*. Mit Press. Cited on pages 12, 13, 15, and 16.

Capannini, G., Lucchese, C., Nardini, F. M., Orlando, S., Perego, R., and Tonellotto, N. (2016). Quality Versus Efficiency in Document Scoring with Learning-to-rank Models. *Information Processing & Management*, 52(6):1161 – 1177. Cited on page 156.

Carterette, B. (2011). System Effectiveness, User Models, and User Utility: A Conceptual Framework for Investigation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 903–912, New York, NY, USA. ACM. Cited on pages 45, 51, 132, and 135.

Carterette, B., Bennett, P. N., Chickering, D. M., and Dumais, S. T. (2008). Here or There: Preference Judgments for Relevance. In *Advances in Information Retrieval. Proceedings of the 30th European Conference on IR Research*, ECIR'08, pages 16–27, Berlin, Heidelberg. Springer-Verlag. Cited on page 20.

Carterette, B., Kanoulas, E., and Yilmaz, E. (2012). Advances on the Development of Evaluation Measures. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 1200–1201, New York, NY, USA. ACM. Cited on page 18.

Carterette, B. and Soboroff, I. (2010). The Effect of Assessor Error on IR System Evaluation. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 539–546, New York, NY, USA. ACM. Cited on pages 81 and 89.

Carterette, Ben and Pavlu, Virgiliu and Fang, Hui and Kanoulas, Evangelos (2009). Million Query Track 2009 Overview. In *The Eighteenth Text Retrieval Conference*, TREC '09, Gaithersburg, MD, USA: Department of Commerce, National Institute of Standards and Technology. NIST Special Publication. Cited on page 27.

Chapelle, O., Joachims, T., Radlinski, F., and Yue, Y. (2012). Large-scale Validation and Analysis of Interleaved Search Evaluation. *ACM Transaction on Information Systems (TOIS)*, 30(1):6:1–6:41. Cited on page 157.

Chapelle, O., Metlzer, D., Zhang, Y., and Grinspan, P. (2009). Expected Reciprocal Rank for Graded Relevance. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 621–630, New York, NY, USA. ACM. Cited on pages 38, 62, 79, 101, and 130.

Chierichetti, F., Kumar, R., and Raghavan, P. (2011). Optimizing Two-dimensional Search Results Presentation. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 257–266, New York, NY, USA. ACM. Cited on pages 130 and 142.

Chuklin, A., Serdyukov, P., and de Rijke, M. (2013). Click Model-based Information Retrieval Metrics. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 493–502, New York, NY, USA. ACM. Cited on page 158.

Clarke, C. L., Craswell, N., Soboroff, I., and Ashkan, A. (2011). A Comparative Analysis of Cascade Measures for Novelty and Diversity. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 75–84, New York, NY, USA. ACM. Cited on page 130.

Cleverdon, C. W. (1960). Report on the First Stage of an Investigation into the Comparative Efficiency of Indexing Systems. Technical report, Aslib Cranfield Research Project, Cranfield, England. Cited on page 18.

Cleverdon, C. W. (1962). Report on the Testing and Analysis of an Investigation into the Comparative Efficiency of Indexing Systems. Technical report, Aslib Cranfield Research Project, Cranfield, England. Cited on pages 18 and 31.

Cleverdon, C. W. and Mills, J. and Keen, M. (1966). Factors Determining The Performance of Indexing Systems. Volume 2, Test results. Technical report, Aslib Cranfield Research Project, Cranfield, England. Cited on pages 19, 28, and 30.

Clough, P., Sanderson, M., Tang, J., Gollins, T., and Warner, A. (2013). Examining the Limits of Crowdsourcing for Relevance Assessment. *IEEE Internet Computing*, 17(4):32–38. Cited on page 80.

Collins-Thompson, K. and Callan, J. (2005). Query Expansion Using Random Walk Models. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 704–711, New York, NY, USA. ACM. Cited on page 129.

Cooper, W. S. (1968). Expected Search Length: A Single Measure of Retrieval Effectiveness Based on the Weak Ordering Action of Retrieval Systems. *American Documentation*, 19(1):30–41. Cited on page 35.

Cooper, W. S. (1973). On Selecting a Measure of Retrieval Effectiveness. *Journal of the American Society for Information Science (JASIS)*, 24(2):87–100. Cited on pages 45 and 51.

Cormack, G. and Lynam, T. (2005). TREC 2005 Spam Track Overview. In *The Fourteenth Text REtrieval Conference Proceedings*, TREC '05. National Institute of Standards and Technology (NIST), Special Publication 500-266, Washington, USA. Cited on page 100.

Daniłowicz, C. and Baliński, J. (2001). Document Ranking Based upon Markov Chains. *Information Processing & Management*, 37(4):623–637. Cited on page 129.

Dato, D., Lucchese, C., Nardini, F. M., Orlando, S., Perego, R., Tonellotto, N., and Venturini, R. (2016). Fast Ranking with Additive Ensembles of Oblivious and Non-Oblivious Regression Trees. *ACM Transaction on Information Systems (TOIS)*, 35(2):15:1–15:31. Cited on page 156.

Dawid, A. P. and Skene, A. M. (1979). Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28. Cited on page 83.

Dennis, B. K., Brady, J. J., and Dovel, J. A. (1962). Index Manipulation and Abstract Retrieval by Computer. *Journal of Chemical Documentation*, 2(4):234–242. Cited on page 22.

Donmez, P., Svore, K. M., and Burges, C. J. (2009). On the Local Optimality of LambdaRank. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 460–467, New York, NY, USA. ACM. Cited on page 153.

Egghe, L. (2008). The Measures Precision, Recall, Fallout and Miss as a Function of the Number of Retrieved Documents and their Mutual Interrelations. *Information Processing & Management*, 44(2):856–876. Cited on page 51.

Eickhoff, C., Harris, C. G., de Vries, A. P., and Srinivasan, P. (2012). Quality Through Flow and Immersion: Gamifying Crowdsourced Relevance Assessments. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 871–880, New York, NY, USA. ACM. Cited on page 81.

Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8):861–874. Cited on page 100.

Fenton, N. and Bieman, J. M. (2014). *Software Metrics: A Rigorous & Practical Approach*. Chapman and Hall/CRC, USA. Cited on pages 42, 44, and 47.

Ferrante, M., Ferro, N., and Maistro, M. (2014a). Injecting User Models and Time into Precision via Markov Chains. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 597–606, New York, NY, USA. ACM. Cited on pages 4 and 7.

Ferrante, M., Ferro, N., and Maistro, M. (2014b). Rethinking How to Extend Average Precision to Graded Relevance. In *Information Access Evaluation – Multilinguality, Multimodality, and Interaction. Proceedings of the Fifth International Conference of the CLEF Initiative*, CLEF '14, pages 19–30. Lecture Notes in Computer Science (LNCS) 8685, Springer, Heidelberg, Germany. Cited on page 120.

Ferrante, M., Ferro, N., and Maistro, M. (2015). Towards a Formal Framework for Utility-oriented Measurements of Retrieval Effectiveness. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, ICTIR '15, pages 21–30, New York, NY, USA. ACM. Cited on pages 2, 6, and 42.

Ferrante, M., Ferro, N., and Maistro, M. (2017). AWARE: Exploiting Evaluation Measures to Combine Multiple Assessors. *ACM Transaction on Information Systems*, 36(2):20:1–20:38. Cited on pages 3, 6, and 79.

Ferro, N. (2017). Reproducibility Challenges in Information Retrieval Evaluation. *ACM Journal of Data and Information Quality (JDIQ)*, 8(2):8:1–8:4. Cited on page 81.

Ferro, N., Fuhr, N., Järvelin, K., Kando, N., Lippold, M., and Zobel, J. (2016a). Increasing Reproducibility in IR: Findings from the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science". *SIGIR Forum*, 50(1):68–82. Cited on page 81.

Ferro, N., Lucchese, C., Maistro, M., and Perego, R. (2017). On Including the User Dynamic in Learning to Rank. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1041–1044, New York, NY, USA. ACM. Cited on pages 4, 7, and 155.

Ferro, N., Silvello, G., Keskustalo, H., Pirkola, A., and Järvelin, K. (2016b). The Twist Measure for IR Evaluation: Taking User's Effort into Account. *Journal of the Association for Information Science and Technology*, 67(3):620–648. Cited on pages 18, 48, and 120.

Finkelstein, L. (2003). Widely, Strongly and Weakly Defined Measurement. *Measurement*, 34(1):39–48. Cited on page 43.

Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons, New York, USA, 2nd edition. Cited on pages 44, 51, and 59.

Friedman, Jerome H (2001). Greedy Function Approximation: a Gradient Boosting Machine. *Annals of statistics*, pages 1189–1232. Cited on page 154.

Fuhr, N. (1989). Optimum Polynomial Retrieval Functions Based on the Probability Ranking Principle. *ACM Transaction on Information Systems (TOIS)*, 7(3):183–204. Cited on page 10.

Fuhr, N. (2010). *IR between Science and Engineering, and the Role of Experimentation*, pages 1–1. Springer Berlin Heidelberg, Berlin, Heidelberg. Cited on pages 41 and 46.

Goker, A. and Davies, J. (2009). *Information Retrieval: Searching in the 21st Century*. John Wiley & Sons. Cited on page 13.

Golub, G. H. and Van Loan, C. F. (2012). *Matrix Computations*. Johns Hopkins University Press, USA, 4th edition. Cited on page 90.

Grady, C. and Lease, M. (2010). Crowdsourcing Document Relevance Assessment with Mechanical Turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 172–179. The Association for Computational Linguistics (ACL), USA. Cited on page 80.

Halvey, M., Villa, R., and Clough, P. (2014). SIGIR 2014 Workshop on Gathering Efficient Assessments of Relevance (GEAR). In *Proceedings of the 37th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 1293–1293, New York, NY, USA. ACM. Cited on page 77.

Harman, D. (1992a). The DARPA TIPSTER Project. *SIGIR Forum*, 26(2):26–28. Cited on page 23.

Harman, D. K. (1992b). Overview of the First TREC Conference. In *Proceedings of the First Text REtrieval Conference*, TREC 1, pages 1–30, Gaithersburg, MD, USA: Department of Commerce, National Institute of Standards and Technology. National Institute of Standards and Technology (NIST), Special Publication 500-225, Washington, USA. Cited on page 10.

Harman, D. K. (1993). Overview of the Second TREC Conference (TREC-2). In *Proceedings of the Second Text REtrieval Conference*, TREC 2, Gaithersburg, MD, USA: Department of Commerce, National Institute of Standards and Technology. National Institute of Standards and Technology (NIST), Special Publication 500-225, Washington, USA. Cited on pages 33, 34, and 139.

Harman, D. K. (1994). Overview of the Third Text REtrieval Conference (TREC-3) . In *Proceedings of the Third Text REtrieval Conference*, TREC 3, pages 1–19. National Institute of Standards and Technology (NIST), Special Pubblication 500-225, Washington, USA. Cited on page 135.

Harman, D. K. (1995). Overview of the Second Text Retrieval Conference (TREC-2). *Information Processing & Management*, 31(3):271–289. Cited on page 25.

Harman, D. K. (2011). *Information Retrieval Evaluation*. Morgan & Claypool Publishers, 1st edition. Cited on pages 1, 3, 19, 22, and 121.

Harris, C. and Srinivasan, P. (2013). Using Hybrid Methods for Relevance Assessment in TREC Crowd'12. In *The Twenty-First Text REtrieval Conference Proceedings*, TREC '12. National Institute of Standards and Technology (NIST), Special Publication 500-298, Washington, USA. Cited on page 80.

He, Y. and Wang, K. (2011). Inferring Search Behaviors Using Partially Observable Markov Model with Duration (POMD). In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 415–424, New York, NY, USA. ACM. Cited on pages 130 and 134.

Hiemstra, D. and Kraaij, W. (1998). Twenty-One at TREC-7: Ad-hoc and Cross-language Track. pages 227–238. Cited on page 16.

Hochberg, Y. and Tamhane, A. C. (1987). *Multiple Comparison Procedures*. John Wiley & Sons, USA. Cited on page 104.

Hofmann, K., Schuth, A., Whiteson, S., and de Rijke, M. (2013a). Reusing Historical Interaction Data for Faster Online Learning to Rank for IR. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 183–192, New York, NY, USA. ACM. Cited on page 133.

Hofmann, K., Whiteson, S., and de Rijke, M. (2013b). Balancing Exploration and Exploitation in Listwise and Pairwise Online Learning to Rank for Information Retrieval. *Information Retrieval*, 16(1):63–90. Cited on page 28.

Hofmann, K., Whiteson, S., Schuth, A., and de Rijke, M. (2014). Learning to Rank for Information Retrieval from User Interactions. *SIGWEB Newsletter*, (Spring):1–7. Cited on page 133.

Hosseini, M., Cox, I. J., Milić-Frayling, N., Kazai, G., and Vinay, V. (2012). On Aggregating Labels from Multiple Crowd Workers to Infer Relevance of Documents. In *Advances in Information Retrieval. Proceedings of the 34th European Conference on IR Research*, ECIR'12, pages 182–194, Berlin, Heidelberg. Springer-Verlag. Cited on pages 27, 80, 81, 82, and 83.

Ipeirotis, P. G. and Gabrilovich, E. (2014). Quizz: Targeted Crowdsourcing with a Billion (Potential) Users. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 143–154, New York, NY, USA. ACM. Cited on page 80.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated Gain-based Evaluation of IR Techniques. *ACM Transaction on Information Systems (TOIS)*, 20(4):422–446. Cited on pages 35, 36, 62, 65, 79, and 101.

Joachims, T. (2002). Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA. ACM. Cited on pages 27, 28, and 133.

Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. (2005). Accurately Interpreting Clickthrough Data As Implicit Feedback. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 154–161, New York, NY, USA. ACM. Cited on pages 4, 27, 29, 122, and 132.

Joachims, T., Swaminathan, A., and Schnabel, T. (2017). Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 781–789, New York, NY, USA. ACM. Cited on page 27.

Josephy, T., Lease, M., Paritosh, P., Krause, M., Georgescu, M., Tjalve, M., and Braga, D. (2014). Workshops Held at the First AAAI Conference on Human Computation and Crowdsourcing: A Report. *AI Magazine*, 35(2):75–78. Cited on page 81.

Jung, H. J. and Lease, M. (2015). A Discriminative Approach to Predicting Assessor Accuracy. In *Advances in Information Retrieval. Proceedings of the 37th European Conference on IR Research*, ECIR '15. Lecture Notes in Computer Science (LNCS) 9022, Springer, Heidelberg, Germany. Cited on page 81.

Kazai, G. (2011). In Search of Quality in Crowdsourcing for Search Engine Evaluation. In *Advances in Information Retrieval. Proceedings of the 33rd European Conference on IR Research*, ECIR '11, pages 165–176. Lecture Notes in Computer Science (LNCS) 6611, Springer, Heidelberg, Germany. Cited on page 81.

Kazai, G., Craswell, N., Yilmaz, E., and Tahaghoghi, S. (2012a). An Analysis of Systematic Judging Errors in Information Retrieval. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 105–114, New York, NY, USA. ACM. Cited on page 81.

Kazai, G., Kamps, J., Koolen, M., and Milic-Frayling, N. (2011). Crowdsourcing for Book Search Evaluation: Impact of Hit Design on Comparative System Ranking. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 205–214, New York, NY, USA. ACM. Cited on page 80.

Kazai, G., Kamps, J., and Milic-Frayling, N. (2012b). The Face of Quality in Crowdsourcing Relevance Labels: Demographics, Personality and Labeling Accuracy. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2583–2586, New York, NY, USA. ACM. Cited on page 80.

Kazai, G., Kamps, J., and Milic-Frayling, N. (2013a). An Analysis of Human Factors and Label Accuracy in Crowdsourcing Relevance Judgments. *Information Retrieval*, 16(2):138–178. Cited on page 81.

Kazai, G., Yilmaz, E., Craswell, N., and Tahaghoghi, S. (2013b). User Intent and Assessor Disagreement in Web Search Evaluation. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 699–708, New York, NY, USA. ACM. Cited on page 81.

Kekäläinen, J. and Järvelin, K. (2002). Using Graded Relevance Assessments in IR Evaluation. *Journal of the American Society for Information Science and Technology (JASIST)*, 53(13):1120—1129. Cited on pages 20 and 48.

Kendall, M. G. (1945). The Treatment of Ties in Ranking Problems. *Biometrika*, 33(3):239–251. Cited on page 144.

Kendall, M. G. (1948). *Rank Correlation Methods*. Griffin, Oxford, England. Cited on pages 41 and 92.

Kenney, J. F. and Keeping, E. S. (1954). *Mathematics of Statistics – Part One*. D. Van Nostrand Company, Princeton, USA, 3rd edition. Cited on page 91.

Kent, A., Berry, M. M., Luehrs, F. U., and Perry, J. W. (1955). Machine Literature Searching VIII. Operational Criteria for Designing Information Retrieval Systems. *American Documentation*, 6(2):93–101. Cited on page 31.

Kiewitt, E. L. (1979). *Evaluating Information Retrieval Systems: The Probe Program*. Greenwood Publishing Group Inc., Westport, CT, USA. Cited on page 2.

King, I., Chen, K.-T., Alonso, O., and Larson, M. (2016). Special Issue: Crowd in Intelligent Systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(4). Cited on page 77.

Kinney, K. A., Huffman, S. B., and Zhai, J. (2008). How Evaluator Domain Expertise Affects Search Result Relevance Judgments. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 591–598, New York, NY, USA. ACM. Cited on page 80.

Knuth, D. E. (1981). *The Art of Computer Programming – Volume 2: Seminumerical Algorithms*. Addison-Wesley, USA, 2nd edition. Cited on page 49.

Koopman, B. and Zuccon, G. (2014). Relevation!: An Open Source System for Information Retrieval Relevance Assessment. In *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '14, pages 1243–1244, New York, NY, USA. ACM. Cited on page 12.

Krantz, D. H., Luce, R. D., Suppes, P., and Tversky, A. (1971). *Foundations of Measurement. Additive and Polynomial Representations*, volume 1. Academic Press, New York, USA. Cited on pages 2, 29, 42, 43, and 44.

Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86. Cited on page 91.

Lafferty, J. and Zhai, C. (2001). Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 111–119. ACM. Cited on page 129.

Law, E., Bennett, P. N., and Horvitz, E. (2011). The Effects of Choice in Routing Relevance Judgments. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1127–1128, New York, NY, USA. ACM. Cited on page 81.

Lawrence, S. and Giles, C. L. (1998). Searching the World Wide Web. *Science*, 280(5360):98–100. Cited on page 22.

Lease, M. and Yilmaz, E. (2013). Crowdsourcing for Information Retrieval: Introduction to the Special Issue. *Information Retrieval*, 16(2):91–100. Cited on page 77.

Lesk, M., Harman, D. K., Fox, E. A., Wu, H., and Buckley, C. (1997). The SMART Lab Report. *SIGIR Forum*, 31(1):2–22. Cited on page 22.

Lesk, M. E. and Salton, G. (1968). Relevance Assessments and Retrieval System Evaluation. *Information Storage and Retrieval*, 4(4):343–359. Cited on pages 21 and 80.

Li, L. and Smucker, M. D. (2014). Tolerance of Effectiveness Measures to Relevance Judging Errors. In *Advances in Information Retrieval. Proc. 36th European Conference on IR Research*, ECIR '14, pages 148–159. Lecture Notes in Computer Science (LNCS) 8416, Springer, Heidelberg, Germany. Cited on pages 81 and 89.

Lipani, A., Lupu, M., and Hanbury, A. (2015). Splitting Water: Precision and Anti-Precision to Reduce Pool Bias. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 103–112, New York, NY, USA. ACM. Cited on page 25.

Liu, T.-Y. (2011). *Learning to Rank for Information Retrieval*. Springer-Verlag Berlin Heidelberg. Cited on pages 5, 10, 12, 16, 123, and 153.

Loni, B., Larson, M., Bozzon, A., and Gottlieb, L. (2013). Crowdsourcing for Social Multimedia at MediaEval 2013: Challenges, Data set, and Evaluation. In *Working Notes Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop*. CEUR Workshop Proceedings (CEUR-WS.org). Cited on page 81.

Lucchese, C., Orlando, S., Perego, R., Silvestri, F., and Tolomei, G. (2013). Discovering Tasks from Search Engine Query Logs. *ACM Transaction on Information Systems (TOIS)*, 31(3):14. Cited on page 121.

Luhn, Hans Peter (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, 1(4):309–317. Cited on page 14.

Maddalena, E. and Mizzaro, S. (2014). Axiometrics: Axioms of Information Retrieval Effectiveness Metrics. In *Proceedings of the 6th International Workshop on Evaluating Information Access*, EVIA '14, pages 17–24. National Institute of Informatics, Tokyo, Japan. Cited on pages 47, 48, 52, 53, and 59.

Maddalena, E., Mizzaro, S., Scholer, F., and Turpin, A. (2015). Judging Relevance Using Magnitude Estimation. In *Advances in Information Retrieval. Proceedings of the 37th European Conference on IR Research*, ECIR '15, pages 215–220. Lecture Notes in Computer Science (LNCS) 9022, Springer, Heidelberg, Germany. Cited on page 48.

Maddalena, E., Mizzaro, S., Scholer, F., and Turpin, A. (2017). On Crowdsourcing Relevance Magnitudes for Information Retrieval Evaluation. *ACM Transaction on Information Systems (TOIS)*, 35(3):19:1–19:32. Cited on pages 20 and 48.

Manmatha, R., Rath, T., and Feng, F. (2001). Modeling Score Distributions for Combining the Outputs of Search Engines. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 267–275, New York, NY, USA. ACM. Cited on page 32.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA. Cited on pages 3, 33, and 77.

Marcus, A. and Parameswaran, A. (2015). Crowdsourced Data Management: Industry and Academic Perspectives. *Foundations and Trends in Databases (FnTDB)*, 6(1–2):1–161. Cited on page 77.

Mari, L. (2000). Beyond the Representational Viewpoint: a New Formalization of Measurement. *Measurement*, 27(2):71–84. Cited on page 43.

Maron, M. E. and Kuhns, J. L. (1960). On Relevance, Probabilistic Indexing and Information Retrieval. *Journal of the ACM*, 7(3):216–244. Cited on page 12.

Maxwell, K. T. and Croft, W. B. (2013). Compact Query Term Selection Using Topically Related Text. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 583–592, New York, NY, USA. ACM. Cited on page 129.

Maxwell, S. and Delaney, H. D. (2004). *Designing Experiments and Analyzing Data. A Model Comparison Perspective*. Lawrence Erlbaum Associates, Mahwah (NJ), USA, 2nd edition. Cited on page 103.

Miller, D. R. H., Leek, T., and Schwartz, R. M. (1999). A Hidden Markov Model Information Retrieval System. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 214–221, New York, NY, USA. ACM. Cited on pages 16 and 129.

Miyamoto, S. (2004). Generalizations of Multisets and Rough Approximations. *International Journal of Intelligent Systems*, 19(7):639–652. Cited on page 50.

Mizzaro, S. (1997). Relevance: The Whole History. *Journal of the Association for Information Science and Technology*, 48(9):810–832. Cited on page 12.

Moffat, A. (2013). Seven Numeric Properties of Effectiveness Metrics. In *Proceedings of the 9th Asia Information Retrieval Societies Conference*, volume 8281 of *AIRS '13*, pages 1–12. Lecture Notes in Computer Science (LNCS), Springer, Heidelberg, Germany. Cited on pages 41, 46, 47, 48, 52, 59, 60, and 71.

Moffat, A., Thomas, P., and Scholer, F. (2013). Users Versus Models: What Observation Tells Us About Effectiveness Metrics. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 659–668, New York, NY, USA. ACM. Cited on page 135.

Moffat, A. and Zobel, J. (2008). Rank-biased Precision for Measurement of Retrieval Effectiveness. *ACM Transactions on Information Systems (TOIS)*, 27(1):2:1–2:27. Cited on pages 37, 50, 62, 130, 135, 139, and 143.

Mooers, C. N. (1950). Information Retrieval Viewed as Temporal Signaling. In *Proceedings of the International Congress of Mathematicians*, pages 572–573, Providence, R.I. American Mathematical Society. Cited on page 9.

Mooers, C. N. (1951). Scientific Information Retrieval Systems for Machine Operation; Case Studies in Design. *Zator Technical Bulletin*, 66:18 L. Cited on page 9.

Moshfeghi, Y., Huertas Rosero, H. F., and Jose, J. M. (2016). A Game-Theory Approach for Effective Crowdsource-Based Relevance Assessment. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(4):55:1–55:XXX. Cited on pages 81 and 89.

Murphy, K. R., Myors, B., and Wolach, A. (2014). *Statistical Power Analysis: A Simple and General Model for Traditional and Modern Hypothesis Tests*. Routledge, Taylor & Francis Group, UK, 4th edition. Cited on page 104.

Norris, J. R. (1998). *Markov chains*. Cambridge University Press, UK. Cited on pages 122, 133, 137, 141, and 150.

Olejnik, S. and Algina, J. (2003). Generalized Eta and Omega Squared Statistics: Measures of Effect Size for Some Common Research Designs. *Psychological Methods*, 8(4):434–447. Cited on page 104.

Pillai, I., Fumera, I., and Roli, F. (2013). Multi-label Classification with a Reject Option. *Pattern Recognition*, 46(8):2256–2266. Cited on page 80.

Ponte, J. M. and Croft, W. B. (1998). A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA. ACM. Cited on pages 12, 13, and 16.

Qin, T. and Liu, T. (2013). Introducing LETOR 4.0 Datasets. *Computing Research Repository (CoRR)*, abs/1306.2597. Cited on page 156.

Qiu, F. and Cho, J. (2006). Automatic Identification of User Interest for Personalized Search. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 727–736, New York, NY, USA. ACM. Cited on page 132.

Radlinski, F., Kurup, M., and Joachims, T. (2008). How Does Clickthrough Data Reflect Retrieval Quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 43–52, New York, NY, USA. ACM. Cited on page 121.

Raykar, V. C. and Yu, S. (2012). Eliminating Spammers and Ranking Annotators for Crowdsourced Labeling Tasks. *Journal of Machine Learning Research*, 13:491–518. Cited on pages 80 and 83.

Raykar, V. C., Yu, S., Zhao, L. H., Jerebko, A., Florin, C., Hermosillo Valadez, G., Bogoni, L., and Moy, L. (2009). Supervised Learning from Multiple Experts: Whom to Trust when Everyone Lies a Bit. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 889–896. ACM Press, New York, USA. Cited on page 108.

Raykar, V. C., Zhao, L. H., Hermosillo Valadez, G., Florin, C., Bogoni, L., and Moy, L. (2010). Learning From Crowds. *Journal of Machine Learning Research*, 11:1297–1322. Cited on pages 80, 83, and 108.

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States. AUAI Press. Cited on page 132.

Robertson, S. (2008a). A New Interpretation of Average Precision. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 689–690, New York, NY, USA. ACM. Cited on pages 122, 132, 135, and 136.

Robertson, S. (2008b). On the History of Evaluation in IR. *Journal of Information Science*, 34(4):439–456. Cited on page 29.

Robertson, S. E. (1977). The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4):294–304. Cited on pages 12 and 15.

Robertson, S. E. and Hancock-Beaulieu, M. M. (1992). On the Evaluation of IR Systems. *Information Processing & Management*, 28(4):457–466. Cited on page 4.

Robertson, S. E., Kanoulas, E., and Yilmaz, E. (2010). Extending Average Precision to Graded Relevance Judgments. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 603–610, New York, NY, USA. ACM. Cited on pages 35, 101, and 120.

Robertson, S. E. and Spärck Jones, K. (1976). Relevance Weighting of Search Terms. *Journal of the Association for Information Science and Technology*, 27(3):129–146. Cited on page 15.

Rocchio, J. (1971). Relevance Feedback in Information Retrieval. *The Smart Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Cited on page 14.

Rorvig, M. E. (1990). The Simple Scalability of Documents. *Journal of the American Society for Information Science*, 41(8):590–598. Cited on page 20.

Rutherford, A. (2011). *ANOVA and ANCOVA. A GLM Approach.* John Wiley & Sons, New York, USA, 2nd edition. Cited on page 103.

Ruthven, I. (2014). Relevance Behaviour in TREC. *Journal of Documentation*, 70(6):1098–1117. Cited on page 80.

Sakai, T. (2005). Ranking the NTCIR Systems Based on Multigrade Relevance. In *Information Retrieval Technology – Asia Information Retrieval Symposium (AIRS 2004)*, pages 251–262. Lecture Notes in Computer Science (LNCS) 3411, Springer, Heidelberg, Germany. Cited on page 130.

Sakai, T. (2006). Evaluating Evaluation Metrics Based on the Bootstrap. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 525–532, New York, NY, USA. ACM. Cited on pages 41 and 72.

Sakai, T. (2007). Alternatives to Bpref. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 71–78, New York, NY, USA. ACM. Cited on pages 143 and 158.

Sakai, T. (2014a). *Metrics, Statistics, Tests*, pages 116–163. Springer Berlin Heidelberg, Berlin, Heidelberg. Cited on pages 45 and 51.

Sakai, T. (2014b). Statistical Reform in Information Retrieval? *SIGIR Forum*, 48(1):3–12. Cited on page 104.

Sakai, T. and Dou, Z. (2013). Summaries, Ranked Retrieval and Sessions: A Unified Framework for Information Access Evaluation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 473–482, New York, NY, USA. ACM. Cited on page 150.

Salton, G. (1968). *Automatic Information Organization and Retrieval*. McGraw Hill Text. Cited on page 32.

Salton, G. (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. Cited on page 10.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA. Cited on pages 12 and 14.

Salton, G. and Yang, C.-S. (1973). On the Specification of Term Values in Automatic Indexing. *Journal of documentation*, 29(4):351–372. Cited on page 14.

Salton, G. and Yu, C. T. (1973). On the Construction of Effective Vocabularies for Information Retrieval. In *Proceedings of the 1973 Meeting on Programming Languages and Information Retrieval*, SIGPLAN '73, pages 48–60, New York, NY, USA. ACM. Cited on page 22.

Sanderson, M. (2010). Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends® in Information Retrieval*, 4(4):247–375. Cited on pages 2, 17, 31, and 35.

Sanderson, M. and Zobel, J. (2005). Information Retrieval System Evaluation: Effort, Sensitivity, and Reliability. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 162–169, New York, NY, USA. ACM. Cited on page 80.

Saracevic, T. (1975). Relevance: A Review of and a Framework for the Thinking on the Notion in Information Science. *Journal of the Association for Information Science and Technology*, 26(6):321–343. Cited on page 12.

Schuth, A., Hofmann, K., Whiteson, S., and de Rijke, M. (2013). Lerot: An Online Learning to Rank Framework. In *Proceedings of the 2013 Workshop on Living Labs for Information Retrieval Evaluation*, LivingLab '13, pages 23–26, New York, NY, USA. ACM. Cited on page 133.

Serdyukov, P., Craswell, N., and Dupret, G. (2012). WSCD 2012: Workshop on Web Search Click Data 2012. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 771–772, New York, NY, USA. ACM. Cited on pages 123, 144, and 155.

Silvestri, F. (2009). Mining Query Logs: Turning Search Usage Data into Knowledge. *Foundations and Trends® in Information Retrieval*, 4(1–2):1–174. Cited on page 121.

Smucker, M. D. and Clarke, C. L. (2012a). Time-based Calibration of Effectiveness Measures. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 95–104, New York, NY, USA. ACM. Cited on pages 18, 123, 131, 135, and 158.

Smucker, M. D. and Clarke, C. L. A. (2012b). Stochastic Simulation of Time-biased Gain. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2040–2044, New York, NY, USA. ACM. Cited on pages 131 and 158.

Smucker, M. D. and Jethani, C. P. (2011a). Measuring Assessor Accuracy: A Comparison of Nist Assessors and User Study Participants. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 1231–1232, New York, NY, USA. ACM. Cited on page 80.

Smucker, M. D. and Jethani, C. P. (2011b). The Crowd vs. the Lab: A Comparison of Crowd-Sourced and University Laboratory Participant Behavior. In *Proceedings of the SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval*. Cited on page 80.

Smucker, M. D., Kazai, G., and Lease, M. (2013). Overview of the TREC 2012 Crowdsourcing Track. In *The Twenty-First Text REtrieval Conference Proceedings*, TREC '12. National Institute of Standards and Technology (NIST), Special Publication 500-298, Washington, USA. Cited on pages 79, 81, and 99.

Smucker, M. D., Kazai, G., and Lease, M. (2014). Overview of the TREC 2013 Crowdsourcing Track. In *The Twenty-Second Text REtrieval Conference Proceedings*, TREC '13. National Institute of Standards and Technology (NIST), Special Publication 500-302, Washington, USA. Cited on pages 81, 93, and 100.

Soboroff, I. (2006). Dynamic Test Collections: Measuring Search Effectiveness on the Live Web. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 276–283, New York, NY, USA. ACM. Cited on page 39.

Soboroff, I., Nicholas, C., and Cahan, P. (2001). 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. ACM. Cited on page 89.

Soboroff, I. and Robertson, S. (2003). Building a Filtering Test Collection for TREC 2002. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 243–250, New York, NY, USA. ACM. Cited on page 25.

Spärck Jones, K. (1973). Collection Properties Influencing Automatic Term Classification Performance. *Information Storage and Retrieval*, 9(9):499 – 513. Cited on page 22.

Spärck Jones, K. (1997). *Readings in Information Retrieval*. Morgan Kaufmann. Cited on pages 1, 2, 9, 11, 17, and 41.

Spärck Jones, K. and Van Rijsbergen, C. J. (1975). *Report on the Need for and Provision of an Ideal Information Retrieval Test Collection*. British Library Research and Development reports. Computer Laboratory, University of Cambridge. Cited on pages 10 and 22.

Spärck Jones, K., Walker, S., and Robertson, S. E. (2000). A Probabilistic Model of Information Retrieval: Development and Comparative Experiments. *Information Processing and Management*, 36(6):779–808. Cited on page 16.

Speicher, M., Both, A., and Gaedke, M. (2013). TellMyRelevance!: Predicting the Relevance of Web Search Results from Cursor Interactions. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 1281–1290, New York, NY, USA. ACM. Cited on page 132.

Speretta, M. and Gauch, S. (2005). Personalized Search Based on User Search Histories. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '05, pages 622–628. Cited on page 132.

Stevens, S. S. (1946). On the Theory of Scales of Measurement. *Science, New Series*, 103(2684):677–680. Cited on pages 47 and 48.

Stoker, B. (1897). *Dracula*, volume 135. New York: Oxford University Press, 1990. Cited on page 1.

Sutton, R. S. and Barto, A. G. (1998). *Introduction to Reinforcement Learning*, volume 135. MIT Press Cambridge. Cited on page 132.

Swets, J. A. (1963). Information Retrieval Systems. *Science*, 141(3577):245–250. Cited on page 31.

Taylor, R. S. (1962). The Process of Asking Questions. *American Documentation*, 13(4):391–396. Cited on page 11.

Teodorescu, I. (2009). Maximum Likelihood Estimation for Markov Chains. *arXiv preprint arXiv:0905.4131*. Cited on page 151.

Turpin, A., Scholer, F., Mizzaro, S., and Maddalena, E. (2015). The Benefits of Magnitude Estimation Relevance Assessments for Information Retrieval Evaluation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 565–574, New York, NY, USA. ACM. Cited on page 20.

van den Bosch, A., Bogers, T., and de Kunder, M. (2016). Estimating Search Engine Index Size Variability: a 9-Year Longitudinal Study. *Scientometrics*, 107:839–856. Cited on pages 22 and 27.

van Rijsbergen, C. J. (1974). Foundation of Evaluation. *Journal of Documentation*, 30(4):365–373. Cited on page 31.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition. Cited on pages 2, 12, 17, 28, and 32.

van Rijsbergen, C. J. (1981). Retrieval effectiveness. In Spärck Jones, K., editor, *Information Retrieval Experiment*, pages 32–43. Butterworths, London, United Kingdom. Cited on pages 46 and 51.

Vaswani, P. K. T. and Cameron, J. B. (1970). *The National Physical Laboratory Experiments in Statistical Word Associations and Their Use in Document Indexing And Retrieval*. National Physical Laboratory Computer Science Division-Publications, National Physical Laboratory, Teddington, UK. Cited on page 22.

Voorhees, E. M. (1998). 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*, SIGIR '98, pages 315–323. ACM Press, New York, USA. Cited on page 80.

Voorhees, E. M. (2000). Variations in Relevance Judgments and the Measurement of Retrieval Effectiveness. *Information processing & management*, 36(5):697–716. Cited on pages 80 and 144.

Voorhees, E. M. (2001). Evaluation by Highly Relevant Documents. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 74–82, New York, NY, USA. ACM. Cited on page 144.

Voorhees, E. M. (2008). On Test Collections for Adaptive Information Retrieval. *Information Processing and Management*, 44(6):1879–1885. Cited on page 29.

Voorhees, E. M. (2015). Overview of the TREC 2004 Robust Track. In *The Twenty-Third Text REtrieval Conference Proceedings*, TREC '14. National Institute of Standards and Technology (NIST), Special Publication 500-308, Washington, USA. Cited on pages 79 and 101.

Voorhees, E. M. and Harman, D. K. (1999). Overview of the eighth text retrieval conference (TREC-8). In *The Eighth Text Retrieval Conference*, TREC-8, pages 1–24, Gaithersburg, MD, USA: Department of Commerce, National Institute of Standards and Technology. NIST Special Publication. Cited on pages 25, 79, 99, and 101.

Vuurens, J. B. P. and de Vries, A. P. (2012). Obtaining High-Quality Relevance Judgments Using Crowdsourcing. *IEEE Internet Computing*, 16(5):20–27. Cited on page 81.

Wakeling, S., Halvey, M., Villa, R., and Hasler, L. (2016). A Comparison of Primary and Secondary Relevance Judgements for Real-Life Topics. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, CHIIR '16, pages 173–182, New York, NY, USA. ACM. Cited on page 80.

Wand, M. P. and Jones, M. C. (1995). *Kernel Smoothing*. Chapman and Hall/CRC, USA. Cited on page 91.

Wang, C., Liu, Y., Wang, M., Zhou, K., Nie, J.-y., and Ma, S. (2015). Incorporating Non-sequential Behavior into Click Models. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 283–292, New York, NY, USA. ACM. Cited on page 132.

Wang, K., Gloy, N., and Li, X. (2010). Inferring Search Behaviors Using Partially Observable Markov (POM) Model. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 211–220, New York, NY, USA. ACM. Cited on pages 129 and 134.

Wang, Z. Y. and Klir, G. J. (1992). *Fuzzy Measure Theory*. Springer-Verlag, New York, USA. Cited on page 59.

Webber, W., Chandar, P., and Carterette, B. (2012). Alternative Assessor Disagreement and Retrieval Depth. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 125–134, New York, NY, USA. ACM. Cited on page 80.

Webber, W., Moffat, A., and Zobel, J. (2010). A Similarity Measure for Indefinite Rankings. *ACM Transactions on Information Systems (TOIS)*, 4(28):20:1–20:38. Cited on page 50.

Webber, W. and Pickens, J. (2013). Assessor Disagreement and Text Classifier Accuracy. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 929–932, New York, NY, USA. ACM. Cited on page 80.

Wei, X. and Croft, W. B. (2006). LDA-based Document Models for Ad-hoc Retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 178–185, New York, NY, USA. ACM. Cited on page 129.

Wilson, E. B. (1952). *An Introduction to Scientific Research*. McGraw-Hill. Cited on page 9.

Wu, C. F. J. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95–103. Cited on page 83.

Wu, Q., Burges, C. J. C., Svore, K. M., and Gao, J. (2010). Adapting Boosting for Information Retrieval Measures. *Information Retrieval*, 13(3):254–270. Cited on page 123.

Yadati, K., Shakthinathan, P. S. N., Ayyanathan, C., and Larson, M. (2014). Crowdsorting Timed Comments about Music: Foundations for a New Crowdsourcing Task. In *Working Notes Proceedings of the MediaEval 2014 Multimedia Benchmark Workshop*. CEUR Workshop Proceedings (CEUR-WS.org). Cited on page 81.

Yang, G. H., Sloan, M., and Wang, J. (2016). Dynamic Information Retrieval Modeling. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 8(3):1–144. Cited on pages 130 and 134.

Yilmaz, E. and Aslam, J. A. (2006). Estimating Average Precision with Incomplete and Imperfect Judgments. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pages 102–111, New York, NY, USA. ACM. Cited on pages 135 and 143.

Yilmaz, E. and Aslam, J. A. (2008). Estimating Average Precision when Judgments are Incomplete. *Knowledge and Information Systems*, 16(2):173–211. Cited on page 41.

Yilmaz, E., Aslam, J. A., and Robertson, S. (2008). A New Rank Correlation Coefficient for Information Retrieval. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 587–594, New York, NY, USA. ACM. Cited on pages 41, 93, and 100.

Yilmaz, E., Shokouhi, M., Craswell, N., and Robertson, S. (2010). Expected Browsing Utility for Web Search Evaluation. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1561–1564, New York, NY, USA. ACM. Cited on page 35.

Yu, J., Tao, D., Wang, M., and Rui, Y. (2015). Learning to Rank Using User Clicks and Visual Features for Image Retrieval. *IEEE Transactions on Cybernetics*, 45(4):767–779. Cited on page 123.

Zhang, E. and Zhang, Y. (2009). *Eleven Point Precision-recall Curve*, pages 981–982. Springer, Boston, MA, USA. Cited on page 34.

Zhang, Y., Park, L. A. F., and Moffat, A. (2010). Click-based Evidence for Decaying Weight Distributions in Search Effectiveness Metrics. *Information Retrieval*, 13(1):46–69. Cited on pages 35 and 150.

Zobel, J. (1998). How Reliable Are the Results of Large-scale Information Retrieval Experiments? In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 307–314, New York, NY, USA. ACM. Cited on page 25.