FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

On Biases in Information Retrieval Models and Evaluation

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

M.Sc. Aldo Lipani
Matrikelnummer 01129624

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Prof. Allan Hanbury
Zweitbetreuung: Dr. Mihai Lupu

Diese Dissertation haben begutachtet:

Maarten de Rijke	Ricardo Baeza-Yates

Wien, 27. Juni 2018

Aldo Lipani

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

On Biases in Information Retrieval Models and Evaluation

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

M.Sc. Aldo Lipani
Registration Number 01129624

to the Faculty of Informatics

at the TU Wien

Advisor: Prof. Allan Hanbury
Second advisor: Dr. Mihai Lupu

The dissertation has been reviewed by:

Maarten de Rijke

Ricardo Baeza-Yates

Vienna, 27th June, 2018

Aldo Lipani

Erklärung zur Verfassung der Arbeit

M.Sc. Aldo Lipani
StrauSSengas. 1A/11

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 27. Juni 2018

Aldo Lipani

Acknowledgements

Since January 2014, my Ph.D. has been an enjoyable life-changing journey.

I dedicate this thesis to my *family* for their tremendous love and support: father Angelo Lipani, mother Mariella Giarrizzo, sisters Arianna and Gaia, and brother Livio; to my *grandparents*, although two no longer with us: Ignazio Giarrizzo†, Maria Scibona†, Cataldo Lipani, and Assunta Giunta; and my *girlfriend*, Samaneh Souzankar, who patiently tolerated my long working hours.

I would like to offer special thanks to my *supervisors*, Allan Hanbury and Mihai Lupu, who inspired and guided me throughout this journey and devoted their time and energy selflessly.

I thank my *colleagues* in Vienna with whom I collaborated and shared quite a number of coffee breaks: Linda Anderson, Ralf Bierig, Allan Hanbury, Mihai Lupu, Joao Palotti, Florina Piroi, Andreas Rauber, and Serwah Sabetghadam; and my *collaborators* from other institutions: Akiko Aizawa, Evangelos Kanoulas, Bevan Koopman, Thomas Roelleke, Calogero Schillaci, Ellen Voorhees, and Guido Zuccon.

During my Ph.D. I spent thirteen months visiting and interning in other institutions. I thank: Akiko Aizawa for hosting and mentoring me during my research visit at the *National Institute of Informatics* in natural language processing; Leif Azzopardi and Colin Wilkie, for hosting and mentoring me during my research visit at the *University of Glasgow* on the retrievability topic; Evangelos Kanoulas for hosting and mentoring me during my research visit at the *University of Amsterdam* in the pool bias estimation topic; Matthew Johnson, for giving me the opportunity to work on challenging accessibility problems in my first *Microsoft Research Cambridge* internship, Alan Lawrence for his wise mentorship, and my colleagues Qiuying Giulia Lai and Santhilata KV with whom I shared this experience; Ian Soboroff and Ellen Voorhees, for inviting me at the *National Institute of Standards and Technology* and mentoring me on the IR evaluation topic; Tony Wieser for hosting me for my second research internship in *Microsoft Research Cambridge*, Sebastian Blohm and Sean Rintel for their mentorship, and my colleague Peter Wirnsberger with whom I worked side by side to solve hard Machine Learning problems.

Last but not the least, I would like to thank Markus Zlabinger, Peter Knees and Andreas Rauber for helping me translating to German and proofreading the abstract of this thesis.

Kurzfassung

Der Einzug der modernen Informationstechnologie in unsere Gesellschaft führte in den letzten fünfzig Jahren zu einer rasant wachsenden Menge von digitalen Inhalten. Während das Informationsangebot stetig steigt, bleiben unsere Fähigkeiten zur Informationsverarbeitung unverändert. Aufgrund dieser Überladung mit Informationen kommt dem Information Retrieval (IR) die wichtige Rolle zu, Systeme zu entwickeln, die relevante Informationen von irrelevanten trennen können. Diese Trennung ist allerdings auf Grund der Komplexität des Verstehens was relevant ist und was nicht, eine schwierige Aufgabe. Um diese Komplexität zu bewältigen, wurde im IR ein empirischer Ansatz gewählt, der zur Entwicklung praktikabler Retrieval-Modelle geführt hat, die einen systematischen Fehler bzw. eine Neigung (Bias) in Richtung relevanter Information aufweisen. Neben diesem Bias treten allerdings auch andere Verzerrungen auf, die problematisch für den Retrieval-Vorgang sind. In dieser Arbeit werden diese problematischen Bias durch die Betrachtung von Retrieval-Systemen als Informationsfilter bzw. Sampling-Prozesse systematisch untersucht.

Es werden Bias erforscht die üblicherweise in zwei Bereichen des IR auftreten: Retrieval-Modelle und Retrieval-Evaluierung. Zunächst wird das Retrieval-Bias von probabilistischen IR-Modellen analysiert und neue Dokument-Prioren entwickelt um die Retrieval-Leistung zu steigern. Im Anschluss wird das Zugänglichkeits-Bias von Retrieval-Modellen erörtert. Für boolesche Retrieval-Modelle wird ein eigens entwickeltes mathematisches Framework beschrieben. Hinsichtlich des Bias für Retrieval-Evaluierung werden Testdatensätze, welche mittels Pooling-Methode erstellt wurden und somit ein charakteristisches Bias enthalten, analysiert. Um die Zuverlässigkeit der Evaluierung zu verbessern, werden neue Pooling-Strategien beschrieben. Diese Strategien reduzieren das Bias bereits während der Erstellung eines Testdatensatzes. SchlieSSlich wird für die MaSSzahlen Precision- und Recall-at-Cutoff (P@n und R@n) ein neuer Pool-Bias-Schätzer entwickelt, welcher das Bias während der Systemevaluierung reduziert.

Um die vorgeschlagenen Methoden dieser Arbeit zu evaluieren, wurden 15 Testdatensätze, vier IR-Metriken und drei Bias-Messverfahren herangezogen. Durch Experimente werden folgende Erkenntnisse gewonnen: durch das Verwenden von Dokument-Prioren basierend auf Verboseness wird die Retrieval-Genauigkeit von probabilistischen IR-Modellen gesteigert; das Zugänglichkeits-Bias von booleschen IR-Modellen verschlechtert sich für konjunktive Anfragen mit steigender Länge der Anfragen (für disjunktive Anfragen

kann eine leichte Verbesserung festgestellt werden); das Testdatensatz-Bias kann bei der Erstellung des Testdatensatzes durch Pooling-Strategien, welche aus dem Bereich des Reinforcement Learning entlehnt sind (Multi-Armed Bandit Problem), verkleinert werden; und das Testdatensatz-Bias kann in der Evaluierung durch die Analyse der Pool-Beteiligung in den einzelnen Durchläufen reduziert werden. Speziell für den letzten Punkt wird gezeigt, dass das Bias für P@n durch die Quantifizierung des neuen Systems gegen die gepoolten Durchläufe und für R@n durch die Auslassung einzelner gepoolter Durchläufe reduziert wird.

Diese Arbeit leistet einen wichtigen Beitrag zum Gebiet des IR, indem ein besseres Verständnis von Relevanz durch die Betrachtung von Bias in Retrieval-Modellen und Retrieval-Evaluierung erreicht wird. Die Identifizierung dieser Bias und deren Nutzung bzw. Reduktion führt zur Entwicklung von performanteren IR-Modellen und zu einer Verbesserung der derzeitigen Vorgehensweise hinsichtlich IR-Evaluierung.

Abstract

The advent of the modern information technology has benefited society as the digitisation of content increased over the last half-century. While the processing capability of our species has remained unchanged, the information available to us has been notably increasing. In this overload of information, Information Retrieval (IR) has been playing a prominent role by developing systems capable of separating relevant information from the rest. This separation, however, is a difficult task rooted in the complexity of understanding of what is and what is not relevant. To manage this complexity, IR has developed a strong empirical nature, which has led to the development of grounded retrieval models, resulting in the development of retrieval systems empirically designed to be biased towards relevant information. However, other biases have been observed, which counteract retrieval performance. In this thesis, the reduction of retrieval systems to filters of information, or sampling processes, has allowed us to systematically investigate these biases.

We study biases manifesting in two aspects of IR research: retrieval models and retrieval evaluation. We start by identifying retrieval biases in probabilistic IR models and then develop new document priors to improve retrieval performance. Next, we discuss the accessibility bias of retrieval models, and for Boolean retrieval models we develop a mathematical framework of retrievability. For retrieval evaluation biases, we study how test collections are built using the pooling method and how this method introduces bias. Then, to improve the reliability of the evaluation, we first develop new pooling strategies to mitigate this bias at test collection build time and then, for two IR evaluation measures, Precision and Recall at cut-off (P@n and R@n), we develop new pool bias estimators to mitigate it at evaluation time.

Through a large scale experimentation involving up to 15 test collections, four IR evaluation measures and three bias measures, we demonstrate that including document priors based on verboseness improves the performance of probabilistic retrieval models; that the accessibility bias of Boolean retrieval models quickly worsens for conjunctive queries with the increase of the query length (while slightly improving for disjunctive queries); that the test collection bias can be lowered at test collection build time by pooling strategies inspired by a well-known problem in reinforcement learning, the multi-armed bandit problem; and that this bias can also be improved at evaluation time by analysing the runs participating in the pool. For this last point in particular, we show

that for P@n, bias reduction is done by quantifying the potential of the new system against the pooled runs, and for R@n, this is done instead by simulating the absence of a pooled run from the set of pooled runs.

This thesis contributes to the IR field by giving a better understanding of relevance through the lens of biases in retrieval models and retrieval evaluation. The identification of these biases, and their exploitation or mitigation, leads to the development of better performing IR models and the improvement of the current IR evaluation practice.

Contents

CHAPTER 1

Introduction

In the era of information abundance, information consumption is mediated by the use of search engines. These not only help to make information accessible but also discern between what is relevant and what is not. This gives search engines utmost importance for the progress of our society. However, this discernment, if not properly investigated, may harm the access to some information, based on factors that have little to do with its degree of relevance. In this thesis we investigate some of the *biases* observed in Information Retrieval (IR). By the term bias is meant any form of deviation from an expectation. In IR we observe various biases that affect retrieval models and their evaluation. The analysis of these biases will lead to a better understanding of the effectiveness of retrieval models and advance the current evaluation practice.

In statistics, the term 'bias' is presented as an undesirable property of an estimator. An estimator is a function that aims to estimate a parameter of a population. An unbiased estimator guaranties that the expectation of its estimates is equal to the parameter of the population. Thereby the term 'bias' in statistics, but also in the English language where it is somewhat used as a synonym of 'unfair', has developed a negative connotation. However, the etymology of this term comes from the Old French 'biais', meaning 'slant, slope, oblique', and, with uncertain origin, from the Greek 'epikarsios', meaning 'athwart, crosswise, and at an angle', suggesting a more appropriate translation of this term to modern English 'incline'. The use of the term 'incline' per se, with respect to the term 'bias', is free of any negative or positive connotation. Indeed, its connotation is carried by to what someone or something is inclined. For example, if a person is inclined to virtues we have a positive inclination, while if a person is inclined to vices we have a negative inclination, where, of course, virtues and vices are embodied by a culture of reference. Accordingly in IR, if a search engine is inclined to retrieve relevant documents we have a positive inclination, but if a search engine is inclined to retrieve irrelevant documents we have a negative inclination. In this thesis, having as reference the IR field, we use the term 'bias' as we would use the term 'incline', but without losing the meaning carried by its definition in Statistics.

1.1 Types of Bias

Biases manifest when a non-negligible factor of a statistical phenomenon is not included in the derivation of its model, thereby leading to a systematic distortion. A bias can exist only if there is missing information, which happens when a sampling procedure is used in order to estimate a parameter of a given population. Herein, we first introduce two types of bias, *model bias* and *selection bias*, which often interplay, and then identify them in IR.

Model bias refers to a bias observed on the distribution of a feature of the sample that does not reflect the distribution of the same feature in the population. For example let us imagine we have a ballot box that contains balls and cubes of two colours, red and white. All categories of items are distributed in equal number: a quarter of red balls, a quarter of white balls, a quarter of red cubes, and a quarter of white cubes. Our task is to design a filter – a sample procedure – to maximise the number of sampled balls, regardless of their colour. To do this, we design a filter that exploits the geometry of spheres. If the filter has been properly designed, and we allow a margin of error, we expect the sample set to contain mostly balls and some cubes due to a sample error. Moreover, since all items in the ballot box are in equal number red and white, we expect the same proportion of colourful items in the sampled set. But, if we observe a greater number of white items than red ones we indicate the filter to be model-biased towards the white items. However, to know this information does not tell us the causes of such bias, but that in the design of the filter some features of the items have not been taken into account. For instance, continuing with the previous example, further investigation discovers that this happened due to the material of which the items are made, which is indicated by their colour; the material of the white items is softer than the red ones so that the vertexes of the cubes are flexible enough to make them, when pushed into the filter, similar to spheres.

Selection bias refers to a bias observed on a sampled set when the sampling procedure fails in performing a proper randomisation, therefore introducing a discrepancy between the distribution of the sample set and the distribution of the population. For example let us imagine we have a ballot box as defined in the previous example. Our task is to design a filter to sample a representative set of items of the ballot box. If the filter has been properly designed, and we allow a margin of error, we expect in the sample set to contain around a quarter of white cubes, a quarter of red cubes, a quarter of white balls, and a quarter of red balls, as in the ballot box. But, if we observe a greater number of white items than red ones we indicate the filter to be selection-biased towards the white items. However, as for the previous example, this information does not tell us the causes of such bias but that in the design of the filter some features of the items have not been taken into account. For instance, continuing with the previous example, further investigation discovers that the softer white items, because of being less dense, tend to emerge to the surface of the ballot box, thereby being over-sampled.

Although we have treated each bias type separately, based on the information we have

about the distributions of the population's features and the constraints we have on the design of the filter, these two types of bias may interplay. For instance, let us imagine that in the first example we cannot observe the distribution of items' categories. To estimate them we can construct a filter, as in the second example, and then use these estimates to describe the distribution of items' categories, which may suffer from selection bias that contributes to the model bias as in the first example. Or, instead, let us imagine that in the second example we cannot design a general filter. To design it we make a composition of two filters, one optimised for cubes and one for balls and then compare these estimates, each one as in the first example, and then use them to describe the population items' distribution. However, both filters may suffer from model bias, which contributes to the selection bias as in the second example. These two examples demonstrate the possible model-selection bias interaction. In this thesis we treat model bias and selection bias analytically, that is we deal with one type of bias at a time, assuming the other one to be negligible.

Another way to distinguish between these two types of bias is to consider on what the bias is observed. If the bias is observed on the targeted feature by the sampling procedure, we talk about selection bias. If the bias is observed on another feature than the targeted one, then we talk about model bias. In general, biases are difficult to identify because it is not always clear on which feature of the data they manifest.

1.2 Information Retrieval

So far we have discussed definitions of biases that we observe in data when adopting a sampling procedure in optimisation and estimation problems. Now, starting from the definition of IR, we, on the one hand, frame these biases in IR and, on the other hand, delimit the boundaries of this exploration. On the definition of IR, Manning et al. [MRS08] write:

> "IR is finding material (usually documents) of an unstructured nature (usually text) that *satisfies an information need* from within large collections (usually stored on computers)." (p. 1)

The main purpose of a retrieval system is to satisfy a user information need. This user satisfaction plays a central role in IR, as also highlighted in its definition by being put in the centre. Naturally, user satisfaction is considered as a criterion of system effectiveness [AS10]. System effectiveness in IR measures how well a search engine performs in a task. Traditionally, this is expressed in terms of the ratio between the number of correctly identified relevant information and the retrieved information, called *precision*, and in terms of the ratio between the number of correctly identified relevant information and all the relevant information, called *recall*. But, what is relevant information? Intuitively, it is anything that satisfies the user's information need. This answer introduces considerable complexity but it shows the dual relationship between user satisfaction and relevant

information, that is the understanding of one leads to the understanding of the other and *vice versa*. To circumvent this complexity, IR has developed a strong empirical foundation, in which we treat users as relevance holders. Thereby users are the main actors in IR experimentation. However, a pure user-based experimentation would make the IR studies expensive. Therefore, the IR community have experimental settings that mitigate its cost, and have developed different approaches to letting the users perform relevance judgements, which can be categorised as: off-line evaluation, on-line evaluation, and hybrid evaluation.

By off-line evaluation, also known as test collection-based evaluation, is meant an evaluation procedure that builds a test collection with the intervention of users, but that once built, can be later used to evaluate other search solutions without requiring user interaction. By on-line evaluation is meant an evaluation procedure that is done while the user is using the service to be evaluated. A/B testing and interleaving are on-line evaluation procedures [HLR16]. By hybrid evaluation is meant any form of evaluation that aims to evaluate search solutions, as in the on-line evaluation, but that at the same time also aims to build a test collection for later use, as in the off-line evaluation. Counterfactual evaluation is a hybrid evaluation procedure [JS16] that consists in the over sampling of user interaction signals in order to build a richer test collection, which allows the testing of later developed hypnotises.

To the eyes of an IR practitioner, the evaluation of the effectiveness of a retrieval system is important because it can lead to intuitive interpretations that easily translate to an economic impact. For instance, the amount of time people are spending in reading useless documents can be expressed in terms of precision, and the number of relevant documents they are missing can be expressed in terms of recall. However, effectiveness measures do not express another important aspect of search engines, which concerns overall their ability in accessing the information available in the document collection. To tackle this issue, another kind of evaluation has been developed, based on measuring how much an IR system makes documents accessible, called accessibility.

In general, we divide the research conducted in IR into three major areas [Lip+14b]: *retrieval models*, *retrieval systems*, and *retrieval evaluation*. In the retrieval models area we deal with search engines in terms of their effectiveness in finding relevant information. In the retrieval systems area we deal instead with their efficiency in finding relevant information. In the evaluation area we deal with how to measure their effectiveness, efficiency, and various biases. In this thesis we present and propose effective solutions to measure and mitigate a set of biases observed in the IR field delimited by the retrieval models and evaluation areas. In particular in the latter we focus on the further development of accessibility measures, and the evaluation of effectiveness in an off-line setting.

1.3 Sources of Bias in Information Retrieval

Among the many biases observed in IR, in this thesis we focus on the ones that, we believe, will mostly contribute to the future of the field, and that are of interest to IR practitioners. We focus on the model bias of retrieval models introduced by an old dichotomy between the multi-topicality and verboseness hypotheses, which tries to justify why documents have different lengths. Following that, we concentrate on a measure of accessibility, called retrievability, which measures the fairness of a search engine in retrieving the documents of a collection of documents. Finally, we focus on a selection bias observed in off-line evaluation, called pool bias.

1.3.1 Multi-Topicality and Verboseness Hypotheses

IR models try to maximise user satisfaction. To do so, retrieval models measure the likelihood that a document is relevant to a given topic using features at the level of the topic, the document, and the collection of documents. Due to the difficulty introduced by the complexity of natural language, the combination of features computed on the text can introduce model biases. A well-known model bias for standard IR retrieval models is caused by the document length. The reason for such model bias can be found in the fact that longer documents have higher prior probabilities of having a term repeated many times, which means higher term frequency, and higher term frequency in these models is interpreted as more relevant.

The most successful IR retrieval models, Best Match 25 (BM25) [Rob+93] and the various Language Models (LMs) [PC98] normalise the term frequency based on the length of the document. Without it, these models would be biased towards long documents. This normalisation makes sense under the hypothesis that documents are long because of their authors being verbose. However, a crude normalisation based on the document length would generate a model bias towards short documents. Additionally, documents can be long because their authors covered more ground about the topic therefore making them more relevant, which suggests to not normalise under any circumstances. This dichotomy is well explained by the two hypotheses introduced by Robertson et al. [Rob+93; RZ09], the multi-topicality and verboseness hypotheses.

In Chapter 4, we present our contribution, which is a systematic modification of BM25 and LM retrieval models. Based on the observation that the two cases previously discussed for length normalisation (multi-topicality and verboseness) are actually three: multi-topicality, verboseness with word repetition and verboseness with synonyms, we propose and test new normalisations. We focus on the verboseness with word repetition and document length because easily measurable by counting words. To theoretically justify the combination, we show the duality between document verboseness and length. In addition, we investigate the duality between verboseness and other components of IR models.

1.3.2 Accessibility Measures

While effectiveness and efficiency measures are respectively user-centric and system-centric, as pointed out by Azzopardi and Vinay [AV08a], both ignore the accessibility of a document. Accessibility studies if a document is or is not accessible by the user through the IR system. Accessibility plays a particularly important role in recall oriented domains. For example, patent examiners are concerned about the fact that certain IR systems are biased towards particular patents rather than others. Also in the medical domain, medical researchers, doing systematic reviews include in their protocol the use of different search engines in order to avoid such a bias.

A measure of accessibility is retrievability. Retrievability is a document-centric measure that computes the a-priori likelihood that a document in a collection is retrieved, no matter for which topic. It allows the researcher, when comparing the documents of a collection, to understand the a-priori unbalance of a retrieval model in selecting documents.

Retrievability analyses are based on empirical studies and are computationally expensive. In essence, a retrievability study consists in automatically generating a huge number of queries, issuing them to an IR system, then counting how many times a document has been retrieved. Each step of the process has different parameters, useful to characterise the IR system: the likelihood of a query, the parameters of the IR model, and the rank at which a document is considered retrieved. Parameters that, if not tuned conscientiously, easily generate billions of queries making the experiment impractical, leading to difficulties when running experiments with modern test collections. More importantly, these parameters lead to aspects of retrievability unexplored (*e.g.* queries of size greater than two). Therefore another approach has to be taken.

In Chapter 5, we show that the retrievability measure can be computed using an analytical approach. We started modelling conjunctive and disjunctive queries in Boolean models, which let us calculate retrievability without the need for generating large sets of synthetic queries. We then bridge the discoveries to the best-match models, thanks to a theoretical result that states that the result found represents an upper bound on the retrievability for all the other best-match models.

1.3.3 Pooling Method

Since the very beginning of standardised IR benchmarking at the Text REtrieval Conference (TREC) in the early 1990s, the *pooling method* has been used to reduce the number of judgements to be performed by relevance assessors, while still preserving the ability of the benchmark to distinguish between two or more retrieval engines [VH05]. The original pooling method was first proposed in 1975 by Spärk Jones and van Rijsbergen [SR75], and first used when TREC started in 1991 [Har93]. This strategy consists in aggregating, for every topic, the top K documents returned by many search engines, and presenting only this set to human assessors for evaluation. Pooling fundamentally relies on the assumption that if sufficiently many and sufficiently diverse systems participate in a pool (*i.e.*, provide lists of documents they consider to be relevant for each topic), a set of

topic and document pairs can be identified that, once evaluated, will be predictive of the future relative performance of two or more retrieval systems.

While the pooling method was introduced with the objective of finding as many relevant documents as possible (under the hidden implication that if a document is not retrieved by any system, it is probably irrelevant for the topic), the realistic objective is in fact to produce an unbiased sample of the set of relevant documents [Spä03]. Since the early days of the pooling method, it has been observed that, in the absence of sufficiently numerous and diverse systems, there is a risk that the identified set of relevant documents will be so limited that future systems, retrieving a new set of relevant (but actually unjudged) documents, will be considered ineffective because they do not primarily find the set of relevant documents found by the systems that were originally pooled [Rob08]. As a result this generates a selection bias called *pool bias*.

Our contribution on this area channels in two directions: On the one hand, in Chapter 6, we reduce bias at test collection build time by considering several pooling strategies. We analyse old and new proposed pooling strategies on existing test collections. On the other hand, in Chapter 7, we reduce the effect of the bias in existing test collections at evaluation time. We develop a set of methods to reduce the pool bias for precision at cut-off (P@n) and recall at cut-off (R@n). There are two reasons to consider such 'simple' measures: first, they are cornerstones for many other developed measures and, second, they are easy to understand by all users. In particular, as previously mentioned, they lead to more intuitive interpretations for practitioners.

1.4 Mathematical Framework and Software

In this thesis we draw a mathematical framework that unifies two main areas of study in IR: retrieval models and retrieval evaluation. This framework mainly consists of symbols that fix concepts and ideas making them easy to manipulate. The choice of intuitive symbols aims to reduce the cognitive load of the reader when moving from one topic to another. Therefore, before making the observations of Chapters 4–7, we believe that this mathematical framework is also a valuable contribution to the field.

The software developed to run the experiments presented in this thesis is open-source and available at the following website: `http://www.aldolipani.com`.

1.5 Publication List

The subsequent chapters of the thesis are heavily based on the following published papers.

Chapter 4 is on the multi-topicality verboseness dichotomy:

- Aldo Lipani, Mihai Lupu, Allan Hanbury, and Akiko Aizawa. "Verboseness Fission for BM25 Document Length Normalization". In: *Proceedings of the 1st ACM International Conference on The Theory of Information Retrieval*. ICTIR '15. Northampton, Massachusetts, USA: ACM, 2015, pp. 385–388. Short paper;
- Aldo Lipani, Thomas Roelleke, Mihai Lupu, and Allan Hanbury. "A Systematic Approach to Normalization in Probabilistic Models". In: *Information Retrieval Journal* (June 2018). Journal paper.

Chapter 5 is on a theoretical exploration of retrievability:

- Aldo Lipani, Mihai Lupu, Akiko Aizawa, and Allan Hanbury. "An Initial Analytical Exploration of Retrievability". In: *Proceedings of the 2015 ACM International Conference on The Theory of Information Retrieval*. ICTIR '15. Northampton, Massachusetts, USA: ACM, 2015, pp. 329–332. Short paper.

Chapter 6 is on the mitigation of the pool bias at test collection build time:

- Aldo Lipani, Guido Zuccon, Mihai Lupu, Bevan Koopman, and Allan Hanbury. "The Impact of Fixed-Cost Pooling Strategies on Test Collection Bias". In: *Proceedings of the 2nd ACM International Conference on the Theory of Information Retrieval*. ICTIR '16. Newark, Delaware, USA: ACM, 2016, pp. 105–108. Short paper;
- Aldo Lipani, Joao Palotti, Mihai Lupu, Florina Piroi, Guido Zuccon, and Allan Hanbury. "Fixed-Cost Pooling Strategies Based on IR Evaluation Measures". In: *Proceedings of the 39th European Conference on IR Research*. ECIR '17. Cham: Springer International Publishing, 2017, pp. 357–368. Full paper;
- Aldo Lipani, Mihai Lupu, Joao Palotti, Guido Zuccon, and Allan Hanbury. "Fixed Budget Pooling Strategies Based on Fusion Methods". In: *Proceedings of the 32nd ACM SIGAPP Symposium On Applied Computing*. SAC '17. Marrakech, Morocco: ACM, 2017, pp. 919–924. Full paper;
- Aldo Lipani, Mihai Lupu, and Allan Hanbury. "Visual Pool: A Tool to Visualize and Interact with the Pooling Method". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '17. Shinjuku, Tokyo, Japan: ACM, 2017, pp. 1321–1324. Demo paper.

Chapter 7 is on the mitigation of the pool bias at evaluation time for P@n and R@n:

- Aldo Lipani, Mihai Lupu, and Allan Hanbury. "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. Santiago, Chile: ACM, 2015, pp. 103–112. Full paper;
- Aldo Lipani, Mihai Lupu, and Allan Hanbury. "The Curious Incidence of Bias Corrections in the Pool". In: *Proceedings of the 38th European Conference on IR Research*. ECIR '16. Cham: Springer International Publishing, 2016, pp. 267–279. Full paper;
- Aldo Lipani, Mihai Lupu, Evangelos Kanoulas, and Allan Hanbury. "The Solitude of Relevant Documents in the Pool". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM '16. Indianapolis, Indiana, USA: ACM, 2016, pp. 1989–1992. Short paper.

1.6 A Reader's Guide

This thesis is structured as follows. Chapter 2 presents the State-of-the-Art. Chapter 3 introduces the notation that is shared over the rest of the thesis. Then Chapter 4 focuses on the normalisation component of some IR models, and Chapter 5 on the retrievability measure. Next, Chapters 6 and 7 are dedicated to the pool bias, the former aims at mitigating it at test collection build time, the latter at the time of application of an IR measure. Finally, we conclude in Chapter 8.

In Figure 1.1 we present the map of this thesis. The reader is now at the end of the introductory chapter, in grey. From now on they can follow 5 main paths from left to right. If the reader is interested only in a specific bias treated in this thesis they can take one of the four paths leading to the four main Chapters, 4, 5, 6 and 7. However, if the reader is interested in a summary of the thesis findings they can directly skip to the conclusion, as suggested by the fifth path.

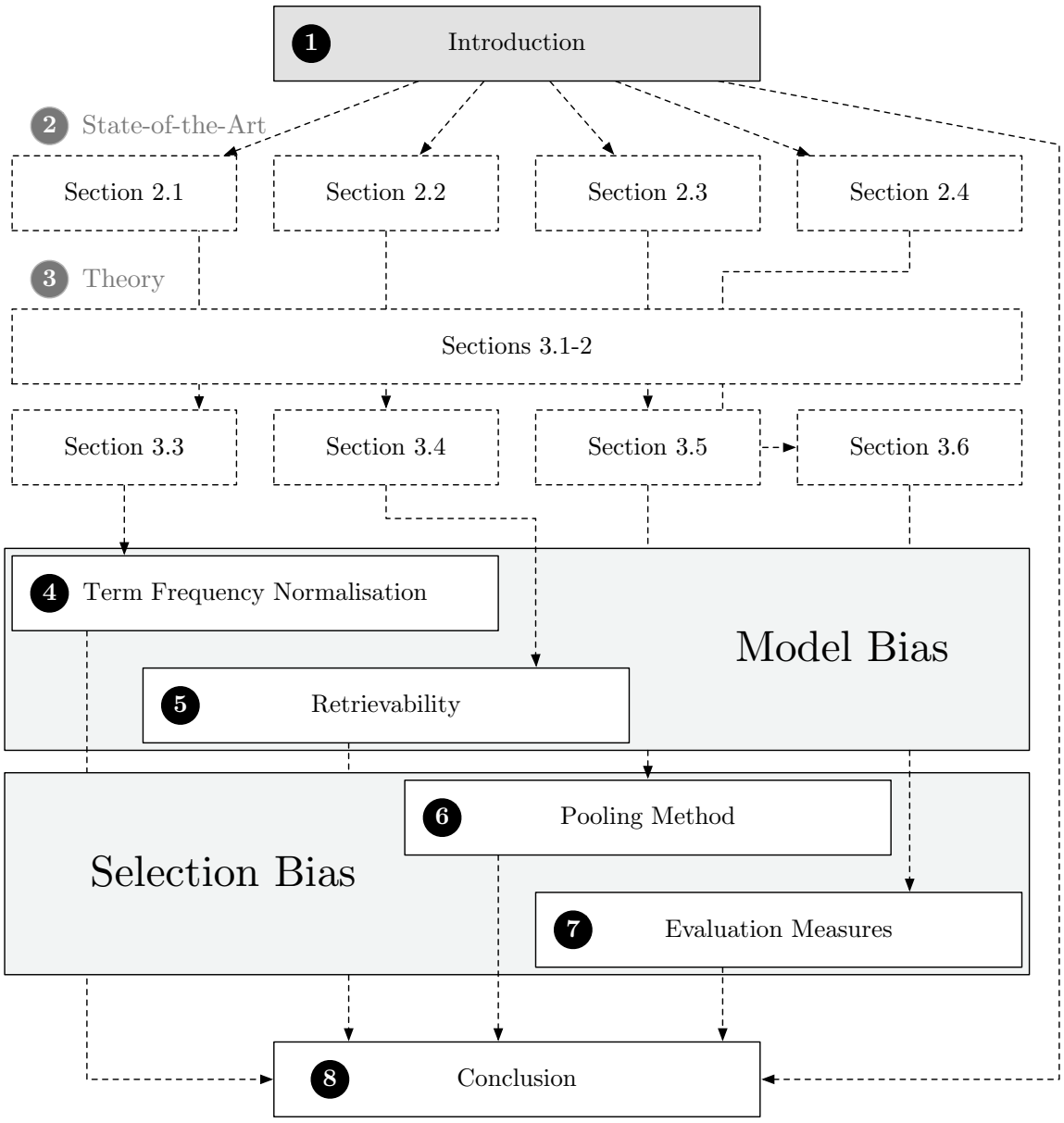

Figure 1.1: Thesis outline.

CHAPTER 2

State-of-the-Art

In this thesis we cover topics on major aspects of IR, retrieval models and retrieval evaluation. Baeza-Yates and Ribeiro-Neto [BR11] give a general overview of IR. Roelleke [Roe13] provides an advanced thorough presentation of retrieval models, including all the retrieval models analysed in this thesis. In this work, Roelleke unifies the presentation and derivation of these retrieval models by creating a common mathematical framework, which has inspired the same approach in this thesis. Sanderson [San10] gives a comprehensive overview of off-line evaluation, including the pool bias topic.

In Section 2.1 we present the state-of-the-art for the term frequency normalisation component of various retrieval models. In Section 2.2 we overview the main work conducted on evaluating the accessibility of retrieval models. We then move onto the work done on the pool bias, with effort channelled in two main directions. On the one hand, prior work has attempted to reduce the pool bias at test collection build time by considering different pooling strategies [Büt+07; CPC98; MWZ07]. On the other hand, for already existing test collections, some studies have adopted measures that reduce the effect of the bias [WP09]. Sometimes, these two directions intertwine, and a new pooling strategy is proposed together with a matching evaluation measure [YKA08], but that significantly restricts the future use of the collection to specific measures. In Section 2.3 we focus on the first direction, while in Section 2.4 we focus on the second one. However, this second direction can be subsequently split into two further directions. On the one hand, prior work has created estimators for correcting the bias of existing evaluation measures [WP09]. On the other hand, new evaluation measures have been developed with the aim of being less sensitive to the pool bias (the work done for bpref [BV04], followed by the work done by Sakai on the condensed lists [Sak07] or by Yilmaz et al. on the inferred measures [YKA08; YA06]). In Section 2.4, we will focus on the former.

2.1 Model Bias: Term Frequency Normalisation

The initiators of the discussion about the term frequency normalisation are the early participants in TREC, with first insights appearing after TREC-3, and the first efforts on document length normalisation showing improved results in TREC-4 [Har95]. To understand why a document is long, Robertson and Zaragoza [RZ09] describe two hypotheses: verboseness and multi-topicality hypotheses. The verboseness hypothesis says that authors use more words than needed to convey the information. The multi-topicality hypothesis says that authors convey the information but including more topics, details, or aspects. These hypotheses have a conflicting effect when treating the normalisation in terms of length, because while the first suggests to normalise the term frequency by the document length, the second suggests the opposite.

Hence, the introduction of a soft normalisation for Best Match 25 (BM25) based on the linear combination between a non-normalisation and a full normalisation based on the average document length, which trade off is controlled by the introduction of the new parameter, b. This is of course not the only method for length normalisation. Among others, Singhal et al. [SBM96] studied it extensively for the TF-IDF model, and justified it experimentally. In their study, they look at how the length distribution of retrieved documents and relevant documents differ, and provide normalisation solutions to correct this discrepancy.

Not much work has been done on the multi-topicality hypothesis, but some for the verboseness hypothesis. Na et al. [NKL08] briefly introduce the concept of verboseness given by the repetitiveness of terms. They compare it with multi-topicality under the language modelling framework. The normalisation factors are corrected based on the assumption that the vocabulary size of a document can be used to estimate the number of topics contained in the document. They show an improvement with respect to other smoothing methods. He and Ounis [HO05a] introduced a new term frequency normalisation following the idea of Amati [AV02], who introduced the use of Dirichlet Priors. He and Ounis point out the relationship between test collection features on term frequency normalisation, and introduce a new parameter, learned from the test collection. They proposed a method for tuning the term frequency normalisation parameters based on the hypotheses that the optimal parameter values are those values that make the normalisation factor give similar normalisation effects across different corpora [HO03; HO05b]. Lv and Zhai pointed out that the length distribution observed on retrieved documents of BM25 does not follow the length distribution observed on relevant documents, as done by Singhal et al. [SBM96] for TF-IDF, biasing the system against long documents. To compensate this bias they introduced a new 'boosting' parameter, δ, which is summed to the normalised term frequency in a first version [LZ11b] and it is summed to the term frequency component in a second version [LZ11a]. Rousseau and Varzirgiannis [RV13] analyse the problem in terms of function composition, comparing BM25 with TF-IDF and combining the two works just mentioned of Lv and Zhai to gain a better understanding of the similarity across the models. This work formalises this term frequency normalisation modification using functional composition, which allowed

the authors to test combinations not yet analysed. Another effort has been made by Cummins and O'Riordan [CO12], but this time on the analysis of the effect of query length on the parameter b. To avoid the over penalisation of long documents, they added, to the classic TF normalisation, the probability that a randomly selected document contains at least one query term (this probability is proportional to the query length) and use this factor to stabilise b at various query size. However this issue does not effect our analysis since we do not change the query size in our experiments. This work, by showing that there exists a relationship between the probability of a document to be retrieved and the effectiveness of a search engine, takes us to the topic of the next section.

The overall criticism of all of these previous works is that the test collections used are always based on News or Web corpora, therefore reflecting only these two domains.

2.2 Model Bias: Retrievability

Accessibility is a well-known concept in the field of transportation planning. Azzopardi and Vinay [AV08a] introduce the concept of accessibility in IR, and advocated the development of document accessibility measures for IR systems. Later they developed, as a measure of accessibility, the retrievability measure [AV08b]. This measure indicates how easily a document could be retrieved by an IR system.

To estimate the retrievability bias the retrievability measure is generally combined with a coefficient of distribution imbalance. An often used coefficient is the Gini coefficient [Gas72], which is a measure of inequality within a population. Many coefficients have been tested [WA15], however they all provide similar information regarding the bias as the Gini coefficient.

Retrievability, despite being still an immature concept in IR, has found already many applications in many contexts. Azzopardi and Owens [AO09] have used it as a tool to assess the bias of search engines on the web. Bache [Bac11] has performed a similar analysis on the patent domain. Zheng and Cox [ZC09] have developed new pruning strategies to improve the efficiency for inverted indices. Pickens et al. [PCG10] have developed the concept of reverted index, where to be indexed are queries rather than documents. This, together with a ranking technique, can be used as high performance query expansion. We have now seen that retrievability has been a widely useful concept in applications, however, we now focus on the work conducted on relating the retrievability to effectiveness of a search engine and in particular how this relates to the term frequency normalisation component of scoring functions, which controls the prior probability of a document to be relevant based on its length.

Azzopardi and Vinay [AV08b] argue that a certain level of retrievability bias is necessary in order to allow the search engine to distinguish between what are relevant and irrelevant documents for a given topic. They show that the effectiveness of a search engine is inversely proportional to the retrievability bias. Bashir and Rauber [BR09], after having designed a pseudo-relevance feedback technique inspired by a retrievability analysis,

observed an increase in retrievability bias and in effectiveness. Then, when they applied this method on the patent domain, they also observed an increase in retrievability bias as well as in recall [BR10]. However, Wilkie and Azzopardi [WA13] show that this is true until a certain extent, showing that this relationship is non-linear and can lead to a decrease in effectiveness if too much retrievability bias is forced on the system.

Wilkie and Azzopardi [WA13] analysed the relationship between retrievability bias and the term frequency normalisation parameter b of BM25. Here, they show that reducing the retrievability bias leads to better effectiveness, however, if this term frequency normalisation is too strong this leads to a degradation in effectiveness and at the same time in an increase in retrievability bias. In this work no aspect of pool bias is analysed or discussed. However, the authors speculate that this inversely proportional behaviour between retrievability bias and effectiveness may be due to a document length bias observed in some standard test collections as discovered by Losada et al. [LAB08].

2.3 Selection Bias: Pooling Method

The pooling method was already used in the first TREC, in 1992, 17 years after its introduction by Spärck Jones and van Rijsbergen [SR75], based on the discussion of building an 'ideal' test collection that would allow reusability. The algorithm [Har93] is described as follows: 1) divide each set of results into results for a given topic; then, for each topic: 2) select the top 200 (subsequently generalised to K) ranked documents of each run, for input to the pool; 3) merge results from all runs; 4) sort results on document identifiers; 5) remove duplicate documents. This strategy is known as *fixed-depth pooling*, here also called Depth@K. This is the most commonly used pooling strategy. Since then other pooling strategies have been proposed.

The aim of the pooling method, as pointed out by Spärck Jones, is to find an unbiased sample of relevant documents [Spä03]. The bias can be minimised via increasing either the number of topics, or the number of pooled documents, or the number and variety of IR systems involved in the process.

With the aim of further reducing the cost of building a test collection, Buckley and Voorhees [BV04] explored the uniformly sampled pool. At the time they observed that P@n had the most rapid deterioration compared to a fully judged pool. The poor behaviour of this strategy for top-heavy measures was confirmed recently in Voorhees' [Voo14] short comparison on pooling strategies.

Another strategy is the stratified pool [YKA08], a generalisation of both the fixed-depth pool and the uniformly sampled pool. The stratified pool consists in layering the pool in different strata based on the highest rank obtained by a document in any of the given runs.

A comparison of the various pooling strategies has been reported by Voorhees [Voo14]. In this paper, it is advocated that for a recall oriented domain to use a stratification with a fully sampled first stratum until rank 10, because this produces test collections that

are less biased. However, in these experiments the number of judged documents is not kept constant.

In order to reduce the amount of budget spent to build a test collection but maintaining the same quality, Cormack et al. [CPC98] introduced the pooling strategy Move To Front (MTF). This strategy improves on the Depth@K by pooling documents based on the retrieval performance of the pooled runs. A similar idea is developed by Moffat et al. [MZ08], who introduce a set of pooling strategies based on the evaluation measure Rank-Biased Precision (RBP). These strategies are evaluated in terms of bias obtained when using fewer relevant judgements. They observed that these strategies perform better than Depth@K. Losada et al. [LPB16] considered a new perspective on pool creation based on multi-armed bandits. The multi-armed bandit problem, studied in reinforcement learning to trade-off between exploration and exploitation, fits well with the characteristics of the pooling method. This paper introduced new pooling strategies, but no evaluation in terms of pool bias was made. However, most of the pooling strategies presented in these articles are more difficult to operationalise because they are adaptive, that is these strategies require to know, every time needing to select a new document, if the last pooled document was relevant or not relevant.

2.4 Selection Bias: Evaluation Measures

In this section we focus on the set of work conducted to create new measures to better handle unjudged documents, and estimate the pool bias to adjust the measured score.

Buckley and Voorhees [BV04] introduced bpref as a measure specifically designed to handle incomplete information, which, as pointed out by Sakai in 2007 [Sak07], is a restricted form of Average Precision (AP) on a so called 'condensed list'. These are condensed versions of the runs where unjudged documents are filtered out. Sakai shows that it is possible to obtain less biased results than bpref when applying the condensed list to well-known IR measures, like AP, Normalised Discounted Cumulative Gain (NDCG) and Q-measure. The concept of condensed list, first denoted as such by Sakai, was however already explored in relation to AP with the measure Induced AP, introduced by Yilmaz and Aslam [YA06]. Induced AP is average precision calculated on condensed lists. The methods explored by these three contributions do not simulate the effect of shallow pooling or of comparing unpooled runs against pooled ones, because they remove the effect of bias sampling from the query relevance set, ending up with an unrealistic use case. This was later addressed by Sakai and Kando [SK08], who demonstrated that the condensed list approach favours new systems.

To deal with incomplete judgements, another measure was introduced by Moffat and Zobel [MZ08], Rank-Biased Precision (RBP). This is expressed by a value and a residual. The residual quantifies the uncertainty introduced by the unjudged documents. Its value is computable thanks to the fact that it is not normalised by the number of relevant documents. This implies that the computation of the measure defines a lower bound for any given run. Moffat and Zobel attempted to make a measure that is naturally

convergent, where the contribution of each rank has a fixed weight. This would have both benefits of a normalised measure and those of a measure averageable over topics with different numbers of relevant documents. However, this attempt was unsuccessful, as pointed out by Sakai and Kando [SK08], who proved this to be inferior with respect to the condensed list.

Moffat and Zobel [MZ08] when presenting RBP, introduce the discussion around the fact that the residual can be used to estimate and correct pool bias. Webber and Park [WP09] continue their work on RBP by adding to the score the average residual calculated against the pool proceeding with a leave-one-run-out approach. To estimate it they span two dimensions: the topics and the systems. Their method follows the assumption that the scores produced by the runs are normally distributed, a probably incorrect but common assumption. Although the method was presented only on RBP, they pointed out that similar results were obtained also with P@n.

CHAPTER 3

Theory

This chapter introduces the reader to the theory shared across the next chapters. The developed theoretical framework will lead to the formalisation of the problems from which the respective chapters will branch out to, on the one hand, if needed, further develop its theory, and on the other hand, through experimentation, present solutions to the problems.

Before indulging into the theory we first unify the mathematical notation used throughout the thesis. We then present how to formalise an IR system and its evaluation. This will guide us to the exposure of the sources of bias studied in this thesis, which will be formalised and tackled in the next chapters.

3.1 Notation

In the following table we present the notation used throughout the thesis. The table includes a set of symbols, functions and operators used to express operations in a compact way.

Symbols	
$\mathcal{U}$	Set of users.
u	A user $u \in \mathcal{U}$.
$\mathcal{Q}$	Set of topics.
q	A topic $q \in \mathcal{Q}$.
$\mathcal{R}$	Set of runs.
$\mathcal{R}_p$	Set of pooled runs $\mathcal{R}_p \subseteq \mathcal{R}$.
$\mathcal{O}$	Set of organisations submitting a set of runs $\subseteq \mathcal{R}$.
r	A run $r \in \mathcal{R}$.
$\mathcal{D}$	Collection of documents.
d	A document $d \in \mathcal{D}$.
$\mathcal{T}$	Set of terms.
t	A term $t \in \mathcal{T}$.
$\mathcal{D}_t$	Set of documents where t occurs.
$\mathcal{D}_r$	Set of documents in r.
$\mathcal{T}_d$	Set of terms in d.
$\mathcal{T}_u$	Set of terms given by a user u.
$\mathcal{J}$	Set of pooled documents ($\mathcal{J}^+ \cup \mathcal{J}^- = \mathcal{J}$ and mutually exclusive).
$\mathcal{J}^+$	Set of relevant pooled documents $\mathcal{J}^+ \subseteq \mathcal{J}$.
$\mathcal{J}^-$	Set of irrelevant pooled documents $\mathcal{J}^- \subseteq \mathcal{J}$.
ϵ	A small number $\ll \max_{r \in \mathcal{R}_p}(\lvert r \rvert)^{-1}$.
Symbolic Values	
$\lvert\mathcal{Q}\rvert$	Number of topics.
$\lvert\mathcal{T}\rvert$	Number of terms.
$\lvert\mathcal{T}_d\rvert$	Number of terms in d.
$\lvert\mathcal{D}\rvert$	Number of documents.
$\lvert\mathcal{D}_t\rvert$	Number of documents where t occurs (aka document frequency).
ℓ_c	Length of the collection (number of term occurrences).
ℓ_d	Length of the document d (number of term occurrences, note $\ell_d \geq \lvert\mathcal{T}_d\rvert$).
ℓ_t	Number of occurrences of the term t in the collection, here also called term length (aka collection frequency).

Expectations	
$\mathrm{E}_{\mathcal{D}_t}[tf_d] = \ell_t/\lvert\mathcal{D}_t\rvert$	Average frequency of term t in the documents in which the term occurs.
$\mathrm{E}_{\mathcal{T}_d}[tf_d] = \ell_d/\lvert\mathcal{T}_d\rvert$	Average term frequency of terms that occur in document d.
$\bar{\ell}_d := \mathrm{E}_{\mathcal{D}}[\ell_d] = \ell_c/\lvert\mathcal{D}\rvert$	Average document length.
$\bar{\ell}_t := \mathrm{E}_{\mathcal{T}}[\ell_t] = \ell_c/\lvert\mathcal{T}\rvert$	Average term length.
Probabilities	
$P(t) = P_L(t) = \ell_t/\ell_c$	Location-based probability of t.
$P(d) = P_L(d) = \ell_d/\ell_c$	Location-based probability of d.
$P_D(t) = \lvert\mathcal{D}_t\rvert/\lvert\mathcal{D}\rvert$	Document-based probability of t.
$P_T(d) = \lvert\mathcal{T}_d\rvert/\lvert\mathcal{T}\rvert$	Term-based probability of d.
Functions	
[Condition]	Returns 1 if the binary condition within the brackets is verified, 0 otherwise (aka Iverson bracket).
$\wp(\mathcal{R})$	Returns the powerset of the set given as its argument (aka Weierstrass p).
$\tau@N(\mathcal{R}_p, s)$	Returns the union of the top N documents retrieved by the set of pooled runs $\mathcal{R}_p$ ordered by the function s.
$\rho(d, r)$	Returns the *rank* at which the document d has been retrieved in run r. If $d \notin r$ Returns the lowest rank possible, which is equal to the size of the collection of documents $\lvert\mathcal{D}\rvert$.
$\sigma(d, r)$	Returns the *score* at which the document d has been retrieved in run r. If $d \notin r$ it returns the lowest score returned by r, $\min_{d \in r}(\sigma(d, r))$.
$\mu(a, b)$	Returns a random number in $[a, b]$.
$\mathrm{id}(r)$	Returns a natural number $n \in \mathbb{N}$ unique for every $r \in \mathcal{R}$ such that $1 \leq n \leq \lvert\mathcal{R}\rvert$.
Sequences	
$a\vert_{n_0}^{n_1}$	Set of elements of the sequence a_n from n_0 to n_1, $\{a_i\}_{n_0 \leq i \leq n_1}$.
$\mathrm{Avg}(a\vert_{n_0}^{n_1})$	Average of the sequence a_n for the values from n_0 to n_1.
$\mathrm{Var}(a\vert_{n_0}^{n_1})$	Variance of the sequence a_n for the values from n_0 to n_1.

3.2 The Anatomy of an IR System

An IR system can be interpreted as a function f that, given as input a topic $q \in \mathcal{Q}$ and a collection of documents $\mathcal{D}$, associates to every document in $\mathcal{D}$ a value $s \in \mathbb{R}$, as part of a set $r \in \mathcal{R}$, where $\mathcal{R} \subseteq \mathcal{D} \times \mathbb{R}$ and $r = \{(d_1, s_1), \ldots, (d_{|\mathcal{D}|}, s_{|\mathcal{D}|})\}$. This can be expressed as $f : \mathcal{Q} \times \mathcal{D} \to \mathcal{R}$. In IR, topics and documents can be any type of information, called modality, *e.g.* a piece of text, a piece of music, a video clip, a formula, or a combination of them (*e.g.* texts and formulae [Lip+14a]). However, in this thesis we focus on text retrieval, therefore we assume a topic to be expressed in textual form, and documents to be pieces of text. Of course, this and the following can be generalised to other modalities or to a combination of them.

A minimal retrieval system is made of a set of components [Lip+14c; Pir+15]. In Figure 3.1 we show how the information flows in a retrieval system throughout its components. Here, we distinguish between optional and required components. Optional components are the ones outside the IR System Core box, and required components are the ones within it. The components are: collection preprocessor (CP), document preprocessor (DP), topic preprocessor (TP), indexer (IN), scorer (SC), ranker (RK), and merger (ME).

Among the required components we have the document and topic preprocessors, which are required to transform the text into comparable entities. These are then indexed by the indexer component for quick scoring. The role of the scorer component is to compute the degree of relevance between the topic and the documents. Then, the ranker outputs them in order of relevance. The optional components are the collection preprocessor, whose task can be cleansing of the collection of documents, and the merger component, which, if needed, merges the ranked lists coming from multiple system cores.

In the next section and in Chapter 4 we focus on the scorer component. This component is characterised by a scoring function also known as Retrieval Status Value (RSV) function. The RSV, given a preprocessed document and a preprocessed topic, returns a value indicating the degree of relevance of the document to the topic, $RSV : \mathcal{Q} \times \mathcal{D} \to \mathbb{R}$. At this point these documents and topics have been already preprocessed. The preprocessing steps required for the models described in this thesis are in order: tokenization, and stemming or lemmatisation. The tokenizer splits text into words following linguistic features, *e.g.* in English spaces are indicator of word boundaries. The stemmer reduces a word to its stem form, while lemmatiser reduces a word to its lemma.

Given these premises, our modelling is about the input of the scorer component, topic and documents. A document is seen as a set of term value pairs ($(t, tf) \in \mathcal{T}_c \times \mathbb{N}$, where $\mathcal{T}_c$ is the set of terms of the collection of documents c). The value associated to the document term is equal to the number of occurrences of the term in the document d, this is also known as *within-document term frequency*. A topic q, usually thought as an item of a set of topics, likewise a document, is also seen as a set of term value pairs ($(t, tf) \in \mathcal{T}_u \times \mathbb{N}$, where $\mathcal{T}_u$ is the set of terms given by the user u). The value associated

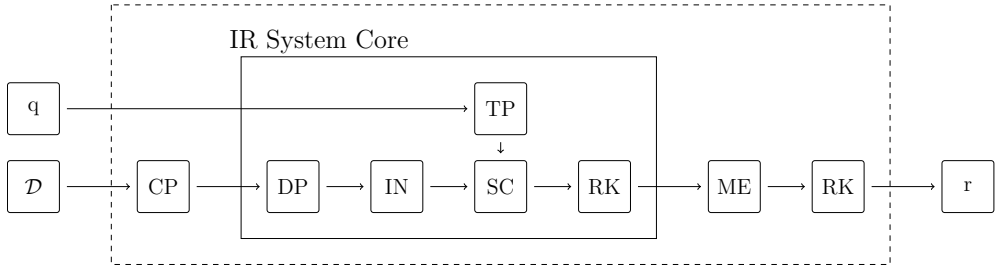

Figure 3.1: Information flow of a retrieval system throughout its components. The components are: collection preprocessor (CP), document preprocessor (DP), topic preprocessor (TP), indexer (IN), scorer (SC), ranker (RK), and merger (ME). This graph uses the plate notation, that is the IR System Core component can be repeated as many times as required.

to each topic term is equal to the number of occurrences of the term in the topic q, this is also known as *within-topic term frequency*. This modelling is also known as bag-of-words.

In the next section we focus on modelling the function of the SC component. In particular, we derive the RSV functions of three common retrieval models.

3.3 Probabilistic Retrieval Models and Term Frequency Normalisation

The description of the IR system presented in the previous section focuses on the functional aspects of an IR system rather than on its efficiency (*e.g.* the test collection is not submitted to the search engine every time a topic is submitted). We are more concerned about its effectiveness. The effectiveness of a search engine is dependent on all its components. However, a big role is played by the retrieval status value function defined by the retrieval model. In this section we will work through the derivation of three common retrieval models.

Most IR models can be derived from measuring the dependence between document and query. The document-query independence (DQI [RW08]) is the point-wise mutual information expressed as:

$$\mathrm{DQI}(d,q) := \log\left(\frac{P(d,q)}{P(d)\cdot P(q)}\right) \tag{3.1}$$

Document and topic are considered as sequences of term events. The decomposition of d leads to TF-IDF (and, for particular assumptions, to BM25), and the decomposition of q leads to language modelling (LM). We first review the decomposition of d. When decomposing d with,

$$P(d,q) = P(d|q)P(q)$$

and then by assuming term events are independent, that is:

$$P(d|q) = \prod_{t \in \mathcal{T}_d} P(t|q)^{tf_d}$$

substituting these two into Eq. (3.1) we obtain:

$$\log\left(\frac{P(d|q)}{P(d)}\right) = \sum_{t \in \mathcal{T}_d} tf_d \cdot \log\left(\frac{P(t|q)}{P(t)}\right) \tag{3.2}$$

Here, $P(t|q)$ is the query term probability, and $P(t)$ is the background model (collection-wide) term probability. The equation makes two independence assumptions: different terms are independent, and also the multiple occurrences of the same term are independent. The first assumption is reflected in applying the sum over different terms, and the second assumption is reflected by the total term frequency count, tf_d. Another assumption one can make is that the occurrence of the same term is semi-subsumed [RKB15], $2 \cdot tf_d/(tf_d + 1)$, which leads to the definition of the $\mathrm{TF}_{\mathrm{BM25}}$.

The question now remains of how to close the gap between $P(t|q)/P(t)$ and IDF, as commonly defined in the literature: IDF=$1/P_{\mathcal{D}}(t)$. Mathematically, we are looking for a justification that leads to the following equation:

$$\log\left(\frac{P(t|q,c)}{P(t|c)}\right) = \begin{cases} \log\left(\frac{1}{P_D(t|c)}\right) & t \in \mathcal{T}_q \\ 0 & t \notin \mathcal{T}_q \end{cases} \tag{3.3}$$

where, in order to avoid confusion in the next derivation steps, the collection symbol c is made explicit. We note that $P(t|c)$ and $P_D(t|c)$ are both in the denominators of the functions. Let us consider what the relation between these two elements is, *i.e.*, $P(t|c)/P_D(t|c)$. We have:

$$\frac{P_D(t|c)}{P(t|c)} = \frac{|\mathcal{D}_t|}{|\mathcal{D}|} \cdot \frac{\ell_c}{\ell_t} = \frac{\ell_c}{|\mathcal{D}|} \cdot \frac{|\mathcal{D}_t|}{\ell_t} = \frac{\bar{\ell}_d}{b_t}$$

where in the right-hand side of the previous equation we use the definition of burstiness [Roe13]:

$$b_t = \frac{\ell_t}{|\mathcal{D}_t|}$$

that is,

$$P_D(t|c) = \frac{\bar{\ell}_d}{b_t} \cdot P(t|c) \tag{3.4}$$

and, substituting in the left side of (3.3), it becomes:

$$\log\left(\frac{P(t|q,c)}{P(t|c)}\right) = \log\left(\frac{P(t|q,c)}{b_t/\bar{\ell}_d \cdot P_D(t|c)}\right) \tag{3.5}$$

If we were to return to Eq. (3.3), we are forced to consider:

$$P(t|q,c) = \begin{cases} b_t/\bar{\ell}_d & t \in \mathcal{T}_q \\ b_t/\bar{\ell}_d \cdot P_D(t|c) & t \notin \mathcal{T}_q \end{cases} = \begin{cases} b_t/\bar{\ell}_d & t \in \mathcal{T}_q \\ P(t|c) & t \notin \mathcal{T}_q \end{cases}$$

Essentially, we have observed that the IDF, in its generic form of $1/P_D(t|c)$ implies that, when the term is not part of the topic q, we estimate $P(t|q)$ as the probability of the term in the collection ($P(t|c)$) and when the term is part of q we estimate it as $P(t|q) = b_t/\bar{\ell}_d$.

The within-document term frequency (tf_d) in IR models is usually not used in pure form due to its bias towards long documents. The step from tf_d towards a quantification function involves a normalisation component, referred to as K_d. The widely known TF_{BM25} normalisation factor is:

$$K_d = k_1 \cdot \left(1 - b + b \cdot \hat{\ell}_d\right) \tag{3.6}$$

Given that k_1 and b are parameters of K_d, one should use the notation $K_{k_1,b,d}$, but for readability, we simplify the notation to K_d.

We now have the all components to define one of the variants of the RSV of TF-IDF:

$$\text{RSV}_{\text{TF-IDF}}(d,q) = \sum_{t \in \mathcal{T}_q \cap \mathcal{T}_d} \frac{tf_d}{K_d} \log\left(\frac{|\mathcal{D}|}{|\mathcal{D}_t|}\right)$$

and of BM25:

$$\text{RSV}_{\text{BM25}}(d,q) = \sum_{t \in \mathcal{T}_q \cap \mathcal{T}_d} \frac{tf_d}{tf_d + K_d} \log\left(\frac{|\mathcal{D}|}{|\mathcal{D}_t|}\right)$$

Until now we have presented the RSV function of TF-IDF and how this leads to BM25. We follow discussing the derivation of the LM model and highlight some commonality with the derivation of TF-IDF. We remember that the discussion of the derivation of TF-IDF and BM25 was started from Eq. (3.1), where we decomposed the DQI using $P(d,q) = P(d|q)P(q)$. Here we review the decomposition of q as:

$$P(d,q) = P(q|d)P(d)$$

We will then have:

$$P(q|d) = \prod_{t \in \mathcal{T}_q} P(t|d)^{tf_q}$$

and:

$$\log\left(\frac{P(q|d,c)}{P(q|c)}\right) = \sum_{t \in \mathcal{T}_q} tf_q \cdot \log\left(\frac{P(t|d,c)}{P(t|c)}\right) \tag{3.7}$$

Using again the observation formalised in Eq. (3.4):

$$\log\left(\frac{P(t|d,c)}{P(t|c)}\right) = \log\left(\frac{P(t|d,c)}{b_t/\bar{\ell}_d \cdot P_D(t|c)}\right) \tag{3.8}$$

As commonly done in language modelling, we estimate the $P(t|d,c)$ as:

$$P(t|d,c) = \lambda_d\, P(t|d) + (1-\lambda_d)\, P(t|c)$$

and substituting to Eq. (3.8) obtain:

$$\log\left(\frac{P(t|d,c)}{b_t/\operatorname{E}_{\mathcal{D}}[\ell_d]\cdot P_D(t|c)}\right) = \log\left((1-\lambda_d) + \lambda_d \frac{P(t|d)}{b_t/\bar{\ell}_d \cdot P_D(t|c)}\right) \tag{3.9}$$

In LM, when applying a Dirichlet-based mixture (D-LM), the value of λ_d is [ZL01]:

$$\lambda_d = \frac{\ell_d}{\ell_d + \mu}$$

where μ is a parameter of the collection. This parameter could be set based on the average document length $\bar{\ell}_d$. Zhai and Lafferty [ZL01] report values of $\mu \approx 2000$, though they note that the range of optimal parameter values in different collections is quite large (500–10000). Later, Fang, Tao and Zhai [FTZ04] posited that μ needs to be at least as large as the average document length ($\bar{\ell}_d$), so a reasonable value form for λ_d has been:

$$\lambda_d = \frac{\ell_d}{\ell_d + \bar{\ell}_d} = \frac{1}{1 + \frac{\bar{\ell}_d}{\ell_d}}$$

The RSV of D-LM is therefore:

$$\operatorname{RSV}_{\text{D-LM}}(d,q) = \sum_{t\in\mathcal{T}_q} tf_q \cdot \log\left(\frac{1}{1+\ell_d/\bar{\ell}_d}\left(1 + tf_d \frac{|\mathcal{D}|}{\ell_t}\right)\right)$$

In summary, in this section we have explored the derivation of the three most common retrieval models in IR, TF-IDF, BM25, and D-LM, and observed a series of symmetries that we will further explain in Chapter 4.

3.4 Retrievability: a Measure of Accessibility

The retrievability measure defines how likely it is that a document is retrieved [AV08b]. Formally, the retrievability (ret) of a document d with respect to a set of topics $\mathcal{Q}$ submitted to a particular retrieval system, is defined as:

$$\operatorname{ret}(d) = \sum_{q\in\mathcal{Q}} o_q f(d,q,K)$$

where o_q is the probability (also called opportunity) of the topic being chosen, and f is a utility function that measures how retrievable the document d is for a topic q given the rank cut-off K. The function f can be defined in many ways, based on where d has been retrieved in the results of the retrieval system.

Before going into the definitions of the potential utility functions, we define the function *retrieval status rank* (RSR) as follows:

$$\mathrm{RSR}(d,q) = |\{d' \in \mathcal{D} : \mathrm{RSV}(d',q) \geq \mathrm{RSV}(d,q)\}| \tag{3.10}$$

This function returns the rank of a document with respect to a collection of documents $\mathcal{D}$ based on a retrieval status value function (RSV), which defines the scoring schema of a retrieval model.

We now define the first utility function:

$$f(d,q,K) = \begin{cases} 1 & \mathrm{RSR}(d,q) \leq K \\ 0 & \text{otherwise} \end{cases} = [\mathrm{RSR}(d,q) \leq K] \tag{3.11}$$

this function returns 1 if the document is retrieved with rank above or equal to the cut-off K, and 0 if below. In the right-hand side we have the same but using the Iverson bracket notation.

A second utility function, known in the literature as the gravity based utility function [AV08b] is defined as follows:

$$f(d,q,K) = \begin{cases} \frac{1}{\mathrm{RSR}(d,q)} & \mathrm{RSR}(d,q) \leq K \\ 0 & \text{otherwise} \end{cases} = \frac{1}{\mathrm{RSR}(d,q)}[\mathrm{RSR}(d,q) \leq K]$$

where the weights to every document change for every rank at which it has been retrieved. This is similar to the evaluation measure reciprocal rank (RR). However, in Chapter 5, we consider only the first utility function.

3.5 Test Collection-Based Evaluation and Pool Bias

Effectiveness of an IR system refers to the ability of a search engine to satisfy user information needs. A way to know if a retrieval system retrieves information that is relevant or irrelevant to the user is to define, for a set of prescribed topics, which documents are relevant or irrelevant. This, given a collection of documents, defines what in IR is called a test collection, and this pair relationship between topics and documents defining their relevance, relevance assessments. This, in combination with a retrieval evaluation measure is the standard setup to assess the quality of an IR system.

To collect relevance assessments is an expensive process. The current size of collections of documents makes judging every document per topic impossible. Therefore a sampling method has been developed, named pooling, consisting in building a test collection making use of the results produced by various search engines. The most used pooling strategy developed in IR, here called Depth@K, goes as follows: 1) given a certain cut-off K, the first K documents from each result are collected in a pool, 2) every document in the pool is then judged by an assessor. This process is repeated for every topic.

We here formalise this strategy by defining a building set function J and a document scoring function $s : \mathcal{D} \times \wp(\mathcal{R}) \to \mathbb{R}$, used by J to select the top scoring documents. The Depth@K strategy is specified by the following definitions of s, which scores every document d retrieved by the set of pooled runs $\mathcal{R}_p$:

$$s(d, \mathcal{R}_p) = \max_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} (-\rho(d, r))$$

and J, which determines the set of pooled documents:

$$J_{\mathcal{R}_p} = \{d \in r, r \in \mathcal{R}_p : s(d, \mathcal{R}_p) \leq K\} \tag{3.12}$$

We indicate the combined result of the previous scoring function s and the set build function J with the symbol Depth@K, which for the sake of clarity is abbreviated with D^K.

When building a test collection, the main factor under the control of the test collection builder is the number of judged documents. This number depends both on the number of pooled runs and on the minimum number of judged documents per run. To show these dependencies we define a relaxation of the Depth@K strategy that instead of pooling the top K documents retrieved by the pooled runs, it randomly samples from each run a fixed number of documents K. We call this strategy RandomDepth@K, abbreviated as $\not{D}^K$. This defines an s function as:

$$s(d, \mathcal{R}_p) = \max_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} (\epsilon)$$

and a building set function J as defined in Eq. (3.12). The following set inequality shows the relation between these two components:

$$\not{D}^{K+1}_{\mathcal{R}_p} \setminus \not{D}^{K}_{\mathcal{R}_p} \supseteq \not{D}^{K+1}_{\mathcal{R}_p \setminus \{r_p\}} \setminus \not{D}^{K}_{\mathcal{R}_p \setminus \{r_p\}} \tag{3.13}$$

this inequality holds deterministically if the result of the function s on every application of the set building function J is equal, that is the pool returned by repeated calls to the pooling strategy $\not{D}^K$ produces the same initially nondeterministic pool $\mathcal{J}$. Otherwise this inequality holds nondeterministically. Now, on the one hand, given an evaluation measure f, monotonically increasing with the number of relevant documents in $\not{D}^K$ like P@n, we obtain:

$$f(r, \not{D}^{K+1}_{\mathcal{R}_p}) - f(r, \not{D}^{K}_{\mathcal{R}_p}) \geq f(r, \not{D}^{K+1}_{\mathcal{R}_p \setminus \{r_p\}}) - f(r, \not{D}^{K}_{\mathcal{R}_p \setminus \{r_p\}})$$

where r is a run, $\mathcal{R}_p$ is the set of runs used to build the pool $\mathcal{J}$ as returned by $\not{D}^K$, $r_p \in \mathcal{R}_p$, K is the minimum number of documents judged per run, and $f(r, \not{D}^K)$ is the score of the run r evaluated on the pool $\mathcal{J}$ created with $\not{D}^K$. The proof is evident if we observe that: $\not{D}^{K}_{\mathcal{R}_p} \subseteq \not{D}^{K+1}_{\mathcal{R}_p}$, $\not{D}^{K+1}_{\mathcal{R}_p \setminus \{r_p\}} \subseteq \not{D}^{K+1}_{\mathcal{R}_p}$, $\not{D}^{K}_{\mathcal{R}_p \setminus \{r_p\}} \subseteq \not{D}^{K+1}_{\mathcal{R}_p \setminus \{r_p\}}$ and $\not{D}^{K}_{\mathcal{R}_p \setminus \{r_p\}} \subseteq \not{D}^{K}_{\mathcal{R}_p}$. When $r_p = r$, the inequality (Eq. 3.13) defines the *reduced pool* bias. This shows that the bias is influenced by K, the minimum number of judged documents per run, and by

$|\mathcal{R}_p|$ the number of runs. However $\mathcal{R}_p$ is not usually under the control of the collection builder, which makes this bias sometimes inevitable. On the other hand, if f is not monotonically increasing with the number of relevant documents in $\mathcal{J}$ like R@n, this inequality is undefined:

$$f(r, \not{D}^{K+1}_{\mathcal{R}_p}) - f(r, \not{D}^{K}_{\mathcal{R}_p}) \gtreqless f(r, \not{D}^{K+1}_{\mathcal{R}_p \setminus \{r_p\}}) - f(r, \not{D}^{K}_{\mathcal{R}_p \setminus \{r_p\}})$$

We can observe that the bias is not dependent on $\mathcal{R}_p$. Thereby increasing the number of pooled runs or the number of pooled documents do not guarantee a reduction in pool bias.

We define the *pool bias* as *the effect that documents that were not selected in the pool created from the original runs will never be considered relevant* [Lip16]. Therefore, this bias affects the evaluation of a system that has not been part of the pool, with any IR evaluation measures, making the comparison with pooled systems unfair. In the following we provide a formal definition of pool bias:

Definition 3.5.1. The pool bias $\beta_f(r, J_{\mathcal{R}_p})$ is a systematic error we observe when performing a measurement with an evaluation measure f on a run r using the pooled documents resulting from a pooling strategy J with input a set of pooled runs $\mathcal{R}_p$, which may or may not contain information about the run r:

$$\beta_f(r, J_{\mathcal{R}_p}) = f(r, J_{\mathcal{R}_p}) - f(r, \mathcal{I}) \tag{3.14}$$

where $\mathcal{I}$ is the *ideal set of judgements* one would obtain when evaluating the entire collection of documents. We say that $f(r, \mathcal{I})$ is the *true measurement* and $f(r, J_{\mathcal{R}_p})$ is the *biased measurement*.

However, the ideal set of judgements $\mathcal{I}$, in reality does not exist, this bias cannot be computed. Instead, in IR we usually dispose of an approximation of this set $\mathcal{I}$, which in the following we indicate as the ground-truth G. The use of G in the measurement introduces a random error in the observed measurement $f(r, J_{\mathcal{R}_p})$, which we define as follows:

Definition 3.5.2. We define as the *random error*, the difference we would observe on a measure f applied on a run r between the actual measurement and the true measurement:

$$\varepsilon = f(r, G) - f(r, \mathcal{I}) \tag{3.15}$$

where $\mathcal{I}$ is the ideal set of judgements, therefore making $f(r, \mathcal{I})$ the true measurement and G the actual ground-truth, therefore making $f(r, G)$ the *actual measurement*.

This difference is defined as the random error because we have no means of control over it. By its definition, the random error goes to zero, if $f(r, G)$ tends to $f(r, \mathcal{I})$ when the number of judged documents $|G|$ tends to $|\mathcal{I}|$.

From this point, in Chapter 6, we develop pooling strategies that aim to build less biased test collection by containing a more unbiased set of documents.

3.6 Pool Bias Estimators

In the previous section we have seen that the pool bias is an artefact of the pooling method and that it is partially under control of the collection builder. In addition to this approach, we can tackle this problem at evaluation time, that is when a retrieval system gets assessed using a retrieval evaluation measure. We do this by developing pool bias estimators. We analyse here the first pool bias estimator introduced to the IR community. In particular we evaluate this bias estimator for the retrieval evaluation measures P@n. In the following f refers to P@n. We do this here rather than in Chapter 7 in order to establish the notation and to prepare the ground for the detailed discussion there.

Webber and Park [WP09] present a method for the estimation of pool bias that computes the error introduced by the pooling method when one of the pooled runs is removed from the pool. This value is computed for each pooled run using a leave-one run-out approach and then averaged and used as a correction coefficient. Their correction coefficient for a run $r \notin \mathcal{R}_p$ is the expectation:

$$\beta_f(r) = \mathop{\mathrm{E}}_{r' \in \mathcal{R}_p} \left[f(r', J_{\mathcal{R}_p}) - f(r', J_{\mathcal{R}_p \setminus \{r'\}}) \right] \tag{3.16}$$

where $J_{\mathcal{R}_p}$ is the set build function of a generic pooling strategy. This pool bias correctors is later referred in Chapter 7 as BS.

To evaluate this bias estimator we use the mean absolute error (MAE). Eq. 3.16 is simple enough that we can attempt to analytically observe how the estimator behaves with respect to the reduced pool, in the context of a Depth@K pool at vary of K. We identify analytically a theoretical limitation of this approach when used with a Depth@K strategy. The quality of the estimator, in expectation, will not get any better in terms of MAE than the reduced pool when increasing the cut-off value n. This means also that if the MAE of BS is worse than the MAE measured on the reduced pool, this will not be able to recover when increasing n over K.

We start analysing the absolute error (AE) of the BS estimator for a run r:

$$\left| f(r, G) - \left[f(r, D^K_{\mathcal{R}_p}) + \mathop{\mathrm{E}}_{r' \in \mathcal{R}_p} \left[f(r', D^K_{\mathcal{R}_p}) - f(r', D^K_{\mathcal{R}_p \setminus \{r'\}}) \right] \right] \right|$$

where G is ground truth[1], $D^K_{\mathcal{R}_p}$ is the pool constructed using a Depth@K strategy where K is its depth and R_p is the set of pooled runs. We compare it to the absolute error of the reduced pool:

$$\left| f(r, G) - f(r, D^K_{\mathcal{R}_p}) \right|$$

We observe that when the depth of the pool K becomes greater or equal than n, $f(r', D^K_{\mathcal{R}_p})$ becomes constant. For the sake of clarity we substitute it with C_n. We substitute $f(r, G)$, which is also a constant, with C_G. Finally, we also rename the

[1] The ground truth is the pool using the maximum depth available in the test collection

components $a(K) = f(r, D^K_{R_p})$, $b(K) = \mathrm{E}_{r' \in R_p}[f(r', D^K_{\mathcal{R}_p \setminus \{r'\}})]$, and call $g(K)$ the AE of the BS estimator, and $h(K)$ the AE of the reduced pool:

$$g(K) = |C_G - [a(K) + C_n - b(K)]| \quad \text{and} \quad h(K) = |C_G - a(K)|$$

To study the behaviour at vary of K, we define $\dot{f}$ as the finite difference of f with respect to K:

$$\dot{f}(r, D^K_{\mathcal{R}_p}) = f(r, D^{K+1}_{\mathcal{R}_p}) - f(r, D^K_{\mathcal{R}_p})$$

We finitely differentiate the previous two equations, and since both are decreasing functions of K, to see where the margin between the two functions shrinks (the benefit decreases), it is sufficient to study when the inequality $\dot{g}(K) \geq \dot{f}(K)$ holds.

$$\dot{g}(K) = \begin{cases} -\dot{a}(K) + \dot{b}(K), & \text{if } C_G - [a(K) + C_n - b(K)] \geq 0 \\ \dot{a}(K) - \dot{b}(K), & \text{if } C_G - [a(K) + C_n - b(K)] < 0 \end{cases} \quad \text{and} \quad \dot{h}(K) = -\dot{a}(K)$$

Therefore,

$$\dot{g}(K) \geq \dot{h}(K) \text{ iff } \begin{cases} \dot{b}(K) \geq 0, & \text{if } C_G - [a(K) + C_n - b(K)] \geq 0 \\ 2\dot{a}(K) \geq \dot{b}(K), & \text{if } C_G - [a(K) + C_n - b(K)] < 0 \end{cases}$$

While the first condition is always verified ($\dot{b}(K)$ is an average of positive quantities), the second tells us that if $\dot{b}(K)$ is less or equal to $2\dot{a}(K)$ the BS estimator decreases more slowly than the reduced pool. This inequality does not say anything about the behaviour of an arbitrary run r as it can be different for each r. Therefore, we study the MAE using its expectation. We define $\mathcal{R}_G$ as the set of runs of the ground truth G, in which $\mathcal{R}_p \subset \mathcal{R}_G$. Using the law of total expectation we can write:

$$\begin{aligned} &\mathop{\mathrm{E}}_{r \in \mathcal{R}_G}[\dot{b}(K)] = \\ &\quad = \mathop{\mathrm{E}}_{r \in \mathcal{R}_G}\left[\mathop{\mathrm{E}}_{r' \in \mathcal{R}_G \setminus \{r\}}\left[f(r', D^{K+1}_{\mathcal{R}_G \setminus \{r, r'\}}) - f(r', D^K_{\mathcal{R}_G \setminus \{r, r'\}})\right]\right] = \\ &\quad = \mathop{\mathrm{E}}_{r_1, r_2 \in \mathcal{R}_G : r_1 \neq r_2}\left[f(r_1, D^{K+1}_{\mathcal{R}_G \setminus \{r_1, r_2\}}) - f(r_1, D^K_{\mathcal{R}_G \setminus \{r_1, r_2\}})\right] \end{aligned}$$

Using the pool inequality in Eq. 3.13:

$$\begin{aligned} &\mathop{\mathrm{E}}_{r_1, r_2 \in \mathcal{R}_G : r_1 \neq r_2}\left[f(r_1, D^{K+1}_{\mathcal{R}_G \setminus \{r_1, r_2\}}) - f(r_1, D^K_{\mathcal{R}_G \setminus \{r_1, r_2\}})\right] \leq \\ &\quad \leq \mathop{\mathrm{E}}_{r_1 \in \mathcal{R}_G}\left[f(r_1, D^{K+1}_{\mathcal{R}_G \setminus \{r_1\}}) - f(r_1, D^K_{\mathcal{R}_G \setminus \{r_1\}})\right] = \\ &\quad = \mathop{\mathrm{E}}_{r \in \mathcal{R}_G}[\dot{a}(K)] \leq \mathop{\mathrm{E}}_{r \in \mathcal{R}_G}[2\dot{a}(K)] \end{aligned}$$

Therefore, in expectation, at increasing of depth of the pool K, for P@n with $n \geq K$, the MAE of the BS estimator decreases more slowly than the MAE of the reduced pool. The BS estimator does not suffer of the same constraint for R@n since it keeps changing in particular it decreases when K is increased.

CHAPTER 4

Model Bias: Term Frequency Normalisation

Every Information Retrieval (IR) model embeds in its scoring function a form of term frequency (TF) quantification. The contribution of the term frequency is determined by the properties of the function of the chosen TF quantification, and by its TF normalisation. The first defines how independent the occurrences of multiple terms are, while the second acts on mitigating the a priori probability of having a high term frequency in a document (estimation usually based on the document length). New test collections, coming from different domains (*e.g.* medical, legal), give evidence that not only document length, but in addition, verboseness of documents should be explicitly considered. Therefore we propose and investigate a systematic combination of document verboseness and length. To theoretically justify the combination, we show the duality between document verboseness and length. In addition, we investigate the duality between verboseness and other components of IR models.

We test these new TF normalisations on four test collections. These test collections have been chosen based on their statistical properties, which are indicative of the use of the English language in their domains. We do this on a well defined spectrum of TF quantifications, which spectrum covers four of the possible degree of independence for the multiple occurrences of the same term in documents. Finally, based on the theoretical and experimental observations, we show how the two components of this new normalisation, document verboseness and length, interact with each other. Our experiments demonstrate that the new models never underperform existing models, while sometimes introducing statistically significantly better results, at no additional computational cost.

4.1 Introduction

The development of retrieval models is one of the key aspects of research in IR. The IR models arise from experimental observations about the use of the language, predominantly on collections of documents primarily composed of news corpora. Today, with the almost total digitisation of most text produced, it is clear that the textual documents are not just news and that different collections require different approaches [HL13]. Consequently, the field has been driven to deal with different kinds of information types, demonstrated by the creation of new and more domain specific initiatives in the main IR evaluation campaigns: TREC, NTCIR, CLEF, and FIRE. Now, thanks to the observations made in the context of these evaluation campaigns, we are able to revisit some of the original assumptions and extend the models to integrate other collection statistics that reflect the different use of the language in different domains.

Every IR model boils down to a scoring function in which we can distinguish a component that increases with the number of occurrences of a term in a document (a term frequency component, TF) and a component that decreases with the commonality of a term (an inverse document frequency component IDF). In this chapter we focus on the TF component. Its normalisation, first introduced by Robertson et al. [Rob+94] for BM25, and then generalised by Singhal et al. [SBM96] for a generic model, consists in adjusting the within-document term frequency (tf_d) based on the ratio between the document length (ℓ_d) and its expectation ($\mathrm{E}_{\mathcal{D}}[\ell_d]$), called pivoted document length normalisation. The work of Singhal et al. is motivated by the experimental observation that the distribution of length of the retrieved documents should match the distribution of length of the relevant documents. Robertson et al. justify this normalisation, later declared as 'soft' for the mitigation effect provided by the division by the mean, by introducing two contrasting hypotheses [RZ09], named *verboseness* and *multi-topicality*, previously discussed in Section 2.1, in which we have observed that while the first hypothesis suggests a document should be normalised by its length, the second suggests the contrary.

We point out that other collection statistics can be embedded in the TF normalisation of probabilistic models, namely verboseness and burstiness. The former quantifies the repetitiveness of terms in a document. The latter quantifies the repetitiveness of terms across documents. In this chapter we focus primarily on verboseness, but we also make some observations on burstiness and its relation with IDF.

4.2 Motivation

In this section we formally introduce the document verboseness and term burstiness. We then motivate their investigation in IR models.

Verboseness is reflected by the ratio $\ell_d/|\mathcal{T}_d|$: the document length divided by the number of (distinct) terms in the document. The ratio corresponds to *the average tf_d (over all*

terms) in document d:

$$v_d := \mathop{\mathrm{E}}_{\mathcal{T}_d}[tf_d] = \frac{\ell_d}{|\mathcal{T}_d|} \tag{4.1}$$

> A document is verbose if few terms are repeated many times; its domain is $[1, \ell_d]$, 1 for non-verbose (no term occurs more than once), and ℓ_d for maximally verbose (one term is repeated ℓ_d times).

Intuitively, the more verbose (repetitive) a document is, the higher is the chance to find a high tf_d. In other words, a document has a high score just because words are repeated (*e.g.* spamming), and therefore, one wants to demote verbose documents in the ranking.

Burstiness is reflected by the ratio $\ell_t/|\mathcal{D}_t|$, that is the term length in the collection c (or number of occurrences of the term in c) divided by the number of the collection's documents where the term t occurs (aka document frequency). The ratio corresponds to *the average tf_d (over the number of documents where the term t occurs) in collection c*:

$$b_t := \mathop{\mathrm{E}}_{\mathcal{D}_t}[tf_d] = \frac{\ell_t}{|\mathcal{D}_t|} \tag{4.2}$$

> A term is bursty if it occurs in few documents many times; its domain is $[1, \ell_t]$, 1 for a non-bursty term (it occurs only once in each document where it is present), ℓ_t for maximally bursty (all the occurrences are only in one document).

Intuitively, the more bursty a term is, the higher is the chance to find a high tf_d. In other words, a bursty term occurs in fewer documents than a non-bursty (a normal) term, and therefore, one wants to promote documents containing bursty terms.

Instead of verboseness and burstiness, scoring functions most often use normalisation of the tf_d based on the document length ℓ_d (*e.g.* in the TF component of BM25 and in some versions of TF-IDF) .

The contribution of the **document length** is smoothed by its average, that corresponds to *the average ℓ_d (over all the documents) in collection c*:

$$\mathrm{avgdl}(c) = \mathop{\mathrm{E}}_{\mathcal{D}}[\ell_d] = \frac{\ell_c}{|\mathcal{D}|} \tag{4.3}$$

This is then used to calculate the pivoted document length (pivotisation is indicated in this chapter by a hat) as follows:

$$\hat{\ell}_d := \frac{\ell_d}{\mathrm{E}_{\mathcal{D}}[\ell_d]}$$

The $\hat{\ell}_d$ is greater than 1 for relatively long documents (greater than the average document length), and smaller than 1 for short documents (lower than the average document length).

It is surprising that IR models are keen to capture the $\hat{\ell}_d$, but seem to hide away verboseness and burstiness, *i.e.*, there is no parameter explicitly associated with these properties. However we observe that some IR models implicitly use these normalisations.

Following, we observe which IR models capture verboseness and burstiness, and how the parameters can be made explicit or added. As a supportive case we present the verboseness dualities with the concept of burstiness [Roe13] and term length (aka collection frequency).

4.3 Term Frequency Normalisation

Before getting into the details of the duality between document verboseness and length, we formally define the standard pivotisation of document length and introduce the pivotisation of verboseness. To do this we start from the foundation of every IR model: the document-term matrix $A \in \mathbb{N}^{|\mathcal{D}|\times|\mathcal{T}|}$, in which each element is a tf_d indicated here by $a_{d,t}$ for convenience of the notation. For any given matrix, we can define two ways to sum the elements of this matrix; one that fixes a column (a term t) and sums over the rows (the $|\mathcal{D}|$ documents) and one that fixes a row (a document d) and sums over the columns (the $|\mathcal{T}|$ terms). Doing this we calculate two lengths: the length of a document and the length of a term[1], as follows:

$$S_t = \sum_{d\in\mathcal{D}} a_{d,t} = \ell_t \qquad S_d = \sum_{t\in\mathcal{T}} a_{d,t} = \ell_d$$

Now, if we want to compute the average of the values on each row or column, we have to divide the sums obtained above by a *value*. For this *value* we actually have two options: the number of columns or rows, and the number of non-zero elements in the columns or rows. The first is what we would call the *average*, and the second the *elite average*. To give an intuition, think of the question *"What is the average number of Ferraris owned by a person?"*. This question has two answers: we can divide the total number of Ferraris (the sum of the elements on a row/column) by the total number of people on the planet (the number of columns/rows); or, we can consider only those people that have at least one Ferrari and then divide the number of Ferraris by the size of this set of people. The first one is the common average, while the second, obviously, is the *elite* average.

Returning to our document-term matrix, we will denote by a bar ($\bar{a}$) a common average

[1] Although the "length of a term" is non intuitive, here it is meant the L1-length of a vector

and by a breve ($\breve{a}$) an elite average:

$$\bar{a}_t = \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} a_{d,t} = \frac{\ell_t}{|\mathcal{D}|}$$

$$\breve{a}_t = \frac{1}{|\{a_{d,t} : a_{d,t} \neq 0\}|}\sum_{d\in\mathcal{D}} a_{d,t} = \frac{1}{|\mathcal{D}_t|}\sum_{d\in\mathcal{D}} a_{d,t} = \frac{\ell_t}{|\mathcal{D}_t|} = b_t$$

$$\bar{a}_d = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} a_{d,t} = \frac{\ell_d}{|\mathcal{T}|}$$

$$\breve{a}_d = \frac{1}{|\{a_{d,t} : a_{d,t} \neq 0\}|}\sum_{t\in\mathcal{T}} a_{d,t} = \frac{1}{|\mathcal{T}_d|}\sum_{t\in\mathcal{T}} a_{d,t} = \frac{\ell_d}{|\mathcal{T}_d|} = v_d$$

in which we observe that the two elite averages just defined $\breve{a}_t$ and $\breve{a}_d$ correspond to the burstiness b_t as defined in Eq. (4.2) and verboseness v_d as defined in Eq. (4.1).

Considering the remaining elements, $\bar{a}_t$ and $\bar{a}_d$, we can think of them as defining an average document $\bar{d} = \{(t_1, \bar{a}_{t_1}) \dots (t_{|\mathcal{T}|}, \bar{a}_{t_{|\mathcal{T}|}})\}$ and an average term $\bar{t} = \{(d_1, \bar{a}_{d_1}) \dots (d_{|\mathcal{D}|}, \bar{a}_{d_{|\mathcal{D}|}})\}$.

So, now, for each row d and for each column t we have a sum, an average, and an elite average. To obtain a collection-level statistic, we have to aggregate again, calculating sums and averages (common and elite averages are identical now, because all rows and all columns have a non-zero aggregated value).

Doing so, we observe that

$$\breve{\ell}_d := \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} \ell_d \qquad \bar{\ell}_d := \sum_{t\in\mathcal{T}} \bar{a}_t = \frac{\ell_c}{|\mathcal{D}|} \qquad \breve{\ell}_d = \bar{\ell}_d$$

i.e., the average document length $\bar{\ell}_d$ is equal to the sum of the elements of the average document $\bar{d}$.

However, the same observation is not valid for verboseness, because it is an elite average. Instead, we have two notations:

$$\breve{v}_d := \frac{1}{|\mathcal{D}|}\sum_{d\in\mathcal{D}} v_d \qquad \bar{v}_d := \sum_{d\in\mathcal{D}} \bar{a}_d = \frac{\ell_c}{|\mathcal{T}|} \qquad \breve{v}_d \neq \bar{v}_d$$

4.3.1 Duality: Document Verboseness and Length

Recalling the definition of verboseness from Eq. (4.1), it is the average number of times a document's term occurs within the document. To observe the duality of document verboseness and length, Eq. (4.3), let us first define the notation to identify the singleton of a document $d \in \mathcal{D}$ as $\mathcal{D}_d = \{d\}$ and the singleton of a term $t \in \mathcal{T}$ as $\mathcal{T}_t = \{t\}$. Obviously $|\mathcal{D}_d| = |\mathcal{T}_t| = 1$ and therefore we can write $\ell_d = \ell_d/|\mathcal{D}_d|$. Let us now consider the pivoted verboseness and pivoted document length, using the two sets of values defined

above: $\bar{\ell}_d = \breve{\ell}_d$, $\bar{v}_d$ and $\breve{v}_d$:

$$\ddot{\ell}_d = \frac{\ell_d}{\bar{\ell}_d} = \frac{\ell_d/|\mathcal{D}_d|}{\ell_c/|\mathcal{D}|} \qquad \hat{\ell}_d = \frac{\ell_d}{\breve{\ell}_d} = \frac{\ell_d/|\mathcal{D}_d|}{\mathrm{E}_{\mathcal{D}}[\ell_d/|\mathcal{D}_d|]}$$

$$\ddot{v}_d = \frac{v_d}{\bar{v}_d} = \frac{\ell_d/|\mathcal{T}_d|}{\ell_c/|\mathcal{T}|} \qquad \hat{v}_d = \frac{v_d}{\breve{v}_d} = \frac{\ell_d/|\mathcal{T}_d|}{\mathrm{E}_{\mathcal{D}}[\ell_d/|\mathcal{T}_d|]}$$

where we indicate the non-elite pivotisation with a double dots and the elite pivotisation with a hat. The duality is obtained substituting $\mathcal{D} \to \mathcal{T}$ to go from ℓ_d to v_d or $\mathcal{T} \to \mathcal{D}$ to go from v_d to ℓ_d.

The pivoted verboseness of a document is with respect to the space of terms ($\mathcal{T}$), whereas the pivoted document length of a document is with respect to the space of documents ($\mathcal{D}$). One can also show the duality between document verboseness and length based on probabilistic expressions, for the average case:

$$\ddot{\ell}_d = \frac{\ell_d}{\bar{\ell}_d} = \frac{P_L(d)}{P_D(d)} = \frac{\ell_d/\ell_c}{|\mathcal{D}_d|/|\mathcal{D}|} \qquad \hat{\ell}_d = \frac{\ell_d}{\breve{\ell}_d} = \frac{P_L(d)/P_D(d)}{\mathrm{E}_D[P_L(d)/P_D(d)]}$$

$$\ddot{v}_d = \frac{v_d}{\bar{v}_d} = \frac{P_L(d)}{P_T(d)} = \frac{\ell_d/\ell_c}{|\mathcal{T}_d|/|\mathcal{T}|} \qquad \hat{v}_d = \frac{v_d}{\breve{v}_d} = \frac{P_L(d)/P_T(d)}{\mathrm{E}_D[P_L(d)/P_T(d)]}$$

$P_L(d)$ is the location-based probability of a document. Dividing this by the term-based probability of d, $P_T(d) = |\mathcal{T}_d|/|\mathcal{T}|$ yields the pivoted verboseness. Dividing by the document-based probability of d, $P_D(d) = |\mathcal{D}_d|/|\mathcal{D}| = 1/|\mathcal{D}|$, yields the pivoted document length.

The dualities between average document verboseness and average document length justify the combination of parameters as formalised in the definition capturing the normalisation variants of K_d:

Definition 4.3.1 (TF Normalisations K_d)**.** $\ddot{K}_d$: the non-elite normalisation comprises the non-elite pivots $\ddot{\ell}_d$ and $\ddot{v}_d$.

$\hat{K}_d$: the elite normalisation comprises the elite pivots $\hat{\ell}_d$ and $\hat{v}_d$.

The expression pivdl, pivoted document length, denotes one of the two:

$$\text{pivdl} = \begin{cases} \ddot{\ell}_d & \text{non-elite pivot} \\ \hat{\ell}_d & \text{elite pivot} \end{cases}$$

Analogously for pivdv, pivoted document verboseness.

Then, the pivotisation components are defined for the disjunctive (linear) and conjunctive (product) combination of the pivots.

$$\text{comb\_piv}_{b,a,\vee}(d) := 1 - b + b \cdot [(1-a) \cdot \text{pivdl} + a \cdot \text{pivdv}]$$

$$\text{comb\_piv}_{b,a,\wedge}(d) := \left[\text{pivdl}^{1-a} \cdot \text{pivdv}^{a}\right]^{b}$$

The combined pivot becomes part of the usual definition of the normalisation parameter K_d.

$$K_d = k_1 \cdot \text{comb\_piv}(d)$$

It is worth pointing out now that for $b = 0$, or $b = 1$ and $a = \{0, 1\}$ these two combinations are the same. In particular we should note that:

$$\text{comb\_piv}_{0,a,\wedge}(d) = \text{comb\_piv}_{0,a,\vee}(d) = 1$$

is the "traditional" K_d, created ignoring both document length, and verboseness ($b = 0$).

To summarise, there are four variants of the pivotisation factor K_d: non-elite disjunctive denoted as $\ddot{K}_\vee$, non-elite conjunctive denoted as $\ddot{K}_\wedge$, and the respective elite variants $\hat{K}_\vee$ and $\hat{K}_\wedge$. The experiments emphasise the analysis of the behaviour of these four variants.

4.3.2 Example of the Calculation of the Pivotisations

This example illustrates the arithmetic to compute the pivoted document verboseness and length.

Example 4.3.1 (Pivoted Document Verboseness and Length)**.** Assume a document d with $\ell_d = 300$ word occurrences, and $|\mathcal{T}_d| = 150$ distinct words. The verboseness is:

$$v_d = \frac{\ell_d}{|\mathcal{T}_d|} = \frac{300}{150} = 2$$

Let the collection contain $\ell_c = 10^7$ word occurrences, and $|\mathcal{T}| = 10^5$ distinct words. The non-elite average document verboseness is 100, that is, on average, a term occurs $\bar{v}_d = 100$.

The elite average verboseness is the average over the verboseness values of the documents. For example, let $\breve{v}_d = 5/2$ be the elite verboseness.

The pivoted verboseness is the verboseness divided by the average verboseness (*e.g.* the non-elite average verboseness).

$$\ddot{v}_d = \frac{v_d}{\bar{v}_d} = \frac{2}{100} = \frac{1}{50}$$

while the pivoted elite verboseness is the verboseness divided by the elite average verboseness:

$$\hat{v}_d = \frac{v_d}{\breve{v}_d} = \frac{2}{5/2} = \frac{4}{5}$$

Regarding the document length, let $\bar{\ell}_d = 400$ be the average document length. Then, the pivoted document length is:

$$\ddot{\ell}_d = \frac{\ell_d}{\bar{\ell}_d} = \frac{300}{400} = \frac{3}{4}$$

Then we can combine the non-elite pivots, for example, in a disjunctive way:

$$\ddot{K}_{\vee,d} = k_1 \cdot \left\{1 - b + b \cdot \left[(1-a) \cdot \frac{3}{4} + a \cdot \frac{1}{50}\right]\right\}$$

or, the elite pivots in a conjunctive:

$$\hat{K}_{\wedge,d} = k_1 \cdot \left[\left(\frac{3}{4}\right)^a \left(\frac{4}{5}\right)^{1-a}\right]^b$$

The other two variants, elite pivots combined in a disjunctive way ($\hat{K}_{\vee,d}$), and non-elite pivots combined in a conjunctive way ($\ddot{K}_{\wedge,d}$) are left to the reader.

4.3.3 Other Dualities

To strengthen the theoretical justifications, we explore two other dualities, namely the duality between document verboseness and term burstiness, and later in the section the duality between term burstiness and term length. Here, the definitions of the first couple:

$$\begin{aligned} \text{document verboseness: } v_d &:= \ell_d / |\mathcal{T}_d| \\ \text{term burstiness: } b_t &:= \ell_t / |\mathcal{D}_t| \end{aligned}$$

The duality is obtained substituting $\mathcal{D} \to \mathcal{T}$ and $d \to t$ to go from v_d to b_t or $\mathcal{T} \to \mathcal{D}$ and $t \to d$ to go from b_t to v_d. Verboseness is the average term frequency when considering the document length ℓ_d over the set $\mathcal{T}_d$ of terms that occur in the respective document. Burstiness is the average term frequency when considering the number of times the term occurs ℓ_t over the set $\mathcal{D}_t$ of documents in which the respective term occurs.

Furthermore, starting from burstiness and substituting $\mathcal{D} \to \mathcal{T}$, we observe another duality, between term length and burstiness:

$$\begin{aligned} \text{term burstiness: } b_t &:= \ell_t / |\mathcal{D}_t| \\ \text{term length: } \ell_t &:= \ell_t / |\mathcal{T}_t| \end{aligned}$$

Table 4.1: List of all four dual properties.

document verboseness	$v_d := \ell_d / \lvert\mathcal{T}_d\rvert$	
document length	$\ell_d := \ell_d / \lvert\mathcal{D}_d\rvert$	(noting that $\lvert\mathcal{D}_d\rvert = 1$)
term burstiness	$b_t := \ell_t / \lvert\mathcal{D}_t\rvert$	
term length	$\ell_t := \ell_t / \lvert\mathcal{T}_t\rvert$	(noting that $\lvert\mathcal{T}_t\rvert = 1$)

These dualities, based fundamentally on substitutions between the set of documents $\mathcal{D}$ and the set of terms $\mathcal{T}$, were briefly explored in the early 1990s, when Knaus et al. [KMS94], and Amati and Kerpedjiev [AK92] talked about ITF (inverse term frequency) and IDF.

Whereas the IDF is applied for reasoning about the similarity between *documents*, the ITF is applied for reasoning about the similarity between *terms*. Viewing the ITF and IDF together shows that ITF is related to verboseness, and IDF is related to burstiness.

$$\mathrm{ITF}(d, c) := -\log P_T(d|c) \qquad \left(= \log \frac{|\mathcal{T}_c|}{|\mathcal{T}_d|}\right)$$

$$\mathrm{IDF}(t, c) := -\log P_D(t|c) \qquad \left(= \log \frac{|\mathcal{D}_c|}{|\mathcal{D}_t|}\right)$$

Overall, the discussion supports the case to consider verboseness as a document-specific parameter, whereas traditional IR focuses on the pivoted document length, only.

4.3.4 Summary

This section justified the systematic combination of pivoted document length and pivoted verboseness, while placing them in the context of other dualities, involving burstiness and term length. Table 4.1 shows the list of all the explored dualities.

4.4 Probabilistic Derivation of IR Models

To discuss the justification of TF quantifications, we consider the probabilistic derivation of IR models, which we have presented in Section 3.3. In this section we have seen how to derive the TF-IDF from DQI in Eq. (3.1) by expanding $P(d, q)$ as $P(d|q)P(q)$. Next, by making two assumptions of independence, the first on the occurrences of different terms and the second on the occurrences of the same term, we have obtained Eq. (3.2). At this point we have seen that to provide a justification for TF-IDF, we are looking for the bridges to close the gap between the probabilistic roots (assuming independence) and the TF-IDF. Expressed as an equation, we are looking for justifications to transform components of Eq. (3.2) to TF-IDF.

$$\begin{array}{ccc} tf_d & \cdot & \log \frac{P(t|q)}{P(t)} \\ \downarrow & & \downarrow \\ \mathrm{TF}(t, d) & \cdot & \mathrm{IDF}(t) \end{array}$$

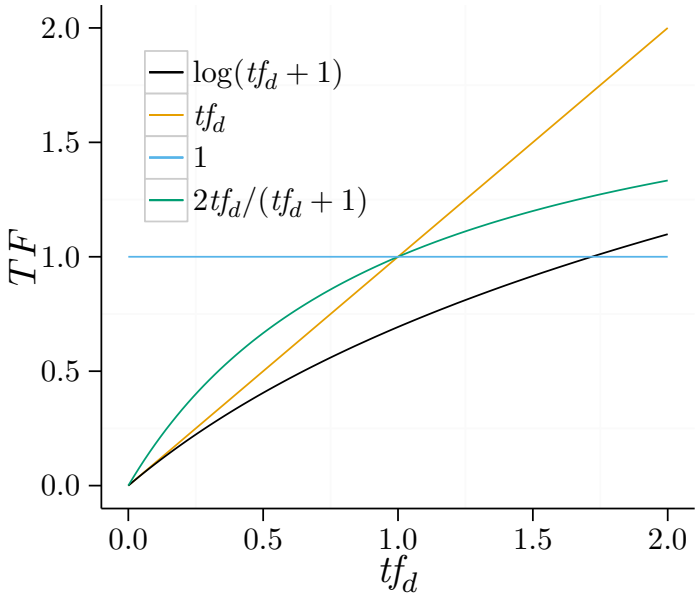

Figure 4.1: TF quantifications when $K_d = 1$.

where TF and IDF are the two components, term frequency and inverse document frequency.

4.4.1 Observations about the TF

Still in Section 3.3, about the within-term frequency (tf_d), we have observed that this is usually not used pure due to its bias towards long documents and we have defined the normalisation component K_d in Eq. (3.6). At this point, we have mentioned that by revisiting the second assumption we can derive TF_{BM25}. In the following we extend this reasoning by considering a well-defined spectrum of TF quantifications [RKB15] defined as follows:

Definition 4.4.1 (TF Quantifications)**.**

$$\text{TF}(t,d) = \begin{cases} tf_d/K_d & \text{TF}_{\text{total}}\text{: independent} \\ \log(tf_d/K_d + 1) & \text{TF}_{\log}\text{: logarithmic} \\ 2 \cdot tf_d/(tf_d + K_d) & \text{TF}_{\text{BM25}}\text{: semi-subsumed} \\ 1/K_d & \text{TF}_{\text{constant}}\text{: subsumed} \end{cases}$$

The shape of the different TF quantifications is shown in Figure 4.1. This spectrum is well-defined because, each of these TFs correspond to an assumption regarding term events [RKB15]. TF_{total} corresponds to assuming independence, and the $\text{TF}_{\log}$ and TF_{BM25} variants assume the occurrences of an event to be dependent.

With this understanding of what the TF stands for, namely a factor modelling a dependence assumption, the role of K_d is to tune the dependence assumption. For $K_d > 1$, that is for long documents, $\mathrm{TF}(t,d)$ decreases, *i.e.*, the dependence increases. This means that in long documents, the multiple term occurrences are more dependent than in short documents. This makes perfect sense when imagining a long document that repeats some terms many times.

This discussion makes evident that it is not just the length of the document that matters. To illustrate, consider two documents of equal length, for example, $\ell_d = 300$ words. The standard K_d will be equal for both documents. One document, however, contains many repetitions of some words (the document is verbose), whereas the other document contains many different words (the document is not verbose). Indeed, it is the verboseness and not simply the document length that leads to high term frequencies, and thus, to dependencies of multiple term occurrences. Therefore, this chapter views K_d as a combination of the pivoted document length (pivdl) and the pivoted document verboseness (pivdv). Where we use a different notation for the pivoted document length to distinguish between its generalisation and its particular implementation ($\hat{\ell}_d$). The following equation indicates the difference between the standard K_d as known for BM25 (as shown in Eq. (3.6)), and the systematic extension proposed and investigated in this thesis:

$$K_d = k_1 \cdot f(\mathrm{pivdl}, \mathrm{pivdv})$$

Here, $f(\mathrm{pivdl}, \mathrm{pivdv})$ is a function combining the two parameters, and this chapter explores both a conjunctive and a disjunctive combination.

4.4.2 Observations about the IDF

Continuing in Section 3.3, we have seen that when answering the question on how to close the gap between $P(t|q)/P(t)$ and IDF, as commonly defined in the literature: IDF=$1/P_D(t)$, we are looking for a justification that leads to Eq. (3.3), where the log probability of a term is equal to 0 when the term does not belong to the set of query terms, and equal to $1/P_D(t)$ when it does. Then, after some mathematical passages, we have obtained Eq. (3.5), in which we have observed that this equation makes burstiness explicit, and in particular its otherwise implicit role in the relationship between IDF and the probabilistic model. At this point, we recall that we were forced to consider:

$$P(t|q,c) = \begin{cases} b_t/\bar{\ell}_d & t \in \mathcal{T}_q \\ b_t/\bar{\ell}_d \cdot P_D(t|c) & t \notin \mathcal{T}_q \end{cases} = \begin{cases} b_t/\bar{\ell}_d & t \in \mathcal{T}_q \\ P(t|c) & t \notin \mathcal{T}_q \end{cases}$$

We now observe that this separation between the cases when $t \in \mathcal{T}_q$ and $t \notin \mathcal{T}_q$ is reminiscent of smoothing in language modelling. We could for instance write:

$$P(t|q,c) = \lambda_q \, b_t/\bar{\ell}_d + (1-\lambda_q)\, P(t|c) \tag{4.4}$$

with

$$\lambda_q = \begin{cases} 1 & t \in \mathcal{T}_q \\ 0 & t \notin \mathcal{T}_q \end{cases}$$

We shall call this an *extreme mixture*.

If we were to continue this inspiration from language modelling, leaving the above for a moment aside, to compute the $P(t|q,c)$ we would estimate it through a linear mixture between the $P(t|c)$ and the $P(t|q)$, as follows:

$$P(t|q,c) = \lambda_q\, P(t|q) + (1-\lambda_q)\, P(t|c) \tag{4.5}$$

This equation is traditionally made because to estimate the probability of a term given the query q, when q is short, is not reliable (even more so than when considering a document d).

Substituting Eq. (4.5) into Eq. (3.5), we have:

$$\log\left(\frac{P(t|q,c)}{b_t/\bar{\ell}_d \cdot P_D(t|c)}\right) = \log\left((1-\lambda_q) + \lambda_q \frac{P(t|q)}{b_t/\bar{\ell}_d \cdot P_D(t|c)}\right) \tag{4.6}$$

where $P(t|q)$ is calculated in a traditional way with a maximum likelihood estimator. However, this would not solve our problem given by the shortness of q. Instead, we need to use the estimation of Eq. 4.4. Then, reintroducing the distinction between $t \in T_q$ and $t \notin T_q$ (*i.e.*, λ_q), we obtain

$$\log\left((1-\lambda_q) + \lambda_q \frac{P(t|q)}{b_t/\bar{\ell}_d \cdot P_D(t|c)}\right) = \begin{cases} \log\left((1-\lambda_q) + \lambda_q \frac{1}{P_D(t|c)}\right) & t \in T_q \\ 0 & t \notin T_q \end{cases}$$

In which if we set $\lambda_q = 1$ then the foreground probability $P(t|c)$ cancels out from the linear mixture assumption ending up with the standard IDF. We shall call this inverse document frequency IDF_L, where L stands for linear mixture, in contrast to the standard IDF (or IDF_E) that is defined by an extreme mixture.

4.4.3 LM and TF-IDF

We already reached with our analysis a point where the border between LM and TF-IDF gets blurred. In this section we recall the derivation of the LM model and highlight some commonality with the derivation of TF-IDF. In Section 3.3, we have shown that to derive LM, we start from Eq. (3.1) like for TF-IDF, but we then decompose $P(d,q) = P(q|d)P(d)$. By doing this we obtain $P(q|d) = \prod_{t\in T_q} P(t|d)^{tf_q}$ and (3.7). Again we observe here that the formalisation in Eq. (3.4), makes explicit burstiness in the Eq. (3.8), as it was in Eq. (3.5).

Analogously for the derivation of TF-IDF for the estimation of $P(t|q,c)$ in Eq. (4.5), as commonly done in language modelling, we have estimated $P(t|d,c) = \lambda_d\, P(t|d) + (1-\lambda_d)\, P(t|c)$, which after substituting to Eq. (3.8) we have obtained (3.9). In which we can now notice the symmetry with Eq. (4.6). We have then observed that in LM, when applying a Dirichlet-based mixture (D-LM), the value of λ_d is [ZL01]:

$$\lambda_d = \frac{\ell_d}{\ell_d + \bar{\ell}_d} = \frac{1}{1 + \frac{\bar{\ell}_d}{\ell_d}} = \frac{\mathrm{pivdl}}{\mathrm{pivdl}+1}$$

Now, just as we did for the normalisation of TF in the TF-IDF derivation, we should consider here not only the presence of the document length but also that of verboseness:

$$\lambda_d = \frac{f(\text{pivdl}, \text{pivdv})}{f(\text{pivdl}, \text{pivdv}) + 1}$$

In a symmetric way we may define for TF-IDF a parameter not strongly dependent by the presence or absence of the term in q (as it was the case in the extreme mixture observed in the previous section) but rather using the Dirichlet based smoothing approach and the maximum likelihood estimation for $P(t|q) = tf_q/\ell_q$:

$$\lambda_q = \frac{f(\text{pivql}, \text{pivqv})}{f(\text{pivql}, \text{pivqv}) + 1}$$

However, the components of this formulation for λ_q are generally not very informative (queries tend to be significantly shorter than documents, and therefore we cannot really talk about the verboseness of a query). Instead, at this place we can exploit the duality of document verboseness and length with term length and burstiness (see Section 4.3.3):

$$\lambda_q = \frac{f(\text{pivtl}, \text{pivtb})}{f(\text{pivtl}, \text{pivtb}) + 1}$$

In summary, in this section we have explored the relationship between TF-IDF and LM. Both models apply a mixture: TF-IDF for estimating $P(t|q, c)$, and LM for estimating $P(t|d, c)$. Moreover, both models involve the component $b_t/\bar{\ell}_d \cdot P_D(t)$ measuring the discriminativeness of the term, where burstiness is made explicit.

The mixture assumptions for $P(t|q, c)$ lead to IDF and it becomes clear why IDF is seen as capturing burstiness in an "implicit" way [CG99]. The Dirichlet-based mixture for $P(t|d, c)$, usually only associated with the document length, is extended with the document verboseness, in this is executed analogously to the way the TF quantification was extended for TF-IDF.

4.5 Experiments

In this section, we first present the material, then the experimental setup. Finally we discuss the results.

4.5.1 Setup and Materials

To test the TF normalisation variants on the different kinds of TF quantifications, we used 4 test collections: TREC HARD 2005, TREC Ad Hoc 8, CLEF eHealth'14, and TREC Web 2002. Details and corpora properties shown in Table 4.2. The test collections have been purposefully chosen with a high degree of variability of $\breve{v}_d$. In this way we can observe the different use of the language in different domains (*e.g.* we observe that in

Table 4.2: Test collection's information about the collection size $|\mathcal{D}|$, number of terms $|\mathcal{T}|$, collection length ℓ_c, average document length $\bar{\ell}_d$, non-elite average verboseness $\bar{v}_d$, elite average verboseness $\breve{v}_d$, average term length $\bar{\ell}_t$, non-elite average burstiness $\bar{b}_t$, and elite average burstiness $\breve{b}_t$. Ordered as indicated by the arrow (↓).

Corpus	EC	Challenge	$\|\mathcal{D}\|$ $\bar{\ell}_d$ $\bar{\ell}_t$	$\|\mathcal{T}\|$ $\bar{v}_d$ $\bar{b}_t$	ℓ_c $\breve{v}_d \downarrow$ $\breve{b}_t$
Aquaint	TREC	HARD'05	1,033,461 273.700 436.995	647,280 436.995 273.700	282,858,247 1.519 1.384
Disks 4&5	TREC	Ad Hoc 8	528,106 295.823 211.699	737,963 211.699 295.823	156,226,039 1.575 1.377
eHealth'14	CLEF	eHealth'14	1,104,298 620.917 308.294	1,103,947 308.294 620.917	685,458,908 1.900 1.349
.GOV	TREC	Web'02	1,214,592 1,457.379 602.645	2,937,251 602.645 1,457.379	1,770,120,644 4.830 3.012

.GOV on average a term is repeated 218% more times than in the Aquaint collection). We developed the tested IR models on the IR platform Terrier[2] 4.2. All the documents have been preprocessed using the English tokenizer and Porter stemmer of the Terrier search engine.

We tested a total of 24 models:

- 16 models based on TF-IDF variants: 4 TF normalisations for each of the 4 TF quantifications defined in Definition 4.4.1. Each model is identified by its TF quantification, $\mathrm{TF}_{\mathrm{total}}$, $\mathrm{TF}_{\log}$, $\mathrm{TF}_{\mathrm{BM25}}$, and $\mathrm{TF}_{\mathrm{constant}}$ and kind of TF normalisation applied: non-elite disjunctive $\ddot{K}_{\vee,d}$, non-elite conjunctive $\ddot{K}_{\wedge,d}$, elite disjunctive $\hat{K}_{\vee,d}$ and elite conjunctive $\hat{K}_{\wedge,d}$.
- 4 models based on D-LM: Each Dirichlet-based mixture is identified by its kind of λ_d normalisation applied: non-elite disjunctive $\ddot{\lambda}_{\vee,d}$, non-elite conjunctive $\ddot{\lambda}_{\wedge,d}$, elite disjunctive $\hat{\lambda}_{\vee,d}$ and elite conjunctive $\hat{\lambda}_{\wedge,d}$.
- 4 models based on the $\mathrm{TF\text{-}IDF}_{\mathrm{L}}$: Each Dirichlet-based mixture is identified by its kind of λ_q normalisation applied: non-elite disjunctive $\ddot{\lambda}_{\vee,q}$, non-elite conjunctive $\ddot{\lambda}_{\wedge,q}$, elite disjunctive $\hat{\lambda}_{\vee,q}$ and elite conjunctive $\hat{\lambda}_{\wedge,q}$. As TF(t,d), we select the non-normalised $\mathrm{TF}_{\mathrm{total}}$.

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

The TF normalisation of each model presents three parameters: k_1, b and the new a introduced in this chapter. Whilst the D-LM and TF-IDF$_{\mathrm{L}}$ based models present two parameters: b and a. Our experiments focus on the parameter a. For k_1 and b, there are two ways of selecting their values: using the standard values from the literature, or identifying trained values. For the models based on the TF-IDF variants, the standard parameters for $\mathrm{TF}_{\mathrm{BM25}}$ are $k_1 = 1.2$ and $b = 0.7$ [Rob+94]. The standard parameter for $\mathrm{TF}_{\mathrm{total}}$ and $\mathrm{TF}_{\mathrm{constant}}$ is $b = 0$ that simplifies K_d to a constant. In this case we set $k_1 = 1$, because it is easy to demonstrate that to change the parameter k_1, as long as $k_1 > 0$, does not change the rank of the retrieved documents for these two quantifications. The same set of parameter values are set for the standard $\mathrm{TF}_{\log}$ ($b = 0$, $k_1 = 1$). For the models based on the D-LM, the standard parameters are $k_1 = 1$ and $b = 0$, which reduces to the standard definition of D-LM [ZL01]. For the models based on the LM variant derived by TF-IDF, the standard parameters are $k_1 = +\infty$, which reduces to standard the TF-IDF model with non normalised $\mathrm{TF}_{\mathrm{total}}$ quantification.

To identify trained values, the parameters of each model have been spanned as follows: $a, b \in [0, 1]$ at steps of 0.1, and $k_1 \in [0, 5]$, from 0 to 1 at steps decided by the function $1/n$ with $n \in \{1, ..., 50\}$, and from 1 to 5 at steps of 0.1. The trained values are obtained by maximising the mean over the topics of the selected evaluation measure. For every model configuration that requires training we perform a 5-fold cross validation.

The IR evaluation measures are AP, NDCG and P@10.

4.5.2 Model Candidates / Structure

Each TF-IDF model candidate is characterised by choosing one of the following options:

1. Pivotisation: elite pivotisation or non-elite pivotisation for document verboseness and length;
2. Normalisation: conjunctive ($\wedge$) or disjunctive ($\vee$) combination of pivoted document verboseness and length into K_d;
3. Quantification: $\mathrm{TF}_{\mathrm{total}}$, $\mathrm{TF}_{\log}$, $\mathrm{TF}_{\mathrm{BM25}}$, or $\mathrm{TF}_{\mathrm{constant}}$;
4. Parameter Settings: standard (S) or trained (T) parameters.

Each D-LM model candidate is characterised by choosing one of the following options:

1. Pivotisation: elite pivotisation or non-elite pivotisation for document verboseness and length;
2. Normalisation: conjunctive ($\wedge$) or disjunctive ($\vee$) combination of pivoted document verboseness and length into λ_d;
3. Parameter Settings: standard (S) or trained (T) parameters.

Each TF-IDF$_{\mathrm{L}}$ model candidate is characterised by choosing one of the following options:

1. Pivotisation: elite pivotisation or non-elite pivotisation for term length and burstiness;
2. Normalisation: conjunctive ($\wedge$) or disjunctive ($\vee$) combination of pivoted document verboseness and length into λ_q;
3. Parameter Settings: standard (S) or trained (T) parameters.

4.5.3 Results

The main results observed are:

1. Document Verboseness vs Length: show a certain independence as shown by the shape of the distributions in Figures 4.2 and 4.3;
2. Pivotisation: for TF-IDF models the elite pivotisation is overall better than the non-elite one; for D-LM models the non-elite pivotisation performs better.
3. Normalisation: for TF-IDF models the combination of document verboseness and length achieves significantly better results, especially when combined in a conjunctive fashion; for D-LM models the combination of document verboseness and length rarely achieves statistical significance;
4. TF-Quantification: TF$_{\mathrm{BM25}}$ appears best, with TF$_{\log}$ close behind;
5. Standard vs Trained parameter: in both parameter configurations, standard and trained, the use of verboseness makes the model achieve better results. On the other hand, the use of term length has most of the time negligible impact.

For each test collections: HARD 2005 in Table 4.3, Ad Hoc 8 in Table 4.4, eHealth'14 in Table 4.5, and Web 2002 in Table 4.6, we present the results obtained with the TF-IDF model variants and the two pivotisations. In these tables we observe each model with either its standard configuration (S), or its trained configuration (T), obtained taking the configuration that maximises the evaluation measure AP. The standard parameters of the normalisations for the TF quantifications: TF$_{\mathrm{total}}$, TF$_{\log}$ and TF$_{\mathrm{constant}}$, have the effect of disabling the normalisation component ($b = 0$). However, for TF$_{\mathrm{BM25}}$ this does not happen. Thereby, we can study the effect of the parameter a in its standard parametrisation. To do this we extract the best value obtained with the standard k_1 and b by selecting the maximum value of the measure AP obtained by varying the parameter a. In case of the trained parameter values instead, for all the TF quantifications, we show in the first row the best result obtained maximising the AP without the use of verboseness in the scoring function ($a = 1.0$), and then we show the result obtained when verboseness is added in the scoring function. The tables distinguish between the conjunctive ($\wedge$) and disjunctive ($\vee$) combinations of document verboseness and length.

Table 4.3: Comparison of the scores obtained with the TF-IDF model candidates with each TF normalisation using the non-elite and elite pivotisation. Column K indicates if standard (S) or trained (T) parameters are used. † indicates statistical significance (paired t-test, $p < 0.05$) against the standard and ‡ against the trained parameters when a is not used.

HARD'05									
P	Q	K	C	k_1	b	a	AP	NDCG	P@10
Non-Elite	TF_{total}	S	-	> 0	0.0	-	0.0721	0.2936	0.1920
		T	-	> 0	0.5	-	0.0900 †	0.3201 †	0.2160
			$\vee$	> 0	0.9	0.9	0.0904 †	0.3223 †‡	0.2200
			$\wedge$	> 0	1.0	0.6	0.0942 †‡	0.3277 †‡	0.2380 ‡
	TF_{log}	S	-	1.0	0.0	-	0.1614	0.4424	0.4160
		T	-	0.2	0.3	-	0.2005 †	0.4799 †	0.4360
			$\vee$	0.2	0.4	0.2	0.2010 †	0.4801 †	0.4320
			$\wedge$	5.0	0.8	0.7	0.2003 †	0.4813 †	0.4400
	TF_{BM25}	S	-	1.2	0.7	-	0.1848	0.4563	0.3660
		T	$\vee$	1.2	0.7	0.6	0.1898	0.4584	0.4280 †
			-	1.5	0.3	-	0.2023 †	0.4797 †	0.4440 †
			$\vee$	1.9	0.4	0.5	0.2030 †	0.4802 †	0.4480 †
			$\wedge$	3.2	0.4	0.3	0.2032 †	0.4812 †	**0.4540** †
	$TF_{constant}$	S	-	> 0	0.0	-	0.0613	0.2436	0.1500
		T	-	> 0	0.1	-	0.0735 †	0.2744 †	0.1620
			$\vee$	> 0	0.2	0.7	0.0742 †	0.2756 †	0.1620
			$\wedge$	> 0	0.1	0.0	0.0740 †	0.2745 †	0.1660
Elite	TF_{total}	S	-	> 0	0.0	-	0.0721	0.2936	0.1920
		T	-	> 0	0.5	-	0.0900 †	0.3201 †	0.2160
			$\vee$	> 0	1.0	0.6	0.0946 †‡	0.3283 †‡	0.2380 ‡
			$\wedge$	> 0	1.0	0.6	0.0942 †‡	0.3277 †‡	0.2380 ‡
	TF_{log}	S	-	1.0	0.0	-	0.1614	0.4424	0.4160
		T	-	0.2	0.3	-	0.2005 †	0.4799 †	0.4360
			$\vee$	0.2	0.6	0.5	0.2013 †	0.4798 †	0.4300
			$\wedge$	0.2	0.8	0.7	0.2003 †	0.4810 †	0.4400
	TF_{BM25}	S	-	1.2	0.7	-	0.1848	0.4563	0.3660
		T	$\vee$	1.2	0.7	0.6	0.2012 †	0.4759 †	0.4480 †
			-	1.5	0.3	-	0.2023 †	0.4797 †	0.4440 †
			$\vee$	1.5	0.5	0.5	0.2034 †	0.4807 †	0.4420 †
			$\wedge$	1.9	0.8	0.7	**0.2037** †	**0.4833** †	0.4400 †
	$TF_{constant}$	S	-	> 0	0.0	-	0.0613	0.2436	0.1500
		T	-	> 0	0.1	-	0.0735 †	0.2744 †	0.1620
			$\vee$	> 0	0.1	0.0	0.0735 †	0.2744 †	0.1620
			$\wedge$	> 0	0.1	0.0	0.0740 †	0.2745 †	0.1660

Table 4.4: Comparison of the scores obtained with the TF-IDF model candidates with each TF normalisation using the non-elite and elite pivotisation. Column K indicates if standard (S) or trained (T) parameters are used. † indicates statistical significance (paired t-test, $p < 0.05$) against the standard and ‡ against the trained parameters when a is not used.

Ad Hoc 8									
P	Q	K	C	k_1	b	a	AP	NDCG	P@10
Non-Elite	TF_{total}	S	-	> 0	0.0	-	0.0635	0.2762	0.1360
		T	-	> 0	0.5	-	0.0977 †	0.3306 †	0.2240 †
			∨	> 0	0.5	0.0	0.0977 †	0.3306 †	0.2240 †
			∧	> 0	1.0	0.5	0.1076 †‡	0.3491 †‡	0.2400 †
	TF_{log}	S	-	1.0	0.0	-	0.1753	0.4568	0.3360
		T	-	0.1	0.3	-	0.2478 †	0.5381 †	0.4280 †
			∨	0.1	0.9	0.9	0.2563 †	0.5415 †	0.4560 †‡
			∧	0.1	0.9	0.5	0.2625 †‡	0.5475 †	0.4620 †‡
	TF_{BM25}	S	-	1.2	0.7	-	0.2433	0.5193	0.4680
		T	∨	1.2	0.7	0.8	0.2614 †	0.5438 †	0.4480
			-	0.6	0.3	-	0.2614 †	0.5447 †	0.4520
			∨	0.6	0.3	0.1	0.2616 †	0.5441 †	0.4620 ‡
			∧	2.7	0.6	0.5	**0.2681** †‡	**0.5523** †‡	**0.4660**
	$TF_{constant}$	S	-	> 0	0.0	-	0.1550	0.4071	0.2060
		T	-	> 0	0.1	-	0.1868 †	0.4387 †	0.3260 †
			∨	> 0	0.1	0.9	0.1880 †	0.4452 †‡	0.3240 †
			∧	> 0	0.2	0.4	0.1922 †	0.4462 †‡	0.3260 †
Elite	TF_{total}	S	-	> 0	0.0	-	0.0635	0.2762	0.1360
		T	-	> 0	0.5	-	0.0977 †	0.3306 †	0.2240 †
			∨	> 0	1.0	0.7	0.1056 †‡	0.3469 †‡	0.2380 †
			∧	> 0	1.0	0.5	0.1076 †‡	0.3491 †‡	0.2400 †
	TF_{log}	S	-	1.0	0.0	-	0.1753	0.4568	0.3360
		T	-	0.1	0.3	-	0.2478 †	0.5381 †	0.4280 †
			∨	0.1	1.0	0.7	0.2521 †	0.5435 †	0.4500 †‡
			∧	0.1	0.8	0.6	0.2562 †‡	0.5474 †‡	0.4540 †‡
	TF_{BM25}	S	-	1.2	0.7	-	0.2433	0.5193	0.4680
		T	∨	1.2	0.7	0.6	0.2535 †	0.5399 †	**0.4700**
			-	0.6	0.3	-	0.2614 †	0.5447 †	0.4520
			∨	0.5	1.0	0.7	0.2638 †	0.5463 †	**0.4700**
			∧	0.6	0.6	0.5	**0.2681** †‡	**0.5524** †‡	0.4680 ‡
	$TF_{constant}$	S	-	> 0	0.0	-	0.1550	0.4071	0.2060
		T	-	> 0	0.1	-	0.1868 †	0.4387 †	0.3260 †
			∨	> 0	0.1	0.4	0.1878 †	0.4418 †‡	0.3320 †
			∧	> 0	0.2	0.4	0.1922 †	0.4462 †‡	0.3260 †

Table 4.5: Comparison of the scores obtained with the TF-IDF model candidates with each TF normalisation using the non-elite and elite pivotisation. Column K indicates if standard (S) or trained (T) parameters are used. † indicates statistical significance (paired t-test, $p < 0.05$) against the standard and ‡ against the trained parameters when a is not used.

eHealth'14									
P	Q	K	C	k_1	b	a	AP	NDCG	P@10
Non-Elite	TF_{total}	S	-	> 0	0.0	-	0.1166	0.3361	0.2640
		T	-	> 0	0.7	-	0.2594 †	0.5206 †	0.5580 †
			$\vee$	> 0	0.8	0.4	0.2610 †	0.5209 †	0.5540 †
			$\wedge$	> 0	1.0	0.4	0.2699 †	0.5322 †	0.5580 †
	TF_{log}	S	-	1.0	0.0	-	0.2106	0.4637	0.4280
		T	-	0.2	0.7	-	0.4222	0.6701 †	0.7960 †
			$\vee$	0.4	0.8	0.5	0.4242	**0.6729** †‡	0.8000 †
			$\wedge$	1.9	1.0	0.4	**0.4260**	**0.6729** †	**0.8040** †
	TF_{BM25}	S	-	1.2	0.7	-	0.3729	0.6310	0.7640
		T	$\vee$	1.2	0.7	0.0	0.3729	0.6310	0.7640
			-	4.5	0.6	-	0.4022 †	0.6595 †	0.7840
			$\vee$	4.5	0.6	0.0	0.4022 †	0.6595 †	0.7840
			$\wedge$	4.5	0.7	0.0	0.4018 †	0.6542 †	0.7880
	$TF_{constant}$	S	-	> 0	0.0	-	0.0474	0.2021	0.1140
		T	-	> 0	0.2	-	0.0755 †	0.2552 †	0.2280 †
			$\vee$	> 0	0.0	0.0	0.0840 †	0.3523 †‡	0.1760 †
			$\wedge$	> 0	0.2	0.2	0.0745 †	0.2551 †	0.2260 †
Elite	TF_{total}	S	-	> 0	0.0	-	0.1166	0.3361	0.2640
		T	-	> 0	0.7	-	0.2594 †	0.5206 †	0.5580 †
			$\vee$	> 0	1.0	0.5	0.2697 †	0.5316 †‡	0.5820 †
			$\wedge$	> 0	1.0	0.4	0.2699 †	0.5322 †	0.5580 †
	TF_{log}	S	-	1.0	0.0	-	0.2106	0.4637	0.4280
		T	-	0.2	0.7	-	0.4222	0.6701 †	0.7960 †
			$\vee$	0.2	1.0	0.4	**0.4239**	0.6713 †	**0.8080** †
			$\wedge$	0.2	1.0	0.4	**0.4239**	**0.6715** †	0.8060 †
	TF_{BM25}	S	-	1.2	0.7	-	0.3729	0.6310	0.7640
		T	$\vee$	1.2	0.7	0.1	0.3742	0.6320	0.7640
			-	4.5	0.6	-	0.4022 †	0.6595 †	0.7840
			$\vee$	5.0	1.0	0.5	0.4079 †‡	0.6635 †‡	0.7900
			$\wedge$	5.0	1.0	0.4	0.4092 †‡	0.6607 †	0.8000
	$TF_{constant}$	S	-	> 0	0.0	-	0.0474	0.2021	0.1140
		T	-	> 0	0.2	-	0.0755 †	0.2552 †	0.2280 †
			$\vee$	> 0	0.2	0.0	0.0755 †	0.2552 †	0.2280 †
			$\wedge$	> 0	0.2	0.2	0.0745 †	0.2551 †	0.2260 †

Table 4.6: Comparison of the scores obtained with the TF-IDF model candidates with each TF normalisation using the non-elite and elite pivotisation. Column K indicates if standard (S) or trained (T) parameters are used. † indicates statistical significance (paired t-test, $p < 0.05$) against the standard and ‡ against the trained parameters when a is not used.

							Web'02		
P	Q	K	C	k_1	b	a	AP	NDCG	P@10
Non-Elite	TF_{total}	S	-	> 0	0.0	-	0.0171	0.1387	0.0260
		T	-	> 0	0.9	-	0.0568 †	0.2642 †	0.0880 †
			∨	> 0	0.9	0.4	0.0577 †	0.2713 †‡	0.0820 †
			∧	> 0	1.0	0.4	0.0563 †	0.2732 †	0.0800 †
	TF_{log}	S	-	1.0	0.0	-	0.0603	0.2719	0.1100
		T	-	0.2	0.8	-	0.1951 †	0.4799 †	0.2420 †
			∨	0.2	0.9	0.6	0.1991 †	0.4803 †	0.2360 †
			∧	0.2	0.9	0.2	0.1974 †	0.4812 †	0.2360 †
	TF_{BM25}	S	-	1.2	0.7	-	0.1948	0.4696	0.2380
		T	∨	1.2	0.7	0.0	0.1948	0.4696	0.2380
			-	4.1	0.7	-	0.2010	0.4777	**0.2520**
			∨	3.1	0.7	0.1	**0.2016**	**0.4816**	0.2420
			∧	5.0	0.8	0.2	0.1923	0.4722	0.2520
	$TF_{constant}$	S	-	> 0	0.0	-	0.0140	0.1514	0.0140
		T	-	> 0	0.1	-	0.0310 †	0.2041 †	0.0500 †
			∨	> 0	0.2	0.3	0.0310 †	0.2008 †	0.0500 †
			∧	> 0	0.1	0.5	0.0311 †	0.1979 †	0.0480 †
Elite	TF_{total}	S	-	> 0	0.0	-	0.0171	0.1387	0.0260
		T	-	> 0	0.9	-	0.0568 †	0.2642 †	0.0880 †
			∨	> 0	1.0	0.4	0.0635 †	0.2860 †‡	0.0940 †
			∧	> 0	1.0	0.4	0.0563 †	0.2732 †	0.0800 †
	TF_{log}	S	-	1.0	0.0	-	0.0603	0.2719	0.1100
		T	-	0.2	0.8	-	0.1951 †	0.4799 †	0.2420 †
			∨	0.1	0.9	0.2	0.1989 †‡	**0.4817** †	0.2360 †
			∧	0.1	0.9	0.2	0.1975 †	0.4816 †	0.2380 †
	TF_{BM25}	S	-	1.2	0.7	-	0.1948	0.4696	0.2380
		T	∨	1.2	0.7	0.0	0.1948	0.4696	0.2380
			-	4.1	0.7	-	0.2010	0.4777	**0.2520**
			∨	3.6	0.8	0.2	**0.2016**	0.4808	0.2460
			∧	3.3	1.0	0.4	0.1966	0.4770	0.2500
	$TF_{constant}$	S	-	> 0	0.0	-	0.0140	0.1514	0.0140
		T	-	> 0	0.1	-	0.0310 †	0.2041 †	0.0500 †
			∨	> 0	0.2	0.3	0.0319 †	0.1988 †	0.0520 †
			∧	> 0	0.1	0.5	0.0311 †	0.1979 †	0.0480 †

Table 4.7: Comparison of the scores obtained with the D-LM model candidates using the non-elite and elite pivotisation. Column K indicates if standard (S) or trained (T) parameters are used. † indicates statistical significance (paired t-test, $p < 0.05$) against the standard parameters.

Ch.	P	K	C	b	a	AP	NDCG	P@10
HARD'05	Non-Elite	S	-	1.0	-	0.1912	0.4680	0.4220
		T	$\vee$	1.0	0.8	0.1970	0.4801 †	**0.4580** †
			$\wedge$	1.0	0.3	**0.1998** †	**0.4806** †	0.4380
	Elite	T	$\vee$	1.0	0.0	0.1912	0.4680	0.4220
			$\wedge$	1.0	0.0	0.1912	0.4680	0.4220
Ad Hoc 8	Non-Elite	S	-	1.0	-	0.2583	0.5420	0.4560
		T	$\vee$	0.9	0.7	**0.2625** †	**0.5481** †	0.4600
			$\wedge$	0.8	0.3	0.2606	0.5448	0.4480
	Elite	T	$\vee$	0.9	0.0	0.2589	0.5410	**0.4680**
			$\wedge$	0.9	0.0	0.2587	0.5415	0.4600
eHealth'14	Non-Elite	S	-	1.0	-	0.3863	0.6444	0.7980
		T	$\vee$	0.8	0.5	0.3965 †	0.6468	0.7900
			$\wedge$	0.7	0.7	**0.4082** †	**0.6616** †	**0.7920**
	Elite	T	$\vee$	0.8	0.0	0.3939 †	0.6467	0.7820 †
			$\wedge$	0.7	0.0	0.3927 †	0.6468	0.7900
Web'02	Non-Elite	S	-	1.0	-	0.1877	0.4617	0.2380
		T	$\vee$	0.8	0.0	0.1984 †	0.4767 †	0.2580
			$\wedge$	0.5	0.1	**0.2039** †	**0.4844** †	0.2600
	Elite	T	$\vee$	0.9	0.3	0.2002 †	0.4785 †	0.2620
			$\wedge$	0.5	0.0	0.2037 †	0.4836 †	**0.2660**

TF_{BM25} works generally better than the other TF quantifications, but not for all test collections. For the test collection eHealth 2014 TF_{log} is better.

We also observe that best configuration is achieved using the elite pivotisation. The conjunctive combination works generally better than the disjunctive case (24 of 32 experiments better than the disjunctive, all 7 unfavourable cases occur when using the Web 2002 test collection).

In Table 4.7, we present the results obtained for every test collection using D-LM with λ_d extended with verboseness. For this model the standard parameter is when $b = 1$, and $a = 0$, which reduces the formula to the standard D-LM without verboseness [ZL01]. This variant is shown on the first row for every test collection. The following rows present the variant of λ_d when combined with verboseness in disjunction and conjunction with non-elite and elite pivots. For this model we observe that the presence of verboseness produces for only one test collection significant improvements. Overall we observe that the non-elite pivotisation should be preferred (all the experiments produce better results

Table 4.8: Comparison of the scores obtained with the TF-IDF$_{\mathrm{L}}$ model candidates using the non-elite and elite pivotisation. Column K indicates if standard (S) or trained (T) parameters are used. † indicates statistical significance (paired t-test, $p < 0.05$) against the standard.

Ch.	P	K	C	b	a	AP	NDCG	P@10
HARD'05		S	-	-	-	0.0721	0.2936	0.1920
	Non-Elite	T	$\vee$	1.0	1.0	**0.0967** †	**0.3329** †	**0.2120**
			$\wedge$	1.0	1.0	**0.0967** †	**0.3329** †	**0.2120**
	Elite	T	$\vee$	1.0	1.0	0.0753 †	0.2994 †	0.1960
			$\wedge$	1.0	1.0	0.0753 †	0.2994 †	0.1960
Ad Hoc 8		S	-	-	-	0.0635	0.2762	0.1360
	Non-Elite	T	$\vee$	1.0	1.0	**0.1500** †	**0.4135** †	**0.2440** †
			$\wedge$	1.0	1.0	**0.1500** †	**0.4135** †	**0.2440** †
	Elite	T	$\vee$	1.0	1.0	0.0688 †	0.2914 †	0.1480 †
			$\wedge$	1.0	1.0	0.0688 †	0.2914 †	0.1480 †
eHealth'14		S	-	-	-	0.1166	0.3361	0.2640
	Non-Elite	T	$\vee$	1.0	1.0	**0.1623** †	**0.4177** †	**0.3220**
			$\wedge$	1.0	1.0	**0.1623** †	**0.4177** †	**0.3220**
	Elite	T	$\vee$	1.0	1.0	0.1231 †	0.3502 †	0.2780
			$\wedge$	1.0	1.0	0.1231 †	0.3502 †	0.2780
Web'02		S	-	-	-	0.0171	0.1387	0.0260
	Non-Elite	T	$\vee$	1.0	1.0	**0.0249** †	**0.1865** †	**0.0460** †
			$\wedge$	1.0	1.0	**0.0249** †	**0.1865** †	**0.0460** †
	Elite	T	$\vee$	1.0	1.0	0.0183 †	0.1456 †	0.0280
			$\wedge$	1.0	1.0	0.0183 †	0.1456 †	0.0280

than the elite one). No difference is observed by using a disjunctive or conjunctive combination of the pivots.

In Table 4.8, we present the results obtained for every test collection using the TF-IDF$_{\mathrm{L}}$ model with λ_q that combines in a LM fashion the term length and burstiness. For this model the standard parameter is when $\lambda_q = 1$, which reduces this IR model to a non TF-normalised TF$_{\mathrm{total}}$-IDF model. This variant is shown on the first row for every test collection. The following rows present the variant of λ_q when combined in disjunction and conjunction with non-elite and elite pivots. We observe that this parametrisation produces significantly better results than the standard case. Also here, as for D-LM, no difference is observed by using a disjunctive or conjunctive combination of the pivots. We also observe that overall the values of the trained parameter a are often equal to 1, which suggests that, for these model variants, the term length does not play an important role in adjusting the document's score. This is a curious behaviour since it is dual to the D-LM model, where the document verboseness did not play an important role either.

Table 4.9: 5-fold cross validation of the trained TF-IDF models candidates observed in Tables 4.3, 4.4, 4.5, and 4.6 for the evaluation measure AP.

P	Q	C	k_1	b	a	HARD'05	Ad Hoc 8	eHealth'14	Web'02
Non-Elite	TF_{total}	-	> 0	$*$	-	0.0873	0.0927	0.2594	0.0543
		$\vee$	> 0	$*$	$*$	0.0873	0.0927	0.2594	0.0543
		$\wedge$	> 0	$*$	$*$	0.0942	0.1058	0.2699	0.0523
	TF_{log}	-	$*$	$*$	-	0.2005	0.2436	0.4136	0.1911
		$\vee$	$*$	$*$	$*$	0.2293	0.2591	0.6081	0.2058
		$\wedge$	$*$	$*$	$*$	0.2257	0.2679	0.5985	0.2048
	TF_{BM25}	$\vee$	1.2	0.7	$*$	0.2228	**0.2718**	0.5679	0.2033
		-	$*$	$*$	-	0.1983	0.2597	0.3987	0.1937
		$\vee$	$*$	$*$	$*$	0.2316	0.2671	0.6050	0.2042
		$\wedge$	$*$	$*$	$*$	0.2006	0.2634	0.3990	0.1892
	$TF_{const.}$	-	> 0	$*$	-	0.0735	0.1868	0.0727	0.0309
		$\vee$	> 0	$*$	$*$	0.1215	0.2087	0.2647	0.0559
		$\wedge$	> 0	$*$	$*$	0.0740	0.1881	0.0735	0.0291
Elite	TF_{total}	-	> 0	$*$	-	0.0873	0.0927	0.2594	0.0543
		$\vee$	> 0	$*$	$*$	0.1495	0.1206	0.5188	0.0965
		$\wedge$	> 0	$*$	$*$	0.0942	0.1058	0.2699	0.0523
	TF_{log}	-	$*$	$*$	-	0.2005	0.2436	0.4136	0.1911
		$\vee$	$*$	$*$	$*$	0.2268	0.2591	0.6070	0.2060
		$\wedge$	$*$	$*$	$*$	0.2265	0.2593	**0.6131**	**0.2062**
	TF_{BM25}	$\vee$	1.2	0.7	$*$	0.2301	0.2573	0.5631	0.2033
		-	$*$	$*$	-	0.1983	0.2597	0.3987	0.1937
		$\vee$	$*$	$*$	$*$	**0.2339**	**0.2718**	0.6028	0.2023
		$\wedge$	$*$	$*$	$*$	0.2010	0.2636	0.4089	0.1926
	$TF_{const.}$	-	> 0	$*$	-	0.0735	0.1868	0.0727	0.0309
		$\vee$	> 0	$*$	$*$	0.1198	0.2075	0.2645	0.0553
		$\wedge$	> 0	$*$	$*$	0.0740	0.1881	0.0735	0.0291

Finally, in Table 4.10 we present the result of the 5-fold cross validation for all the trained case of the these last two models, D-LM and TF-IDF_L.

4.6 Discussion

Finally we make some observations across the experimental results about the behaviour of the parameter a. Before that however, let us make an observation on the nature of the data at our disposal. Figures 4.2 and 4.3 show the distribution of the document verboseness versus document length for the elite and non-elite pivotisations. In both cases we see that verboseness brings additional information compared to document length: the plotted distributions are well spread, away from the first diagonal.

Table 4.10: Comparison of the 5-fold cross validation of the trained D-LM and TF-IDF$_{\mathrm{L}}$ model candidates observed in Tables 4.7 and 4.8.

Challenge	P	C	D-LM	TF-IDF$_{\mathrm{L}}$
HARD'05	Non-Elite	$\vee$	**0.2288**	**0.1523**
		$\wedge$	0.1998	0.0967
	Elite	$\vee$	0.2258	0.1369
		$\wedge$	0.1912	0.0753
Ad Hoc 8	Non-Elite	$\vee$	**0.2679**	**0.1600**
		$\wedge$	0.2539	0.1500
	Elite	$\vee$	0.2653	0.0821
		$\wedge$	0.2556	0.0688
eHealth'14	Non-Elite	$\vee$	0.5740	**0.4545**
		$\wedge$	0.4060	0.1623
	Elite	$\vee$	**0.5769**	0.4116
		$\wedge$	0.3927	0.1231
Web'02	Non-Elite	$\vee$	0.2051	**0.0450**
		$\wedge$	0.2011	0.0250
	Elite	$\vee$	**0.2092**	0.0393
		$\wedge$	0.2010	0.0183

Comparing the two distributions, it is interesting to observe that the non-elite pivotisation is significantly more skewed than the elite one: the x-axis of the left plot has a scale in the (0,0.02) range, while the one on the right plot has a scale that matches the y-scale: (0, 4). This supports and grounds our hypothesis that elite pivotisation should provide us with a better means to balance verboseness and document length with parameter a.

The a parameter controls the contribution of elite pivoted verboseness and elite pivoted document length. When $a < 0.5$, the contribution of the document verboseness is higher then the contribution of the document length, and *vice versa* when $a > 0.5$. Looking at the distribution for the elite pivotisations of the documents, redefining the origin to the point $(1, 1)$ we split the distributions in four quadrants enumerated as in Figure 4.5. We know that whatever a we fix, the documents in the I quadrant will be always demoted to some degree, and in the III quadrant the documents will be always promoted to some degree. So here the question is what happens to the documents in the IV and II quadrant. When to be preferred is the contribution of document verboseness ($a > 0.5$) more documents with low verboseness ($\hat{v}_d < 1$) and high length ($\hat{\ell}_d > 1$) will be promoted against the documents of the IV quadrant, and when preferred is the contribution of the document length ($a < 0.5$) the contrary happens. Therefore, the a values, previously listed, should anti-correlate with the ratio of the number of relevant documents between the II quadrant and the IV quadrant. Here the two lists of values sorted by test collection, of a extracted from Tables 4.3, 4.4, 4.5, and 4.6, for the standard BM25 case with

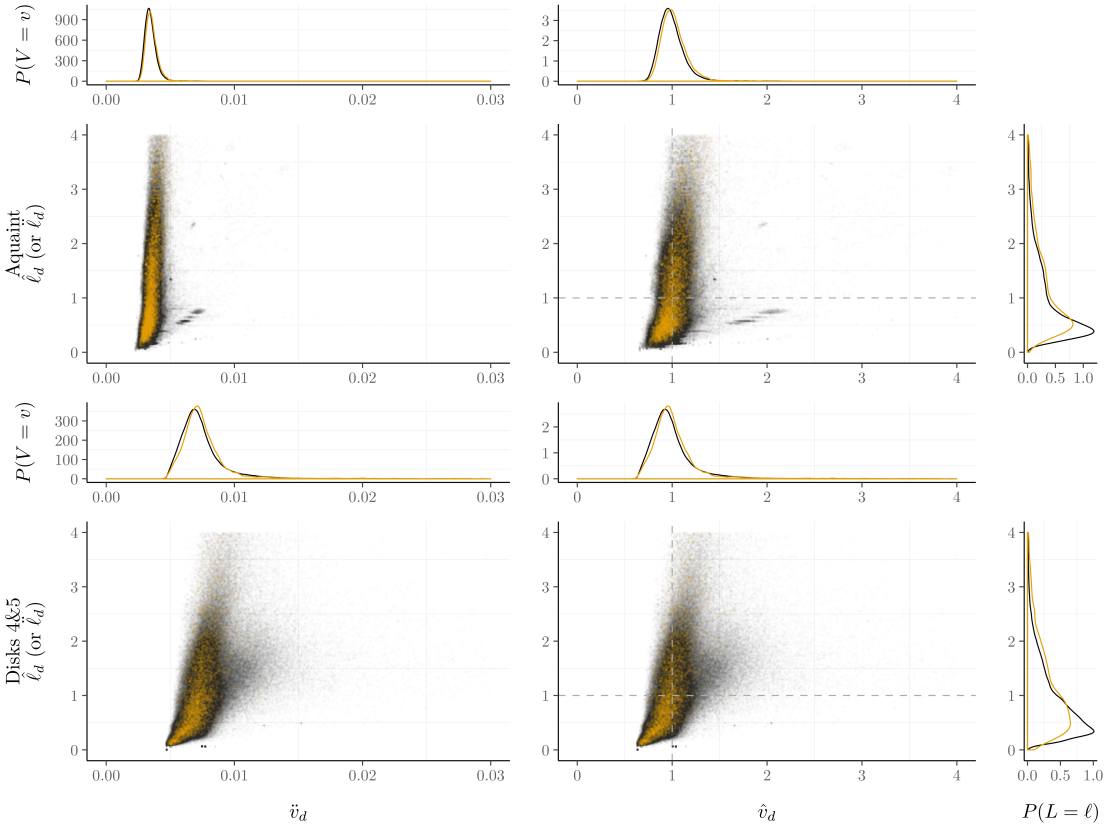

Figure 4.2: Distribution of verboseness on the x-axis and document length on the y-axis of the relevant documents (in gold) and all the documents (in black). The left plot shows the non-elite pivotisation case of verboseness ($\ddot{v}_d$) and length ($\ddot{d}_v$) and the right plot shows the elite pivotisation case of verboseness ($\hat{v}_d$) and length ($\hat{\ell}_d$).

trained a: 0.8, 0.6, 0.4, and 0.0 and ratios: 0.63, 0.86, 1.16 and 4.20, where we observe that they anti-correlate. Therefore if we think that all the documents of the collection should be relevant we should find the a value that mostly balances the proportion of non verbose but long documents with the short but verbose documents. All the test collections but Disks 4&5 have been crawled from the Web. For all of them we can observe that the plots manifest a visible noise. These black dots that could be caused by duplicated documents would not be visible if to be used would be just one of the residual distributions. Especially for eHealth'14 in which is well-known in the eHealth IR community the presence of duplicated documents.

In Tables 4.3, 4.4, 4.5, and 4.6 we observe that the best performing configuration, for both TF_{log} and TF_{total}, uses the trained parameters combined in disjunction, in particular in Table 4.4 these configurations also show statistical significance against both standard configuration and trained configuration when verboseness is not present ($a = 0$). The

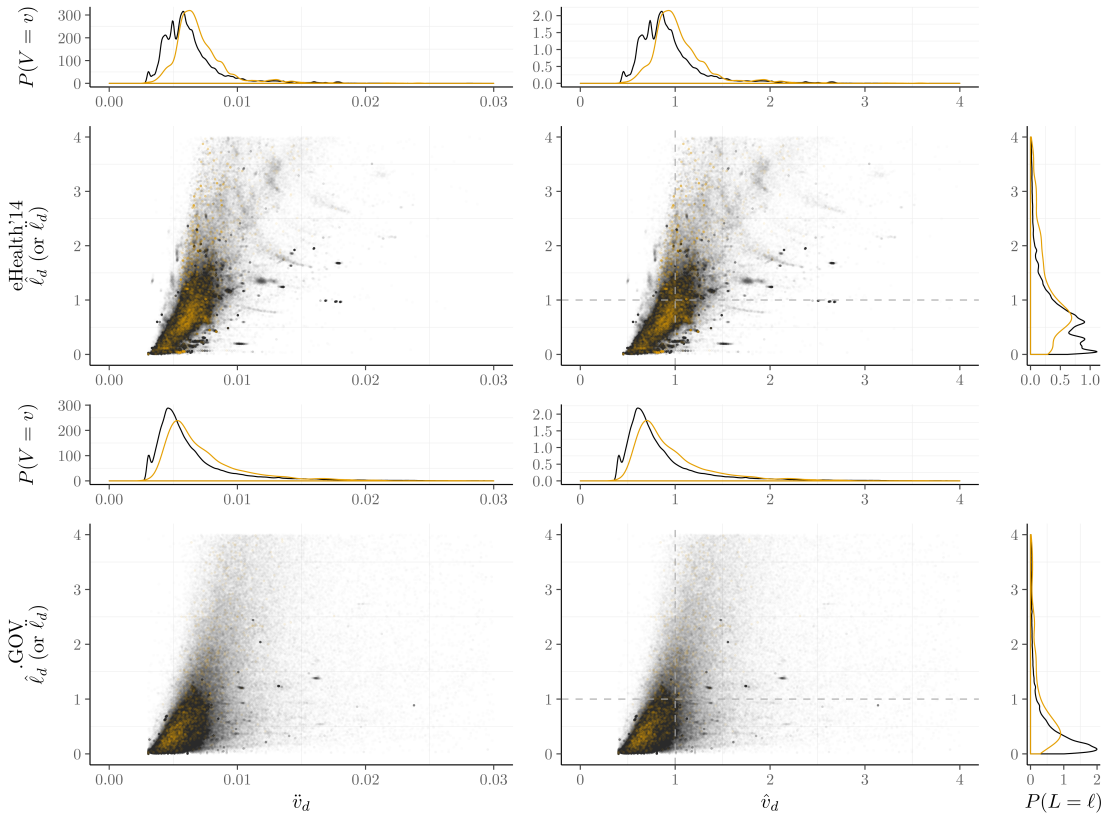

Figure 4.3: Continuation of Figure 4.2 for the rest of test collections.

elite pivotisation performs generally better than the non-elite pivotisation. The best performing configurations are with elite pivotisation, trained parameters in conjunction. We observe also that in general the elite pivotisation weighting role is taken by a ($b = 1$ means that a full document verboseness and length normalisation is applied).

In Figure 4.4 we further analyse the best configuration on a per topic bases. Here, we show the difference in AP between the AP of the trained TF_{BM25}-IDF with verboseness combined in conjunction with elite pivots, and the trained classic TF_{BM25}-IDF. If the difference is positive the variant with verboseness is better than the classic version.

4.7 Summary

This chapter presents an extensive study of TF quantifications and normalisations. The quantifications are with respect to a well-defined spectrum comprising TF_{total}, $\text{TF}_{\log}$, TF_{BM25}, and $\text{TF}_{\text{constant}}$. Each of these TF quantifications reflects a dependence assumption. In particular, TF_{total} and $\text{TF}_{\text{constant}}$ are the extremes of the quantification spectrum, assuming independence for the former and subsumption for the latter. TF_{BM25} is a

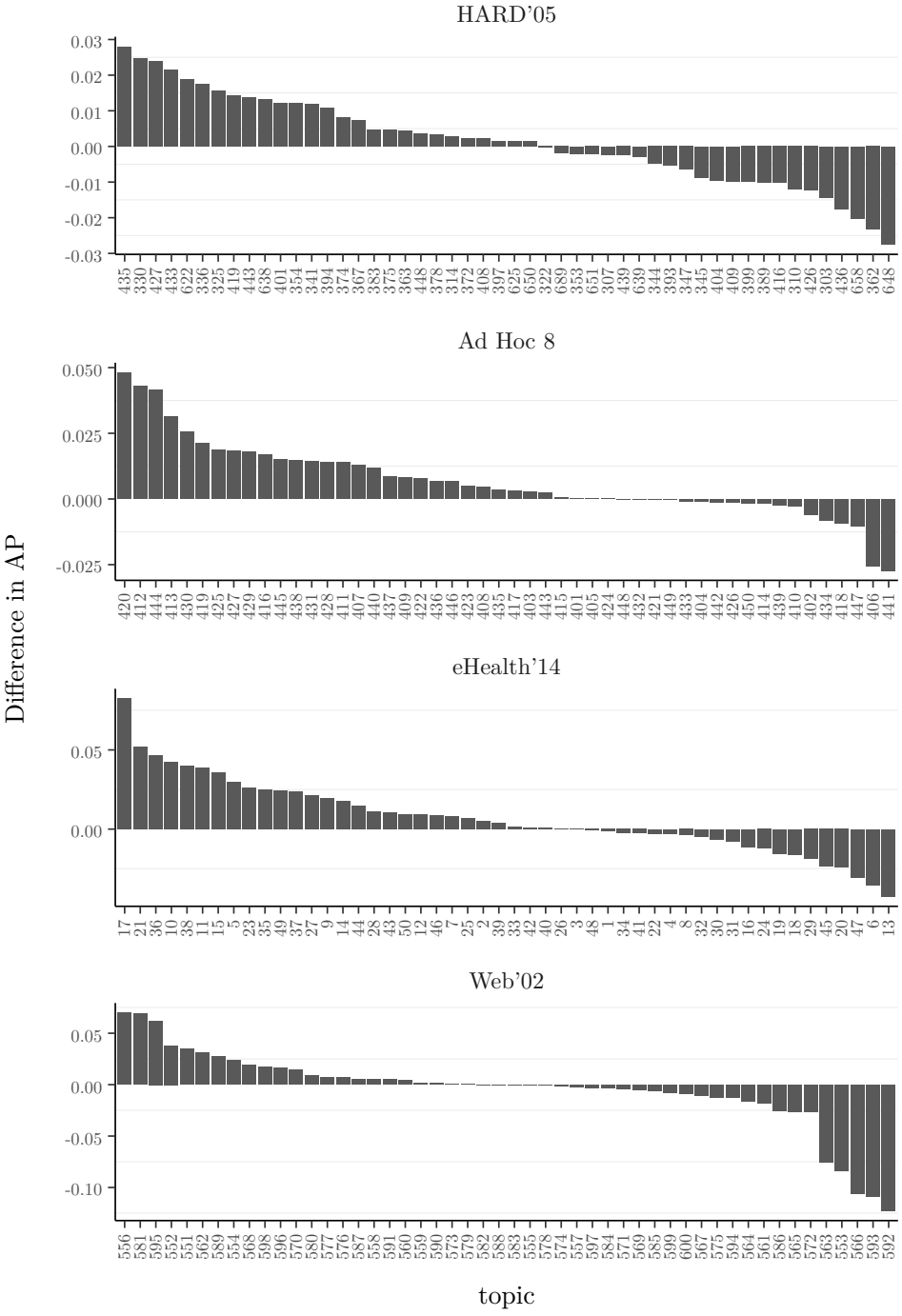

Figure 4.4: Difference on a per topic based between the AP of the trained TF_{BM25}-IDF with verboseness combined in conjunction with elite pivots, and the trained classic TF_{BM25}-IDF. When the difference is positive the variant with verboseness performs better than the classic version.

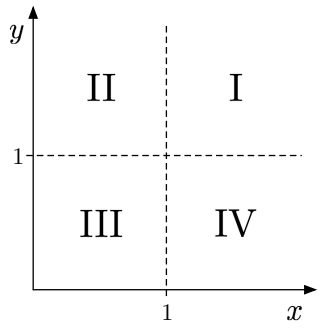

Figure 4.5: Enumeration of the quadrants.

relatively strong dependence assumption, and $\mathrm{TF}_{\log}$ is in the middle between $\mathrm{TF}_{\mathrm{total}}$ and $\mathrm{TF}_{\mathrm{BM25}}$. Each of these quantifications incorporates a *TF normalisation* parameter, usually denoted as K_d.

Whereas current approaches regarding K_d consider only the document length as parameter of K_d, this chapter makes the case for K_d to be a combination of *document verboseness and length*. There are many heuristic options for how to combine the parameters, and this chapter contributes the theoretical foundations leading to a systematic combination of document verboseness and length.

The chapter reports results of an experimental study investigating the effect of various settings of K_d for the four main TF quantifications. The overall finding is that combining document verboseness with document length (either in a conjunctive or disjunctive way) improves retrieval quality when compared to results considering document length only.

We expand this in two directions, first by exploring a similar normalisation in the context of LM and second a similar normalisation in the context of TF-IDF. For the former, we include document verboseness into the Dirichlet smoothing where a non-significant effect is observed, which signifies that document verboseness can be neglected. For the latter, in Section 4.4.3 we have observed the duality between document verboseness and document length on one side, and term burstiness and term length on the other side, and we observed the effect of these normalisations on the query side with respect to LM. Here, significant improvements are observed, however these improvements are obtained primarily by the use of term burstiness, while the term length can be neglected. In both directions improvements are observed given by the new parametrisations, and their results show a dual behaviour, given by the exclusion of document verboseness in the former, and by the exclusion of term length in the latter.

In summary in this chapter we have provided an exhaustive study of normalisation factors in IR probabilistic models using four test collections. Based on the observations made on these test collections, we have made the case that different domains, having different text statistics, can be directly factored into the existing probabilistic models. We have thus

provided a quantification of the various document and term statistics into one factor that balances different prior probabilities that all of these models, more or less explicitly, rely on.

CHAPTER 5

Model Bias: Retrievability

In the previous chapters we have seen that a major issue in Information Retrieval (IR) is the evaluation of retrieval systems. We recall that the general understanding of evaluation is about efficiency and effectiveness. By efficiency is meant all the performing aspects of an IR system (*e.g.* indexing time, memory consumed by its index, response time); and by effectiveness is meant the ability of the IR system to satisfy the information needs of the users within a domain. But we have also seen that while effectiveness and efficiency measures are respectively *system-centric* and *user-centric*, as pointed out by Azzopardi and Vinay [AV08a], both ignore the accessibility of a document. Accessibility studies if a document is or is not accessible by the user through the IR system, which its concept becomes concrete with the definition of a measure of accessibility called retrievability.

Retrievability is a *document-centric* measure that computes the a-priori likelihood that a document in a collection is retrieved, no matter for which topic. This measure quantifies the model bias of an IR system in selecting documents. A quick quantification of this bias may lead to the development more effecting and also efficient IR systems.

In this thesis, we approach the problem of retrievability from an analytical perspective. We start modelling conjunctive and disjunctive queries in a Boolean model. Then, we show that this represents an upper bound on retrievability for all other best-match models. We follow this with an observation of imbalance in the distribution of retrievability, using the Gini coefficient. Simulation-based experiments show the behaviour of the Gini coefficient for retrievability under different types and lengths of queries, as well as different assumptions about the document term-size distribution in a collection.

5.1 Introduction

Retrievability allows the researcher, when comparing the documents of a collection, to understand the a-priori unbalance of retrieval models in retrieving documents. Moreover, recent discoveries have shown that there is a relation between the retrievability and the effectiveness evaluation measures [WA14], allowing a glimpse of the ability of the retrievability analyses to predict the performance of retrieval models without the need of expensive test collections. However, as we have already discussed in Section 1.3.2, retrievability analyses are based on empirical studies and are computationally expensive.

In this chapter we develop a mathematical framework that will allow us to compute the retrievability of a document under certain IR models. We start with the analysis of the perfect-match models (Boolean models). We then bridge the discoveries to the best-match models, thanks to a small theoretical result that states their relationship. Finally, inspired by the experimental discoveries in which it has been pointed out that given an IR model, the length of the document influences its accessibility [WA13], we explore, under some assumptions, to which degree this happens. We do so analytically and through simulations.

5.2 The Retrievability Measure

In this Section we briefly recall the definitions discussed in Section 3.4. The retrievability measure quantifies how likely is that a document is retrieved by a retrieval system. Formally, the retrievability (ret) of a document d with respect to a set of topics $\mathcal{Q}$ submitted to a retrieval system, is defined as:

$$\text{ret}(d) = \sum_{q \in \mathcal{Q}} o_q f(d, q, K) \tag{5.1}$$

where o_q is the opportunity of the topic being chosen, q a topic, and f a utility function that measures how retrievable the document d is for a topic q given the rank cut-off K. It is common to use as utility function f as defined in Eq. (3.11). This function returns 1 if the document is retrieved with rank above or equal to the cut-off K, and 0 if below. However, in this chapter we focus mostly on Boolean models. In this context, the outcome of the system is not a ranked list of documents but rather a set – we can neglect the cut-off K.

In previous retrievability studies, the topics $\mathcal{Q}$ have been generated following one of two strategies:

1. starting from the indexed terms, for single term queries, all the terms that appear in the collection at least 5 times; for bi-term topics, each bi-gram in the collection [AV08b];
2. starting from the documents, extracting all the bi-grams from the collection, and selecting those that appear more than 20 times [BR09; BR10].

In both cases, the adopted procedure is an approximation of the entire set of possible topics.

The study of Boolean models does not require the generation of all the possible topics. We only need some assumption about the class of topic used. There are only a few characteristics of a topic: its length in terms, whether it is a uni-gram or n-gram, and whether it is conjunctive or disjunctive. Therefore, given the type of topics, we set off to analytically calculate the expected $\text{ret}(d)$ for each document that has a specific number of unique terms.

5.3 Retrievability in Perfect-Match Models

A Boolean model is defined in the usual way: it considers relevant (and returns) a document matching the (sub)set of terms in the query. A best-match model is essentially a ranking model applied on top of a Boolean model. Therefore, in this study we do not consider those ranking models which bypass individual terms and do their similarity computation in an abstract semantic space (*e.g.* Latent Semantic Indexing and Latent Dirichlet Allocation). In other words, a best-match model here is any model where the implementation can be done using an inverted list and a weighting method.

5.3.1 The Conjunctive Case

For conjunctive queries all the topic terms are required in order to retrieve a specific document. Given $|\mathcal{T}_q|$, the size of the topic, we can calculate $\text{ret}(d)$ by interpreting the components of Eq. 5.1. The opportunity to use topic q, o_q, is generally fixed to 1 in Azzopardi's and colleagues' work [AV08b; WA15]. In this case, we can focus on the function f. We shall come back to o_q shortly.

The utility function f is essentially an indicator function with codomain in $\{0, 1\}$ if its parameter is false or true. For a Boolean model, the utility function is therefore:

$$f_B(d, q, K) = \begin{cases} 1 & \text{if } \mathcal{T}_q \subseteq \mathcal{T}_d \\ 0 & \text{otherwise} \end{cases} = [\mathcal{T}_q \subseteq \mathcal{T}_d] \tag{5.2}$$

where in the right-hand side of the second equation we use the Iverson bracket, which is a more compacted way of expressing the formula in its left-hand side. The Iverson notation returns 1 if the condition within the squared brackets is true, and 0 otherwise.

For a random document d and topic q, in the case of the Boolean model, the expectation of the utility function is the probability $P(\mathcal{T}_q \subseteq \mathcal{T}_d)$, which can be calculated by considering all possible sets of n terms ($|\mathcal{T}_q| = n$) from the collection dictionary:

$$P(\mathcal{T}_q \subseteq \mathcal{T}_d) = \binom{|\mathcal{T}_d|}{n} \binom{|\mathcal{T}|}{n}^{-1}$$

Therefore, in the case of $o_q = 1$ and by defining $\mathcal{Q}_n$ as the set of topics of length n generated with the term of the collection $\mathcal{T}$ as follows:

$$\mathcal{Q}_n = \{q \in \{\mathcal{T}_{q'} \in \wp(\mathcal{T}) : |\mathcal{T}_{q'}| = n\} \times \mathbb{1}^n\}$$

where $\wp(\mathcal{T})$ is the powerset of $\mathcal{T}$ and $\mathbb{1}^n$ is the set of vectors of ones of size n, we obtain:

$$\text{ret}(d) = \sum_{q \in \mathcal{Q}_n} \binom{|\mathcal{T}_d|}{n} \binom{|\mathcal{T}|}{n}^{-1}$$

given that

$$|\mathcal{Q}_n| = \binom{|\mathcal{T}|}{n}$$

we finally have:

$$\text{ret}(d) = \binom{|\mathcal{T}_d|}{n} \tag{5.3}$$

However, if o_q was considered 1 for practical reasons in simulations, in this theoretical exercise where we already assumed that the vocabulary is limited by the collection vocabulary, we can estimate the probability of a topic of length n as $1/|\mathcal{Q}_n|$. Feeding that into the equation above, we obtain:

$$\text{ret}(d) = \sum_{q \in \mathcal{Q}_n} \binom{|\mathcal{T}|}{n}^{-1} \binom{|\mathcal{T}_d|}{n} \binom{|\mathcal{T}|}{n}^{-1}$$

and following the same motivation as above:

$$\text{ret}(d) = \binom{|\mathcal{T}_d|}{n} \binom{|\mathcal{T}|}{n}^{-1}$$

This is closer to a probabilistic perspective of retrievability, but in what follows we shall continue to use the form of Eq. 5.3 because, on one hand, it is simpler, and on the other hand, it is closer to what related empirical studies have been working with. Now, let us consider all possible topic sizes, that is with n that goes from 1 to $|\mathcal{T}|$ the size of the terms in the document collection. The retrievability of a document d in case of using any combinations of n terms as conjunctive queries is:

$$\text{ret}(d) = \sum_{n=1}^{|\mathcal{T}|} \binom{|\mathcal{T}_d|}{n} = \sum_{n=1}^{|\mathcal{T}_d|} \binom{|\mathcal{T}_d|}{n} = 2^{|\mathcal{T}_d|} - 1 \tag{5.4}$$

The second equality is possible because for any $n > |\mathcal{T}_d|$ the binomial coefficient is 0.

5.3.2 The Disjunctive Case

In this section we explore the case when the terms' topics are submitted in disjunction, which means that at least one topic term is required to retrieve a document. Given n, the size of topic, we can calculate $\text{ret}(d)$ similarly to the conjunctive case above. In this case, the utility function is:

$$f_B(d, q, K) = \begin{cases} 1 & \text{if } \mathcal{T}_q \cap \mathcal{T}_d \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Again, the expectation of this function for a query of size $|\mathcal{T}_q| = n$ is given by the probability:

$$P(\mathcal{T}_q \cap \mathcal{T}_d \neq \emptyset) = \left[\binom{|\mathcal{T}|}{n} - \binom{|\mathcal{T}| - |\mathcal{T}_d|}{n} \right] \binom{|\mathcal{T}|}{n}^{-1}$$

where the first factor is the difference between the number of combinations of topic size n minus the number of combinations of size n that do not retrieve the document d (all the combinations without a document term), and the second factor is the number of possible combinations of topic size n. Consequently for any topic of length n, we have:

$$\text{ret}(d) = \binom{|\mathcal{T}|}{n} - \binom{|\mathcal{T}| - |\mathcal{T}_d|}{n} \tag{5.5}$$

Now, let us consider all possible topic sizes, that is with n that goes from 1 to $|\mathcal{T}|$, the size of the terms in the document collection. The retrievability of a document d in case of using any combinations of n terms as disjunctive queries is:

$$\text{ret}(d) = \sum_{n=1}^{|\mathcal{T}|} \left[\binom{|\mathcal{T}|}{n} - \binom{|\mathcal{T}| - |\mathcal{T}_d|}{n} \right] = (2^{|\mathcal{T}_d|} - 1) \cdot 2^{|\mathcal{T}| - |\mathcal{T}_d|} \tag{5.6}$$

The second equality is obtained by dividing the summation into two sums and simplifying as done in Eq. 5.4. This retrievability, if computed, can easily exceed the precision of a calculator. This, due to the constant factor $2^{|\mathcal{T}|}$, where $|\mathcal{T}|$ in collections of documents is usually in the order of millions. However, if these values are going to be used on a normalised coefficient of imbalance (the Gini coefficient is one of them), these can be dived by $2^{|\mathcal{T}|}$ obtaining a rank equivalent form equal to $1 - 2^{-|\mathcal{T}_d|}$.

5.3.3 Summary

In Table 5.1 we summarize the analysed retrievability cases of a perfect-match model by presenting the formulae to compute the retrievability score of a document based on the query type, conjunctive or disjunctive, and on two specific cases: 1) when considering fixed-length queries of length n, or 2) when considering all queries of length from 1 to $|\mathcal{T}|$, the number of terms in the collection of documents.

Table 5.1: Summary of the analysed retrievability cases for IR perfect-match models based on query type and query-size.

Query Type	Query Size	ret
Disjunctive	n	Eq. (5.3)
	$1, \ldots, \lvert\mathcal{T}\rvert$	Eq. (5.4)
Conjunctive	n	Eq. (5.5)
	$1, \ldots, \lvert\mathcal{T}\rvert$	Eq. (5.6)

5.4 Bridging the Best-Match Models

Now that we know how to compute analytically the retrievability of a document for perfect-match models, we move to the best-match models. For these models the analytical computation of their retrievability becomes more complicated. However, a first observation is given in the theorem and corollary below.

Before going into this first result, we define the utility function f_S for a best-match model like we have done for the utility function of the perfect-match models, f_B in Eq. (5.2). We first recall the definition of the function *retrieval status rank* (RSR), because useful for the proof of the Theorem below, as given in Eq. (3.10) in Section. 3.4:

$$\text{RSR}(d, q) = |\{d' \in \mathcal{D} : \text{RSV}(d', q) \geq \text{RSV}(d, q)\}| \tag{5.7}$$

This function returns the rank of a document with respect to a collection of documents D based on a retrieval status value function (RSV), which defines the scoring schema of a best-match model. Following the definition of f_S:

$$f_S(d, q, K) = \begin{cases} 1 & RSR(d, q) \leq K \\ 0 & \text{otherwise} \end{cases} = [RSR(d, q) \leq K]$$

This function returns 1 if d is among the top K documents of a collection $\mathcal{D}$ ordered in terms of their RSV, otherwise it returns 0. The right-hand side of the second equation is again obtained by using the Iverson bracket.

Theorem 1. *The retrievability of a document under a Boolean retrieval model B is an upper bound for the retrievability of the same document and the same topic types, under any ranking system S.*

Proof. From the definition of retrievability we have:

$$\text{ret}_R(d) \leq \text{ret}_B(d) \Leftrightarrow \sum_{q \in \mathcal{Q}} o_q f_S(d, q, K) \leq \sum_{q \in \mathcal{Q}} o_q f_B(d, q, K)$$

Therefore,

$$\sum_{q \in \mathcal{Q}} o_q f_S(d, q, K) \leq \sum_{q \in \mathcal{Q}} o_q f_B(d, q, K) \Leftrightarrow f_S(d, q, k) \leq f_B(d, q, K)$$

For the conjunctive case, we have that the above is equivalent to:

$$[\text{RSR}(d,q) \leq K] \leq [\mathcal{T}_q \subseteq \mathcal{T}_d]$$

Now, assuming the contrary,

$$[\text{RSR}(d,q) \leq K] > [\mathcal{T}_q \subseteq \mathcal{T}_d] \Leftrightarrow [\text{RSR}(d,q) \leq K] = 1 \ \wedge \ [\mathcal{T}_q \subseteq \mathcal{T}_d] = 0$$

Similarly, for the disjunctive case we would have:

$$[\text{RSR}(d,q) \leq K] > [\mathcal{T}_q \subseteq \mathcal{T}_d] \Leftrightarrow [\text{RSR}(d,q) \leq K] = 1 \ \wedge \ [\mathcal{T}_q \cap \mathcal{T}_d \neq \emptyset] = 0$$

Both contradict our definition of the ranking function in Eq. 5.7 □

Naturally, from this result, one obtains the following corollary:

Corollary 1. *When there is no cut-off ($K = |\mathcal{D}|$), the retrievability of a document in any of the best-match models is equal to its retrievability in the perfect-match model.*

In other words, this corollary says that, when no information about the ranking is taking into account, perfect-match models and best-match models are equivalent.

In the analysis so far we have only considered queries of various sizes, but not with multi-word terms (n-grams). However, since n-grams are essentially terms in themselves, the only thing that would change is the scale of the calculation, rather than the observations about the nature of retrievability itself. We would agree that a more in-depth study into retrievability with n-grams is desirable, if only to prove our statement above, but we do make this simplification for this particular study.

5.5 The Gini Coefficient

The purpose of this section is to observe the distribution of retrievability not over documents but rather over document lengths, counted in unique terms. This is because we want to observe the effect of retrievability on this document lengths distribution, but also because in the current analytical view, two documents with the same number of unique terms are indistinguishable.

To assess the bias of an IR model it is possible to observe the Lorenz curve, which visualises the inequality among documents within a collection. The Lorenz curve has already been introduced in retrievability studies as the cumulative distribution of $\text{ret}(d)$ ordered in non-decreasing order with varying of d. The Gini coefficient was proposed as a way to summarise with a single value the amount shown by the Lorenz curve [AV08b; WA15]. It is defined as:

$$G = \frac{n+1}{n} - \frac{2\sum_{i=1}^{n}(n+1-i)y_i}{n\sum_{i=1}^{n} y_i} \tag{5.8}$$

where y_i is the population indexed in non-decreasing order ($y_i \leq y_{i+1}$), and n is the size of the population. The domain of this function is $[0, 1]$. A Gini coefficient of 1 indicates maximal inequality, where all the documents are irretrievable but one. A Gini coefficient of 0 indicates perfect equality, where all the documents have the same likelihood of being retrieved. For these reasons we consider this as a measure of *fairness*.

As we have observed in the previous analysis, $\text{ret}(d)$ for a perfect-match model is a function of the number of unique terms in the document. It can be in fact shown that $\text{ret}(d)$ is monotonically increasing with $|\mathcal{T}_d|$. Therefore, given a distribution of document lengths (based on unique terms) in a collection of documents, with probability mass function $u(s) = P(S = s)$, where S is the length of a document counted in unique terms, and $n = |\mathcal{D}|$, the numerator in Eq. 5.8 is:

$$\sum_{i=1}^{|\mathcal{D}|}(|\mathcal{D}| + 1 - i)\text{ret}(d_i) = \sum_{i=1}^{\infty}\sum_{j=1}^{\phi(i)}\left[|\mathcal{D}| + 1 - \left(j + \sum_{s=1}^{i-1}\phi(s)\right)\right]\text{ret}(d_{\phi(i)}) \tag{5.9}$$

where $\phi(i) = \lfloor |\mathcal{D}|u(i) + 1/2 \rfloor$ is the expected number of documents of length i, and $d_{\phi(i)}$ is a document of length i. The denominator is substituted by:

$$\sum_{i=1}^{|\mathcal{D}|}\text{ret}(d_i) = \sum_{i=1}^{\infty}\sum_{j=1}^{\phi(i)}\text{ret}(d_{\phi(i)}) \tag{5.10}$$

Substituting Eq. (5.9) and Eq. (5.10) to Eq. (5.8), and simplifying, we obtain:

$$G = \frac{|\mathcal{D}| + 1}{|\mathcal{D}|} - \frac{2\sum_{i=1}^{\infty}\left[|\mathcal{D}| + \frac{1}{2} - \left(\sum_{s=1}^{i-1}\phi(s) + \frac{\phi(i)}{2}\right)\right]\phi(i)\text{ret}(d_{\phi(i)})}{|\mathcal{D}|\sum_{j=1}^{\infty}\phi(j)\text{ret}(d_{\phi(j)})}$$

This, with respect to the original formulation in Eq. (5.8), computes the Gini coefficient giving as input the distribution of document lengths. This formulation is useful for the analysis we will conduct in the next section.

5.6 Discussion

With this definition of the Gini coefficient (G), we can now observe the effects of the query term-size and distribution of document lengths in the collection. We do this by exploring three different shapes of distributions: the uniform distribution in Figure 5.1, the Poisson distribution in Figure 5.2, and the Gamma distribution in Figure 5.3. We use them to test the behaviour of the Gini coefficient on these three distributions to observe how the fairness in terms of retrievability changes when varying a property of the test collection, the document lenght distribution. For each shape of distribution, three instances are drawn by keeping the average document length constant (50, 150, 250), their variances increase but without a prefixed criteria.

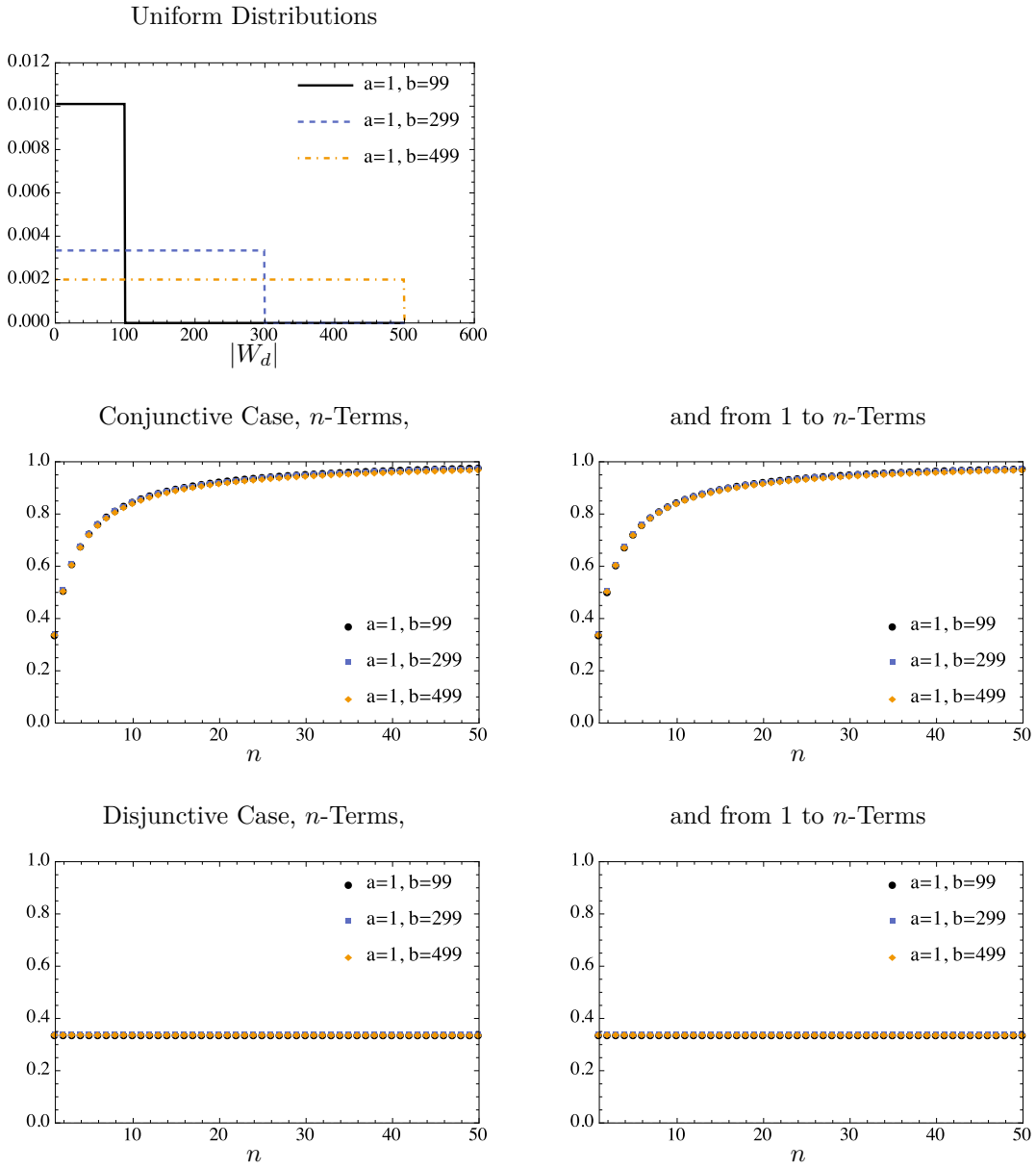

Figure 5.1: Gini coefficient, given a uniform term-size distribution of a collection of documents vs. different cases with varying of n of n-term queries. The top row shows the three term-size distributions tested. a and b are the parameters of the uniform distribution. The middle row shows the two conjunctive cases, with n-terms queries and with n-terms queries from 1 to n. The last row shows the disjunctive case, with n-terms queries and with n-terms queries from 1 to n

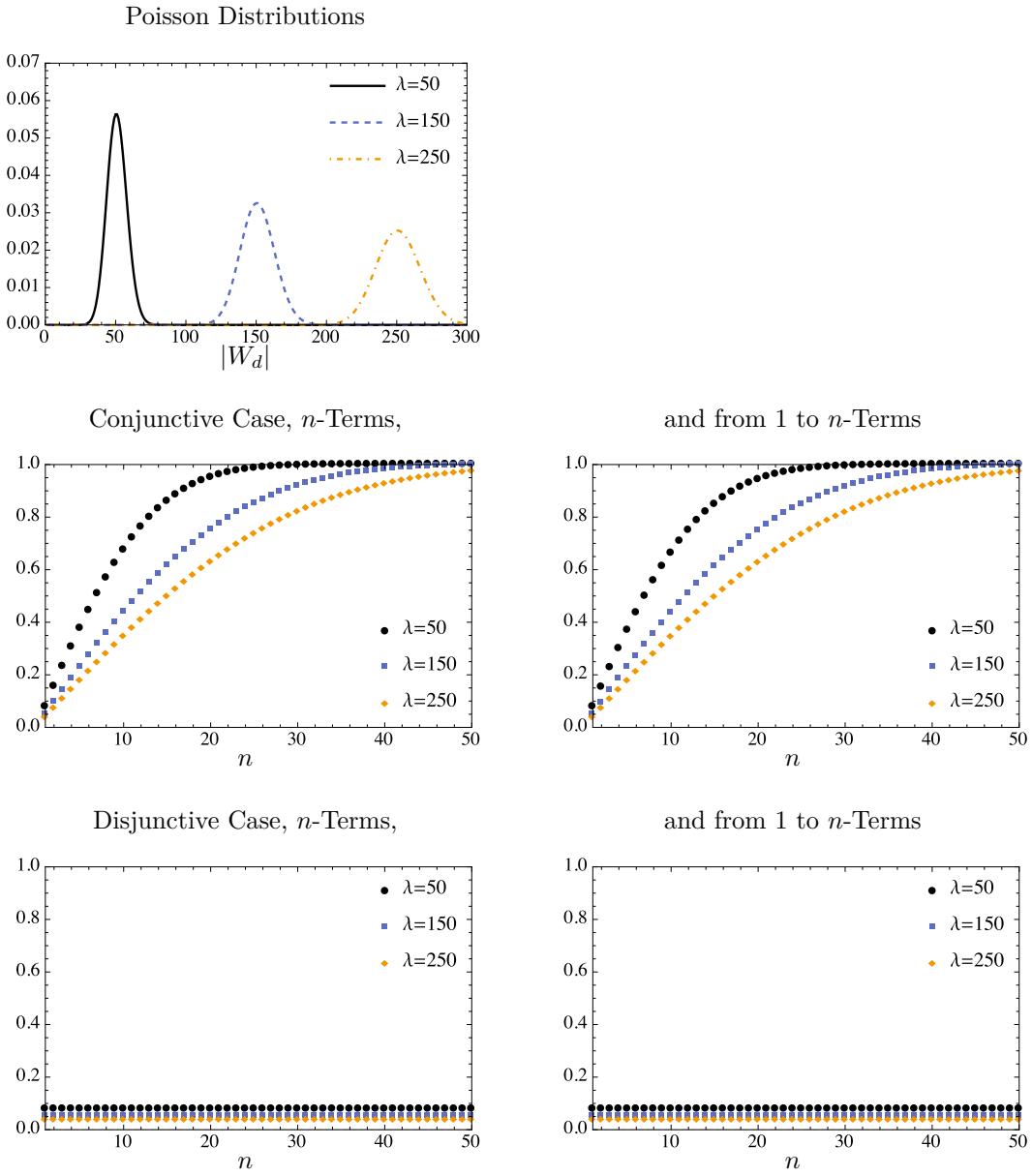

Figure 5.2: Gini coefficient, given a Poisson term-size distribution of a collection of documents vs. different cases with varying of n of n-term queries. The top row shows the three term-size distributions tested. λ is the parameter of the Poisson distribution. The middle row shows the two conjunctive cases, with n-terms queries and with n-terms queries from 1 to n. The last row shows the disjunctive case, with n-terms queries and with n-terms queries from 1 to n

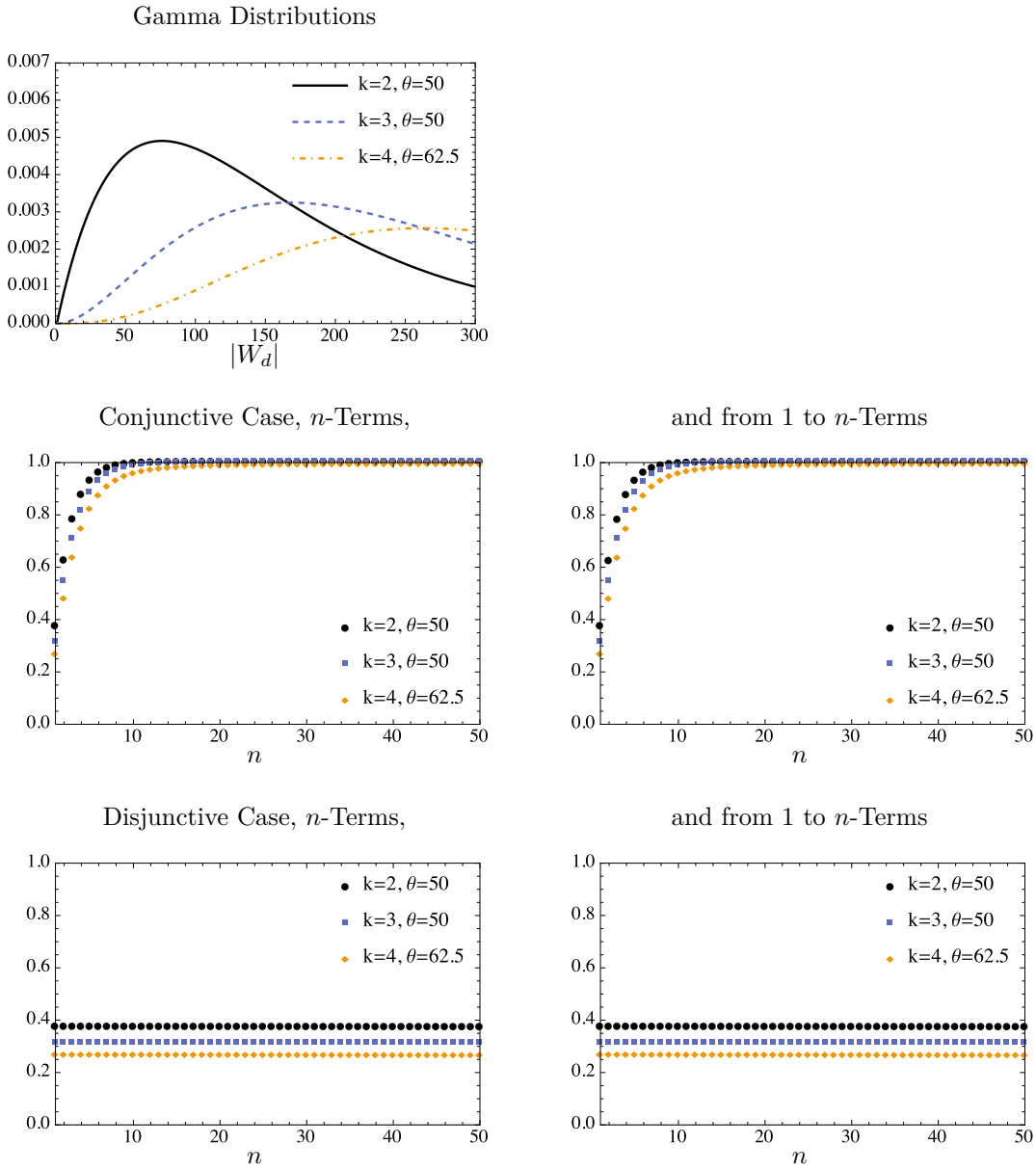

Figure 5.3: Gini coefficient, given a Gamma term-size distribution of a collection of documents vs. different cases with varying of n of n-term queries. The top row shows the three term-size distributions tested. k and θ are the parameters of the Gamma distribution. The middle row shows the two conjunctive cases, with n-terms queries and with n-terms queries from 1 to n. The last row shows the disjunctive case, with n-terms queries and with n-terms queries from 1 to n

We first compare the behaviour of the Gini coefficients between the conjunctive and disjunctive case, because they are similar across the distribution shapes. We observe that when using conjunctive queries the Boolean system becomes more biased when the topic term-size n increases, until converging to 1, the maximum value of the Gini coefficient – point of maximum imbalance. While when using disjunctive queries the bias decreases slightly with increasing of n, making the Boolean system more fair. When comparing between the two cases, fixed term-size n queries and queries with term-sizes from 1 to n, we do not observe any visible difference between the two. This suggests that when studying retrievability it is sufficient to explore the former case, the fixed term-size n queries case, because it is as informative as considering all the query term-sizes from 1 to $|\mathcal{T}|$ and easier to calculate.

We now observe the behaviour of the Gini coefficient with different distribution shapes. A property of the uniformly distributed document term-size case, in Figure 5.1, is that it guarantees an equal number of documents for every term-size $|\mathcal{T}_d|$. However, we observe that the Gini coefficient values computed on the three instances of the distribution are insensible to this property. They are all almost overlapping – increasing the mean and variance of the distribution does not change the fairness of the Boolean retrieval model.

A more localised distribution like the Poisson distribution instances produce an observable effect when measuring the Gini coefficient on the conjunctive case. A test collection with longer documents are overall more retrievable than shorter documents and therefore the Boolean model in the conjunctive case behaves more fairly. This is also observable in the case of the Gamma distribution where the instance with the highest mean and variance is more fair for the conjunctive case. However, for this distribution shape, the three instances produce different results in the disjunctive case. The instance that makes a Boolean model more fair is the one with an higher mean and variance.

Juxtaposing these distribution shapes, we observe that the worst shape for the conjunctive case is the uniform distribution, followed by the Gamma distribution. This is because they converge to 1 more quickly than the instances of Poisson distribution. While for the disjunctive case the worst distribution shape is again the uniform distribution, followed by the Gamma distribution. The Poisson distribution is again the best case. However, for the case of the Gamma distribution, having a higher mean makes the disjunctive case better than the disjunctive case for the uniform distribution.

In Figure 5.4 we show how the retrievability changes when varying n in the four cases for a document term size $|\mathcal{T}_d| = 50$. In particular we observe that for the conjunctive case by considering only queries of size n $ret(d)$ increases until reaching a maximum and then decreases to zero. This happens as soon as the topic size becomes larger than the document size – the query contains at least a term not contained in the document. When considering all query sizes from 1 to n, $ret(d)$ is equal to the cumulative sum of the retrievability as measured in the previous case, therefore since this previous function converges to zero this cumulation converges with an asymptote. For the disjunctive case we observe that the retrievability increases exponentially. This happens in the same way in both cases, for n and for the cumulative sum from 1 to n.

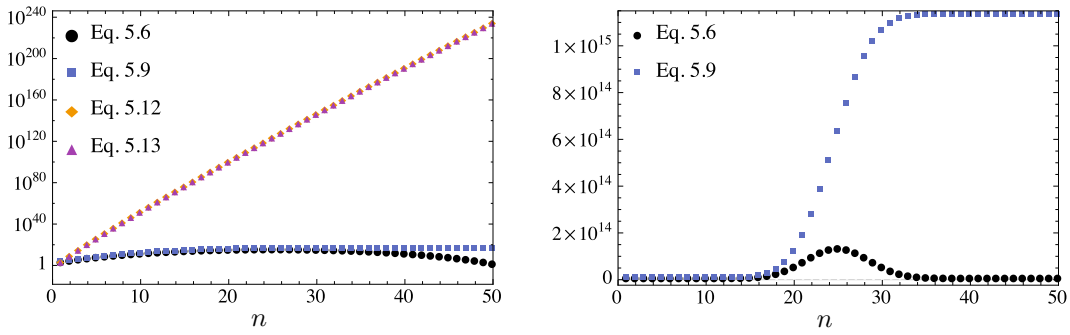

Figure 5.4: Retrievability ret(d) in the four cases for a document of $|\mathcal{T}_d| = 50$ with varying of n query terms. The y-axis of the first plot to the left is in log-scale; the second plot shows the same but only for the conjunctive case but with y-axis in linear-scale.

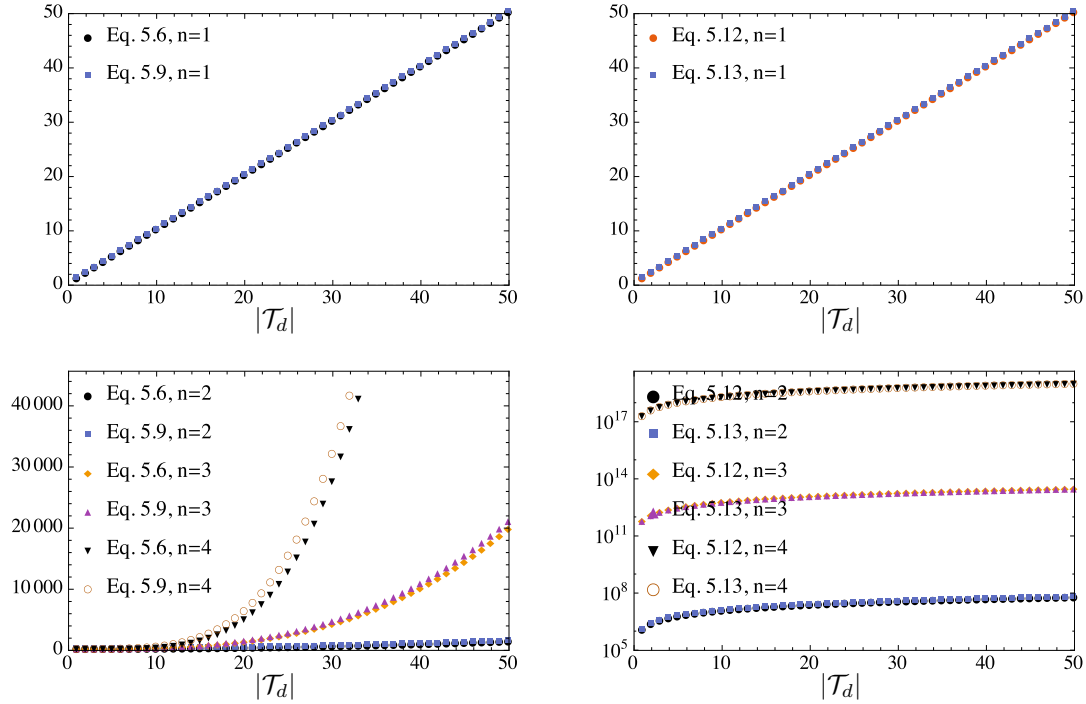

Figure 5.5: The first row shows how the retrievability ret(d) varies for the four cases with varying of $|\mathcal{T}_d|$ for single term queries, the first plot on the left for the conjunctive case, and the second plot on the right for the disjunctive case; The two plots on the second row are similar to the two plots in the first row but with n equal to 2, 3, and 4, the first plot on left for the conjunctive case, the second plot on the right for the disjunctive case.

In Figure 5.5 we show how the retrievability changes when varying the term-size of a document $|\mathcal{T}_d|$. We provide examples for different ns (1, 2, 3, and 4). We observe that if $n = 1$ the retrievability is the same, regardless of the case, and increases linearly. This is also demonstrated by the previous Figure 5.4 by observing that they all start from the same point. We observe that when $n > 1$ the retrievability of the various cases increases much faster than linear, in particular the disjunctive case increases faster than the conjunctive case.

5.7 Summary

We have shown that retrievability for the Boolean model can be approached analytically. While in this study we considered different probability distributions for document lengths, the method can also be used in the presence of an actual test collection to calculate accessibility without the need for generating large sets of synthetic queries. Furthermore, the relationship between document term-size and retrievability, even in this particular retrieval model, may provide insights into new normalisation factors for best-match models.

CHAPTER 6

Selection Bias: Pooling Method

The empirical nature of Information Retrieval (IR) mandates strong experimental practices. A keystone of such experimental practices is the Cranfield/TREC evaluation paradigm. Within this paradigm, the collection of relevance judgements has been the subject of intense scientific investigation. This is because, on one hand, consistent, precise, and numerous judgements are keys to reducing evaluation uncertainty and test collection bias; on the other hand, however, relevance judgements are costly to collect. The selection of which documents to judge for relevance, known as the *pooling method*, has therefore a great impact on IR evaluation. In this chapter we focus on the bias introduced by the pooling method, known as *pool bias*, which affects the reusability of test collections, in particular when building test collections with a limited budget. We formalise and evaluate a set of 22 pooling strategies based on: traditional strategies, voting systems, retrieval fusion methods, evaluation measures, and multi-armed bandit models. To do this we run a large-scale evaluation by considering a set of 9 standard TREC test collections, in which we show that the choice of the pooling strategy has significant effects on the cost needed to obtain an unbiased test collection. We also identify the least biased pooling strategy in terms of pool bias according to three IR evaluation measures: AP, NDCG, and P@10.

6.1 Introduction

The effectiveness of an IR system is evaluated with the use of test collections. A test collection consists of a collection of documents, a set of topics (expressions of information needs), and a set of relevance assessments, which express the relevance relationship between topics and documents.

This set of relevance assessments is, in the vast majority of cases, by necessity a very small subset of the Cartesian product between the set of documents and the set of topics. If we were to consider even a relatively small test collection, with 500,000 documents and 50 topics (this is approximately the size of the Ad Hoc 8 test collection [VH99b]), the total relevance judgements to be made would be 5×10^6. At a very optimistic rate of 120 seconds/judgement, this represents the equivalent of 95 years of work for one person [KZ14]. Therefore, since the very beginning of standardised IR benchmarking at the Text Retrieval Conference (TREC) in the early 1990s, *"pooling"* has been used to reduce the number of judgements, while still preserving the ability of the benchmark to distinguish between two or more retrieval engines [VH05].

Since the proposal of the Depth@K pooling strategy, substantial research effort has gone into improving the evaluation procedures, reducing the associated costs, increasing the reliability of test collections, and devising alternative pooling strategies [San10] (*e.g.* [CPC98; Büt+07; MWZ07; WP09]).

Reliability is understood here as the opposite of *bias* in a test collection. Since the early days of pooling, it has been observed that, in the absence of sufficiently numerous and diverse systems, there is a risk that the identified set of relevant documents will be so limited that future systems, retrieving a new set of relevant (but at this point unjudged) documents, will be considered ineffective because they do not primarily find the set of relevant documents found by the systems that were originally pooled [Rob08]. Incomplete judgements, *i.e.*, the presence among the retrieved results of unjudged documents, have little impact on the small newswire collections used in early TREC years; however, they do lead to uncertainty in the evaluation quality on larger, web-size collections, thus rendering evaluation on these collections invalid [Buc+07; Zob98].

In this chapter, we focus on reducing the bias at test collection build time, exploring different pooling strategies to identify the most efficient way to create the pool, while controlling the bias. We focus on a specific case of pooling: when the pool has to respect a financial constraint (budget) that limits the number of documents to be pooled to a set value (N documents). We call this *fixed-cost pooling*. Moreover, these N documents to be judged are fairly distributed across topics (equally divided when possible). Both are typical constraints in most IR evaluation exercises like TREC, CLEF and NTCIR. While a number of isolated studies have analysed and proposed a number of pooling strategies, a complete picture of their effectiveness and bias is still lacking, and little has been analysed about these strategies in the context of fixed-cost pooling. This chapter extends and complements the body of evidence regarding pooling by providing: a synthesis of a substantial line of research done on the pooling method; a coherent mathematical

framework to describe pooling strategies; the identification of theoretical similarities between the analysed strategies; and a large-scale evaluation using 9 test collections. Based on this, we provide guidelines for building more stable test collections. In addition to the traditional Depth@K pooling strategy, we analyse the pool bias of a set of 22 previously identified pooling strategies.

6.2 Pooling Strategies

We examine each of the pooling strategies that we empirically investigate in this chapter as alternative to the standard Depth@K strategy. These pooling strategies can be classified in different ways. In this chapter we do it by (1) their origin: *classic pooling*, *voting systems*, *retrieval fusion methods*, *IR evaluation measures*, and *multi-armed bandit models*; and (2) the pooling strategy type: *adaptive* or *non-adaptive*. By adaptive we refer to those pooling strategies that adapt their behaviour based on knowledge acquired in the previous selection step(s), and by non-adaptive we refer to those pooling strategies that do not adapt.

As mentioned in the introduction, in this chapter we are mainly concerned with pools formed by exactly N documents, but the methods may be further generalised to variable-size pools (*e.g.*, by implementing different stopping criteria; this is left for future work). Moreover, the fair distribution of the N documents across topics by equally dividing them may be also further generalised to variable-size topic pools (*e.g.*, by implementing different topic allocation strategies, this is also left for future work).

Each pooling strategy takes the pooled runs and outputs the set of pooled documents. We formally analyse the pooling strategies by defining a scoring function (s) on all candidate documents, and a set-building function (J) that uses s to obtain the set of pooled documents ($\mathcal{J}$). This aims at highlighting the differences and commonalities among pooling strategies. Each strategy is identifiable by the properties of s and J. This formalisation will naturally lead to the taxonomy of pooling strategy types: *non-adaptive*, and *adaptive*. In particular the latter will be subdivided into *adaptive with run allocation*, and *adaptive without run allocation*. By *with run allocation* we refer to an adaptive strategy that, to select the next document, it first selects a run – allocates a judgement to be performed to this run – then selects a document from this run. While *without run allocation* refers to an adaptive strategy that selects a document by aggregating information across runs. In Figure 6.1 we show the taxonomy of the pooling strategies analysed in this chapter.

Herein, we make the effort to unify all the pooling strategies under this framework in order to be able to formally assess their similarities and differences. Some of the pooling strategies below are relatively new to IR, although the underlying intuitions have been extensively used in IR as evaluation measures, and retrieval fusion methods [AM01; Mac09; MA02]. We also present the recently developed multi-armed bandit-based strategies [LPB16], and classic pooling strategies [CPC98].

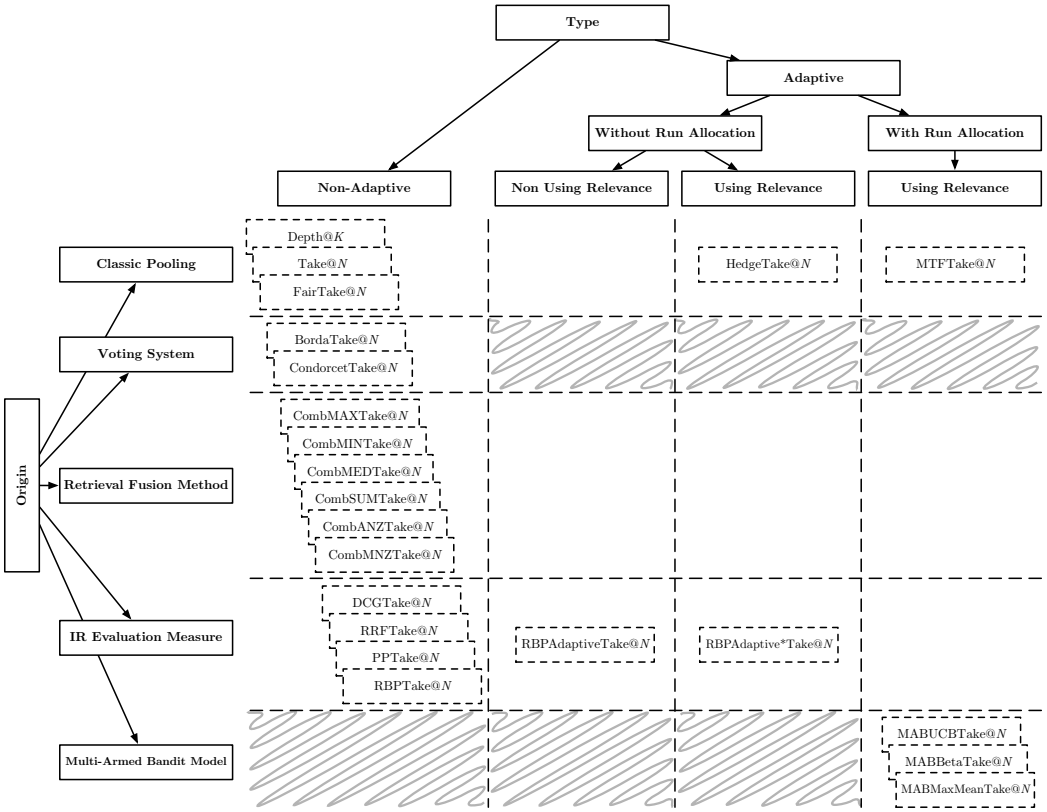

Figure 6.1: Taxonomy of the pooling strategies analysed in this chapter based on the pooling strategy type and their origin. Every cell represents a combination of these two classifications. The cells marked with a squiggly line are those cells for which a pooling strategy cannot exist.

In what follows, we introduce the non-adaptive pooling strategies, next the adaptive ones.

6.2.1 Non-Adaptive Pooling Strategies

Non-adaptive pooling strategies do not modify their behaviour based on the current pooled documents, regardless of whether these documents have been judged or not. The strategy Depth@K belongs to this category. The following subsections group the pooling strategies by their origin: *classic pooling*, *voting systems*, *retrieval fusion methods*, and *IR evaluation measures*.

Classic Strategies

Before analysing the considered pooling strategies, we start recalling the formalisation discussed in Section 3.5 of the most common strategy: Depth@K. Then, we present

some natural variants, Take@N and FairTake@N, which consider the number of required documents (N) as a parameter.

Depth@K (D). This strategy creates, for each topic, a global ranked list of documents where each document is scored based on its highest position across all $\mathcal{R}_p$ runs. Given this ranked list, the top ranked documents are selected to form the pool. The Depth@K strategy is specified by the following definitions of s, which scores every document d retrieved by the set of pooled runs $\mathcal{R}_p$:

$$s(d, \mathcal{R}_p) = \max_{r \in \mathcal{R}_p} (-\rho(d, r)) \tag{6.1}$$

and J, which determines the set of pooled documents:

$$J_{\mathcal{R}_p} = \{d \in \mathcal{D}_r : r \in \mathcal{R}_p : s(d, \mathcal{R}_p) \geq -K\} \tag{6.2}$$

A primary feature of this pooling strategy is its *fairness* to the pooled runs. A strategy is *fair* when the probability of a document to be judged at a given position is constant across runs. This is guaranteed by selecting the top K documents from every run. However, although this pooling strategy takes into consideration the contribution of all pooled runs, it has no control on the exact size of the final set of pooled documents ($|\mathcal{J}|$). It is therefore not a fixed-cost pooling strategy. Moreover, the number of documents selected per topic can vary depending on the size of the overlapping retrieved documents the pooled runs share on a per topic basis.

We introduce a natural extension of Depth@K that guarantees a given number N of pooled documents, called Take@N, effectively turning Depth@K into a fixed-cost pooling strategy. We now formalise this strategy, show its limitation, and introduce a new version that addresses it.

Take@N (T). This strategy creates, for each query, a global ranked list of documents using a new definition of s:

$$s(d, \mathcal{R}_p) = \max_{r \in \mathcal{R}_p} (-\rho(d, r) - \epsilon \cdot \mathrm{id}(r)) \tag{6.3}$$

This definition of s is similar to the definition in Eq. 6.1. However, it differs by the small deterministic contribution ($\epsilon \cdot \mathrm{id}(r)$) that is used to provide a unique score for every d in order to break ties. This contribution is small enough to not change the order defined by the document's ranks, and it is deterministic because it is based on the *id*s of the runs. The top n ranked documents, fraction of the size of the pool N, are selected to be pooled as follows:

$$J_{\mathcal{R}_p} = \tau@n(\mathcal{R}_p, s) \tag{6.4}$$

where $\tau@n$ is always well defined, *i.e.*, there is no ambiguity on which documents to return first. Compared to Depth@K, this strategy presents a drawback: it does not guarantee fairness with the pooled runs. With Depth@K all runs contribute equally to

the pool (first K documents). With Take@N, not all runs may contribute the same. The contributions are however only slightly unbalanced: the maximum difference between the number of documents contributed by two runs is equal to one. This strategy also compared with Depth@K behaves differently across topics, because while Depth@K can vary based on the size of the overlapping retrieved documents the pooled runs share, Take@N distributes the N documents to be judged uniformly. That is, to every topic is assigned, if possible, the same fraction of documents to be judged ($n \cdot |\mathcal{Q}| = N$).

FairTake@N (F). This strategy aims to address the lack of fairness of Take@N by introducing a nondeterministic selection of the documents to be judged. This strategy shares some of the characteristics of the *Stratified* pooling strategy [YKA08]. The *Stratified* strategy defines multiple strata, each characterised by a depth and a sample rate. This strategy is defined in two steps. First, each document is assigned to a stratum based on its highest rank across the pooled runs. Then, documents are sampled based on the sample rate of the stratum. FairTake@N is akin to having a stratification composed of two strata, a stratification with sample rate 1 as deep as the number of documents to be judged $n_{q,0}$ does not exceed n_q, the fraction of documents to be judged assigned to the topic q, and a second stratification of depth 1 with sample rate equal to $(n_q - n_{q,0})/|\mathcal{R}_p|$, which guarantees eventually to have exactly n_q judged documents. By definition, this strategy is fair with the pooled runs because any document at a given position has the same probability to be judged. In this strategy s is defined as:

$$s(d, \mathcal{R}_p) = \max_{r \in \mathcal{R}_p} (-\rho(d, r) - \epsilon \cdot \mu(0, 1)) \tag{6.5}$$

J is defined as in Eq. 6.4. Fairness is achieved by introducing a small random component to the score s. This value breaks potential ties and is small enough to not influence the ranking. Its random nature ensures that any document has equal opportunity to be sampled from any run. In this way, the strategy selects n_q documents to be judged in a fair way because every run will have in expectation (across topics) the same number of judged documents.

Voting System-Based Strategies

These strategies are based on the intuitions underlying voting systems. In general, voting systems take one of two forms: (1) positional voting systems that rely on the rank at which a document is retrieved (*e.g.* to assign a voting score to that document), and (2) majority voting systems that assign document weights based on pairwise comparisons between candidate documents.

BordaTake@N (B). This strategy is a positional voting strategy in which candidate documents are ranked in order of preference. For this strategy, s is defined as:

$$s(d, \mathcal{R}_p) = \sum_{r \in \mathcal{R}_p} B(d, r) + \epsilon \cdot \mu(0, 1) \tag{6.6}$$

where B defines the particular implementation of the Borda count. In this case, because we are dealing with truncated ballots (*i.e.*, not every document is ranked by each run), we follow the method also used by [AM01]: for a document d, the strategy assigns a score equal to the size of the collection of documents ($|\mathcal{D}|$) minus the rank at which d has been retrieved in the r ($\rho(d,r)$) if d has been retrieved by r, or else, if d has not been retrieved by r, the average score the strategy would have assigned to the documents retrieved between the last ranked document (equal to the size of the run $|\mathcal{D}_r|$) and the size of the collection of documents ($|\mathcal{D}|$). Formally, B is defined as follows:

$$B(d,r) = \begin{cases} |\mathcal{D}| - \rho(d,r) & \text{if } d \in r \\ \text{Avg}_{|\mathcal{D}_r|<n\leq|\mathcal{D}|}\left(|\mathcal{D}| - n\right) & \text{if } d \notin r \end{cases} \simeq - \begin{cases} \rho(d,r) & \text{if } d \in r \\ \frac{|\mathcal{D}|+|\mathcal{D}_r|+1}{2} & \text{if } d \notin r \end{cases}$$

where the symbol $\simeq$ indicates rank equivalence, and the expression on the right side of $\simeq$ is a simplified rank equivalent form of the same strategy. J is defined as in Eq. 6.4. Comparing this equation with Eq. 6.3 we observe that BordaTake@N is different from Take@N in that it considers the sum of all ranks at which a document has been retrieved, while Take@N only considers the highest rank (the earliest rank).

CondorcetTake@N (C). This majority voting strategy ensures that pooled documents are those that, when compared to not-pooled documents, have been retrieved at higher ranks by more systems. Strategies that fulfil this condition satisfy the *Condorcet criterion*, and it is easy to prove that Depth@K, Take@N, FairTake@N and BordaTake@N do not satisfy this condition. Specifically, this strategy starts by forming a list containing the set of all documents retrieved by the pooled systems. Then, it sorts the list according to the following procedure. Each document pair d_i and d_j is then compared as follows. We iterate through the document rankings of each system and increment a counter if d_i is ranked above d_j (or decrement the counter in the converse situation). When all systems have been considered, if the counter is positive, then d_i should be ranked above d_j; if it is negative, then the opposite ranking should be enforced. This leads to the definition of the following comparative function:

$$C(d_0, d_1, R_p) = \sum_{r \in R_p} \text{sign}(\rho(d_1,r) - \rho(d_0,r)) \tag{6.7}$$

This function does not define a total order, leading to the so-called Condorcet paradox. Imagine three documents, d_a, d_b, and d_c, such that d_a is preferred over d_b, d_b over d_c, and d_c over d_a. This cycle is a paradox because the conclusions are in conflict with each other. A solution is to adopt a method that still respects the Condorcet condition but that does not lead to this paradox. In our case we use what is known as Copeland's method, which counts the number of times a document beats the other documents. This

leads to the following definition of s:

$$s(d, \mathcal{R}_p) = \sum_{d' \in \mathcal{D}} \begin{cases} 1 & C(d, d', \mathcal{R}_p) > 0 \\ 0 & \text{otherwise} \end{cases} + \epsilon \cdot \mu(0,1) \simeq$$

$$\simeq \sum_{d' \in \bigcup_{r \in \mathcal{R}_p} \mathcal{D}_r} \begin{cases} 1 & C(d, d', \mathcal{R}_p) > 0 \\ 0 & \text{otherwise} \end{cases} + \epsilon \cdot \mu(0,1) \quad (6.8)$$

where the expression on the right side is a simplified rank equivalent form of the same strategy. This strategy is related to the Borda voting system. It can be proven that the relaxation of the Condorcet criterion used in the Copeland method leads to the Borda strategy (see Appendix A.1). This observation illustrates why this method is majority-based. It only counts when a document in the majority of the cases, across runs, has a higher score than another document, rather than counting its contribution per each individual run, like in BordaTake@N.

Retrieval Fusion Method-Based Strategies

Another class of non-adaptive pooling strategies is based on retrieval fusion methods. The main difference with the other strategies is that these are based on the score each ranker gives to a document (rather than the rank). To allow the comparison of scores between runs, score normalisation is required, otherwise the pooling strategy would be biased towards the runs that produce larger scores. Following existing practice in fusion for retrieval [AM01; Cro00; Lee97; MA01], we apply the following feature scaling:

$$\overline{\sigma}(d, r) = \frac{\sigma(d, r) - \min_{d' \in \mathcal{D}_r}(\sigma(d', r))}{\max_{d' \in \mathcal{D}_r}(\sigma(d', r)) - \min_{d' \in \mathcal{D}_r}(\sigma(d', r))}$$

which normalises all the values into the range $[0, 1]$. To be noted that for any document not retrieved by the run r by the definition of σ, which returns the minimum value observed in the run, $\overline{\sigma}$ returns 0.

CombMAXTake@N (MAX). This strategy assigns to each document the maximum retrieval score that the document has across all systems. In general, a document may be retrieved by multiple systems, and this likely happens with different scores. s is therefore defined as:

$$s(d, \mathcal{R}_p) = \max_{r \in \mathcal{R}_p} (\overline{\sigma}(d, r)) + \epsilon \cdot \mu(0,1) \simeq \max_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} (\overline{\sigma}(d, r)) + \epsilon \cdot \mu(0,1) \quad (6.9)$$

where on the right side of $\simeq$ we can observe a simplified rank equivalent form of the same strategy. After constructing a new document ranking with the maximum scores, the pool is obtained as for FairTake@N, *i.e.*, only the documents with the highest n_q scores are included in the pool $\mathcal{J}$, where n_q is the fraction of documents to be judged assigned to the topic, as defined in Eq. 6.4. The *CombMAX* retrieval fusion method,

which shares the same underlying intuition of CombMAXTake@N, is a commonly used strong baseline in the literature of fusion methods for retrieval. This strategy minimises the probability to discover relevant documents being poorly ranked. This definition of s and the definition in Eq. 6.3 are similar, while the former uses documents' ranks, the latter documents' scores.

CombMINTake@N (MIN). While the previous strategy minimises the probability to discover relevant documents being poorly ranked, this strategy minimises the probability to discover irrelevant documents ranked at early ranks. This strategy also combines the scores from different runs (by extracting the minimum score of each document across all runs). s is therefore defined as:

$$s(d, \mathcal{R}_p) = \min_{r \in \mathcal{R}_p} (\overline{\sigma}(d, r)) + \epsilon \cdot \mu(0, 1) \tag{6.10}$$

J is defined as in Eq. 6.4.

CombMEDTake@N (MED). This strategy takes a middle-ground approach to the selection of pooling documents based on fusion, by selecting the median score (as opposed to the maximum or minimum score as in CombMAXTake@N and CombMINTake@N, respectively). s is defined as follows:

$$s(d, \mathcal{R}_p) = \operatorname*{Med}_{r \in \mathcal{R}_p}(\overline{\sigma}(d, r)) + \epsilon \cdot \mu(0, 1) \tag{6.11}$$

J is defined as in Eq. 6.4.

CombSUMTake@N (SUM). Instead of selecting a single score as in CombMAXTake@N, CombMINTake@N, and CombMEDTake@N, CombSUMTake@N sums all the available document's scores. s is therefore defined as:

$$\begin{aligned} s(d, \mathcal{R}_p) = \sum_{r \in \mathcal{R}_p} \overline{\sigma}(d, r) + \epsilon \cdot \mu(0, 1) \simeq \\ \simeq \operatorname*{Avg}_{r \in \mathcal{R}_p} (\overline{\sigma}(d, r)) + \epsilon \cdot \mu(0, 1) \simeq \\ \simeq \sum_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} \overline{\sigma}(d, r) + \epsilon \cdot \mu(0, 1) \end{aligned} \tag{6.12}$$

where we observe that the expression on the right side of the first $\simeq$ demonstrates the rank equivalence of this strategy with a strategy defined by the arithmetic mean across runs, differing only by a constant $(1/|\mathcal{R}_p|)$; and the expression on the right side of the second $\simeq$ presents a simplified rank equivalence form of the same strategy. Comparing this equation with Eq. 6.6, we observe that CombSUMTake@N is the counterpart of the Borda strategy, but for scores (Borda uses ranks). Q is defined as in Eq. 6.4.

CombANZTake@N (ANZ). This strategy computes the average of the non-zero document scores. This strategy effectively eliminates the effect of a single run failing to retrieve a document (and thus assigning a zero score to that document). s is therefore defined as:

$$s(d, \mathcal{R}_p) = \frac{1}{|\{r \in \mathcal{R}_p : \overline{\sigma}(d, r) > 0\}|} \sum_{r \in \mathcal{R}_p} \overline{\sigma}(d, r) + \epsilon \cdot \mu(0, 1) \tag{6.13}$$

J is defined as in Eq. 6.4.

CombMNZTake@N (MNZ). This strategy aims to give higher weights to documents retrieved by multiple systems. This is achieved by multiplying the sum of scores of a document by the number of runs that retrieved that document. s is defined as:

$$s(d, \mathcal{R}_p) = |\{r \in \mathcal{R}_p : \overline{\sigma}(d, r) > 0\}| \sum_{r \in \mathcal{R}_p} \overline{\sigma}(d, r) + \epsilon \cdot \mu(0, 1) \tag{6.14}$$

J is defined as in Eq. 6.4.

IR Evaluation Measure-Based Strategies

This section presents several strategies inspired by IR evaluation measures. These pooling strategies accumulate evidence of the importance of a document d for a given topic based on both a) the rank $\rho(d, r)$ at which d has been retrieved in the pooled run $r \in \mathcal{R}_p$, and b) the specific characteristics of the considered IR evaluation measure.

All the pooling strategies below share the same generalisation of s, in which the contribution from every rank is replaced by a gain function related to the evaluation measure. s is defined as follows:

$$s(d, \mathcal{R}_p) = \sum_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} G(\rho(d, r)) + \epsilon \cdot \mu(0, 1) \tag{6.15}$$

where G is the gain defined by the evaluation measure.

DCGTake@N (DCG). This strategy uses the gain function defined in the discounted cumulative gain (DCG) to rank candidate documents [JK02]. The gain is characterised by an inverse log_2 decay function, as follows:

$$G(\rho) = \frac{1}{\log_2(\rho + 1)} \tag{6.16}$$

Candidate documents are ranked in decreasing order of the sum of values computed by G in s. J is defined as in Eq. 6.4.

RRFTake@N (RRF). This strategy is rooted in the reciprocal rank (RR) evaluation measure, which is commonly used to assess system effectiveness in tasks such as known item search, question answering, or query auto completion [Dum+02]. A variant of RR, the reciprocal rank fusion (RRF), has been used as retrieval fusion method [CCB09]. RRF makes use of an additional parameter, α, that controls the decay of the document contribution score as a function of the rank. In this pooling strategy we employ the same idea, with $\alpha = 60$ as in [CCB09]; other values will be investigated in future work. Its G is defined as follows:

$$G(\rho) = \frac{1}{\rho + \alpha} \tag{6.17}$$

Candidate documents are ranked in decreasing order of the sum of values computed by G in s. J is defined as in Eq. 6.4.

PPTake@N (PP). This strategy (*PP*, for *perfect precision*) is inspired by the family of measures that count the number of relevant documents found at rank ρ and divide it by the number of documents up to rank ρ. Average Precision [BV00] and Sakai's Q-Measure [Sak04] are examples of metrics belonging to this family. To define the G function for these class of IR evaluation measures, we assume to compute these IR evaluation measures on a ranked list as if all documents up to rank ρ are relevant, therefore the rank score attributed to a document retrieved by runs in $\mathcal{R}_p$ is the number of runs that have retrieved that document:

$$G(\rho) = 1 \tag{6.18}$$

This leads to a set-based majority voting procedure to rank documents and select the top N. It is set-based because the order in which the documents are retrieved does not count. This can be seen as a relaxation of the Borda strategy. Candidate documents are ranked in decreasing order of the sum of values computed by G in s. J is defined as in Eq. 6.4.

RBPTake@N (RBP). This strategy (named *Method A* in the original article [MWZ07]) computes document scores based on Rank Biased Precision (RBP) [MZ08]. The RBP formula is characterised by a parameter p that models the user persistence, *i.e.*, the likelihood that the user examines a document. The persistence parameter is effectively used to discount the contribution of a relevant document, similarly to other gain-discount based measures [Car11]. The gain function is defined as follows:

$$G(\rho) = (1 - p)p^{\rho - 1} \tag{6.19}$$

In our experiments we use $p = 0.8$; this is akin to previous work that relied on RBP for evaluation [PZ07; ZPM10] and for pooling [MWZ07]. The use of RBP as a document discount factor in weighting the contribution of documents to the pool creates a family of 3 pooling strategies [MWZ07], one being RBPTake@N. We present the other two in the next subsection. Candidate documents are ranked in decreasing order of the sum of values computed by G in s. J is defined as in Eq. 6.4.

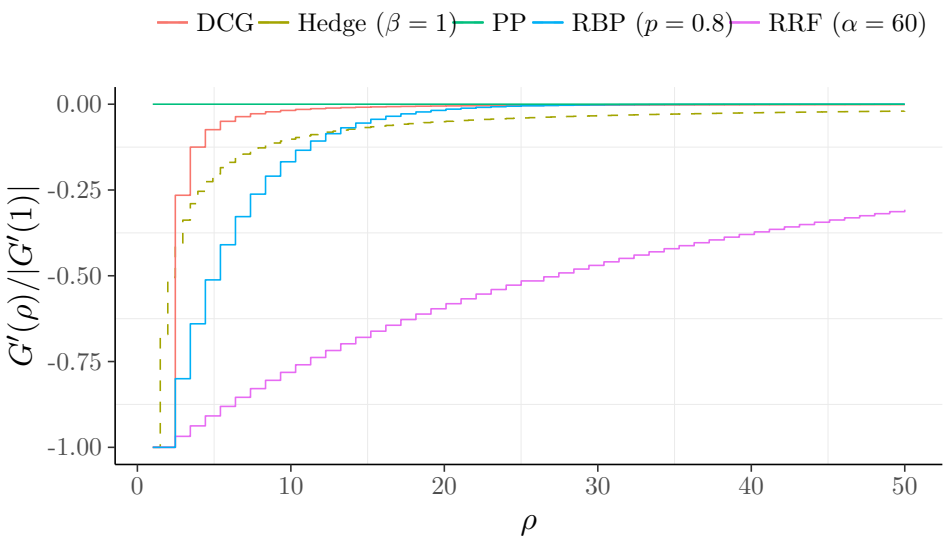

Figure 6.2: Derivative of gain functions G normalised by $|G'(1)|$, for DCGTake@N, RRFTake@N, PPTake@N, and RBPTake@N as functions of the rank position, for a run r. The figure also shows a special case of an adaptive pooling strategy, HedgeTake@N.

Figure 6.2 shows the normalised derivative of the gain functions (G). For the sake of comparison we observe the normalised derivative, because we are not interested in the values of the functions, but on their sensitivity to change with respect to a change in the rank (ρ) of the retrieved document. We normalise by dividing the derivatives by their first values ($G'(1)$). This is possible because multiplying by a constant value generates new but rank equivalent strategies. In this plot we can observe that for *PP* the function does not depend on the rank position at which the document has been retrieved, while for *RRF* the function increases almost linearly. For *DCG* and *RBP* the documents are almost indistinguishable when retrieved after rank 10 for the first, and 20 for the second.

6.2.2 Adaptive Pooling Strategies

So far we have discussed the non-adaptive pooling strategies. These strategies are characterized by first computing a score for each candidate document, ranking the documents in decreasing score, and selecting the top N documents. They are non-adaptive because the score of a document is not affected by the previously selected documents.

Another class of pooling strategies is adaptive. Pooling strategies in this class recompute the scores used by the ranking function s based on the last document selected. This is formalised by having s taking as input the current set of pooled documents and iteratively changing the scores of the documents.

First, the definition of s is expanded to consider the documents that have already been

pooled. The superscript $\mathcal{J}$ indicates that we now receive the pooled documents as an input. The new definition of s, which will be denoted as $s_+^{\mathcal{J}}$, ensures that documents that have been pooled in the previous iteration are not re-scored:

$$s_+^{\mathcal{J}}(d, \mathcal{R}_p) = \begin{cases} s^{\mathcal{J}}(d, \mathcal{R}_p) & d \notin \mathcal{J} \\ -\infty & d \in \mathcal{J} \end{cases} \tag{6.20}$$

Setting $s_+^{\mathcal{J}}(d, \mathcal{R}_p)$ to $-\infty$ ensures that already pooled documents do not get selected again. The specific definition of $s^{\mathcal{J}}(d, \mathcal{R}_p)$ will be determined by each pooling strategy.

The set $\mathcal{J}$ grows as documents are pooled. The pooled documents after the n-th iteration of judgements will be referred to as $\mathcal{J}_n$. The construction of the $\mathcal{J}_n$s is achieved recursively:

$$\begin{aligned} \mathcal{J}_1 &= \tau@1(\mathcal{R}_p, s_+^{\emptyset}) \\ \mathcal{J}_n &= \mathcal{J}_{n-1} \cup \tau@1(\mathcal{R}_p, s_+^{\mathcal{J}_{n-1}}) \\ J_{\mathcal{R}_p} &= \mathcal{J}_n \end{aligned} \tag{6.21}$$

$\mathcal{J}_1$ contains the top-ranked document (beginning of the assessment process), and $\mathcal{J}_n$ contains all previously judged documents ($\mathcal{J}_{n-1}$) together with a newly selected document that depends on how $s_+^{\mathcal{J}_{n-1}}$ re-scores the documents. This definition of a pooling strategy generalises the non-adaptive definition previously presented in Eq. 6.4.

There exists another type of adaptive strategy: the adaptive with run allocation. These adaptive strategies also specify which runs should be pooled (*e.g.*, by iteratively choosing documents from one run or another). This is formalised by a sequence r_n that determines from which run r the documents have to be pooled.

We have seen that in the adaptive pooling strategies without run allocation, s is defined by the pooling strategy using as input the previous pooled documents. In the adaptive pooling strategies with run allocation, s is defined as follows:

$$s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p) = -\rho(d, r_n) \tag{6.22}$$

where the effect of the run pooling strategy is only observed in the run allocation sequence ($\{r_n\}_{n \in \mathbb{N}_1}$), which is different for every strategy. This definition of s scores every document of the allocated run in order of their retrieved rank position, and by substituting it into Eq. (6.20), it allows s_+ to re-rank to the end of the list all the documents already pooled.

Adaptive pooling strategies modify their behaviour based on the current pooled documents. These strategies can be further divided into two categories based on which kind of document information is required in the adaptive stage: *non relevance-based*, and *relevance-based*. All the pooling strategies listed below are relevance-based pooling strategies, except for RBPAdaptiveTake@N, which is a non relevance-based one. The adaptive pools are incrementally built using the recursive definition of J in Eq. 6.21. We now describe the pooling strategies that belong to the adaptive class, classified by their origin: *classic strategies*, *IR evaluation measures*, and *multi-armed bandit models*.

Classic Strategies

This category includes traditional strategies developed in IR. In this category, two strategies exhibit adaptive behaviour, the Move-To-Front strategy (MTFTake@N), and the Hedge strategy (HedgeTake@N).

MTFTake@N (MTF). *MTF* is an heuristic developed by Cormack et al. [CPC98], which associates a priority to each run. Initially, all runs have maximum priority. At every iteration of J, this strategy selects a random run among the maximum priority runs. Then, it takes the first document retrieved by this run and judges it for relevance. At the next iteration, if the document was relevant ($\mathcal{J}^+_{n-1} \setminus \mathcal{J}_{n-2} \neq \emptyset$) then MTFTake@$N$ will continue selecting and judging documents from the same run. Otherwise, the priority of the current run is decreased and the method randomly selects another maximum priority run. We first define the following function that returns the number of times a run r has been sampled:

$$\#(r, r|_1^n) = |\{i \in \{1, 2, 3, \ldots, n\} : r = r_i\}|$$

The run selection sequence is defined as follows:

$$r_1 = \arg\min_{r \in \mathcal{R}_p} (\mu(0,1))$$

$$r_n = \begin{cases} r_{n-1} & \text{if } \mathcal{J}^+_{n-1} \setminus \mathcal{J}_{n-2} \neq \emptyset \\ \arg\min_{r \in \mathcal{R}_p} \left(\left| \{d \in \mathcal{D}_r : \rho(d, r) \leq \#(r, r|_1^{n-1})\} \cap \mathcal{J}^-_{n-1} \right| + \right. & \\ \left. +\epsilon \cdot \mu(0,1) \right) & \text{otherwise} \end{cases} \tag{6.23}$$

r_1 makes an initial random selection (all runs have the maximum priority), and r_n either continues on the current run because the last document was relevant (r_{n-1}), or jumps to another maximum priority run. s is as defined in Eq. 6.22 and J in Eq. 6.21.

HedgeTake@N (H). This strategy is an online learning algorithm proposed by Aslam et al. [APS03] for metasearch and pooling. It associates a set of losses to the contributing runs. These losses depend on the relevance outcomes and the positions in the runs of the judged documents. For example, a run's loss is increased (decreased) if the run retrieved a irrelevant (relevant) document at a high position. After each assessment, the run's losses are updated and the next pick (next assessed document) depends on the run's losses and the positions of the unjudged documents in the runs. For each document-run pair, the following function takes the document's position and estimates the loss we would obtain if the document is deemed irrelevant:

$$G(\rho) = \ln(|\mathcal{D}|/\rho)$$

This loss needs to be computed for all documents (including those that do not belong to the run). This is achieved by extending G as follows:

$$G^*(d, r) = \begin{cases} G(\rho(d, r)) & \text{if } d \in \mathcal{D}_r \\ \text{Avg}_{|\mathcal{D}_r| < i \leq |\mathcal{D}|} G(i) & \text{otherwise} \end{cases}$$

If the document does not belong to the run then the loss is estimated as the average loss the document would get if retrieved in positions from $|\mathcal{D}_r| + 1$ to $|\mathcal{D}|$. As we obtain relevance assessments, we iteratively accumulate the loss induced by each run ($L(r, \mathcal{J})$). These runs' losses depend on the relevance outcomes and the positions in the runs of the judged documents (as defined by G^*). The loss of run r is defined as follows:

$$L(r, \mathcal{J}_{n-1}) = \frac{1}{2} \sum_{d \in \mathcal{J}^-_{n-1}} G^*(d, r) - \frac{1}{2} \sum_{d \in \mathcal{J}^+_{n-1}} G^*(d, r)$$

Next, the loss is normalised by:

$$\overline{L}(r, \mathcal{J}_{n-1}) = \frac{\beta^{L(r, \mathcal{J}_{n-1})}}{\sum_{r' \in \mathcal{R}_p} \beta^{L(r', \mathcal{J}_{n-1})}} \tag{6.24}$$

This normalisation has a parameter $\beta \in [0, +\infty[$ that controls the way in which new judgements change the weights. We set $\beta = 0.1$ as in Losada et al. [LPB16]; other values will be investigated in future work. Finally, s is defined as the weighted average of the documents' losses across all runs:

$$s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p) = \sum_{r \in \mathcal{R}_p} \left(\overline{L}(r, \mathcal{J}_{n-1}) \cdot G^*(d, r) \right) + \epsilon \cdot \mu(0, 1) \tag{6.25}$$

J is defined as in Eq. 6.21. It is interesting to observe that this strategy takes into account also the irrelevant documents. Now, we make some observations about how this pooling strategy changes behaviour as we vary β. In particular we analyse three special values of β, when $\beta \to 0$, $\beta = 1$, and $\beta \to +\infty$ (see Appendix A.2). When $\beta = 1$ we observe that this strategy reduces to a non-adaptive evaluation measure-based strategy with G defined as follows:

$$G(\rho) = \log\left(\frac{1}{\rho}\right) + \frac{\log(|\mathcal{D}|!)}{|\mathcal{D}|}$$

When β tends to $+\infty$, we observe that this strategy reduces to a MTFTake@N like pooling strategy. This observation derives from the fact that when β tends to $+\infty$ the normalisation in Eq. 6.24 will select the run that has the largest L score. From this run, due to Eq. 6.25, the document with the highest rank, not yet pooled, is selected. Now, if the document was relevant, a new document will be picked from the same run because it is still the run with the largest score; if the document is not relevant, the score of the run is reduced, and a new document will be picked potentially from a run with a larger score, like the MTFTake@N strategy. However there is a main difference between these two strategies, for MTFTake@N the run is kept the same every time a picked document is judged relevant, in this case this is embedded in the definition of selection of the run by increasing the score for the run. When β tends to 0, we observe an opposite behaviour than the one observed when β tends to $+\infty$: the score for a run is increased if the retrieved document is irrelevant and decreased if the document is relevant. This generates a pooling strategy that instead of continuing to sample from runs that retrieved relevant documents like MTFTake@N, it continues to sample from runs that retrieved irrelevant documents.

IR Evaluation Measure-Based Strategies

The next two strategies are extensions of RBPTake@N. Thanks to the convergent behaviour of RBP, [MWZ07] have naturally extended RBPTake@N to include additional information into the scoring function s.

RBPAdaptiveTake@N ($RBP^{\mathbf{A}}$). This strategy is an adaptive version of *RBP* (named *Method B* in the original article [MWZ07]), which adds documents to the pool in an incremental way. For each run $r \in \mathcal{R}_p$, it computes its residual $e(r, \mathcal{J})$, *i.e.*, a value proportional to the number of not judged documents in the run. The residual is defined as:

$$e(r, \mathcal{J}_{n-1}) = p^{|r|} + (1-p) \sum_{d \in \mathcal{D}_r : d \notin \mathcal{J}_{n-1}} p^{\rho(d,r)-1}$$

s is defined as follows:

$$s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p) = \sum_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} (G_{\text{RBP}}(\rho(d,r)) \cdot e(r, \mathcal{J}_{n-1})) + \epsilon \cdot \mu(0,1) \tag{6.26}$$

With each new selection, the runs' residuals change and the score $s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p)$ needs to be recomputed (thus, the adaptive nature of RBPAdaptiveTake@N). J is defined as in Eq. 6.21.

RBPAdaptive*Take@N ($RBP^{\mathbf{A}*}$). This pooling strategy (named *Method C* in the original article [MWZ07]) is also an adaptive pooling strategy that uses both the RBP residuals, as RBPAdaptiveTake@N, and the actual RBP score $b(r, \mathcal{J})$ of a run r, computed using binary relevance:

$$b(r, \mathcal{J}_{n-1}) = \sum_{d \in \mathcal{D}_r : d \in \mathcal{J}^+_{n-1}} G_{\text{RBP}}(\rho(d,r))$$

The candidate documents for pooling are ranked by decreasing:

$$\begin{aligned} s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p) = \\ = \sum_{r \in \mathcal{R}_p : d \in \mathcal{D}_r} \left[G_{\text{RBP}}(\rho(d,r)) \cdot e(r, \mathcal{J}_{n-1}) \cdot \left(b(r, \mathcal{J}_{n-1}) + \frac{e(r, \mathcal{J}_{n-1})}{2} \right)^3 \right] + \\ + \epsilon \cdot \mu(0,1) \end{aligned} \tag{6.27}$$

At each iteration n, this strategy uses the information about the relevance of the last selected document (observe the set of judged relevant documents $\mathcal{J}^+_{n-1}$ in Eq. 6.27). Being an adaptive strategy, J is defined as in Eq. 6.21.

Multi-Armed Bandit Models-Based Strategies

These strategies model pooling as a multi-armed bandit problem [LPB16]. The bandit-based strategies are adaptive. As we select and judge documents, we gain knowledge on the quality of the contributing runs. Run selection is driven by the classical exploration versus exploitation dilemma, which works as follows. At any point, we can opt for *exploiting* our current knowledge (*i.e.*, choose the run that has supplied the highest average number of relevant documents) or, alternatively, we can opt for *exploring* (*i.e.*, choose a suboptimal run). Exploitation maximises the expected reward on the next pick, but exploration may produce the greater total reward over a long period of time (the runs that are currently inferior can eventually become good suppliers of relevant documents). Every bandit-based strategy implements a specific bandit allocation method. A bandit allocation method chooses the next pick (next run) based on past actions and obtained rewards (relevance of judged documents).

MABGreedyTake@N (BG). This strategy is based on the ϵ-greedy bandit allocation method. A greedy approach consists of always selecting the run with the largest average of judged relevant documents. This greedy approach, which is similar to MTFTake@N, has been shown to be sub-optimal. A simple variant consists of behaving greedily most of the time and sometimes selecting a random (suboptimal) run. A simple strategy that implements this idea is ϵ_n-greedy [SB98]. At any point, ϵ_n-greedy plays with probability $1-\epsilon_n$ the run with the highest average of judged relevant documents, and with probability ϵ_n a randomly chosen run. ϵ_n is known as the exploration probability. It is good practice setting ϵ_n such that it decreases with the number of picks (n). This is because estimates become more accurate as more evidence is encountered and, therefore, the exploration probability should decrease. We employ the following definition of $\epsilon_n = \min(1, c_0|R_p|/(c_1^2(n-1)))$, where c_0 and c_1 are parameters. Following Losada et al. [LPB16], we set c_0 to 0.01, and c_1 to 0.1. For each run, we first compute the proportion of the run's judged documents that were deemed as relevant:

$$P(r, r|_1^{n-1}, \mathcal{J}_{n-1}) = \begin{cases} 1/2 & \#(r, r|_1^{n-1}) = 0 \\ \frac{|\{d \in \mathcal{D}_r : \rho(d,r) \leq \#(r, r|_1^{n-1})\} \cap \mathcal{J}^+_{n-1}|}{\#(r, r|_1^{n-1})} & \text{otherwise} \end{cases} \tag{6.28}$$

following the run succession used by s as defined in Eq. 6.22:

$$r_n = \begin{cases} \arg\max_{r \in \mathcal{R}_p}(\mu(0,1)) & \mu(0,1) < \min\left(1, \frac{c_0|\mathcal{R}_p|}{c_1^2(n-1)}\right) \\ \arg\max_{r \in \mathcal{R}_p}\Big(P(r, r|_1^{n-1}, \mathcal{J}_{n-1}) + \epsilon \cdot \mu(0,1)\Big) & \text{otherwise} \end{cases} \tag{6.29}$$

The second line of the equation above encodes the greedy action, which selects the run with the highest average ($\epsilon \cdot \mu(0,1)$, again, is incorporated here to break the ties), while the first line encodes the exploration action (random run selection). J is defined as in Eq. 6.21.

MABUCBTake@N (UCB). This strategy implements a version of UCB, the UCB1-Tuned method [ACF02]. UCB associates an *upper confidence index* to each run. This index estimates the uncertainty about the quality of the run (average relevance of documents from the run). After n rounds of judgement, we would like to sample from the *leading* run (the one with the largest proportion of judged relevant documents). But we need to be sure that the other runs have been sampled enough. Otherwise, we cannot be sure that they are indeed inferior. MABUCBTake@N (UCB) computes upper confidence bounds for the proportions of relevant documents supplied by the runs and compares the upper confidence bounds of apparently inferior runs with the estimated mean of the leading run. The index of the UCB1 strategy is the sum of two components: the current estimated mean and a quantity related to the size of the one-sided confidence interval for the estimated mean. UCB1-Tuned is an evolution of UCB1 that takes into account the variance of each run. In this strategy we use the probability of extracting relevant documents as defined in Eq. 6.28, and we define its average by renaming the function P in Eq. (6.28) defined in the previous strategy as follows:

$$P_{\mu}(r, r|_1^{n-1}, \mathcal{J}_{n-1}) = P(r, r|_1^{n-1}, \mathcal{J}_{n-1})$$

The definition of its variance is:

$$P_{\sigma^2}(r, r|_1^{n-1}, \mathcal{J}_{n-1}) = P(r, r|_1^{n-1}, \mathcal{J}_{n-1})(1 - P(r, r|_1^{n-1}, \mathcal{J}_{n-1}))$$

S defines the reward to maximise:

$$S(r, \mathcal{J}_{n-1}, r|_1^{n-1}) = P_{\mu}(r, r|_1^{n-1}, \mathcal{J}_{n-1}) + \\ + \sqrt{\frac{\ln(n-1)}{\#(r, r|_1^{n-1})} \min\left(\frac{1}{4}, P_{\sigma^2}(r, r|_1^{n-1}, \mathcal{J}_{n-1}) + \sqrt{\frac{2\ln(n-1)}{\#(r, r|_1^{n-1})}}\right)} + \epsilon \cdot \mu(0,1)$$

Here, we observe that, for the reward to be properly defined, $\#(r, r|_1^{n-1})$ must always be ≥ 1. To guarantee this, all the runs get the first document evaluated. Therefore, in the definition of P in Eq. (6.28) used to define P_{μ} and P_{σ^2}, we can ignore the first case when $\#(r, r|_1^n) = 0$. The initialisation is achieved by defining F as follows:

$$F(r, \mathcal{J}_{n-1}) = \max_{d \in \mathcal{D}_r : d \notin \mathcal{J}_{n-1}} (-\rho(d, r)) + \epsilon \cdot \mu(0,1)$$

and the run allocation policy is defined as:

$$r_n = \begin{cases} \arg\max_{r \in \mathcal{R}_p} \left(F(r, \mathcal{J}_{n-1})\right) & \text{if } \exists\, r \in \mathcal{R}_p, \exists\, d \in \mathcal{D}_r : \rho(d, r) = 1 \\ \arg\max_{r \in \mathcal{R}_p} \left(S(r, \mathcal{J}_{n-1}, r|_1^{n-1})\right) & \text{otherwise} \end{cases} \tag{6.30}$$

J is defined as Eq. 6.21.

MABBetaTake@N (BB). This strategy is based on a heuristic called Thompson sampling [Tho33]. It represents each run with a probability of supplying a relevant document, and each run's probability is associated with a probability distribution under a Bayesian framework. The process begins with no knowledge of these probabilities. This is encoded by applying a uniform prior for each run. This uniform initialisation, which is equivalent to the Beta distribution when assigning its shape parameters $\alpha = 1$ and $\beta = 1$ (Beta$(1, 1)$), represents the lack of knowledge about the chances of extracting relevant documents from each run. Run selection is done by extracting a sample from each distribution ($|\mathcal{R}_p|$ samples, one from each Beta distribution) and the run yielding the largest sample is chosen. This selection approach tends to select runs that have a high mean (*i.e.*, high likelihood of yielding relevant documents). Next, the top ranked unjudged document of the chosen run is judged for relevance, and the relevance outcome is used for updating the run's Beta distributions. With binary relevance, the relevance outcome can be modelled as a Bernoulli variable. This is a mathematical convenience because it guarantees that the update leads to posterior distributions (after incorporating the new evidence) that are also Beta distributed. So, we iteratively update the parameters of the Beta distributions based on the relevance of the judged documents. The run allocation sequence used by s in Eq. 6.22 is defined as follows:

$$r_n = \arg\max_{r \in \mathcal{R}_p} (\text{Beta}(1 + |r \cap \mathcal{J}^+_{n-1}|, \\ 1 + |r \cap \mathcal{J}^-_{n-1}|)) \tag{6.31}$$

J is defined as Eq. 6.21. To be noted that here the small random component ($\epsilon \cdot \mu(0, 1)$), useful to break the ties, is not necessary since it is already a stochastic process.

MABMaxMeanTake@N (MM). This is another Bayesian solution that represents the runs with Beta probabilities and updates the probability distributions based on the relevance assessments. The difference between MABBetaTake@N and MABMaxMeanTake@N is that MABMaxMeanTake@N does not make run selection by sampling from the Beta distributions. The run selected by MABMaxMeanTake@N is simply the one that has the maximum mean of the Beta distributions. The run allocation sequence, used in s as in Eq. 6.22, is defined as:

$$r_n = \arg\max_{r \in \mathcal{R}_p} \left(\frac{1 + |r \cap \mathcal{J}^+_{n-1}|}{2 + |r \cap \mathcal{J}^-_{n-1}|} + \epsilon \cdot \mu(0, 1) \right) \tag{6.32}$$

J is defined as in Eq. 6.21.

Losada et al. [LPB16] also describe a non-adaptive version of a multi-armed bandit based-strategy, which randomly allocates the runs from which to select the documents to be pooled. However, this strategy, as expected, performs similarly to FairTake@N, therefore it has not been considered in this thesis.

6.3 Experiments and Results

We do a large-scale evaluation in terms of pool bias of the 22 pooling strategies presented above on 9 test collections using 3 measures of bias and 3 IR evaluation measures. In this section we first present the experimental design. Next, we present the material and experiment setup. We then introduce the measures of bias; and finally, we present the results.

6.3.1 Experimental Design

In Section 3.5, we have presented that the pooling method is used to build test collections in evaluation efforts like TREC. In these evaluation efforts, an evaluation challenge is instantiated and the set of topics $\mathcal{Q}$ to be evaluated defined. Next, participating organizations $\mathcal{O}$ are invited to submit a set of runs of a given size, of which a subset per $\mathcal{O}$ is then used to form the set of pooled runs $\mathcal{R}_p$. Next, a pooling strategy is used to pool the documents to be judged by human relevance assessors. At the end of this building process, a test collection is released that is then used in laboratory experiments that, unavoidably, will suffer from pool bias.

In order to compare the effectiveness of the pooling strategies presented above in mitigating the effect of pool bias, we run a series of simulation experiments in which we simulate the process of building a test collection. One simulation consists in, given a set $\mathcal{O}$, building a test collection with the runs submitted by $|\mathcal{O}| - 1$ organizations and measure the bias on the runs submitted by the leftover organization. This can be formally expressed as follows: Given a set of organizations $\mathcal{O}$, a set of runs $\mathcal{R}_p$ submitted by $\mathcal{O}$, and an *ideal* set of judgements $\mathcal{I}$ that has a relevance value for each document, we can compute an ideal mean absolute error for a pooling strategy J as follows:

$$\frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| f(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}}) - f(r, \mathcal{I}) \right| = \\ = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| \beta_f(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}}) \right| \quad (6.33)$$

where the right-hand side is obtained by substituting the pool bias as defined in Eq. (3.14). This is referred to in the literature as a *leave-one organization-out* approach. This approach is preferred to a *leave-one run-out* approach because it better simulates the case that the retrieval model used by the organization has not contributed to the pool. However, due to the presence (in Eq. (6.33)) of the ideal set of judgements $\mathcal{I}$, which in reality does not exist, this error cannot be computed. Instead, in IR we usually dispose of an approximation of this set $\mathcal{I}$, which in the following we indicate as the ground-truth G. The use of G in the measurement introduces a random error in the observed measurement $f(r, J_{\mathcal{R}_p})$, as defined in Eq. (3.15). Substituting the random error to Eq. (6.33) we can

define the actual Mean Absolute Error (MAE) as:

$$\begin{aligned}\mathrm{MAE}(J_{\mathcal{R}_p}) = \\ &= \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| f(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}}) - f(r, G) \right| = \\ &= \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| \beta_f(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}}) + \varepsilon \right| \end{aligned} \quad (6.34)$$

Therefore, when using G in the simulations that calculate MAE, the absolute value we are measuring is a composition of the pool bias and random error. However, we claim that this random error is not an issue for our comparison because:

1. this is an error measured between G and $\mathcal{I}$, which makes it independent and constant across the set of tested pooling strategies Js;
2. the presence of this error is in line with standard evaluation praxis in IR, because this is the same error we would observe every time we test a run on an existing test collection;
3. the random error is 0 for some combination of f and G, *e.g.* this happens when f is P@n and at least the first n documents retrieved by r are contained in G.

In order to measure the difference in pool bias we must have perfect knowledge of all the documents that appear in any of the runs. The objective of these experiments is to quantify the effect of missing information (introduced by the pooling strategy) — therefore, we cannot allow missing information to exist at the onset of the experimental process. In this context, the best test collections are those originally built with Depth@K, because this requirement is easily satisfiable by using the pooled runs $\mathcal{R}_p$ and resizing them to a depth equal to $|r| = K$.

This process of test collection transformation is depicted in Figure 6.3. Essentially, the newly created test collections are "clean" in the sense that no information is kept for any of the runs for ranks above K. This cleaning is essential in order to ensure the validity of the experiments with different pooling strategies. If we were not to do this cleaning, when using $f(r, G)$ to observe the pool bias resulting of the use of a particular pooling strategy we would be confounding it with the pool bias of the original test collection.

This experimental design raises three potential issues:

1. the effect of experimenting with fixed-cost pooling strategies with test collections that were originally built with a Depth@K strategy;
2. the selection of too few documents to be judged (low N) may cause the reduction of the number of judged documents per topic at the level that makes any analysis based on this judgements inconclusive;

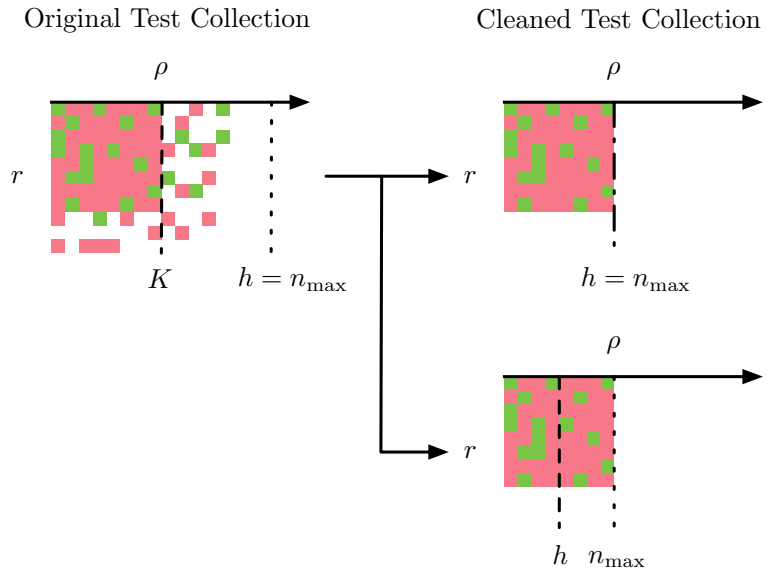

Figure 6.3: In the top left corner we illustrate the shape and setup of the original test collection. The y-axis indicates the runs, the x-axis the rank, every block represents a pooled document, which colour indicates its status: green if relevant, red if irrelevant, and white for unjudged. K indicates the depth of the pooling strategy used to build the original test collection; h indicates the horizon of the pooling strategy; and $n_{\max}$ the maximum evaluation depth available. In the right corner we present the shape and setup of the three experiments. At the top, the shape and setup used to compare the performance of the different pooling strategies and compare the expected number of judged documents. At the bottom, the shape and setup used to verify the consistency of the results of the first experiment varying h.

3. the resizing of the runs may have unexpected effects on the conclusion of the simulation experiments, *i.e.*, would a certain pooling strategy be preferred for a lower runs' size and another one for a higher one?

To address these questions we design three additional experiments. In the first experiment, we compare the FairTake@N strategy against the Depth@K strategy to verify that these two strategies manifest a similar behaviour. To allow the comparison of these two different kinds of strategies we set the parameter N of FairTake@N strategy in function of the number of documents judged, by setting the parameter K of the Depth@K strategy. In the second experiment, for every pooling strategy J, we measure the average number of

judged documents (AJ) for the pair run-topic, which we define as follows:

$$\mathrm{AJ}(J_{\mathcal{R}_p}) = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| \left\{ d \in \mathcal{D}_r : d \in J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}} \right\} \right|$$

this is then average across topics. AJ measures the expected number of judged documents we would expect on a new run. In the third experiment we verify the consistency of the results when used in a real setting. To do this with the same test collections we test the same case but reducing what we call the horizon of the pooling strategies. The horizon (h) is defined as the depth of the runs available to the pooling strategy. If the results found are not consistent with the ones found by the designed experiment, we have to reconsider our previous conclusions, if they are consistent, it means that the horizon effect is a negligible effect in our comparison. To illustrate our methodology we provide a graphical representation of both experiments, the designed one and this new one in Figure 6.3.

6.3.2 Material

To test the effectiveness of the different pooling strategies we selected 9 test collections from TREC [VH05]: Ad Hoc 3 [Har94], Ad Hoc 8 [VH99a], Web 9 [Haw00], Web 2014 [Col+15], Robust 2005 [Voo05], Genomics 2005 [Her+05], Legal 2006 [BLO06], Blog 2006 [Oun+06], and Microblog 2011 [Oun+11]. We selected these test collections because of: 1) the diverse origin – in fact they cover 6 different domains: News, Web, Genomics, Legal, Blog, and Microblog; 2) the large number of judged documents in the collections; 3) the large number of organizations that contributed to the pools – we assume that the number of participating organizations is directly proportional to the variety of the submitted runs, and 4) the pooling strategy used to build the collections, *i.e.*, *fixed depth at cut-off K* pooling strategy (Depth@K). The last point makes the collections suitable for testing new pooling strategies. As explained in the sample design, these test collections require to be normalized to a clean Depth@K. In addition, due to the prototypical nature of the tracks organized to build the test collections, we filtered out the 25% of lowest performing runs from our experimentation. This filtering is done to remove those runs that are likely to contain bugs or very exploratory methods. This procedure is in line with standard practices in the IR field [VB02]. The details about this normalization process and the test collection statistics are described in Table 6.1.

6.3.3 Measures of Pool Bias

The measures of pool bias take as input an IR evaluation measure f. We have already presented the first measure of bias in Eq. 6.34, the mean absolute error (MAE). This measure estimates the expected observed pool bias plus random error on the score of a non-pooled run. This is done by averaging the difference in score of the every $r \in \mathcal{R}_p$ when pooled with the ground truth G, and when non-pooled, together with the runs submitted by its same organization, with a fixed-cost pooling strategy (J). A low MAE

Table 6.1: Pool properties of test collections, for the original pool, and the synthesized "cleaned" pool. The cleaned pool is equivalent to a Depth@K with K equal to the one used to build the original pool.

Test Collection Properties						
	Ad Hoc 3		Ad Hoc 8		Web 9	
$\lvert\mathcal{D}\rvert$	263,509		528,155		1,692,096	
$\lvert\mathcal{R}\rvert$	40		130		104	
$\lvert\mathcal{R}_p\rvert$	23		74		62	
$\lvert\mathcal{O}\rvert$	22		41		23	
$\lvert\mathcal{Q}\rvert$	50		50		50	
K	200		100		100	
	Original →	Cleaned	Original →	Cleaned	Original →	Cleaned
$\lvert\mathcal{J}\rvert$	97,319	75,378	86,830	86,830	70,070	70,030
$\lvert\mathcal{J}^+\rvert$	9,805	9,287	4,728	4,728	2,617	2,616
	Robust 2005		Genomics 2005		Legal 2006	
$\lvert\mathcal{D}\rvert$	1,033,461		4,591,008		6,910,192	
$\lvert\mathcal{R}\rvert$	74		62		34	
$\lvert\mathcal{R}_p\rvert$	18		55		6	
$\lvert\mathcal{O}\rvert$	17		32		8	
$\lvert\mathcal{Q}\rvert$	50		49		38	
K	55		60		100	
	Original →	Cleaned	Original →	Cleaned	Original →	Cleaned
$\lvert\mathcal{J}\rvert$	37,798	22,173	39,958	38,604	31,041	18,929
$\lvert\mathcal{J}^+\rvert$	22,173	4,563	4,584	4,387	3,931	2,386
	Blog 2006		Microblog 2011		Web 2014	
$\lvert\mathcal{D}\rvert$	509,137		16,000,000		733,019,372	
$\lvert\mathcal{R}\rvert$	54		184		30	
$\lvert\mathcal{R}_p\rvert$	28		98		27	
$\lvert\mathcal{O}\rvert$	14		58		10	
$\lvert\mathcal{Q}\rvert$	50		49		50	
K	100		30		25	
	Original →	Cleaned	Original →	Cleaned	Original →	Cleaned
$\lvert\mathcal{J}\rvert$	67,382	60,207	60,129	26,370	14,432	12,334
$\lvert\mathcal{J}^+\rvert$	19,891	18,425	2,965	2,549	5,665	4,895

means that the score obtained by a run with J strategy when not pooled is close to the score obtained by the run when evaluated with the ground-truth.

The second measure of bias we present is system rank error (SRE). This measure counts the number of rank positions lost or gained by runs in the system ranking with respect to when it is pooled with the ground truth G, defined by the test collection, and not

Table 6.2: List of the pooling strategies analysed in this thesis where the columns refer, in order, to the pooling strategy type, the full name of the pooling strategy, its abbreviation, and the references to the equations of the document scoring function (s) and the set-building function (J) that formally define the pooling strategy.

Type	Pooling Strategy	Abbr.	s	J
Non-Adaptive	Depth@K	D	Eq. 6.1	Eq. 6.2
	Take@N	T	Eq. 6.3	Eq. 6.4
	FairTake@N	F	Eq. 6.5	Eq. 6.4
	BordaTake@N	B	Eq. 6.6	Eq. 6.4
	CondorcetTake@N	C	Eq. 6.8	Eq. 6.4
	CombMAXTake@N	MAX	Eq. 6.9	Eq. 6.4
	CombMINTake@N	MIN	Eq. 6.10	Eq. 6.4
	CombMEDTake@N	MED	Eq. 6.11	Eq. 6.4
	CombSUMTake@N	SUM	Eq. 6.12	Eq. 6.4
	CombANZTake@N	ANZ	Eq. 6.13	Eq. 6.4
	CombMNZTake@N	MNZ	Eq. 6.14	Eq. 6.4
	DCGTake@N	DCG	Eq. 6.15 & 6.16	Eq. 6.4
	RRFTake@N	RRF	Eq. 6.15 & 6.17	Eq. 6.4
	PPTake@N	PP	Eq. 6.15 & 6.18	Eq. 6.4
	RBPTake@N	RBP	Eq. 6.15 & 6.19	Eq. 6.4
Adaptive	MTFTake@N	MTF	Eq. 6.22 & Eq. 6.23	Eq. 6.21
	HedgeTake@N	H	Eq. 6.25	Eq. 6.21
	RBPAdaptiveTake@N	RBP^A	Eq. 6.26	Eq. 6.21
	RBPAdaptive*Take@N	RBP^{A*}	Eq. 6.27	Eq. 6.21
	MABGreedyTake@N	BG	Eq. 6.22 & Eq. 6.29	Eq. 6.21
	MABUCBTake@N	UCB	Eq. 6.22 & Eq. 6.30	Eq. 6.21
	MABBetaTake@N	BB	Eq. 6.22 & Eq. 6.31	Eq. 6.21
	MABMaxMeanTake@N	MM	Eq. 6.22 & Eq. 6.32	Eq. 6.21

pooled with a fixed-cost pooling strategy (J). We define SRE as:

$$\begin{aligned}\mathrm{SRE}(J_{\mathcal{R}_p}) = \sum_{r\in\mathcal{R}_p} \Big|\Big\{ & r' \in \mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\} : \\ & : f(r, J_{\mathcal{R}_p\setminus\{r''\in\mathcal{R}_p:o_{r''}=o_r\}}) \le f(r', G) < f(r, J_{\mathcal{R}_p}) \vee \\ & \vee f(r, J_{\mathcal{R}_p}) < f(r', G) \le f(r, J_{\mathcal{R}_p\setminus\{r''\in\mathcal{R}_p:o_{r''}=o_r\}})\Big\}\Big|\end{aligned}$$

A low SRE means that the rank position of the runs when not pooled using J is close to the rank position of the runs when pooled with the ground-truth. In IR when comparing ranking of runs, it is common practice to evaluate their significance. We implemented this in the next measure named system rank error with statistical significance (SRE*). SRE* is similar to SRE but instead of counting all the position differences of a run

against all the other runs, it counts only if significant according to a paired t-test with $p < 0.05$ calculated on the ground-truth. SRE* is defined as follows:

$$\mathrm{SRE}^*(J_{\mathcal{R}_p}) = \sum_{r \in \mathcal{R}_p} \Big| \Big\{ r' \in \mathcal{R}_p \setminus \{ r'' \in \mathcal{R}_p : o_{r''} = o_r \} : \\ : \Big(f(r, J_{\mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\}}) \leq f(r', G) < f(r, J_{\mathcal{R}_p}) \vee \\ \vee f(r, J_{\mathcal{R}_p}) < f(r', G) \leq f(r, J_{\mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\}}) \Big) \wedge \\ \wedge \text{ t-test}_{\text{paired}}(r, r', G) < 0.05 \Big\} \Big|$$

Juxtaposing these measures of bias we can observe that a zero MAE value implies that SRE and SRE* are also equal to zero. However, the contrary is not true. We can also observe that this is true between SRE and SRE*, where a zero SRE corresponds to a zero SRE*, but not *vice versa*.

6.3.4 Experimental Setup

In this paragraph we present the setup of the first three experiments, the first, designed to compare the pooling strategies, the second, where we compare the two different kind of pooling strategies, fixed-cost strategy and fixed-depth strategy, and the third, where we measure the expectation of the number of judged documents per run. For these three experiments, each pooling strategy takes as parameter the pool size, *i.e.*, the number of judged documents. To test how the different strategies behave for different values of this parameter, we repeated the experiment varying the pool size, for the first and third experiments, from 5,000 in steps of 5,000 till all the judgements of the test collection were used. We did this for Ad Hoc 3, Ad Hoc 8, and Web 9. For Blog 2006 we varied the pool size from 2,000 in steps of 2,000, and for Genomics 2005, Legal 2006, Microblog 2011, Robust 2005, and Web 2014 we varied the pool size from 1,000 in steps of 1,000 due to the smaller size of these test collections. For the second experiment we set the parameter N of the fixed-cost strategy in function of the number of documents judged, by setting the parameter K of the non fixed-cost strategy, varying K from 10 in steps of 10 to the maximum possible K.

In the forth experiment, when we verify the stability of the first experiment, for each pooling strategy we fix $N = 10,000$ we then repeated the experiment varying the horizon h from 10, in steps of 10 till the size of the original test collection K. We did this for all the test collections.

In all four experiments, the pool size N, when possible, is equally divided across the topics. Due to an imbalance of documents judged in the original Depth@K strategy among the topics, for big Ns and for some topics we would not find enough documents to cover the number of allocated judgements for these topics, $N/|\mathcal{Q}|$, where $|\mathcal{Q}|$ is the number of topics. In this case the number of judged documents available per topic can vary. Therefore, the aggregated number of documents to be judged for a fixed-cost

pooling strategy would not equal the desired pool size of N judged documents. To avoid this, we implement a heuristic that redistributes the remaining judgements, when needed, fairly across the rest of the topics that still have available documents. Given as input the set of pooled runs ($\mathcal{R}_p$) this heuristic does, in order to achieve the prefixed N pooled documents across topics, a search on the space of possible per-topic sizes. This search space is constrained by the fact that every per-topic size cannot be greater than the number of available judged documents per topic. The heuristic first starts by assigning to each topic q a per-topic size n_q equal to N divided by the number of topics ($N/|\mathcal{Q}|$). So for example, if we have a N of 10,000 documents for 50 topics the heuristic assigns to every topic an $n_q = 200, \forall q \in \mathcal{Q}$. Now, if for some topics the assigned n_qs are too large, for example there is a lack of documents to be judged for these topics the heuristic then reduces the n_qs of these topics to the maximum allowed (that is of course smaller than $N/|\mathcal{Q}|$) and reassigns the remaining judgements to the other topics for which there are still available documents. The reassignment is done by incrementing by 1 each topic n_q until one of the two conditions is verified: 1) the topic has been exhausted, that is no more documents are available, in this case the topic is excluded and the algorithm continues with the other topics, or 2) the sum of the n_qs has reached N ($n_1 + ... + n_{|\mathcal{Q}|} = N$), in this case the algorithm stops returning the found solution. However if this second condition is not verified before all the topics get exhausted the heuristic returns an error. This means that there are not enough documents already judged in the original pool to achieve a solution of size N.

The IR evaluation measures we selected for this study are AP, NDCG, and P@10. The reason for this selection is twofold: (a) these measures are widely used in IR, and (b) they encompass common features of most IR measures: top-heaviness, precision based, recall based, and utility based.

6.3.5 Results

In Figure 6.4 we show the bias evaluation obtained using the non-adaptive pooling strategies and in Figure 6.5 the bias evaluation obtained using the adaptive ones for the Ad Hoc 8 test collection. In the figures, each column is an IR evaluation measure while each row is a measure of bias. The x-axis in each of the plots is the number of judged documents, while the y-axis is the scale of the respective measure bias. Every line is a pooling strategy. In Figures 6.6 and 6.7 we show the same for all the other test collections but measuring MAE on the evaluation measure AP. In these figures we can observe that for some test collections like Blog 2006, Web 2014, and Legal 2006, the measured bias increases when increasing the number of judged documents N, notably also for the FairTake@N. This behaviour does not exist when tested on P@10, and it has to be because of the recall component of the measures AP and NDCG. While this is apparently disturbing, in fact, for the purposes of selecting which strategy to apply in the future, it does not change our conclusions.

In these figures we can observe that all lines converge to a pool bias value of the test collection (or zero when the baseline is subtracted) for large pool size values. This is

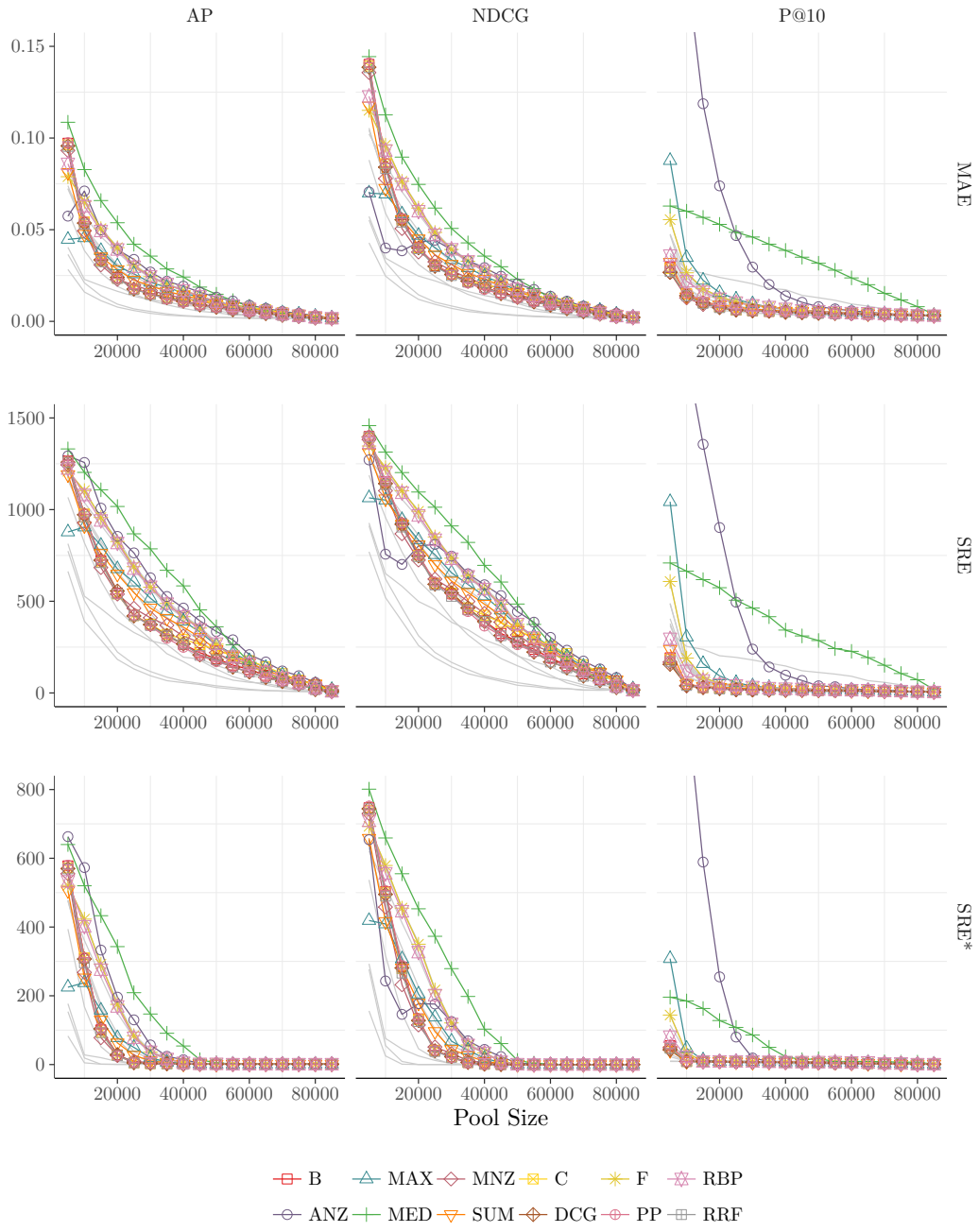

Figure 6.4: Pool bias measured for the *non-adaptive* pooling strategies in terms of the measures of bias (row): MAE, SRE, and SRE*, and IR evaluation measures (columns): AP, NDCG, and P@10. This is plotted by using the Ad Hoc 8 test collection, and for different pool sizes (*i.e.*, aggregated number per topic of documents that require relevance judgement). The lines in grey are the *adaptive* pooling strategies (in Figure 6.5) for comparison.

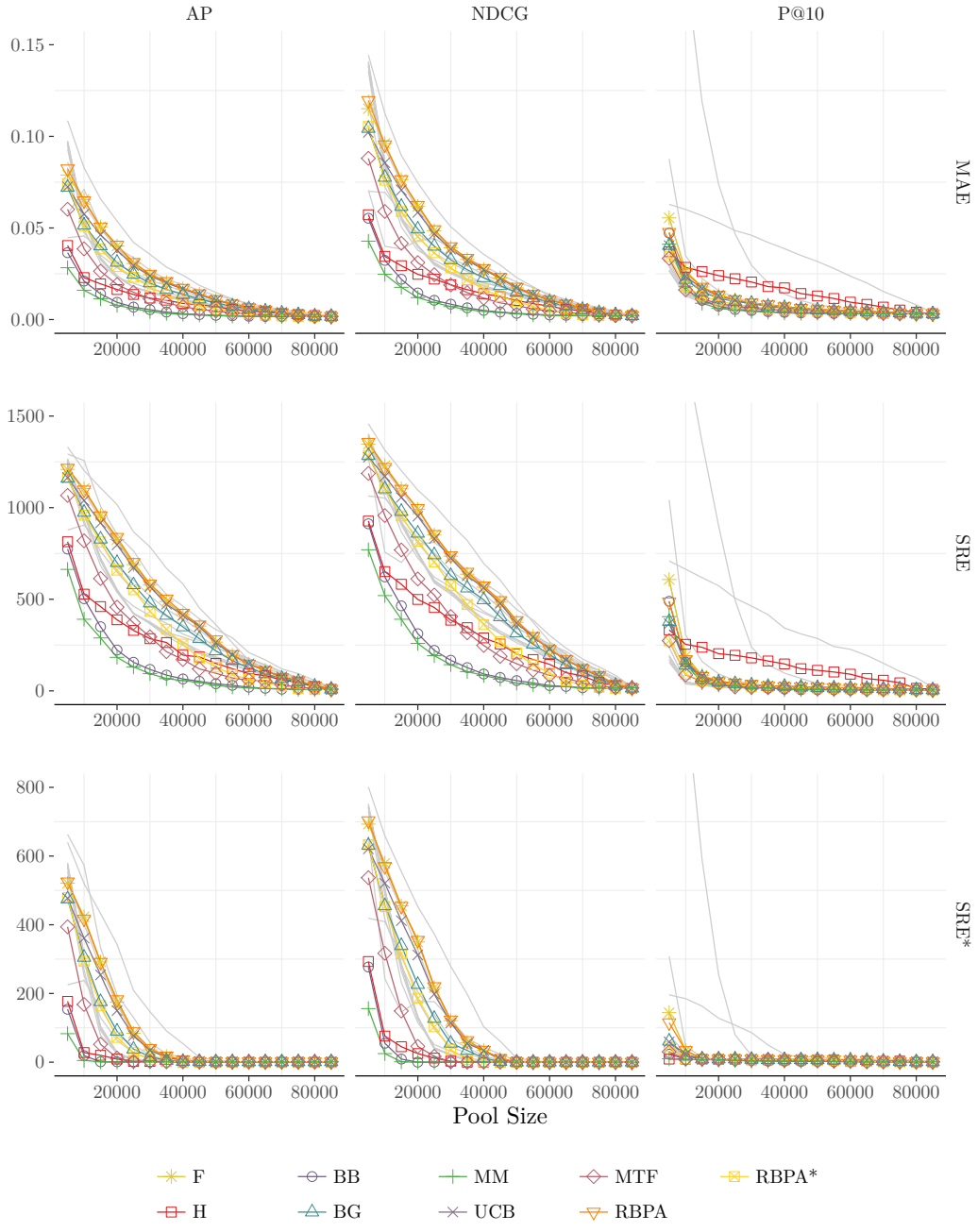

Figure 6.5: Pool bias measured for the *adaptive* pooling strategies in terms of the measures of bias (row): MAE, SRE, and SRE*, and IR evaluation measures (columns): AP, NDCG, and P@10. This is plotted by using the Ad Hoc 8 test collection, and for different pool sizes (*i.e.*, aggregated number per topic of documents that require relevance judgement). The lines in grey are the *non-adaptive* pooling strategies (in Figure 6.4) for comparison.

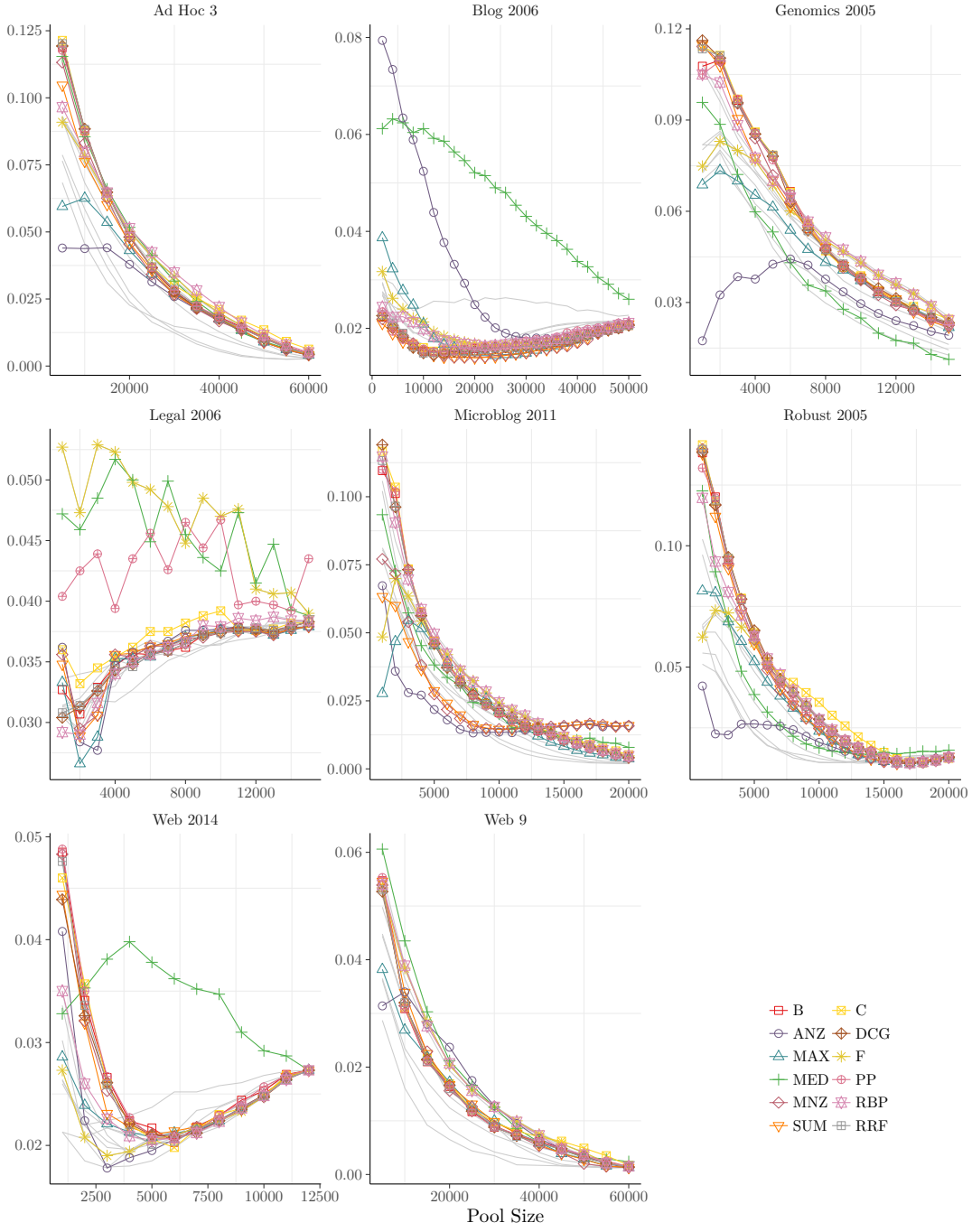

Figure 6.6: Pool bias measured for the *non-adaptive* pooling strategies in terms of the measure of bias MAE and IR evaluation measure AP, and for different pool sizes (*i.e.*, aggregated number per topic of documents that require relevance judgement). The lines in grey are the *adaptive* pooling strategies (in Figure 6.7) for comparison.

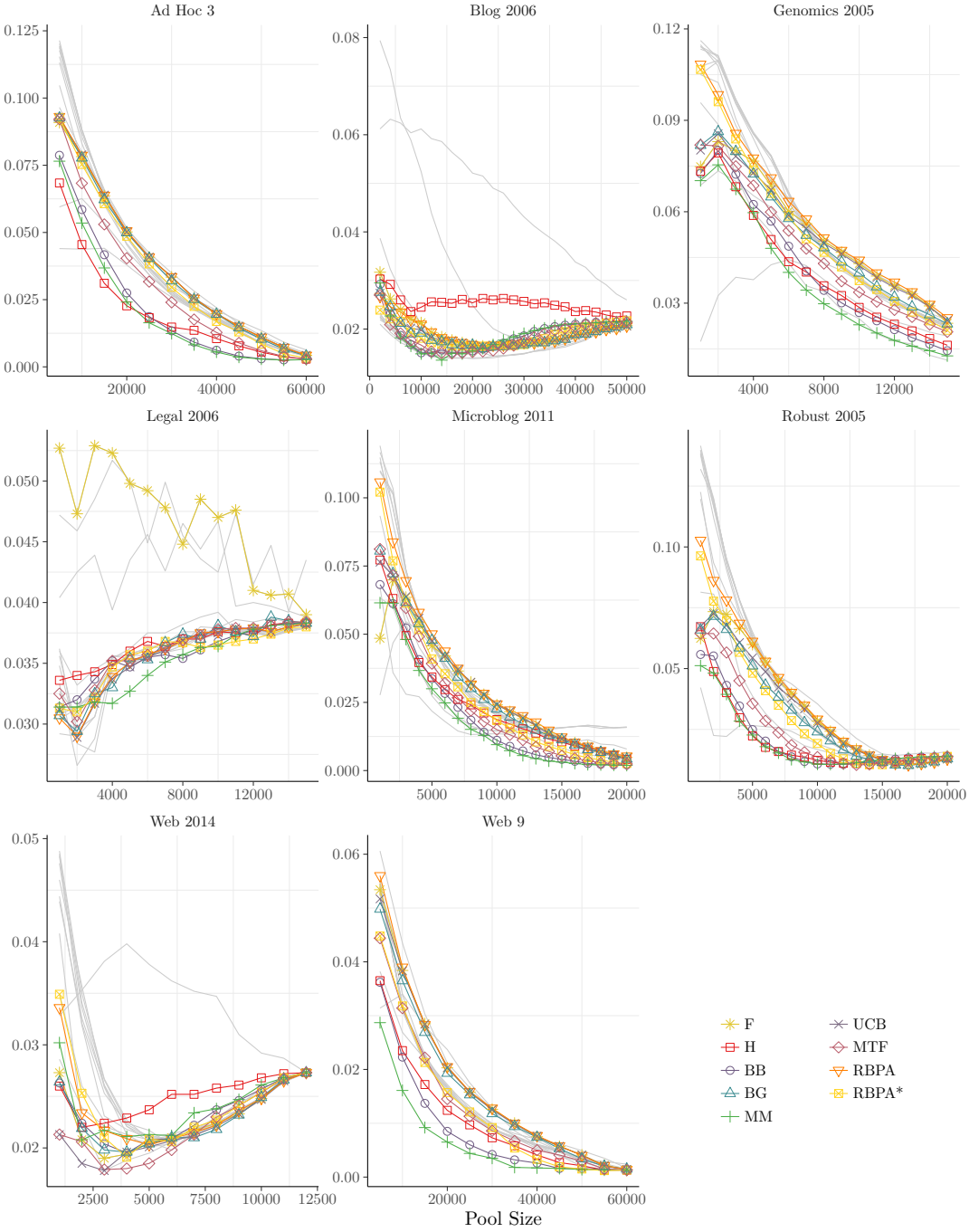

Figure 6.7: Pool bias measured for the *adaptive* pooling strategies in terms of the measure of bias MAE and IR evaluation measure AP, and for different pool sizes (*i.e.*, aggregated number per topic of documents that require relevance judgement). The lines in grey are the *non-adaptive* pooling strategies (in Figure 6.6) for comparison.

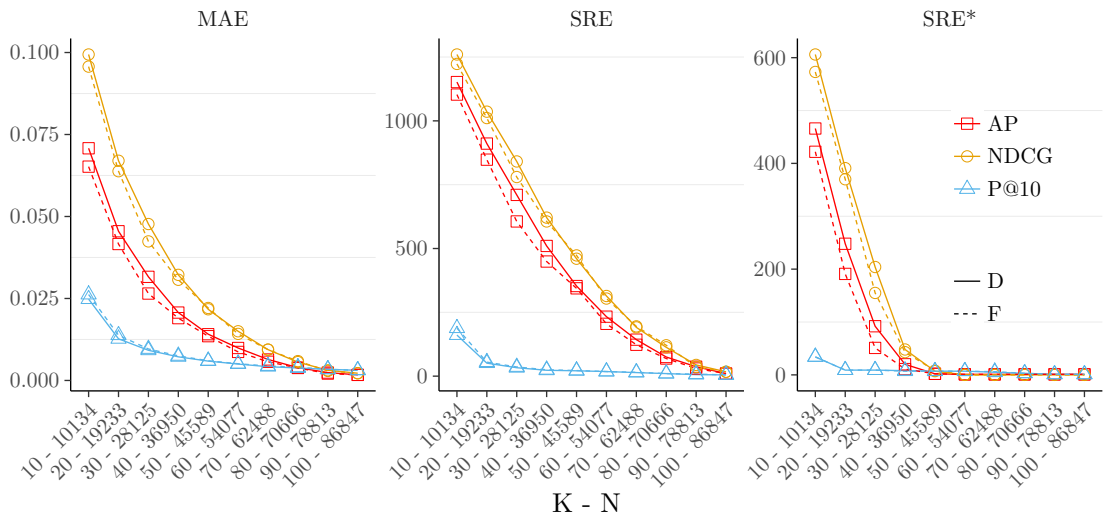

Figure 6.8: Pool bias measured for the Depth@K (D) strategy and FairTake@N (F) strategy in terms of the measure of bias (left to right): MAE, SRE, SRE*, and the IR evaluation measures: AP, NDCG, P@10. This is plotted by using the Ad Hoc 8 test collection.

because all the pooling strategies are constrained to select documents for which we have relevance assessments. This is done as explained previously by only including in the analysis pooled runs and by resizing them to the same depth of the Depth@K pooling strategy used to build the test collection. Thereby, all alternative pooling strategies will reduce to the original Depth@K strategy with K defined by the test collection.

In Tables 6.3, 6.4, 6.5, 6.6 and 6.7, we show the performance of each pooling strategy for $N = 10,000$.

In Figure 6.8 we show the comparison between the Depth@K strategy against the FairTake@N strategy for Ad Hoc 8. The values N are reported in the x-axis text beside the K values.

In Figures 6.9 and 6.10, we show the expected number of judged documents across runs and topics (JD) for Ad Hoc 8. The JD values give us an estimate of how many documents we should expect to be judged for a non pooled run and for a single topic. Every line is a pooling strategy, and the x-axis in each of the plots is the total number of judged documents, while the y-axis is the scale of JD.

In Figures 6.11 and 6.12, we show the stability of the results when varying the horizon of the pooling strategies for a fixed pool size $N = 10,000$ for Ad Hoc 8. Every line is a pooling strategy, and the x-axis in both figures is the horizon, while the y-axis is MAE, SRE, and SRE* measured on AP, NDCG, and P@10.

Table 6.3: Performance of the pooling strategies for N equal to 10,000.

C	Strat.	$\lvert\mathcal{J}^+\rvert$	AP			NDCG			P@10		
			MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Ad Hoc 3	*F*	2962	0.0782	137	76	0.1295	173	104	0.0347	34	**0**
	B	3801	0.0877	145	84	0.1353	171	104	0.0208	19	**0**
	C	3810	0.0877	146	85	0.1351	171	104	0.0203	19	**0**
	MAX	3243	0.0626	122	63	0.1058	155	88	0.0492	55	1
	MIN	1294	0.1383	221	154	0.1586	209	139	0.4212	254	165
	MED	2769	0.0855	148	86	0.1326	173	107	0.0619	70	5
	SUM	3690	0.0763	131	71	0.1193	162	95	0.0296	28	**0**
	ANZ	2352	**0.0438**	97	42	**0.0739**	126	63	0.0966	121	40
	MNZ	3744	0.0831	141	80	0.1270	165	98	0.0227	19	**0**
	DCG	3747	0.0884	148	86	0.1366	173	105	0.0202	18	**0**
	RRF	3736	0.0882	147	85	0.1360	172	104	**0.0187**	**17**	**0**
	PP	3788	0.0874	146	85	0.1353	171	104	0.0219	21	**0**
	RBP	2996	0.0797	141	79	0.1308	173	104	0.0328	32	**0**
	RBP^A	2960	0.0787	138	76	0.1299	171	103	0.0347	33	**0**
	RBP^{A*}	3201	0.0753	133	72	0.1235	167	99	0.0300	28	**0**
	H	4256	0.0455	**84**	**27**	0.0807	**125**	**58**	0.0428	46	**0**
	MTF	3676	0.0683	124	63	0.1104	156	89	0.0283	25	**0**
	BG	3151	0.0777	134	73	0.1262	170	101	0.0317	30	**0**
	UCB	3003	0.0789	137	76	0.1294	171	103	0.0342	33	**0**
	BB	3950	0.0585	108	47	0.0959	146	78	0.0292	28	**0**
	MM	**4265**	0.0535	98	39	0.0866	135	68	0.0330	35	**0**
Ad Hoc 8	*F*	1681	0.0655	1104	423	0.0961	1229	579	0.0265	190	34
	B	2193	0.0541	974	309	0.0858	1150	503	0.0150	46	**9**
	C	2193	0.0542	976	311	0.0860	1148	501	0.0150	47	**9**
	MAX	1939	0.0456	905	238	0.0694	1052	409	0.0348	305	47
	MIN	557	0.1333	2066	1356	0.1823	1987	1313	0.3728	2481	1697
	MED	1221	0.0828	1203	520	0.1126	1314	659	0.0601	664	185
	SUM	2328	0.0475	912	252	0.0726	1059	416	0.0134	48	**9**
	ANZ	675	0.0716	1267	583	0.0413	802	267	0.1929	1834	1051
	MNZ	2258	0.0494	928	268	0.0779	1103	458	**0.0128**	**38**	**9**
	DCG	2195	0.0536	972	307	0.0841	1142	495	0.0140	40	**9**
	RRF	2205	0.0530	961	296	0.0834	1136	490	0.0140	41	**9**
	PP	2188	0.0545	976	311	0.0864	1153	506	0.0154	50	**9**
	RBP	1782	0.0628	1080	402	0.0932	1206	556	0.0219	120	22
	RBP^A	1690	0.0649	1097	417	0.0954	1220	570	0.0255	171	34
	RBP^{A*}	2084	0.0511	959	294	0.0761	1100	453	0.0182	91	10
	H	2635	0.0229	528	28	0.0345	651	76	0.0285	254	**9**
	MTF	2464	0.0386	819	168	0.0590	957	317	0.0162	87	**9**
	BG	2053	0.0515	974	305	0.0776	1102	455	0.0219	140	17
	UCB	1903	0.0576	1039	361	0.0856	1171	521	0.0236	157	23
	BB	3019	0.0210	503	19	0.0323	622	55	0.0197	157	**9**
	MM	**3267**	**0.0160**	**391**	**5**	**0.0247**	**520**	**25**	0.0179	147	**9**

Table 6.4: Continuation of Table 6.3 for the rest of the test collections.

C	Strat.	$\lvert\mathcal{J}^+\rvert$	AP			NDCG			P@10		
			MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Web 9	*F*	947	0.0379	577	36	0.0504	510	56	0.0170	177	4
	B	1250	0.0309	476	21	0.0419	448	44	0.0088	103	1
	C	1249	0.0311	478	21	0.0424	454	45	**0.0086**	104	1
	MAX	1098	0.0268	448	17	0.0372	421	40	0.0188	206	4
	MIN	380	0.0790	1427	686	0.1366	1434	861	0.1967	1798	1057
	MED	675	0.0449	652	82	0.0529	567	119	0.0525	663	103
	SUM	1162	0.0339	511	25	0.0459	472	48	0.0098	112	**0**
	ANZ	595	0.0340	549	30	0.0232	262	9	0.0629	794	153
	MNZ	1227	0.0314	476	21	0.0428	452	46	0.0089	100	**0**
	DCG	1223	0.0320	491	22	0.0434	457	47	0.0088	98	1
	RRF	1229	0.0320	487	22	0.0431	457	45	0.0087	**97**	**0**
	PP	1256	0.0308	472	21	0.0419	446	43	0.0090	108	1
	RBP	977	0.0390	582	37	0.0513	511	54	0.0148	148	2
	RBP^A	945	0.0390	580	38	0.0518	511	55	0.0168	178	4
	RBP^{A*}	1122	0.0318	497	22	0.0425	462	45	0.0122	127	1
	H	1319	0.0236	400	9	0.0301	342	23	0.0151	162	1
	MTF	1197	0.0313	489	21	0.0410	445	44	0.0115	121	1
	BG	1021	0.0372	562	34	0.0478	495	53	0.0162	174	3
	UCB	973	0.0382	578	38	0.0506	508	54	0.0171	179	3
	BB	1415	0.0228	380	10	0.0296	327	19	0.0115	126	1
	MM	**1555**	**0.0156**	**269**	**2**	**0.0216**	**242**	**8**	0.0115	131	3
Web 2014	*F*	4058	**0.0247**	113	25	0.0353	110	**30**	0.1343	155	41
	B	4046	0.0249	118	26	0.0366	113	31	0.1354	157	41
	C	4024	0.0248	115	25	0.0353	**108**	**30**	0.1369	157	41
	MAX	4101	0.0249	115	25	0.0355	110	32	0.1364	155	43
	MIN	3974	0.0306	147	38	0.0433	142	42	0.1650	177	54
	MED	3963	0.0299	146	38	0.0419	134	39	0.1570	172	50
	SUM	4107	0.0249	117	26	0.0355	110	32	0.1355	155	42
	ANZ	4089	0.0248	114	25	0.0356	110	32	0.1360	155	42
	MNZ	4097	0.0249	117	25	0.0356	110	32	0.1353	154	42
	DCG	4056	**0.0247**	115	25	**0.0351**	109	**30**	0.1341	155	41
	RRF	4054	**0.0247**	114	25	**0.0351**	109	**30**	0.1342	155	41
	PP	4039	0.0253	121	27	0.0369	111	31	0.1379	156	41
	RBP	4078	0.0248	114	26	0.0353	109	**30**	0.1340	155	41
	RBP^A	4053	0.0248	113	25	0.0353	110	**30**	0.1344	155	41
	RBP^{A*}	4192	0.0254	121	27	0.0367	116	33	0.1314	153	41
	H	4169	0.0266	126	27	0.0390	124	34	0.1400	163	44
	MTF	4300	0.0254	122	26	0.0365	117	32	**0.1309**	**152**	41
	BG	4141	**0.0247**	118	**24**	0.0355	110	31	0.1332	154	41
	UCB	4099	0.0248	**112**	**24**	0.0355	110	31	0.1339	155	41
	BB	4332	0.0257	122	26	0.0375	118	33	0.1314	153	**40**
	MM	**4366**	0.0260	124	27	0.0384	120	34	0.1321	**152**	41

Table 6.5: Continuation of Table 6.4 for the rest of the test collections.

C	Strat.	$\lvert\mathcal{J}^+\rvert$	AP			NDCG			P@10		
			MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Robust 2005	*F*	2620	0.0285	51	11	0.0404	50	12	0.0447	35	11
	B	2889	0.0286	52	9	0.0395	51	12	0.0358	**26**	**10**
	C	2708	0.0354	60	17	0.0488	58	15	0.0389	27	**10**
	MAX	2775	0.0235	45	7	0.0342	41	7	0.0419	32	11
	MIN	2117	0.0625	106	57	0.0706	88	41	0.2317	121	66
	MED	2377	0.0174	36	8	0.0248	33	7	0.0969	77	29
	SUM	2848	0.0242	45	7	0.0345	42	7	0.0383	31	**10**
	ANZ	2673	0.0190	39	6	0.0280	37	6	0.0556	49	13
	MNZ	2907	0.0247	44	7	0.0350	41	7	0.0367	27	**10**
	DCG	2890	0.0284	50	9	0.0395	50	12	0.0359	**26**	**10**
	RRF	2891	0.0284	50	9	0.0395	51	12	0.0359	**26**	**10**
	PP	2917	0.0271	51	8	0.0382	48	10	0.0366	27	**10**
	RBP	2633	0.0285	51	11	0.0407	50	12	0.0431	34	11
	RBP^A	2607	0.0292	51	11	0.0412	52	13	0.0446	37	11
	RBP^{A*}	3034	0.0192	38	**5**	0.0274	36	5	0.0374	31	**10**
	H	3168	0.0125	25	**5**	0.0163	18	**2**	0.0506	42	11
	MTF	3374	0.0134	26	**5**	0.0191	24	**2**	0.0357	27	**10**
	BG	2805	0.0239	45	7	0.0347	43	9	0.0424	33	**10**
	UCB	2680	0.0276	50	11	0.0390	49	11	0.0452	35	11
	BB	3712	**0.0107**	23	**5**	0.0141	**16**	3	0.0340	27	**10**
	MM	**3791**	**0.0107**	**21**	**5**	**0.0135**	17	3	**0.0330**	27	**10**
Genomics 2005	*F*	1870	0.0437	785	88	0.0542	699	103	0.0288	398	2
	B	2170	0.0384	734	61	0.0488	662	77	0.0206	**291**	**0**
	C	2168	0.0388	735	62	0.0493	665	80	0.0212	308	**0**
	MAX	1952	0.0372	715	53	0.0480	646	68	0.0302	421	2
	MIN	1284	0.0622	1224	380	0.0854	1113	401	0.2347	1502	643
	MED	1485	0.0247	584	34	**0.0266**	471	23	0.0979	1113	269
	SUM	2055	0.0382	721	61	0.0497	666	84	0.0229	320	1
	ANZ	1662	0.0296	592	23	0.0359	523	25	0.0600	740	48
	MNZ	2142	0.0373	718	55	0.0484	657	76	0.0205	295	**0**
	DCG	2147	0.0377	724	56	0.0482	662	76	0.0208	304	**0**
	RRF	2170	0.0379	725	57	0.0484	656	73	**0.0203**	292	**0**
	PP	2178	0.0379	723	57	0.0485	659	76	0.0213	311	**0**
	RBP	1914	0.0436	788	90	0.0537	700	104	0.0268	367	1
	RBP^A	1876	0.0441	794	94	0.0545	709	108	0.0288	397	3
	RBP^{A*}	2034	0.0374	724	55	0.0468	637	59	0.0258	365	2
	H	2115	0.0289	616	22	0.0333	497	20	0.0409	541	2
	MTF	2184	0.0335	675	39	0.0430	613	48	0.0244	351	**0**
	BG	2016	0.0390	740	63	0.0494	654	67	0.0268	378	1
	UCB	1960	0.0423	773	80	0.0523	681	92	0.0275	386	1
	BB	2335	0.0267	560	13	0.0345	506	15	0.0266	372	**0**
	MM	**2422**	**0.0227**	**505**	**5**	0.0295	**448**	**8**	0.0253	348	**0**

Table 6.6: Continuation of Table 6.5 for the rest of the test collections.

C	Strat.	$\lvert\mathcal{J}^+\rvert$	AP			NDCG			P@10		
			MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Legal 2006	*F*	1280	0.0433	10	4	0.1006	8	1	0.1092	14	5
	B	1352	0.0378	**7**	**1**	0.0865	**7**	**0**	0.0961	**9**	**1**
	C	1317	0.0392	8	2	0.0891	**7**	**0**	0.0978	10	2
	MAX	1440	0.0377	**7**	**1**	0.0860	**7**	**0**	0.0939	**9**	**1**
	MIN	1220	0.0487	11	3	0.0989	9	2	0.1132	15	6
	MED	1253	0.0473	11	3	0.0980	9	2	0.1088	14	5
	SUM	1442	0.0375	**7**	**1**	0.0857	**7**	**0**	0.0939	**9**	**1**
	ANZ	1426	0.0378	**7**	**1**	0.0861	**7**	**0**	0.0939	**9**	**1**
	MNZ	1448	0.0374	**7**	**1**	**0.0854**	**7**	**0**	0.0930	**9**	**1**
	DCG	1361	0.0375	**7**	**1**	0.0864	**7**	**0**	0.0961	**9**	**1**
	RRF	1357	0.0374	**7**	**1**	0.0866	**7**	**0**	0.0965	10	2
	PP	1382	0.0456	11	3	0.0957	**7**	**0**	0.1066	15	6
	RBP	1327	0.0385	**7**	**1**	0.0875	**7**	**0**	0.0956	**9**	**1**
	RBP^A	1334	0.0376	**7**	**1**	0.0869	**7**	**0**	0.0961	**9**	**1**
	RBP^{A*}	1436	0.0365	**7**	**1**	0.0868	**7**	**0**	0.0912	**9**	**1**
	H	1570	0.0371	**7**	**1**	0.0887	**7**	**0**	0.0943	**9**	**1**
	MTF	1569	0.0378	**7**	**1**	0.0878	**7**	**0**	0.0917	**9**	**1**
	BG	1360	0.0383	**7**	**1**	0.0873	**7**	**0**	0.0978	10	2
	UCB	1340	0.0375	**7**	**1**	0.0872	**7**	**0**	0.0961	**9**	**1**
	BB	1719	0.0368	**7**	**1**	0.0870	**7**	**0**	0.0877	**9**	**1**
	MM	**1749**	**0.0364**	**7**	**1**	0.0866	**7**	**0**	**0.0868**	**9**	**1**
Blog 2006	*F*	4061	0.0213	117	19	0.0190	62	4	0.2205	219	96
	B	5612	0.0153	75	9	0.0272	85	12	0.1758	192	69
	C	5502	0.0160	78	10	0.0277	86	14	0.1833	193	70
	MAX	3189	0.0209	109	26	0.0206	68	6	0.2533	242	119
	MIN	3015	0.0681	264	161	0.0774	206	100	0.3961	278	155
	MED	3130	0.0612	255	152	0.0650	182	76	0.3761	269	146
	SUM	4898	0.0154	79	8	0.0288	82	11	0.1810	195	72
	ANZ	1783	0.0524	233	130	0.0533	169	63	0.3735	270	147
	MNZ	5587	0.0153	78	9	0.0282	87	12	**0.1669**	**185**	**62**
	DCG	5494	**0.0149**	74	9	0.0271	84	11	0.1734	190	67
	RRF	5525	**0.0149**	**70**	9	0.0276	86	12	0.1738	191	68
	PP	5629	0.0157	78	10	0.0267	82	12	0.1794	192	69
	RBP	4153	0.0198	110	15	0.0196	58	4	0.2139	217	94
	RBP^A	4035	0.0210	116	18	0.0190	60	4	0.2200	221	98
	RBP^{A*}	4814	0.0181	103	12	0.0189	**56**	3	0.1956	203	80
	H	5171	0.0245	114	23	0.0200	74	3	0.2449	242	119
	MTF	5657	0.0167	99	9	0.0179	59	3	0.1821	194	71
	BG	4582	0.0192	109	16	0.0190	62	4	0.2057	212	89
	UCB	4393	0.0198	113	19	0.0194	60	4	0.2130	216	93
	BB	6191	0.0151	93	8	0.0164	61	3	0.1764	190	67
	MM	**6623**	**0.0149**	93	**7**	**0.0156**	**56**	**2**	0.1791	197	74

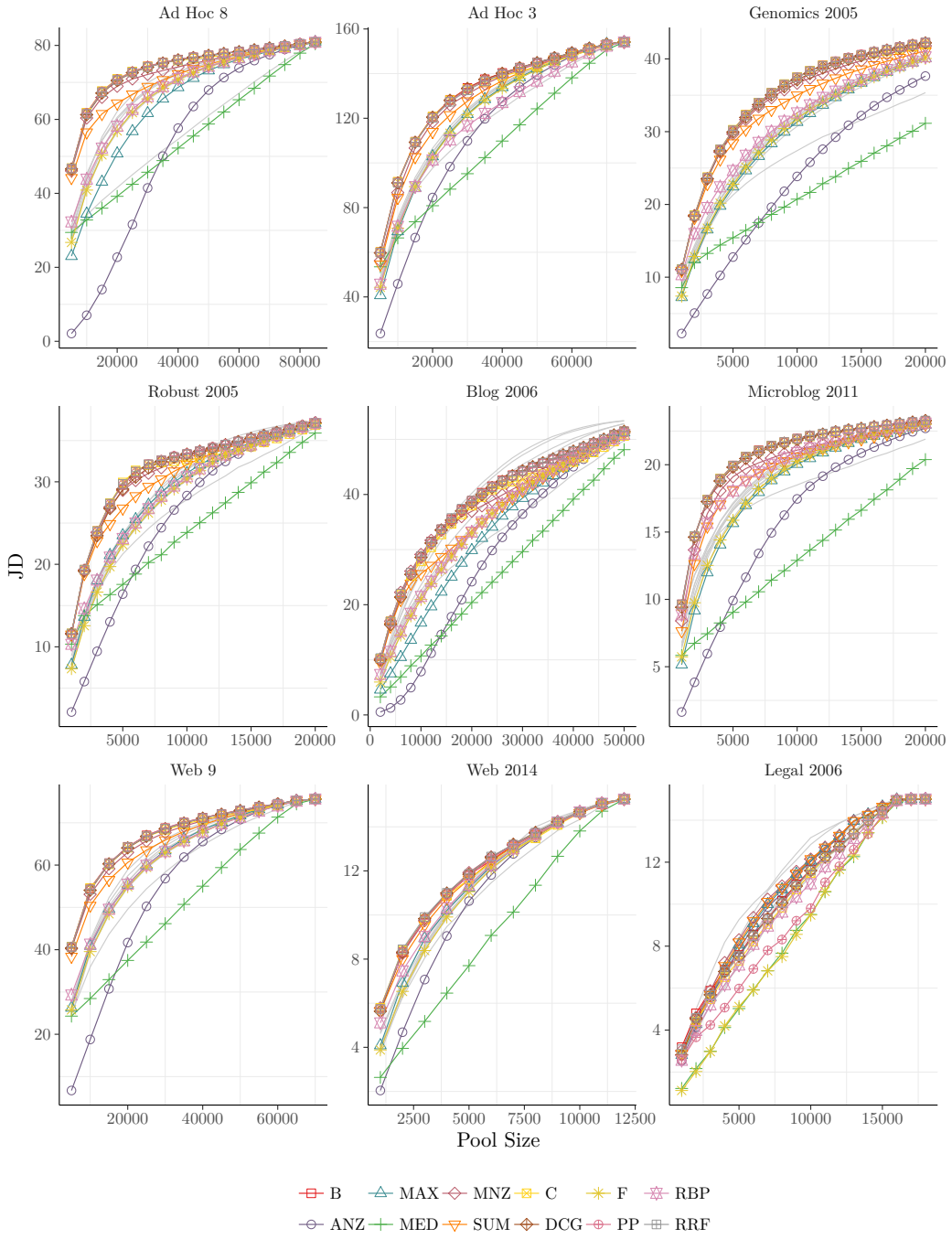

Figure 6.9: Expected number of judged documents for the pair run-topic (JD), for non pooled runs tested on all 9 test collections against all *non-adaptive* pooling strategies. This is plotted in function of the different pool sizes (*i.e.*, aggregated number per topic of documents that require relevance judgement). The lines in grey are the *adaptive* pooling strategies (in Figure 6.10) for comparison.

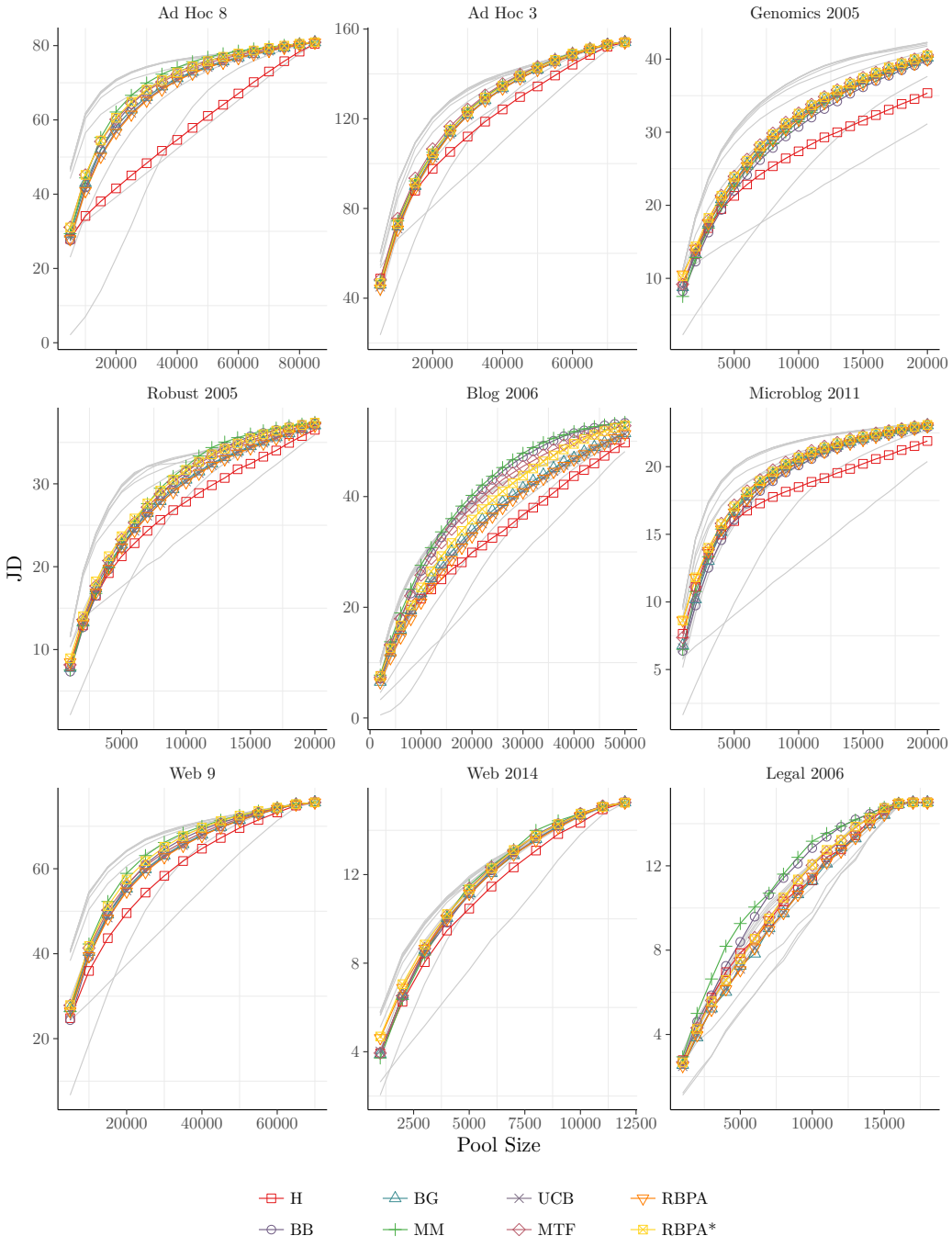

Figure 6.10: Expected number of judged documents for the pair run-topic (JD), for non pooled runs tested on all 9 test collections against all *adaptive* pooling strategies. This is plotted in function of the different pool sizes (*i.e.*, aggregated number per topic of documents that require relevance judgement). The lines in grey are the *non-adaptive* pooling strategies (in Figure 6.9) for comparison.

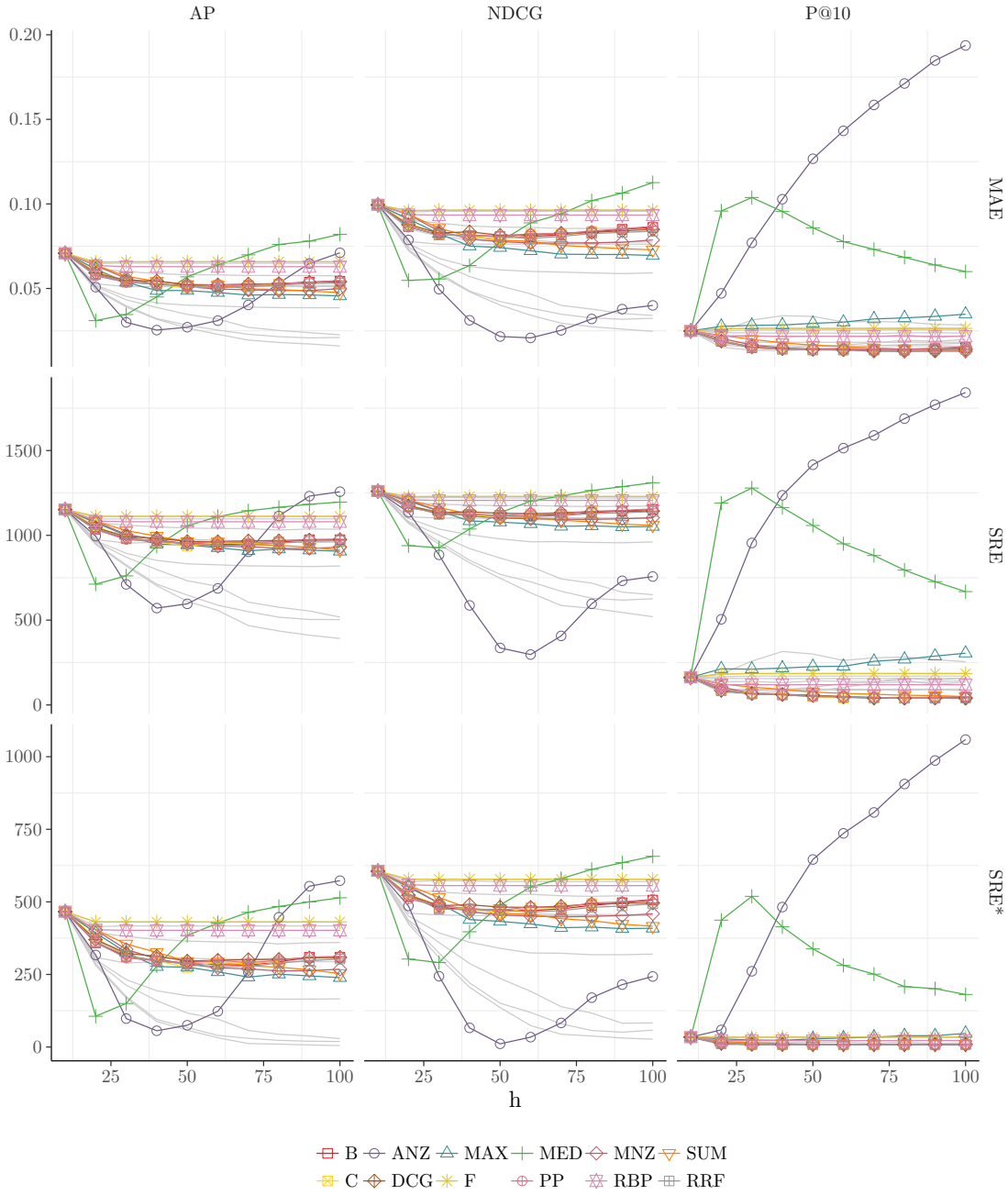

Figure 6.11: Pool bias measured for the *non-adaptive* pooling strategies in terms of the measures of bias (row): MAE, SRE, and SRE*, and IR evaluation measures (columns): AP, NDCG, and P@10. This is plotted by using the Ad Hoc 8 test collection, and for different horizons (*i.e.*, depth of the runs used by the pooling strategies). The lines in grey are the *adaptive* pooling strategies (in Figure 6.12) for comparison.

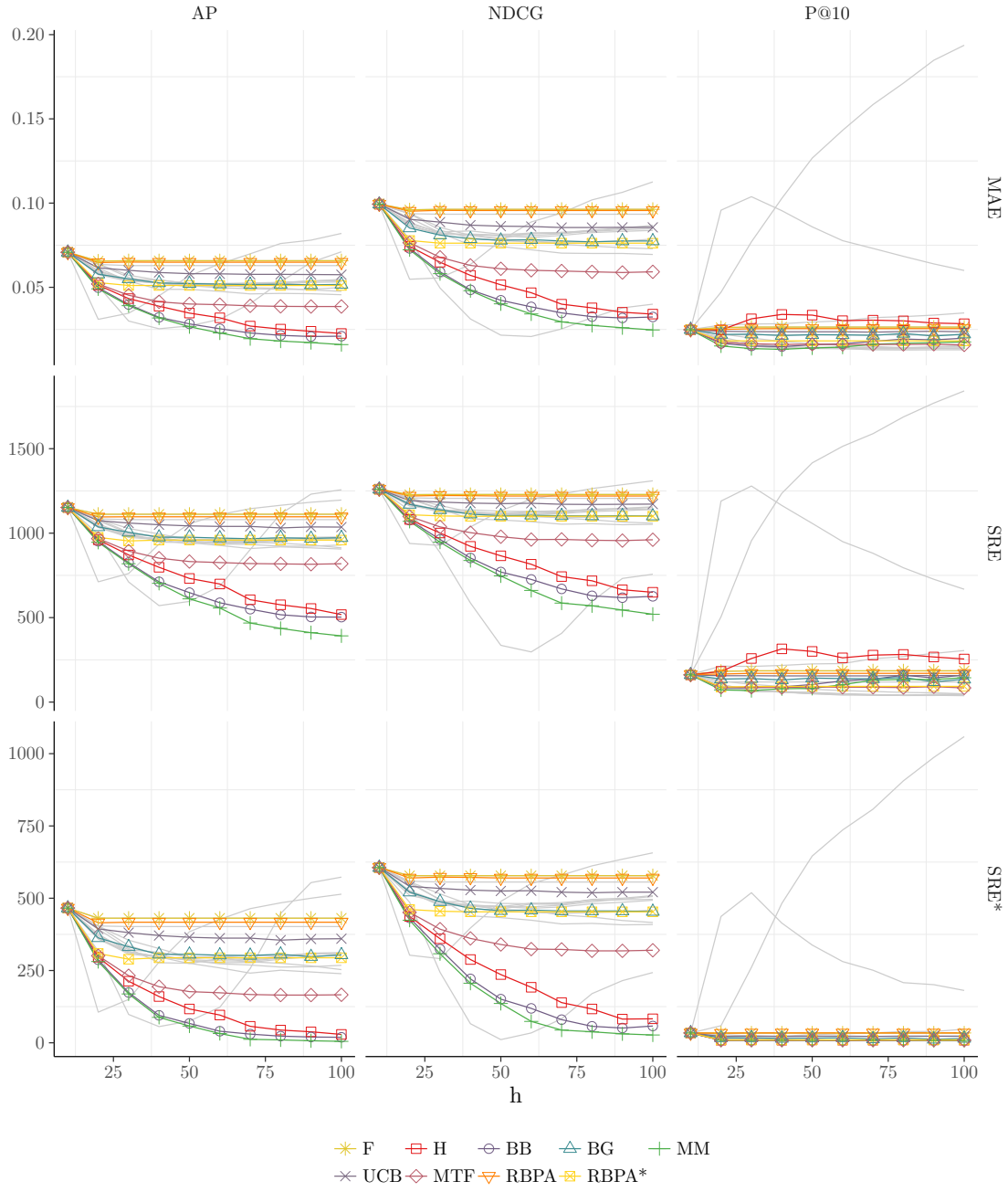

Figure 6.12: Pool bias measured for the *adaptive* pooling strategies in terms of the measures of bias (row): MAE, SRE, and SRE*, and IR evaluation measures (columns): AP, NDCG, and P@10. This is plotted by using the Ad Hoc 8 test collection, and for different horizons (*i.e.*, depth of the runs used by the pooling strategies). The lines in grey are the *non-adaptive* pooling strategies (in Figure 6.11) for comparison.

Table 6.7: Continuation of Table 6.6 for the rest of the test collections.

C	Strat.	$\lvert\mathcal{J}^+\rvert$	AP			NDCG			P@10		
			MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Microblog 2011	*F*	1679	0.0241	1049	10	0.0317	1146	16	0.0181	427	6
	B	1864	0.0207	915	7	0.0271	996	12	0.0107	253	4
	C	1870	0.0211	934	7	0.0274	1007	13	0.0110	262	5
	MAX	1740	0.0216	964	9	0.0281	1043	14	0.0187	457	6
	MIN	1067	0.0717	3337	1246	0.0902	3307	1312	0.2179	4344	2410
	MED	1196	0.0201	1078	22	0.0246	992	31	0.1094	2959	1073
	SUM	1793	0.0145	749	2	0.0163	695	1	0.0247	598	5
	ANZ	1524	0.0135	683	1	0.0155	652	**0**	0.0521	1416	57
	MNZ	1825	0.0146	760	2	0.0166	710	1	0.0232	572	6
	DCG	1849	0.0215	947	7	0.0277	1010	13	0.0107	255	**2**
	RRF	1866	0.0211	931	7	0.0274	1006	12	0.0108	252	3
	PP	1871	0.0202	894	7	0.0269	992	11	0.0107	256	6
	RBP	1741	0.0244	1063	12	0.0317	1140	16	0.0136	328	4
	RBP^A	1682	0.0244	1065	12	0.0317	1150	20	0.0173	414	7
	RBP^{A*}	1875	0.0183	816	6	0.0240	880	4	0.0122	288	**2**
	H	1880	0.0187	828	6	0.0230	854	4	0.0163	402	4
	MTF	1989	0.0150	662	2	0.0201	776	3	0.0103	244	**2**
	BG	1764	0.0226	997	9	0.0295	1070	14	0.0142	340	5
	UCB	1723	0.0242	1052	10	0.0316	1141	17	0.0155	376	3
	BB	2127	0.0111	530	1	0.0143	574	1	0.0096	239	**2**
	MM	**2186**	**0.0095**	**465**	**0**	**0.0123**	**496**	**0**	**0.0089**	**225**	**2**

6.4 Discussion

In the following we discuss the results reported above. We consider the FairTake@N strategy as our baseline. While this strategy is slightly different from Depth@K (see Section 6.2), FairTake@N is the strategy closest to Depth@K that guarantees full control over the number of documents to be assessed.

We start our discussion analysing the operationability of a pooling strategy. Next, we focus on the non-adaptive strategies, then we analyse the adaptive ones. Finally, we compare them to each other.

6.4.1 Pooling Operationability

The operationalisation of a pooling strategy refers to the flexibility that a strategy gives to the test collection builder in gathering the relevance assessments. If a pooling strategy does not impose a constraint on how to gather this information, then we say that this pooling strategy is operationalisable. The advantage of such strategies is that the two processes, pooling and assessing of the documents, are independent. This lack of interdependency, since the assessments are performed by human beings, makes it easier to tackle the cognitive biases that may affect the assessors while performing the

judgements. The standard way to address these biases is to make the assessors judge a randomised sample of the pooled documents. In general we identify the following operationability properties of a pooling strategy: aggregable, ordinable, and parallelisable. In the following discussion we will be primarily concerned with distinguishing those pooling strategies that do not have one or more of these properties.

A pooling strategy is *aggregable* when the collection builder is able to aggregate relevance assessments for a document across judgements from *multiple assessors*. Pooling strategies that do not present this quality put an additional burden on the collection builder. This is because these strategies require information about the relevance of documents already assessed to decide which documents to pool next. Thereby a non aggregable strategy requires that the assessment process is coordinated such that the assessment and selection of the next document to assess cannot start until all assessors have judged the current document: this may happen at different times due to different assessor cognitive abilities, workload, and work scheduling.

A pooling strategy is *ordinable* when the collection builder is able to control in which order the relevance assessments are performed. The absence of such a property may introduce cognitive biases. For example, some pooling strategies may favour such a bias because it requires the judgement of documents in order of their predicted relevance. This bias is instead usually overcome by the ordinable pooling strategies by randomising the pooled documents before presenting them to the assessors.

For the *parallelisable* property of a pooling strategy we can distinguish two parallelisation forms, cross-topic and per-topic parallelisations. The former refers to parallelising the assessments by judging at the same time multiple topics, *i.e.*, exclusively assign each topic to an assessor, but assigning different topics to different assessors. The latter refers to parallelising, given a topic, the assessments for this topic, *i.e.*, distributing documents that are retrieved for the same topic across multiple assessors to speed up the assessment process. While the former is often possible, the latter, always preferable, is sometimes difficult to obtain.

All non-adaptive pooling strategies are aggregable, ordinable, and fully parallelisable; for the adaptive pooling strategies, all but RBPAdaptiveTake@N are only cross-topic parallelisable.

6.4.2 Non-Adaptive Strategies

Among the voting system-based strategies, we observe that BordaTake@N performs slightly better than the CondorcetTake@N in all evaluation measures, although BordaTake@N is a relaxation of CondorcetTake@N. Both strategies are better than FairTake$N$ when used with P@10 and only initially worse when used for AP and NDCG. CondorcetTake@N has the issue that when comparing pairs of documents, if the two are not in the top K of the run, it neither adds nor subtracts anything from the value this strategy computes for the pair. This may lead to situations where it is impossible to compute a complete ordering of documents, *e.g.*, in the situation where a document d_i is preferred

to d_j, d_j to d_k, and also d_k to d_i. To bypass this theoretical limitation Montague and Aslam [MA02] implemented a sorting method that avoids this limit case, but also does not guarantee an optimal result (compare Algorithms 3 and 2 in the original paper [MA02]), while in this chapter a better solution was found by using Copeland's method.

Among the retrieval fusion based-pooling strategies, as expected, we observe a poor performance of the CombMINTake@N strategy. In fact it clearly performs worse than the FairTake@N baseline across all IR evaluation measures and measures of bias for all test collections. The strategy CombMINTake@N prefers the lowest scoring documents and is therefore likely to identify mostly irrelevant items, making the final (evaluation) scores highly unstable. This happens also to CombMEDTake@N for all but one test collection (Microblog 2011). The strategy CombANZTake@N usually performs poorly with all measures of bias except when computed on the IR evaluation measure NDCG. The strategy CombMAXTake@N performs consistently better than the baseline with all the IR evaluation measures but one, P@10. The strategies CombMNZTake@N and CombSUMTake@N behave similarly across both evaluation and bias measures. These strategies are better than FairTake@N when used with P@10 and only initially worse when used for AP and NDCG.

Among the evaluation measure based-pooling strategies, DCGTake@N, PPTake@N, and RRFTake@N correlate with each other, while RBPTake@N does not. They all tend to be better than the baseline only for P@10 and worse initially for NDCG and AP. RBPTake@N is the most conservative. Based on Figure 6.2, we observe that the rank of the non-adaptive strategies is perfectly correlated with their speed of discount (change in reward for popularity) for RRFTake@N, DCGTake@N, and PPTake@N, with the exception of RBPTake@N. The linear and logarithmic discounts remove the rank information from the documents rewarding more popular documents amongst the runs. The relationship between the discount and the top-heaviness of the evaluation measures AP and NDCG also explains the twist in preference, where FairTake@N is preferred for low N, then for higher N almost all non-adaptive methods outperform it, before they all converge to the same value. For P@10 we observe that DCGTake@N, RRFTake@N, and PPTake@N are the best, followed by RBPTake@N.

Juxtaposing all the non-adaptive strategies we observe that the voting system-based strategy BordaTake@N behaves similarly to the retrieval fusion method-based strategy CombMNZTake@N; and voting system-based strategies and IR evaluation measure-based strategies partially correlate with the retrieval fusion method-based strategy CombSUMTake@N.

For the non-adaptive pooling strategies we can conclude that the most stable strategy is CombMAXTake@N. However, if the measure to be optimised is only P@10, DCGTake@N should be preferred. This is not only based on Ad Hoc 8 (Figure 6.4), but is clearly visible for all test collections in Tables 6.3, 6.4, 6.5, 6.6 and 6.7. However, although a selected non-adaptive pooling strategy performs better than the baseline, the collection builder, at the cost of losing some operationability properties, can move to lesser biased pooling strategies in the next category, the adaptive ones.

6.4.3 Adaptive Pooling Strategies

Between the two classic pooling strategies we observe that the traditional MTFTake@N pooling strategy outperforms the baseline in every evaluation measure and test collection. This strategy is one of the most stable pooling strategies across IR evaluation measures, and on average discovers over 25% of relevant documents more than the baseline. The HedgeTake@N strategy outperforms MTFTake@N in all IR evaluation measures but P@10, and in all test collections but Blog 2006 where HedgeTake@N fails for AP and NDCG when compared against FairTake@N. We can observe that although HedgeTake@N discovers on average 27% more relevant documents than the baseline, it is not effective in reducing the bias. This happens in the case of Blog 2006 where the strategy is worse than the baseline. The reason for this failure has to be found in the parameter β that has been trained using test collections with a lower rate of relevant documents. In fact we predicted that increasing β from 0.1 to 0.9 would have increased the performance of HedgeTake@N to become higher than the baseline. This can be observed by the fact that when $\beta = 1$ this strategy reduces to an unbounded RRFTake@N like strategy (see Appendix A.2), whose performance for AP is better than the baseline.

Between the two IR evaluation measure-based pooling strategies we observe that the performance of the RBPAdaptiveTake@N strategy is comparable to the FairTake@N. The RBPAdaptive*Take@N strategy outperforms the baseline in every evaluation measure and test collection.

Among the multi-armed bandit-based strategies the MABUCBTake@N strategy performs comparably to the FairTake@N strategy. Among MABGreedyTake@N, MABBetaTake@N, MABMaxMeanTake@N, they all outperform the baseline for all IR evaluation measures and bias measures. In particular MABMaxMeanTake@N is the best performing pooling strategy in terms of bias.

Comparing all the adaptive pooling strategies, we observe that RBPAdaptive*Take@N, MTFTake@N, MABGreedyTake@N, and MABMaxMeanTake@N are always better than the baseline for every IR evaluation measure. For the adaptive pooling strategies we can draw the following conclusion: the least biased pooling strategy is MABMaxMeanTake@N. It is interesting to observe that this pooling strategy is the one that discovers the highest number of relevant documents, above 45% more than the baseline.

6.4.4 Non-adaptive vs. Adaptive Pooling Strategies

We now consider all the tested pooling strategies together. We observe that the best pooling strategy is MABMaxMeanTake@N for all test collections. However if some operationalisation properties are required, the CombMAXTake@N should be preferred. Overall the adaptive pooling strategies demonstrate to be more stable across IR evaluation measures. In fact RBPAdaptive*Take@N, MTFTake@N, MABGreedyTake@N, and MABMaxMeanTake@N always perform better than the baseline.

6.4.5 Accuracy and Stability of the Results

As discussed in Section 6.3.1, this experimental design raises three potential issues.

About the comparison between Depth@K strategy and FairTake@N, Figure 6.8 shows that the behaviour of the bias of these two pooling strategies is similar despite the substantial difference that exist between them, *i.e.*, the first lets the number of documents to be judged to vary on a per topic basis, while for the second strategy this number is fixed for all the topics. We can observe that FairTake@N is a stronger baseline when the measures of bias are computed on the IR evaluation measures AP and NDCG, and as good as Depth@N when the measures of bias are computed on P@n, in particular for SRE*.

About the inconclusiveness of the results due to having too few documents judged in the non-pooled runs, Figure 6.10 tells us, indeed, about the accuracy of the computation of the term, $f(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}})$, which is present in all three bias measures. For example, if we consider the non-adaptive pooling strategy CombANZTake@N, we observe that for Ad Hoc 8, with a pool size $N = 5,000$, the expected number of documents judged per run per topic is around 2.10, which means that when computing an IR evaluation measure on these non pooled runs, their accuracy is probably compromised. However, because such pool sizes are still used, and there are no available guidelines in the literature on how many judged documents are really necessary, we chose to provide these plots to let the readers assess the results by themselves.

About the stability of the results when changing the horizon of the pooling strategies, Figures 6.12 and 6.12 shows that all the best pooling strategies but two are consistent with the results discussed above. In fact, the best strategies continue to be the best also when changing horizon. The two pooling strategies that show an unstable behaviour are CombMEDTake@N and CombANZTake@N, which favour lower horizons. This experiment shows that the pooling strategies are stable when increasing their horizon. Based on this observation, we expect them to be consistent when increasing their horizon beyond the tested one.

6.4.6 Limitations

There are still a number of limitations and possible extensions to this work. First and foremost, we are constrained by the data available to us. As we have detailed in the beginning of Section 6.3, we do not see an alternative to a proper investigation of pool bias without "cleaning" the test collections and generating runs that have no documents beyond what we know to be evaluated. Nevertheless, this does significantly reduce the "knowledge" available to us as we have to discard a non-negligible percentage of the ground-truth. We see addressing this as a significant effort, to be perhaps undertaken as a new evaluation effort in TREC. Our study would hopefully serve as a initial step, to identify those pooling strategies that should be further tested in the context of such a large scale evaluation exercise.

Beyond this, a limitation that has appeared as we were presenting the various pooling strategies is the setting of their parameters. Throughout this chapter we have considered only parameters that have been published in previous works, but often enough these parameters were used for different purposes (retrieval fusion methods, IR evaluation measures) and maybe different values would be better fitted for pooling strategies.

There are a number of decisions that are taken in every evaluation campaign, that complement the pooling strategy itself. The number of runs, the number of topics, the distribution of evaluation effort over topics are all elements that are worth further investigation in relation to the pooling strategy. Finally, as the title clearly indicates, we focus here on pools of a fixed size. While this is often a real-world constraint, the study of variable-sized pools and the balance between the effort to assess another document and the bias reduction expected from this effort is equally worth pursuing.

6.5 Summary

In this chapter we have explored a large array of pooling strategies, from the standard Depth@K (closely approximated here by FairTake@N in the context of fixed-cost pooling) to recent strategies based on voting systems, retrieval fusion methods, IR evaluation measures, and multi-armed bandits methods. In doing so, we have observed parallels between strategies that had been developed independently (*e.g.* BordaTake@N and CondorcetTake@N, or HedgeTake@N and RRFTake@N) and distinguished between adaptive and non-adaptive pooling strategies, with their different operationalisations.

The baseline, FairTake@N remains a solid candidate, but it can be improved upon. If we have constraints on operationalisation and are therefore mandated to use a non-adaptive method, then CombMAXTake@N (using the maximum score obtained by a document across the runs) would be recommended, particularly when top-heavy metrics like AP and NDCG are the target evaluation metrics. There is one exception, the Blog 2006 test collection, where the RBPTake@N provided better results. However, the Blog 2006 collection is quite unusual: compared to all other test collections, it has an extremely high percentage of relevant documents being judged and the runs are very diverse (because topical relevance was not the main objective of the evaluation in that collection). However, for all test collections, if the measure to be optimised is P@10, DCGTake@N should be preferred.

If, however, adaptive pool generation is operationalisable (*i.e.*, including feedback from assessors in the pool generation process), we should use a multi-armed bandit-based method, MABMaxMeanTake@N, which is the least biased among all the tested pooling strategies; moreover, it is the strategy that discovers the highest number of relevant documents, on average 40% more than the baseline.

In the course of this study we have also observed that the ability of a pooling strategy in discovering a high number of relevant documents is somewhat correlated with the less biased ones, but not completely, *e.g.* the best non-adaptive strategy, CombMAXTake@N,

discovers a number of relevant documents comparable to the baseline but performs better in terms of bias than other non-adaptive strategies that discover on average even more than 15% relevant documents than the baseline. This verifies the statement made by Spärck Jones about the aim of the pooling strategy: a pooling strategy's objective is not to discover the highest number of relevant documents, but to discover an unbiased set of documents [Spä03].

CHAPTER 7

Selection Bias: Evaluation Measures

The increasing informatisation of our society is the spawn of many information rich domains. The strong empirical nature of IR and the variety of the solutions required for these domains has led to the partition of the IR community into smaller groups with more niche interests. This separated effort to bridge as many domains as possible has resulted in the building of lower quality, but sometimes unique, test collections. In this chapter we analyse the pool bias observed in these test collections and present solutions to mitigate it for two well-known IR evaluation measures, P@n and R@n. We chose these evaluation measures because they are cornerstones of IR evaluation practice and to satisfy the emerging need of having more valid and easy to interpret evaluation measures. In particular, in this chapter, we start by explaining how the pool bias affects P@n and R@n. We then present the bias estimators under a coherent mathematical framework to ease their comparison. To evaluate these estimators, we run a large scale experimentation using 15 purposefully chosen TREC test collections, and three measures of bias.

7.1 Introduction

A test collection is a valuable resource for Information Retrieval (IR) researchers because it gives the IR community a common ground to facilitate the development of search models. Numerous test collections have been developed in the field since the first Cranfield experiments in the 1960s. Since the start of TREC in the 1990s, this creation happens at a rate of approximately 25 test collections per year. A test collection is composed of: a set of documents, a set of topics and a set of relevance assessments for each topic, derived from the collection of documents. The number of documents in the collection generally makes the full judgement of the document set for every topic infeasible. Therefore, the relevance assessment process is generally optimised by pooling the top K documents for each run. The pool is constructed from systems taking part in the challenge for which the collection was made, at a specific point in time, after which the collection is generally frozen in terms of relevance judgements. The pooling method aims to identify an unbiased sample of relevant documents. Nevertheless, pool bias negatively affects the score of unpooled runs – those of systems not present at the time of test collection creation. This is a drawback that ultimately affects the reliability of the test collection. The variables controlling this reliability are [LH05]: the number of topics and their representativeness of the information needs of the target user, the number of documents assessed per run, and, last but not least, the diversity of the pooled systems (often however only assessed as the cardinality of the set of runs).

In the last decades the IR community have branched out significantly in a variety of domains and applications, with the creation of specific IR test collections focusing on specific problems. At the same time, benchmarking techniques developed in the IR community are being implemented in industry. Information aware companies request measures to quantify the quality of their information access systems in general, and search systems in particular. With a narrower focus however, the effort to successfully solve the challenges facing the creators of test collections takes on new significance. Most notably, it is often difficult to acquire a sufficient number of participants and diverse systems in order to fulfil the required run diversity to guarantee a reliable test collection.

In this chapter, we analyse the problem of pool bias at the evaluation measure level by presenting multiple pool bias estimators, under the same mathematical framework. We do this based for Precision at cut-off (P@n) and Recall at cut-off (R@n). There are two reasons to consider such a simple measures. First, they are a cornerstone for many other measures developed for the most popular of user models at present: the web user [HJ10]. Second, they are easy to understand by all users. This understandability of the IR measures has drawn moderate attention from our community recently [CS14]. Our own experience in the industry leads us to believe that when results are not presented as simply precision and recall, any numbers are just assumed to be precision or recall. Decision makers at lower or higher levels, trying to make sense of AP, or any other commonly used measure in our community, will most often read 0.12 as 12% and simply assume that either 12% of documents are relevant or 12% of relevant documents have been returned on average. Of course, we do not forget why all the other measures have

been invented to replace, or complement, precision at cut-off and recall at cut-off: (1) for an ideal run, if the topic has fewer relevant documents than n, P@n and R@n do not reach 1; P@n is not normalised by the number of relevant documents, therefore it is difficult to average over topics, (2) both measures partially neglect the position of the documents. Nevertheless, there are many cases where these measures are useful in particular P@n, which is most often, but not only, for the user modelled as considering blocks of 10 documents at a time on the web. This is also demonstrated by its continued use and reporting throughout a majority of evaluation challenges at TREC, CLEF, NTCIR or FIRE. In short, the contributions of this chapter are as follows: (1) a new perspective on P@n and R@n; (2) an extensive analysis of the pool bias estimators in the literature; and (3) novel bias estimators for P@n and R@n.

7.2 Pool Bias in IR Evaluation Measures

In Section 3.5, we have defined the pool bias and we have seen that the degree of bias observed in a run tested on a test collection depends on many factors. An important one is the pooling strategy used. To understand the other factors we need to review the pooling method, and in particular in this chapter we focus on the Depth@K pooling strategy, because it is the most used pooling strategy in IR.

In the same Section 3.5, we have also shown that the pool bias can be minimised via increasing either the number of pooled documents, or the number and variety of pooled systems. But albeit the first one is a controllable parameter that largely depend on the budget invested in the creation of the test collection, the second, the number and variety of the involved IR systems depends on the interest and participation of the IR community in the issued challenge. This problem is more evident in domain specific IR, where a sufficient participation is almost always unreached. Such lack of diversity not only yields to a greater pool bias but more importantly to a not follow-up challenge, making the already built test collections unique in their kind, therefore precious.

In the following, we formally define the IR evaluation measures analysed in this chapter and how the pool bias manifests at the measure level.

7.2.1 Estimating Precision at Cut-off

In evaluating IR systems, Precision (P) is one of the two fundamental measures. We recall its definition: given $\mathcal{D}$ a set of documents, $\mathcal{D}_r$ a subset of $\mathcal{D}$ (the documents in a run r), $\mathcal{J}^+$ the set of relevant documents, P is defined as:

$$P(r) = \frac{|\mathcal{D}_r \cap \mathcal{J}^+|}{|\mathcal{D}_r|}$$

Precision represents the proportion of relevant and retrieved documents against the retrieved ones. From this definition of P we derive the definition of Precision at cut-off n

(P@n), used to better handle ranked retrieval systems:

$$P@n(r) = \frac{|\{d \in \mathcal{D}_r \cap \mathcal{J}^+ : \rho(d,r) \leq n\}|}{n}$$

where the function ρ returns the rank of a document d in a run r. The measure takes into account only the relevant documents because it is supposed to be used when there is a complete knowledge of the relevance function over the documents in the run. When we consider the problem of missing relevance assessments this assumption is not true, ending up considering unjudged documents as irrelevant. To overcome this problem and take into account the missing information about the run, we define the complement of precision, called anti-precision ($\overline{P}$). Anti-Precision measures the proportion of irrelevant and retrieved documents against the retrieved documents. In statistics, a similarly defined quantity is referred to as the False Discovery Rate (FDR) [BH95]. It is used in quantifying the results of multiple hypothesis testing experiments. However, given the very different use of it here, we continue to refer to it as anti-precision in this study, and define it as:

$$\overline{P}(r) = \frac{|\mathcal{D}_r| - |\mathcal{D}_r \setminus \mathcal{J}^-|}{|\mathcal{D}_r|}$$

where $\mathcal{J}^-$ is the set of irrelevant documents. We define $\overline{P}$ in this less intuitive way than the $|\mathcal{D}_r \cap \mathcal{J}^-|/|\mathcal{D}_r|$ because this will be useful later when comparing it against the definition of recall in the next section. As well as for precision, we can define also its cut-off version:

$$\overline{P}@n(r) = \frac{n - |\{d \in \mathcal{D}_r \setminus \mathcal{J}^- : \rho(d,r) \leq n\}|}{n}$$

Indeed, when a run is fully judged, $r \in \mathcal{R} : \forall d \in \mathcal{D}_r, d \in \mathcal{J}^+ \cup \mathcal{J}^-$, the following equation holds:

$$P(r) + \overline{P}(r) = 1$$

When it is not, and unjudged documents are present in the run, the sum of P and $\overline{P}$ is lower than 1, reduced by a quantity that represents the proportion of retrieved and unjudged documents against the retrieved documents. We refer to this as ƛ_P.

$$P(r) + \overline{P}(r) = 1 - \text{ƛ}_P(r)$$

This quantity represents the uncertainty of the measurement. Just as for P and $\overline{P}$, ƛ_P can be also defined at cut-off ($\text{ƛ}_P@n$), as follows:

$$P@n(r) + \overline{P}@n(r) = 1 - \text{ƛ}_P@n(r)$$

Now that we have defined P and its related functions, we define the bias observed when measuring P on a run. Then we quantify the range of this bias and its expected behaviour. All of this is in order to understand the causes of bias for P.

We start by defining a quantity β_P that indicates the pool bias observed on the measure P, as:

$$\hat{P}(r) - P(r) = \beta_P(r) \tag{7.1}$$

where $\hat{P}$ represents the unbiased value of P for the run r, which in this case is the same as saying that the run r has been totally judged. As for the previous quantities β_P can also be defined at cut-off ($\beta_P@n$):

$$\hat{P}@n(r) - P@n(r) = \beta_P@n(r) \tag{7.2}$$

where $\hat{P}@n$ still represents the unbiased value of $P@n$ for the run r, but in this case it needs to be judged till the cut-off n.

To know the range of these β_Ps we solve the following inequalities for P:

$$P(r) \leq \hat{P}(r) \leq P(r) + \bar{k}_P(r) \tag{7.3}$$

or for P@n:

$$P@n(r) \leq \hat{P}@n(r) \leq P@n(r) + \bar{k}_P@n(r) \tag{7.4}$$

Then, by substituting Eq. (7.1) into Eq. (7.3) we obtain that:

$$0 \leq \beta_P(r) \leq \bar{k}_P(r)$$

Similarly substituting Eq. (7.2) into Eq. (7.4):

$$0 \leq \beta_P@n(r) \leq \bar{k}_P@n(r)$$

These inequalities define the lower and upper bounds of the β_Ps and $\beta_P@n$s. Thus, these βs are positive numbers (≥ 0) bounded from above by the ratio of unjudged documents in the run $\bar{k}_P$, or $\bar{k}_P@n$ for the cut-off case.

To correct the pool bias means to estimate β_P or $\beta_P@n$. In particular, in this chapter we focus on estimating $\beta_P@n$.

7.2.2 Estimating Recall at Cut-off

The second fundamental IR measure to evaluate the performance of search engines is Recall (R). We recall its definition: given $\mathcal{D}_r$ a subset of $\mathcal{D}$ (the documents in a run r), $\mathcal{J}^+$ the set of relevant documents, R is defined as:

$$R(r) = \frac{|\mathcal{D}_r \cap \mathcal{J}^+|}{|\mathcal{J}^+|} \tag{7.5}$$

Recall represents the proportion of relevant and retrieved documents against the relevant ones. This is a complementary measure with respect to precision because it emphasises the retrieval of a relevant document, while P emphasises the relevance of a retrieved document. For recall, this is the case because it is normalised by the number of relevant documents, retrieved and non-retrieved, for precision, this is the case because it is normalised by the number of retrieved documents, relevant and irrelevant. From R we derive the definition of Recall at cut-off n (R@n), used to better handle ranked retrieval

systems. Given ρ a function that returns the rank of a document d in a run r, R@n is defined as:

$$R@n(r) = \frac{|\{d \in \mathcal{D}_r \cap \mathcal{J}^+ : \rho(d,r) \leq n\}|}{|\mathcal{J}^+|}$$

Similarly to precision, the measure takes into account only the relevant documents because it is supposed to be used when there is a complete knowledge of the relevance function over the documents in the run. When we consider the problem of missing relevance assessments this assumption is not true, ending up considering unjudged documents as irrelevant. To overcome this problem and take into account the missing information about the run, we define the complement of recall, called anti-recall ($\overline{R}$). Anti-Recall measures the proportion of relevant but non-retrieved documents against the relevant documents:

$$\overline{R}(r) = \frac{|\mathcal{J}^+| - |\mathcal{D}_r \setminus \mathcal{J}^-|}{|\mathcal{J}^+|}$$

As well as for recall, we define also its cut-off version ($\overline{R}@n$):

$$\overline{R}(r) = \frac{|\mathcal{J}^+| - |\{d \in \mathcal{D}_r \setminus \mathcal{J}^- : \rho(d,r) \leq n\}|}{|\mathcal{J}^+|}$$

Indeed, when a run is fully judged, $r \in \mathcal{R} : \forall d \in \mathcal{D}_r, d \in \mathcal{J}^+ \cup \mathcal{J}^-$, the following equation holds:

$$R(r) + \overline{R}(r) = 1$$

When it is not, and unjudged documents are present in the run, the sum of R and $\overline{R}$ is less than 1, decreased by a quantity that represents the proportion of retrieved and unjudged documents against the relevant documents. We refer to this as $\bar{k}_R$.

$$R(r) + \overline{R}(r) = 1 - \bar{k}_R(r)$$

Just as for R and $\overline{R}$, $\bar{k}_R$ can be also defined at cut-off ($\bar{k}_R@n$), as follows:

$$R@n(r) + \overline{R}@n(r) = 1 - \bar{k}_R@n(r)$$

Now that we have defined R and its related functions, we define the bias observed when measuring R on a run. Then, we quantify the range of this bias and its expected behaviour. All of this in order to understand the causes of bias for R.

We start by defining a quantity β_R that indicates the pool bias observed on the measure R, as:

$$\hat{R}(r) - R(r) = \beta_R(r) \tag{7.6}$$

where $\hat{R}$ represents the unbiased value of R for the run r, which in this case it is the same as saying that the run r has been totally judged. As for the previous quantities also β_R can be defined at cut-off ($\beta_R@n$):

$$\hat{R}@n(r) - R@n(r) = \beta_R@n(r) \tag{7.7}$$

where $\hat{R}@n$ still represents the unbiased value of $R@n$ for the run r.

To know the range of these β_Rs we solve the following inequalities for R:

$$R(r) \leq \hat{R}(r) \leq (R(r) + \bar{k}_R(r)) \frac{1}{1 + \bar{k}_R(r)} \tag{7.8}$$

Similarly to what done for P and $P@n$, the inequality on left-hand side is simply obtained by assuming that non of the unjudged documents retrieved by r are relevant, while the inequality of the right-hand side is obtained by assuming the contrary, that is all of the unjudged documents retrieved by r are relevant. While the result of the former assumption is simply $R(r)$, the latter assumption produces a less intuitive outcome. To make this explicit we present the mathematical passages performed to obtain it. We start from the definition of R as in Eq. 7.5:

$$R(r) = \frac{|\mathcal{D}_r \cap \mathcal{J}^+|}{|\mathcal{J}^+|} \Rightarrow \frac{|\mathcal{D}_r \cap \mathcal{J}^+| + |\mathcal{D}_r \setminus \mathcal{J}|}{|\mathcal{J}^+| + |\mathcal{D}_r \setminus \mathcal{J}|} = \\ = \frac{\frac{|\mathcal{D}_r \cap \mathcal{J}^+|}{|\mathcal{J}^+|} + \frac{|\mathcal{D}_r \setminus \mathcal{J}|}{|\mathcal{J}^+|}}{1 + \frac{|\mathcal{D}_r \setminus \mathcal{J}|}{|\mathcal{J}^+|}} = (R(r) + \bar{k}_R(r)) \frac{1}{1 + \bar{k}_R(r)}$$

After recalling the definition of R, we show the effect of the assumption on R. This assumption means that a quantity equal to $|\mathcal{D}_r \setminus \mathcal{J}|$ should be added to both the numerator and denominator of the definition of R. We add this to the numerator because this quantity, now considered relevant, adds to the number of relevant documents retrieved by the run. Likewise, we add this also to the denominator because this quantity adds to the the overall number of relevant documents, since those documents were not considered relevant. Finally, by multiplying and dividing this expression by $|\mathcal{J}^+|$ and performing the needed substitutions we obtain the right-hand side of the equation. Following the same reasoning, for R@n we obtain:

$$R@n(r) \leq \hat{R}@n(r) \leq (R@n(r) + \bar{k}_R@n(r)) \frac{1}{1 + \bar{k}_R@n(r)} \tag{7.9}$$

Then, by substituting Eq. (7.6) to Eq. (7.8) we obtain that:

$$0 \leq \beta_R(r) \leq (1 - R(r)) \frac{\bar{k}_R(r)}{1 + \bar{k}_R(r)}$$

Similarly substituting Eq. (7.7) to Eq. (7.9):

$$0 \leq \beta_R@n(r) \leq (1 - R@n(r)) \frac{\bar{k}_R@n(r)}{1 + \bar{k}_R@n(r)}$$

These inequalities define the lower and upper bounds of the β_Rs and $\beta_R@n$s. Thus, these βs are positive numbers (≥ 0) bounded from above by a number in function of the ratio

of irrelevant documents in the pool ($1 - R(r)$, or $1 - R@n(r)$ for the cut-off case) and the ratio of unjudged documents in the run ($\bar{k}_R(r)$, or $\bar{k}_R@n(r)$ for the cut-off case).

To correct the pool bias means to estimate the β_Rs. In particular, in this chapter we focus on estimating $\beta_R@n$.

However, when the cut-off of the recall is lower than the one guaranteed by a pooling strategy, for example if runs are pooled using a Depth@K pooling strategy with $n < K$, the range of the bias is defined as follows:

$$R@n(r)\frac{1}{1+\bar{k}_R@K(r)} \leq \hat{R}@n(r) \leq (R@n(r)+\bar{k}_R@n(r))\frac{1}{1+\bar{k}_R@K(r)} \tag{7.10}$$

Substituting again Eq. (7.7) to Eq. (7.10):

$$-R@n(r)\frac{\bar{k}_R@K(r)}{1+\bar{k}_R@K(r)} \leq \beta_R@n(r) \leq \left(\frac{\bar{k}_R@n(r)}{\bar{k}_R@K(r)} - R@n(r)\right)\frac{\bar{k}_R@K(r)}{1+\bar{k}_R@K(r)}$$

we can observe that when there is a discrepancy between the size of the run pooled and the cut-off of recall, the lower bound is translated back of a value equal to $-R@n(r)\frac{\bar{k}_R@K(r)}{1+\bar{k}_R@K(r)}$, while the upper bound is also reduced but in a more complicated way. This translation affects every pooled run.

7.2.3 Summary

In this section we have shown what the pool bias is and its causes. Then, we analysed how it is propagated when evaluating retrieval systems with P@n and R@n. In particular, we have observed that the pool bias for R@n is also affected by the pooling strategy used. In the next section we formally introduce the pool bias estimators for P@n and R@n.

7.3 Pool Bias Estimators

In this section we generalise the estimators presented in this chapter as a special form of rotation estimator (aka cross-validation or leave-one-out). We then provide a classification of the studied estimators.

An estimator is a function that, given a run r, a pooling strategy J, and the set of runs $\mathcal{R}_p$ used to build $\mathcal{J} = J_{\mathcal{R}_p}$, returns an estimation of the bias of the run r. We generalise a pool bias estimator as follows:

$$\beta(r) = A_r \mathop{\mathrm{E}}_{r' \in \mathcal{R}_p}\left[\frac{1}{a_{r'}} C(r', r, J_{\mathcal{R}_p})\right] \tag{7.11}$$

$$C(r', r, J_{\mathcal{R}_p}) = \begin{cases} f(r', J_{\mathcal{R}_p}) - f(r', J_{\mathcal{R}_p \setminus \{r'\}}) & \text{simulation-based} \\ f(r' \circ r, J_{\mathcal{R}_p}) - f(r', J_{\mathcal{R}_p}) & \text{perturbation-based} \end{cases} \tag{7.12}$$

where E is an expectation over the pooled runs $\mathcal{R}_p$; A_r and $a_{r'}$ are normalisation constants for the run to be estimated r, and the pooled run r'; f quantifies a feature of r given a set of relevance assessments $\mathcal{J}$ (*i.e.*, an IR evaluation measure); and C is a function used to define the transformation of the input before the application of f.

This generalisation is a specialisation of the rotation estimation. Since we are quantifying the bias of a run r due to its absence from the set of pooled runs, we can adopt two approaches to obtain an estimation:

Simulation-based estimators. These estimators simulate the absence of a pooled run from the pool, and compute the difference between the score when it is pooled and when it is not pooled;

Perturbation-based estimators. These estimators perturb a pooled run using the run r to indirectly measure the potential of r, *i.e.*, by measuring the performance improvement on the perturbed pooled run.

These two classes of estimators are identified by the function C, which can take one of the two forms presented in (7.12).

The constants A_r and a_r are useful to compensate the potential bias of the measured feature f. The expectation E refers to the Arithmetic mean unless otherwise specified. For the sake of clarity we define here the two expectations used in this chapter, as AM for the arithmetic mean:

$$\mathrm{AM}(V) = \frac{1}{|V|} \sum_{v \in V} v$$

and GM for the geometric mean:

$$\mathrm{GM}(V) = \sqrt[|V|]{\prod_{v \in V} v}$$

In the next sections we will introduce the bias estimators for P@n and R@n. However, before delving into their formalisations, we observe that if a test collection has been built using a Depth@K pooling strategy given an estimator for P@n we can always compute an estimator for R@n. To achieve this conclusion we start from observing how to compute R given P:

$$R(r) = \frac{|\mathcal{D}_r \cap \mathcal{J}^+|}{|\mathcal{J}^+|} = \frac{|\mathcal{D}_r \cap \mathcal{J}^+| \cdot \frac{|\mathcal{D}_r|}{|\mathcal{D}_r|}}{|\mathcal{J}^+|} = \frac{P(r) \cdot |\mathcal{D}_r|}{|\mathcal{J}^+|} \tag{7.13}$$

similarly for R@n given P@n:

$$R@n(r) = \frac{|\{d \in \mathcal{D}_r \cap \mathcal{J}^+ : \rho(d, r) \leq n\}|}{|\mathcal{J}^+|} = \\ = \frac{|\{d \in \mathcal{D}_r \cap \mathcal{J}^+ : \rho(d, r) \leq n\}| \cdot \frac{n}{n}}{|\mathcal{J}^+|} = \frac{P@n(r) \cdot n}{|\mathcal{J}^+|} \tag{7.14}$$

Now, recalling the definition of the estimation β_R in Eq. (7.6) and β_P in Eq. (7.1) and substituting them to Eq. (7.13) we obtain:

$$\beta_R(r) = \hat{R}(r) - R(r) = \frac{\hat{P}(r) \cdot n}{|\mathcal{J}^+| + \beta_P(r) \cdot n} - R(r) = \frac{(P(r) + \beta_P(r)) \cdot n}{|\mathcal{J}^+| + \beta_P(r) \cdot n} - R(r)$$

similarly for $P@n$, by recalling the definition of the estimation $\beta_R@n$ in Eq. (7.7) and $\beta_P@n$ in Eq. (7.7) and substituting them to Eq. (7.14) we obtain:

$$\begin{aligned}\beta_R@n(r) &= \hat{R}@n(r) - R@n(r) = \\ &= \frac{\hat{P}@n(r) \cdot n}{|\mathcal{J}^+| + \beta_P@n(r) \cdot n} - R@n(r) = \frac{(P@n(r) + \beta_P@n(r)) \cdot n}{|\mathcal{J}^+| + \beta_P@n(r) \cdot n} - R@n(r)\end{aligned}$$

However, when the test collection is built with the Depth@K strategy, this score needs to be adjusted as follows:

$$\beta_R@n(r) = \frac{(P@n(r) + \beta_P@n(r)) \cdot n}{|\mathcal{J}^+| + \beta_P@n(r) \cdot n + \beta_P@K(r) \cdot \max(K - n, 0)} - R@n(r) \tag{7.15}$$

where $\beta_P@n(r)$ is the estimation for $P@n$ and $\beta_P@K(r)$ is the estimation for $P@K$. This is also theoretically justified by interpreting $P@n$ plus its correction as a probability [GG05]. In the experimental section we will indicate these estimators with the name of the estimator for $P@n$ but with P as superscript.

7.3.1 Simulation-based Estimators

The intuition at the base of such estimators is that we can observe, by simulating the absence of a pooled run from the pool, how a quantity associated to the run changes when is pooled and is not pooled. A bad pool would make this variation in quantity large, while a good pool would make this variation minimal.

Estimating Pool Bias for Precision at Cut-off

In this section we give an overview of two pool bias estimators. Webber and Park [WP09] introduced the first presented estimator to mitigate the pool bias of Rank-Based-Precision (RBP) and P@n, we call this the *basic simulation estimator*. Next, breaking down the main assumption of this estimator and studying P@n behaviour, we introduce the second estimator named *k-normalised simulation estimator*.

Basic Simulation (BS). This estimator consists in adding to the score of a new run a coefficient equal to the mean difference between the score obtained when a run, initially part of the pool $\mathcal{R}_p$, is pooled and not-pooled. Webber and Park tested this estimator on RBP but claimed to be working also with P@n [WP09]. The correction coefficient for a run ($r \notin \mathcal{R}_p$) is defined by the following A_r and $a_{r'}$:

$$A_r = 1 \qquad (7.16) \qquad\qquad a_{r'} = 1 \qquad (7.17)$$

and a function C defined as:

$$C(r', r, J_{\mathcal{R}_p}) = P@n(r', J_{\mathcal{R}_p}) - P@n(r', J_{\mathcal{R}_p \setminus \{r'\}})$$

Substituting these elements into Eq. (7.11) we obtain the basic simulation estimator:

$$\beta_P@n(r) = \underset{r' \in \mathcal{R}_p}{\text{AM}} \Big(P@n(r', J_{\mathcal{R}_p}) - P@n(r', J_{\mathcal{R}_p \setminus \{r'\}}) \Big) \qquad (7.18)$$

From this formulation we can already observe two limitations. The first limitation is that the correction is not bounded by r, thereby we may have a score that may exceed the upper limit of the P@n codomain, which is $[0, 1]$. The second limitation is that it computes a coefficient that is constant and therefore does not depend on the actual status of r.

There are two main assumptions behind this estimator. The first, a more general assumption, which is present in all estimators is that any given new run is sampled from the same distribution as the pooled ones. This is of course not always true, because runs are selected based on their performance by human intervention. The second assumption is that the distribution of differences $P@n(r', J_{\mathcal{R}_p}) - P@n(r', J_{\mathcal{R}_p \setminus \{r'\}})$ is normally distributed, as indicated by the use of the arithmetic mean in the last equation. To empirically demonstrate the groundlessness of this hypothesis, we observe in Figure 7.1, on the left side, that the distribution of differences is not normally distributed. This makes the estimate for the correction biased for a new run because the distribution generated by the pooled runs is not centred on the calculated mean.

However, while the first assumption is not under our control, but it is more about the quality of the test collection, with the next estimator we focus on tackling the second assumption that arises when applying this estimator to P@n.

k-Normalised Simulation (ƙNS). This estimator solves the main issue of the previous estimator: the assumption of normality for the distribution of the differences. To find a better prior, we look at the ratio between the number of uniquely identified relevant documents discovered by pooling the run r and the number of unjudged documents that the run would have if it had not been pooled. This quantity may be interpreted as the probability of the unjudged documents of the run to be relevant:

$$P(d \in \mathcal{D}_r \setminus J_{\mathcal{R}_p \setminus \{r\}},\, d \in J^+_{\mathcal{R}_p}) = \frac{P@n(r, J_{\mathcal{R}_p}) - P@n(r, J_{\mathcal{R}_p \setminus \{r\}})}{ƙ_P(r, J_{\mathcal{R}_p \setminus \{r\}})}$$

We observe empirically that the distribution of this quantity is log-normal. Indicating with X this distribution and with Y its log-transformation $Y = log(X)$, Y is normally distributed. In the Q-Q plot in Figure 7.1 on the right side, we observe how the theoretical

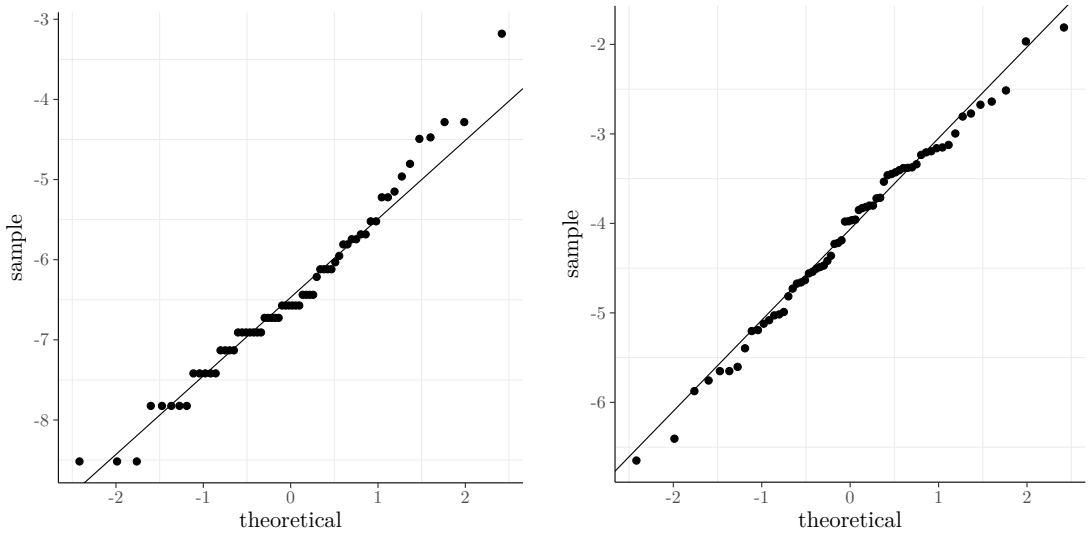

Figure 7.1: Q-Q Plots of a normal distribution against, on the left, the distribution of differences $P@n(r', J_{\mathcal{R}_p}) - P@n(r', J_{\mathcal{R}_p \setminus \{r'\}})$, on the right, the log transformation of the distribution of the probability of providing new relevant documents to the pool, for the test collection Ad Hoc 8.

normal distribution correlates with the sample distribution. To calculate a meaningful mean prior, we compute the mean of the distribution Y and then transform it back to the domain of the distribution X, which leads to the geometric mean of the X.

This leads to redefining the normalisation constants with respect to the previous definition as:

$$A_r = \bar{k}@n(r, J_{\mathcal{R}_p}) \qquad\qquad a_{r'} = \bar{k}@n(r', J_{\mathcal{R}_p \setminus \{r'\}})$$

Substituting these two last equations into Eq. (7.18), and using the geometric mean, we obtain:

$$\beta_P@n(r) = \bar{k}@n(r, J_{\mathcal{R}_p}) \underset{r' \in \mathcal{R}_p}{\text{GM'}} \left(\frac{P@n(r', J_{\mathcal{R}_p}) - P@n(r', J_{\mathcal{R}_p \setminus \{r'\}})}{\bar{k}@n(r', J_{\mathcal{R}_p \setminus \{r'\}})} \right) \tag{7.19}$$

where GM' is a slightly modified version of the geometric mean:

$$\text{GM'}(V) = \text{GM}(\{v \in V : v \neq 0\}) \tag{7.20}$$

This is done because in order to have a well defined formulation, we need to remove the cases when the difference is null. This is reasonable because: first, if this difference is zero it can be shown that $\bar{k}_r = 0$ and consequently the fraction in Eq. (7.19) is undefined. Second, such zero values bring no information to our estimate of the contribution of the run. In fact, one could generate an unbounded number of runs with difference equal to zero.

Comparing Eq. (7.18) and Eq. (7.19) we notice that the numerator is the same and that with respect to the second equation the difference is that now, every difference in P@n gets divided by the number of uniquely identified documents provided to the pool, and then multiplied by the same but for r.

This estimator solves the two limitations of the previous estimator. The first limitation was about the fact that the correction may make the score of the run exceed the upper limit of the P@n codomain. This is no longer possible because the maximum value the estimator can output is equal to $\bar{k}@n$. The second limitation was about the fact that the estimator computes a coefficient that is constant for any run. This is also no longer true since it is multiplied by A_r, which is a value computed on the non pooled run. Therefore it no longer corrects based on a constant prior probability of the run to find relevant documents among its unjudged ones.

Estimating Pool Bias for Recall at Cut-off

In this section we present three simulation-based estimators for R@n. The first estimator is based on the estimator presented by Webber and Park [WP09], like in the previous section called *basic simulation estimator*. We will observe that also in this case, for R@n, the assumption of normality for the distribution of the difference is not hold. From this observation we develop the second estimator, named *geometric simulation estimator*. However, these estimators are independent of the target run, *i.e.*, any run would be corrected with the same value. We present a third estimator, the *$\bar{k}$-normalised simulation estimator*, which overcomes this limitation.

Basic Simulation (BS). Similarly to what done for the BS estimator for P@n, to correct the bias of a run for R@n we redefine C as follows:

$$C(r', r, J_{\mathcal{R}_p}) = R@n(r', J_{\mathcal{R}_p}) - R@n(r', J_{\mathcal{R}_p \setminus \{r'\}})$$

where A_r and $a_{r'}$ are defined as in Eq. (7.16) and Eq. (7.17). Then, substituting these elements into Eq. (7.11) we obtain the following basic simulation estimator:

$$\beta_R@n(r) = \underset{r' \in \mathcal{R}_p}{\mathrm{AM}} \left[R@n(r', J_{\mathcal{R}_p}) - R@n(r', J_{\mathcal{R}_p \setminus \{r'\}}) \right]$$

This estimator suffers of the same limitations listed for P@n: the estimation is constant for every unpooled run, and this estimation being unbounded can lead to a value outside the codomain of R@n, which is $[0, 1]$.

Geometric Simulation (GS). In the previous estimator, it is assumed that the distribution of the differences is normal. However, analysing this distribution shows that this assumption is again groundless. This results in a behaviour similar to the previous observed for the $\bar{k}$NS– the distribution of the differences is log-normal. However, these differences computed on R@n can be negative therefore these values cannot be directly log-transformed. We perform instead a similar transformation by first translating the

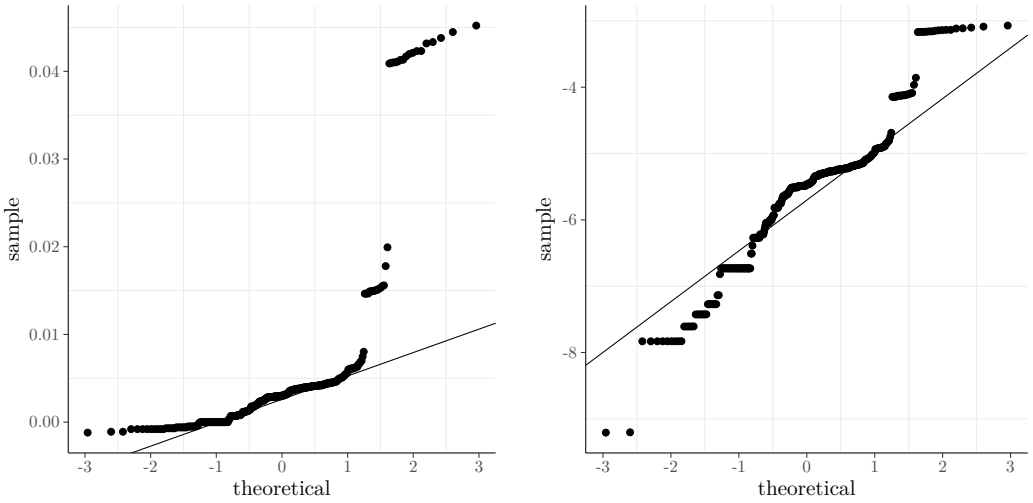

Figure 7.2: Q-Q Plots of a normal distribution against, on the left, the distribution of differences $R@n(r', J_{\mathcal{R}_p}) - R@n(r', J_{\mathcal{R}_p \setminus \{r'\}})$, on the right, the log transformation of the same quantities, for the test collection Ad Hoc 8.

distribution by a quantity equal to the minimum of the distribution, this in order to avoid the presence of negative values, and then log-transform. This observation is made in Figure 7.2. Therefore, we redefine this estimator as follows:

$$\beta_R@n(r) = \underset{r' \in \mathcal{R}_p}{\text{GM'}} \left[R@n(r', J_{\mathcal{R}_p}) - R@n(r', J_{\mathcal{R}_p \setminus \{r'\}}) \right]$$

This definition of GM' is different to the definition in Eq. (7.20). However this new definition generalises the previous one and can be used also in that case. The new version of GM' is defined as follows:

$$\text{GM'}(V) = \text{GM}(\{v \in V : v \neq \min(V)\} - \min(V)) + \min(V)$$

This formula translates the distribution by a constant value equal to the min value of V, computes the geometric mean and translates back by the same quantity.

$\bar{k}$-Normalised Simulation ($\bar{k}$NS). Following the same idea in the definition of $\bar{k}$NSfor P@n, we define a similar estimator for R@n. To do this we define A_r and $a_{r'}$ based on the maximum value obtainable by R@n, as defined when explaining R@n.

$$A_r = (1 - R@n(r, J_{\mathcal{R}_p})) \frac{\bar{k}@n(r, J_{\mathcal{R}_p})}{1 + \bar{k}@n(r, J_{\mathcal{R}_p})}$$

and,

$$a_{r'} = (1 - R@n(r', J_{\mathcal{R}_p \setminus \{r'\}})) \frac{\bar{k}@n(r', J_{\mathcal{R}_p \setminus \{r'\}})}{1 + \bar{k}@n(r', J_{\mathcal{R}_p \setminus \{r'\}})}$$

Substituting this into Eq. (7.18) we obtain:

$$\beta_R@n(r) = (1 - R@n(r, J_{\mathcal{R}_p}))\frac{\bar{k}@n(r, J_{\mathcal{R}_p})}{1 + \bar{k}@n(r, J_{\mathcal{R}_p})} \cdot \underset{r' \in \mathcal{R}_p}{\mathrm{AM}} \left[\frac{R@n(r', J_{\mathcal{R}_p}) - R@n(r', J_{\mathcal{R}_p \setminus \{r'\}})}{1 - R@n(r, J_{\mathcal{R}_p \setminus \{r'\}})} \frac{1 + \bar{k}@n(r, J_{\mathcal{R}_p \setminus \{r'\}})}{\bar{k}@n(r, J_{\mathcal{R}_p \setminus \{r'\}})} \right]$$

This solves the limitations listed for the two previous estimators: this estimation is bounded between 0 and the maximum obtainable value for R@n, and the estimation is no longer constant for every unpooled run.

In addition to these R@n estimators presented above we also include all the estimators defined for P@n but combined as in Eq. 7.15. These are indicated as follows, BS^P and $\bar{k}\mathrm{NS}^P$.

7.3.2 Perturbation-based Estimators

The intuition at the base of such estimators is that we can observe how a new, unpooled run impacts the existing, pooled runs. Given such an existing run, we can imagine to perturb it based on the ranks of its documents in the unpooled run. A "bad" new run will tend to bring down known relevant documents and push up irrelevant ones. Quantifying these changes we create a measure of the potential quality of the new run.

Before going on to the details of the perturbation-based estimators, let us perform an imagination exercise in order to better understand the information content of a partially judged run. As in a deck of cards, a shuffling changes the order of the documents of a run and produces a new run that we will indicate as r'. This run has the same set of documents as before. We want to observe the variation in score the run obtains in the two states, original and shuffled. Therefore, it is necessary to use an evaluation measure sensitive to document rank change, *i.e.*, P would not be suitable in this case. Now, let us define f to be an evaluation measure with this property. Given a run r and its shuffled version r' we define:

$$\delta f(r') = f(r') - f(r)$$

δf is the variation of the measure f after a shuffle of the run r'. The measure f increases with the increase in number of relevant documents at higher positions, and decreases *vice versa*, as mostly IR measures do. An increase in value of δf is the result of the combination of the following two related effects: the shuffle moved up relevant documents, or moved down irrelevant or unjudged documents (with the consequential moving up of potential relevant documents in the run). It decreases if the opposite happens. We can also define $\delta\overline{f}$ as following:

$$\delta\overline{f}(r') = \overline{f}(r') - \overline{f}(r)$$

$\delta\overline{f}$ is the variation of the anti-measure f of the run after a shuffle. Its increase in value is the result of the combination of the following two related effects: the shuffle moved up irrelevant documents, or moved down relevant or unjudged documents (with the

consequential moving up of potential irrelevant documents in the run). It decreases if the opposite happens.

After measuring the variation of relevant documents with δf, and irrelevant documents with $\delta\overline{f}$, we conclude with measuring the variation of unjudged documents with $\delta\bar{k}$ for f:

$$\delta\bar{k}(r') = \bar{k}(r') - \bar{k}(r)$$

$\delta\bar{k}$ is the variation of *unjudged* documents on a given run. Its increase in value is the result of the combination of the following effects: the shuffle moved up unjudged documents or moved down relevant and irrelevant documents (with the consequential moving up of potential unjudged documents in the run).

To summarise this imaginary exercise, when a run changes the order of its documents, δf, $\delta\overline{f}$, and $\delta\bar{k}$ are indicators of the direction of the judged relevant, judged irrelevant, and unjudged documents in the run.

Now let us make a step further and consider not the relationship between a run and a random shuffle of itself, but between a run and another run. In the particular case where each run ranks completely the entire collection, this is the same as above. In general however, the systems only provide runs down to a certain limit (say 1000). To study this effect, we need to define a perturbation function between the two runs. The unpooled run will have an effect on the pooled run, measured by the quantities described above. In the next paragraphs we analyse the meaning of perturbing a run, understanding what the previous introduced measures express in this context.

Such a perturbing function can simply be based on the rank of the documents in the run. The aim here is not to add or remove documents from a run, we must keep in mind that all we need to do here is transfer only the information about the rank of the documents. We do this by combining the ranks if the two runs share the same document.

In the following formula, by r we denote the new, previously unseen and unpooled run, whose effect on r', an existing run, we want to study. This effect we represent as a new, synthetic run r'', which consists exclusively of documents present in r', potentially re-ordered.

$$r'' = r' \circ r$$

where the perturbing function $\circ$ is defined by the perturbing estimator. As any functional composition operator, our perturbing operator $\circ$ is not commutative and always represents the effect of its right member on its left member.

Estimating Pool Bias for Precision at Cut-off

In this section we apply to P@n the concepts discussed previously, and interpret their meaning in this context. After this, we present two bias estimators that make use of these concepts.

As we have seen, if we would use P to compute the δ functions, since there is no information about the position of the documents in the formula, we would measure a

change of 0. However, P@n does not suffer from this issue because it preserves ranking information given by the cut-off, which distinguishes between what happens before and after it. Given a run r' and its shuffled version r'' we can therefore define:

$$\delta P@n(r'') = P@n(r'') - P@n(r')$$

where $\delta P@n$ has domain $[-1, 1]$.

We also define $\delta\overline{P}@n$ as following:

$$\delta\overline{P}@n(r'') = \overline{P}@n(r'') - \overline{P}@n(r')$$

$\delta\overline{P}@n$ has domain $[-1, 1]$.

Finally, $\delta\bar{k}$ for $P@n$ that can be derived as following:

$$\begin{aligned}\delta\bar{k}@n(r'') &= \bar{k}@n(r'') - \bar{k}@n(r') = \\ &= 1 - (P@n(r'') + \overline{P}@n(r'')) - [1 - (P@n(r') + \overline{P}@n(r'))] = \\ &= -\delta P@n(r'') - \delta P@n(r') \quad (7.21)\end{aligned}$$

$\delta\bar{k}@n$ has domain $[-1, 1]$. An interesting property of this function, which is possible to prove, is that if r' has been judged to depth $K : K \geq n$, then the domain of the function $\delta\bar{k}@n$ is $[0, 1]$. This property always holds for pooled runs because they verify the condition (provided of course that no mistakes occurred in the pooling process).

We now present two estimators of this class. The first is a relaxation of the second.

k-Linear Perturbation ($\bar{k}$LP). This estimator defines the simplest perturbing operator $\circ$ as a linear combination of ranks, as follows:

$$r'' = r' \circ r = \{d \in r' : \rho(d, r'') = \mu(d, r', r)\} \quad (7.22)$$

where μ is defined by the following linear combination of ranks:

$$\mu(d, r', r) = \begin{cases} \rho(d, r') \cdot (1 - \alpha) + \rho(d, r) \cdot \alpha & \text{if } d \in r \\ \rho(d, r') & \text{otherwise} \end{cases}$$

μ is the weighted arithmetic mean between the rank of the document in r' and the rank of the document in r, with $0 \leq \alpha \leq 1$. When the same rank is assigned by μ to two different documents, which can happen in some cases for a pair of documents of which one is also in r and the other one is not, the common document is inserted after the r'-exclusive document. In other words, the original run rank has priority.

Now that we have an understanding of which runs are suffering from pool bias, with respect to precision at cut-off, we proceed by presenting the estimator to adjust the score.

To correct the pool bias we want to add a quantity that stays within its uncertainty limit $\bar{k}_r$. In other words, our growth potential in terms of P@n is bounded by $\bar{k}_r$. We are interested in estimating the missing precision of the unjudged documents in the run r.

The adjustment is based on the average effect of this run r on the existing runs, in terms of $\bar{k}$. We do this by computing the $\delta\bar{k}_{r'}$ produced by r on a pooled run r' via the run perturbing function defined in Eq. 7.22. This measures the aggregated change in precision and anti-precision, as described by Eq. 7.21. Therefore, we define C as:

$$C(r', r, J_{\mathcal{R}_p}) = \bar{k}@n(r' \circ r, J_{\mathcal{R}_p}) - \bar{k}@n(r, J_{\mathcal{R}_p})$$

a_r, and A_r as follows:

$$a_{r'} = 1 \qquad\qquad A_r = \bar{k}@n(r, J_{\mathcal{R}_p})$$

Substituting these into Eq. (7.11) we obtain the following estimator:

$$\beta_P@n(r) = \bar{k}@n(r, J_{\mathcal{R}_p}) \underset{r' \in \mathcal{R}_p}{\text{AM}} \left[\bar{k}@n(r' \circ r, J_{\mathcal{R}_p}) - \bar{k}@n(r, J_{\mathcal{R}_p})\right]$$

The average in this estimator, if the runs have been pooled using a fixed-depth at cut-off K pooling strategy and $n \leq K$, is always positive and it acts as a maximum likelihood estimator for the position in $[0,\bar{k}_r]$. Therefore, the correction quantity is the product between $\Delta\bar{k}_r$ and $\bar{k}_r$. However, if the pooling strategy allows to have unjudged documents at ranks lower than K or the cut-off of the measure n is too large, we constrain this average to positive values by taking the maximum between the average and 0, like this:

$$\beta_P@n(r) = \bar{k}@n(r, J_{\mathcal{R}_p}) \max\left(\underset{r' \in \mathcal{R}_p}{\text{AM}} \left[\bar{k}@n(r' \circ r, J_{\mathcal{R}_p}) - \bar{k}@n(r, J_{\mathcal{R}_p})\right], 0\right)$$

λ-Triggered k-Linear Perturbation (λT$\bar{k}$LP). This estimator is an extension of the previous estimator. This estimator arises by the observation that the δ computed on the perturbed run can be used to develop an indicator that can be used to trigger its correction.

$\delta P@n$ and $\delta\overline{P}@n$ can be used to analyse the quality of an unpooled run against a pooled one. An increase in $\delta P@n$ is the result of two forces, one direct and one indirect: 1) direct, if the relevant documents in the top n of r are the same documents found at the bottom of r', they will be pushed up; 2) indirect, if the r has irrelevant or unjudged documents in the bottom that are in the top n documents of r', they will be pushed down. The contribution decreases if the contrary happens. For $\delta\overline{P}@n$ as well, the contribution is: 1) direct, if the irrelevant documents in the top n of r are shared with documents in the bottom of r'; 2) indirect, if the r has relevant or unjudged documents in the bottom that are in the top n documents of r'. If the run r' would be judged in its totality, these two effects would be perfectly correlated and it would be possible to calculate one just knowing the other from the following equation:

$$\delta P@n + \delta\overline{P}@n = 0$$

However, when r' contains unjudged documents at ranks below n, their sum becomes $-\delta\bar{k}@n$, as shown in Eq. 7.21.

Table 7.1: Measures computed for the run `sab05ror1` when it is not part of the pool

P@10	$\bar{k}$@10	ΔP@10	$\Delta\overline{P}$@10
0.4220	0.444	0.0065	-0.1053

As explained above, $\delta\bar{k}@n$ represents the ratio of unjudged documents brought to the top n of the run r' by the run r. Moreover, it is possible to prove that $\delta P@n = 0$ and $\delta\overline{P}@n = 0$ if and only if one of the following two conditions occurs: 1) the two runs r' and r do not share any documents with each other in their top n documents, or 2) the two runs are identical in the top n. These are the two cases where our method will not say anything about the new run r just by using the existing run r' (but this method might be based on other pooled runs).

Let us now take an example to illustrate how this indicator could be useful to understand the behaviour of a run and predict its quality. We use the test collection Robust 2005 and in particular we focus our attention on a special run that presents an unusual effect, the routing run `sab05ror1`. It has the peculiarity of being strongly discounted when it is not in the pool. Buckley et al. [Buc+07] studied it at length, pointing out that the reason for its behaviour was related to the size of the test collection. For this run let us calculate P@10 and $\bar{k}$@10. Let us also consider the average of δP@10 and $\delta\overline{P}$@10, which we denote as follows:

$$\Delta P@10(r) = \underset{r' \in \mathcal{R}_p}{\mathrm{AM}} \left[\delta P@10(r' \circ r)\right]$$

$$\Delta\overline{P}@10(r) = \underset{r' \in \mathcal{R}_p}{\mathrm{AM}} \left[\delta\overline{P}@10(r' \circ r)\right]$$

where $\mathcal{R}_p$ is the set of runs used in the creation of the test collection.

Table 7.1 shows these values for this particular run. When the run is not part of the pool, P@10 assigns it the 11th position in 18th runs. $\bar{k}$@10 says that there are many documents that are unjudged and that therefore there is a high potential to grow. ΔP@10 indicates a low average positive contribution to the pooled runs, and shows that among the relevant documents there is little intersection. $\Delta\overline{P}$@10 instead is negative which suggests that many irrelevant documents have been ranked lower than before, therefore suggesting a good ability of this special run to discriminate relevant documents from irrelevant ones.

In Figure 7.3 we show the resulting ΔP@10, $\Delta\overline{P}$@10 against the residual error ($\hat{\varepsilon}$, the difference between the true score and the unpooled score), generated with a leave-one organisation-out approach. Here we can observe that just using ΔP@10 is not enough because it takes into account only one of the two positive contributions of the run, the other one being the reduction in $\Delta\overline{P}$@10.

Let us now return to the general case. When the average negative contribution of the unpooled run to other runs is reduced (*i.e.*, $\Delta\overline{P} < 0$) and the run has a positive contribution (*i.e.*, $\Delta P > 0$), the run suffers from pool bias and its score should be adjusted. More problematic is the case when $\Delta\overline{P}$ and ΔP have the same sign (*i.e.*, the

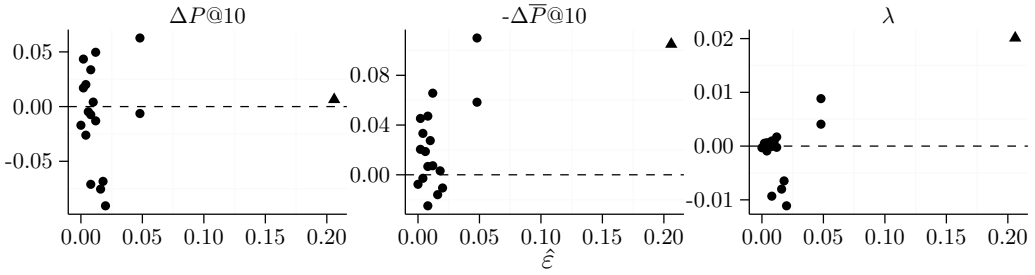

Figure 7.3: Plot of $\Delta P@10$, $\Delta\overline{P}@10$ and λ against the residual ($\hat{\varepsilon}$) in a leave-one organisation-out experiment, for the Robust 2005 test collection. The run indicated as ▲ is the unusual run `sab05ror1`.

run has both a negative and a positive contribution, on average). Indeed, on one hand, if we have $\Delta\overline{P} > 0$ and $\Delta P > 0$ we would improve the P@n score of the run only if their ratio is greater than the ratio of P to $\overline{P}$, because it means that there is a chance to improve the existing score. On the other hand, if we have $\Delta\overline{P} < 0$ and $\Delta P < 0$ we would improve only if their ratio is lower than the ratio of P to $\overline{P}$ because it means that the contribution of the run is more able to discriminate the irrelevant documents.

From these observations we derived a single value indicator that merges the information of all the indicators defined:

$$\lambda_r = \Delta P@n(r) \cdot \overline{P}@n(r) - \Delta\overline{P}@n(r) \cdot P@n(r)$$

For all runs where $\lambda_r > 0$ we apply our correction method.

Returning briefly to the example of the `sab05ror1` run, we can now see in Figure 7.3 that λ_r clearly distinguishes this run from the rest.

Before delving into the definition of this indicator, let us recall the definition of the step function[1] χ that will be used in the formalisation of the estimator.

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases} = [x \in A]$$

This function returns 1 if the argument belongs to the set defined by A, and 0 otherwise. In the right-hand side we have the same but using the Iverson bracket notation.

We now have all the components to define this bias estimator. This indicator with respect to the previous one changes only the definition of A_r:

$$A_r = [\lambda_r > 0] \cdot \bar{k}@n(r, J_{\mathcal{R}_p})$$

[1]This function is also called the indicator function, but for sake of clarity we call it step function in order to not confuse the reader with the indicator functions we have previously defined.

Thereby, the final indicator is:

$$\beta_P@n(r) = [\lambda_r > 0] \cdot \bar{k}@n(r, J_{\mathcal{R}_p}) \max\left(\underset{r' \in \mathcal{R}_p}{\mathrm{AM}} \left[\bar{k}@n(r' \circ r, J_{\mathcal{R}_p}) - \bar{k}@n(r', J_{\mathcal{R}_p})\right], 0\right)$$

This estimator is similar to the definition of the previous estimator except for the component A_r that now works as a trigger based on the value provided by the indicator function λ_r.

Estimating Pool Bias for Recall at Cut-off

To estimate the bias for R@n we use all the estimators defined above but combined as in Eq. 7.13. These are indicated as follows, $\bar{k}\mathrm{LP}^P$ and $\lambda\mathrm{T}\bar{k}\mathrm{LP}^P$.

7.4 Experiments and Results

To test the performance in terms of pool bias of the bias estimators developed in the previous section we perform a large-scale experimentation using 15 test collections. In Table 7.2 we show a summary of the estimators introduced in the previous section. This material and the experimental setup are presented in the next section. Next, we introduce the measures of bias. Finally, we present the results.

7.4.1 Material and Experimental Setup

To test the pool bias estimators developed in the previous section we used 15 test collections sampled from TREC [VH99b]: 7 test collections from the Ad Hoc track, 3 from the Web track, and 5 from more domain specific IR tracks: Genomics, Robust, Legal, Medical and Microblog. Details about the test collections are presented in Table 7.3.

To test all the estimators against each other we could perform a leave-one run-out approach. As baseline we could consider the traditional evaluation against the reduced pool. We call this the *reduced* pool to distinguish it from the ground truth pool — the one also containing documents exclusively contributed by the removed runs. This would be the leave-one run-out experiment as firstly described by Zobel [Zob98], one run at a time is exited from the pool. This is done by removing all the documents uniquely identified by it from the relevance assessments. However, to avoid potential run dependencies across runs submitted by the same organisation we perform instead a *leave-one organisation-out* instead as introduced by Büttcher et al. [Büt+07]. This is similar to the *leave-one run-out*, with the difference that not only is one run removed from the pool, but also all the runs generated by the same organisation. This is done by removing all the documents uniquely identified by the organisation's runs from the relevance assessments. This second approach simulates better the testing of a new run, since in most cases it has been observed that the runs produced by the same organisation come from the same system, with only some parameter variation. Therefore, they often bring to the pool the same relevant documents.

Table 7.2: List of pool bias estimators for P@n and R@n with their defining equations to be substituted into the generalised definition of a pool bias estimator in Eq. (7.11).

Bias Estimators for P@n					
Name	Abbr.	$A_{\mathcal{R}}$	$a_{r'}$	C	E
Basic S	BS	1	1	$P@n(r', J_{\mathcal{R}_p}) - P@n(r', J_{\mathcal{R}_p \setminus r'})$	AM
k-Normalised S	kNS	$k@n(r, J_{\mathcal{R}_p})$	$k@n(r', J_{\mathcal{R}_p \setminus r'})$	$P@n(r', J_{\mathcal{R}_p}) - P@n(r', J_{\mathcal{R}_p \setminus r'})$	GM
k-Linear P	kLP	$k@n(r, J_{\mathcal{R}_p})$	1	$k@n(r' \circ r, J_{\mathcal{R}_p}) - k@n(r', J_{\mathcal{R}_p})$	AM
λ-Triggered **k**-Linear P	λTkLP	$[\lambda_r > 0] \cdot k@n(r, J_{\mathcal{R}_p})$	1	$k@n(r' \circ r, J_{\mathcal{R}_p}) - k@n(r', J_{\mathcal{R}_p})$	AM

Bias Estimators for R@n					
Basic S	BS	1	1	$R@n(r', J_{\mathcal{R}_p}) - R@n(r', J_{\mathcal{R}_p \setminus \{r'\}})$	AM
Geometric S	GS	1	1	$R@n(r', J_{\mathcal{R}_p}) - R@n(r', J_{\mathcal{R}_p \setminus \{r'\}})$	GM
k-Normalised S	kNS	$(1 - R@n(r, J_{\mathcal{R}_p}))\frac{k@n(r, J_{\mathcal{R}_p})}{1+k@n(r, J_{\mathcal{R}_p})}$	$(1 - R@n(r, J_{\mathcal{R}_p \setminus \{r'\}}))\frac{k@n(r, J_{\mathcal{R}_p \setminus \{r'\}})}{1+k@n(r, J_{\mathcal{R}_p \setminus \{r'\}})}$	$R@n(r', J_{\mathcal{R}_p}) - R@n(r', J_{\mathcal{R}_p \setminus \{r'\}})$	AM

Table 7.3: Pool properties of test collections, for the original pool, and the synthesized "cleaned" pool. The cleaned pool is equivalent to a Depth@K with K equal to the one used to build the original pool.

Test Collection Properties						
	Ad Hoc 2		Ad Hoc 3		Ad Hoc 4	
$\lvert\mathcal{R}\rvert$	38		40		33	
$\lvert\mathcal{R}_p\rvert$	36		23		32	
$\lvert\mathcal{O}\rvert$	22		22		19	
$\lvert\mathcal{Q}\rvert$	50		50		50	
K	100		200		100	
	Original →	Cleaned	Original →	Cleaned	Original →	Cleaned
$\lvert\mathcal{J}\rvert$	62,620	49,381	97,319	75,378	87,069	55,949
$\lvert\mathcal{J}^+\rvert$	11,645	10,224	9,805	9,287	6,503	5,457
	Ad Hoc 5		Ad Hoc 6		Ad Hoc 7	
$\lvert\mathcal{R}\rvert$	61		74		103	
$\lvert\mathcal{R}_p\rvert$	60		28		76	
$\lvert\mathcal{O}\rvert$	21		29		42	
$\lvert\mathcal{Q}\rvert$	50		50		50	
K	100		100		100	
	Original →	Cleaned	Original →	Cleaned	Original →	Cleaned
$\lvert\mathcal{J}\rvert$	133,681	78,505	72,270	57,257	80,345	79,133
$\lvert\mathcal{J}^+\rvert$	5,524	5,022	4,611	3,931	4,674	4,584
	Ad Hoc 8		Web 9		Web 2001	
$\lvert\mathcal{R}\rvert$	130		104		97	
$\lvert\mathcal{R}_p\rvert$	74		62		59	
$\lvert\mathcal{O}\rvert$	41		23		29	
$\lvert\mathcal{Q}\rvert$	50		50		50	
K	100		100		100	
	Original →	Cleaned	Original →	Cleaned	Original →	Cleaned
$\lvert\mathcal{J}\rvert$	86,830	86,830	70,070	70,030	70,400	70,400
$\lvert\mathcal{J}^+\rvert$	4,728	4,728	2,617	2,616	3,363	3,363

Table 7.4: Continuation of Table 7.3 for the rest of the test collections.

	Test Collection Properties					
	Web 2002		Legal 2006		Microblog 2011	
$\lvert\mathcal{R}\rvert$	69		34		184	
$\lvert\mathcal{R}_p\rvert$	69		31		98	
$\lvert\mathcal{O}\rvert$	16		8		58	
$\lvert\mathcal{Q}\rvert$	50		38		49	
K	50		10		30	
	Original →	Cleaned	Original →	Cleaned	Original →	Cleaned
$\lvert\mathcal{J}\rvert$	56,650	55,798	31,041	5,693	60,129	26,371
$\lvert\mathcal{J}^+\rvert$	1,574	1,554	3,931	906	2,965	2,548
	Medical 2011		Genomics 2005		Robust 2005	
$\lvert\mathcal{R}\rvert$	127		62		74	
$\lvert\mathcal{R}_p\rvert$	41		55		18	
$\lvert\mathcal{O}\rvert$	29		32		17	
$\lvert\mathcal{Q}\rvert$	34		49		50	
K	10		60		55	
	Original →	Cleaned	Original →	Cleaned	Original →	Cleaned
$\lvert\mathcal{J}\rvert$	8,865	5,049	39,958	38,604	37,798	22,173
$\lvert\mathcal{J}^+\rvert$	1,765	1,437	4,584	4,387	6,561	4,563

Finally, as in previous studies [BL07; SZ05; SZ05; UMM13; Voo09; VB02] to avoid buggy implementations of some of the systems that took part in the challenges, we also tested again with only the top 75% of best performing runs of each test collection.

7.4.2 Measures of Pool Bias

These measures of bias are the same as to the ones presented in Section 6.3.3, but because their use is slightly different, it is worth presenting them again here.

The measures of bias take as input an IR evaluation measure f. The first measure we present is Mean Absolute Error (MAE). This measure estimates the expected observed bias of a test collection. This is computed by averaging over the runs $\mathcal{R}_p$ the absolute difference in score between a run r when it is pooled and not pooled. Given a pooling strategy J, which has been used to build the ground truth $G = J_{\mathcal{R}_p}$, f an IR evaluation measure, and $\hat{f}$ its estimation, we define MAE as:

$$\text{MAE}(J_{\mathcal{R}_p}) = \frac{1}{|\mathcal{R}_p|} \sum_{r \in \mathcal{R}_p} \left| \hat{f}(r, J_{\mathcal{R}_p \setminus \{r' \in \mathcal{R}_p : o_{r'} = o_r\}}) - f\,(r, G) \right|$$

A low MAE means that the score obtained by the estimator is close to the score obtained by the runs when pooled.

The second measure we present is System Rank Error (SRE). This measure counts the number of rank positions lost or gained by the runs among the other pooled runs $\mathcal{R}$ when it is pooled and not pooled. Given f an IR evaluation measure, and $\hat{f}$ its estimation, we define SRE as:

$$\begin{aligned}\mathrm{SRE}(J_{\mathcal{R}_p}) = \sum_{r\in\mathcal{R}_p} \Big|\Big\{ r' \in \mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\} : \\ : \hat{f}(r, J_{\mathcal{R}_p\setminus\{r''\in\mathcal{R}_p : o_{r''}=o_r\}}) \le f(r',G) < f(r,G) \vee \\ \vee\, f(r,G) < f(r',G) \le \hat{f}(r, J_{\mathcal{R}_p\setminus\{r''\in\mathcal{R}_p : o_{r''}=o_r\}})\Big\}\Big|\end{aligned}$$

A low SRE means that the rank position of the runs when not pooled is close to the rank position of the runs when pooled.

In IR, when comparing ranking of runs, it is common practice to evaluate their significance. We implemented this in the next bias measure named System Rank Error with Statistical Significance. Its difference is that instead of counting all the runs gaining or losing rank positions against the runs, it counts them only if significant according to a paired t-test with $p < 0.05$. SRE* is defined as follows:

$$\begin{aligned}\mathrm{SRE}^*(\mathcal{J},\mathcal{R}) = \sum_{r\in\mathcal{R}_p} \Big|\Big\{ r' \in \mathcal{R}_p \setminus \{r'' \in \mathcal{R}_p : o_{r''} = o_r\} : \\ : \Big(\hat{f}(r, J_{\mathcal{R}_p\setminus\{r''\in\mathcal{R}_p : o_{r''}=o_r\}}) \le f(r',G) < f(r,G) \vee \\ \vee\, f(r,G) < f(r',G) \le \hat{f}(r, J_{\mathcal{R}_p\setminus\{r''\in\mathcal{R}_p : o_{r''}=o_r\}})\Big) \wedge \\ \wedge\, \text{t-test}_{\text{paired}}(r,r',G) < 0.05\Big\}\Big|\end{aligned}$$

Juxtaposing the measures of bias, we observe that a zero MAE value implies that SRE and SRE* are equal to zero too. However, the contrary is not true. Moreover, a zero SRE implies a zero SRE*, but not *vice versa*.

7.4.3 Results

The results presented in this chapter are divided into two sets: results regarding the evaluation of the estimators for P@n, and the evaluation of the estimators for R@n. Moreover, the latter is subdivided into two subsets, the estimator originally developed for R@n, and adapted P@n estimators for R@n as shown in Eq. (7.13).

We start with P@n by comparing, in Tables 7.5 and 7.6, the results of all P@n estimators against the baseline, the 'reduced pool'. In Table 7.7 and 7.8 we present the same but only using the 75% best performing runs. These results are also presented in Figure 7.4 for MAE, and 7.5 for SRE.

Continuing with R@n we compare, in Tables 7.9 and 7.10, the results of all R@n estimators against the baseline, the 'reduced pool'. In Table 7.11 and 7.12 we present the same but

Table 7.5: Summary of the results for P@n of the Reduced Pool and its four presented estimators. These are generated through a leave-one organisation-out approach using all the pooled runs. The dotted lines represent the point when $n \leq K$ becomes false, where K is the depth of the Depth@K strategy used to build the test collection.

C	n	Pool			BS			kNS			kLP			λTkLP		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Ad Hoc 2	5	0.0071	16	**0**	0.0071	17	**0**	0.0090	14	**0**	**0.0056**	**13**	**0**	0.0068	16	**0**
	10	0.0087	32	**0**	0.0084	27	**0**	0.0089	**16**	**0**	**0.0071**	30	**0**	0.0084	32	**0**
	15	0.0099	39	**0**	0.0092	24	**0**	0.0088	**23**	**0**	**0.0083**	33	**0**	0.0095	35	**0**
	20	0.0113	50	**0**	0.0096	45	**0**	0.0097	**32**	**0**	**0.0092**	49	**0**	0.0108	49	**0**
	30	0.0130	42	**0**	0.0102	38	**0**	**0.0097**	**28**	**0**	0.0101	38	**0**	0.0123	39	**0**
	100	0.0239	113	8	**0.0137**	58	2	0.0143	**48**	**1**	0.0146	80	2	0.0191	86	4
Ad Hoc 3	5	0.0023	3	**0**	0.0038	3	**0**	0.0083	5	**0**	**0.0022**	**2**	**0**	0.0023	3	**0**
	10	**0.0023**	**0**	**0**	0.0032	**0**	**0**	0.0052	1	**0**	**0.0023**	**0**	**0**	**0.0023**	**0**	**0**
	15	0.0031	3	**0**	0.0039	4	**0**	0.0045	2	**0**	**0.0030**	3	**0**	**0.0030**	3	**0**
	20	0.0036	2	**0**	0.0042	3	**0**	0.0046	3	**0**	**0.0035**	2	**0**	**0.0035**	2	**0**
	30	0.0046	9	**0**	0.0052	8	**0**	0.0043	**5**	**0**	**0.0040**	8	**0**	0.0043	8	**0**
	100	0.0071	14	**0**	0.0067	17	**0**	**0.0051**	**6**	**0**	0.0056	11	**0**	0.0061	11	**0**
Ad Hoc 4	5	0.0052	17	**0**	0.0059	20	**0**	0.0115	20	**0**	**0.0051**	**16**	**0**	**0.0051**	**16**	**0**
	10	0.0064	23	**0**	0.0063	20	**0**	0.0091	**19**	**0**	0.0062	22	**0**	**0.0059**	23	**0**
	15	0.0072	26	**0**	0.0066	23	**0**	0.0073	**19**	**0**	0.0068	23	**0**	**0.0062**	26	**0**
	20	0.0082	33	**0**	0.0075	32	**0**	0.0079	**31**	**0**	0.0076	33	**0**	**0.0069**	**31**	**0**
	30	0.0084	35	**0**	0.0076	31	**0**	0.0074	**28**	**0**	0.0078	**28**	**0**	**0.0067**	31	**0**
	100	0.0129	51	**0**	0.0093	33	**0**	0.0091	34	**0**	0.0112	**25**	**0**	**0.0078**	27	**0**
Ad Hoc 5	5	0.0053	38	**0**	0.0057	38	**0**	0.0137	49	3	0.0053	**37**	**0**	**0.0051**	38	**0**
	10	0.0056	50	**0**	0.0056	50	**0**	0.0080	**45**	1	0.0056	51	**0**	**0.0053**	50	**0**
	15	0.0059	59	**0**	0.0059	61	**0**	0.0082	**52**	1	0.0058	55	**0**	**0.0054**	56	**0**
	20	0.0060	66	**0**	0.0057	65	**0**	0.0075	69	2	0.0057	63	**0**	**0.0052**	**60**	**0**
	30	0.0064	72	**0**	0.0061	76	**0**	0.0071	70	1	0.0057	65	**0**	**0.0054**	**61**	**0**
	100	0.0079	138	**0**	0.0068	123	**0**	0.0068	109	1	0.0068	117	**0**	**0.0044**	**92**	**0**
Ad Hoc 6	5	0.0077	13	**0**	0.0093	15	**0**	0.0088	**5**	**0**	0.0071	12	**0**	**0.0070**	11	**0**
	10	0.0064	**8**	**0**	0.0076	9	**0**	0.0072	**8**	**0**	0.0061	**8**	**0**	**0.0055**	**8**	**0**
	15	0.0065	6	**0**	0.0073	8	**0**	0.0064	**5**	**0**	0.0060	7	**0**	**0.0054**	6	**0**
	20	0.0069	8	**0**	0.0076	8	**0**	0.0066	**4**	1	0.0062	**4**	1	**0.0055**	**4**	**0**
	30	0.0069	6	**0**	0.0068	8	**0**	0.0057	**5**	1	0.0059	6	1	**0.0051**	**5**	**0**
	100	0.0090	25	**0**	0.0066	17	**0**	0.0059	**14**	2	0.0073	20	2	**0.0040**	15	**0**
Ad Hoc 7	5	**0.0010**	**3**	**0**	0.0015	**3**	**0**	0.0156	81	7	0.0011	**3**	**0**	**0.0010**	**3**	**0**
	10	0.0014	**7**	**0**	0.0019	**7**	**0**	0.0050	26	**0**	0.0015	**7**	**0**	**0.0013**	**7**	**0**
	15	0.0017	**10**	**0**	0.0022	11	**0**	0.0057	30	**0**	0.0019	**10**	**0**	**0.0016**	**10**	**0**
	20	0.0018	**11**	**0**	0.0022	21	**0**	0.0048	30	1	0.0021	12	**0**	**0.0017**	**11**	**0**
	30	0.0020	25	**0**	0.0024	31	**0**	0.0038	20	**0**	0.0023	**19**	**0**	**0.0018**	21	**0**
	100	0.0029	45	**0**	0.0029	45	**0**	0.0027	**38**	1	0.0050	68	1	**0.0025**	48	**0**
Ad Hoc 8	5	**0.0033**	**9**	**1**	0.0040	**9**	**1**	0.0058	16	**1**	0.0042	10	**1**	**0.0033**	**9**	**1**
	10	0.0031	**5**	**1**	0.0036	**5**	**1**	0.0050	13	2	0.0040	10	2	**0.0030**	**5**	**1**
	15	0.0031	**3**	**1**	0.0036	**3**	**1**	0.0040	6	2	0.0039	6	2	**0.0029**	**3**	**1**
	20	0.0032	**6**	**1**	0.0036	12	**1**	0.0037	8	**1**	0.0041	10	**1**	**0.0030**	**6**	**1**
	30	0.0031	5	**1**	0.0034	12	**1**	0.0035	10	**1**	0.0043	8	**1**	**0.0030**	**4**	**1**
	100	0.0036	33	**2**	0.0038	36	**2**	0.0033	**21**	3	0.0069	47	6	**0.0031**	29	**2**

Table 7.6: Continuation of Table 7.5 for the rest of the test collections.

C	n	Pool			BS			kNS			kLP			λTkLP		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Web 9	5	0.0017	**15**	**0**	0.0023	**15**	**0**	0.0122	63	7	0.0030	17	**0**	**0.0015**	**15**	**0**
	10	0.0019	17	**0**	0.0021	17	**0**	0.0059	28	1	0.0034	24	**0**	**0.0016**	**14**	**0**
	15	0.0020	14	**0**	0.0019	14	**0**	0.0032	17	**0**	0.0038	21	1	**0.0017**	**9**	**0**
	20	0.0023	26	**0**	0.0020	20	**0**	0.0049	38	1	0.0042	28	**0**	**0.0018**	**16**	**0**
	30	0.0028	42	**0**	**0.0020**	35	**0**	0.0032	**34**	1	0.0049	60	4	0.0024	38	**0**
	100	0.0043	142	**0**	**0.0030**	109	**0**	0.0032	**101**	**0**	0.0114	268	77	0.0055	155	53
Web 2001	5	**0.0014**	4	**0**	0.0021	4	**0**	0.0057	18	1	0.0024	5	**0**	**0.0014**	**3**	**0**
	10	0.0015	**3**	**0**	0.0020	**3**	**0**	0.0034	9	**0**	0.0026	4	**0**	**0.0014**	**3**	**0**
	15	0.0018	11	**0**	0.0020	11	**0**	0.0042	21	**0**	0.0030	15	**0**	**0.0017**	**10**	**0**
	20	0.0018	**12**	**0**	0.0019	**12**	**0**	0.0035	26	**0**	0.0030	17	**0**	**0.0017**	**12**	**0**
	30	0.0021	18	**0**	**0.0019**	**11**	**0**	0.0023	13	**0**	0.0031	22	**0**	**0.0019**	15	**0**
	100	0.0037	92	**0**	0.0025	65	**0**	**0.0023**	**57**	**0**	0.0065	136	3	0.0038	112	**0**
Web 2002	5	0.0043	80	1	0.0042	80	1	0.0073	116	**0**	0.0048	83	**0**	**0.0041**	**75**	1
	10	0.0049	113	**0**	0.0044	113	**0**	**0.0036**	**87**	**0**	0.0055	116	1	0.0046	112	**0**
	15	0.0052	125	1	0.0043	111	1	**0.0035**	**84**	**0**	0.0061	135	3	0.0049	110	1
	20	0.0050	128	**0**	0.0039	112	**0**	**0.0036**	**90**	**0**	0.0071	192	7	0.0047	126	**0**
	30	0.0050	164	1	**0.0035**	**119**	**0**	0.0036	120	3	0.0088	284	44	0.0045	162	1
	100	0.0027	151	3	0.0017	**96**	3	**0.0016**	99	**2**	0.0111	519	258	0.0030	150	3
Legal 2006	5	0.1044	336	35	0.0774	283	9	0.0445	195	**1**	**0.0433**	**162**	11	0.0482	170	15
	10	0.1138	358	88	0.0747	265	37	**0.0471**	**177**	**12**	0.0703	260	35	0.0703	260	35
	15	0.0758	293	41	0.0498	235	13	**0.0397**	**195**	**8**	0.0557	238	23	0.0535	223	17
	20	0.0569	266	24	0.0373	227	6	**0.0342**	211	**4**	0.0490	225	29	0.0441	**186**	9
	30	0.0379	239	18	**0.0249**	200	8	0.0251	198	**7**	0.0388	215	32	0.0352	**182**	13
	100	0.0114	161	13	**0.0075**	**122**	8	0.0083	138	9	0.0156	188	29	0.0134	167	13
Microblog 2011	5	**0.0047**	110	**0**	0.0052	110	**0**	0.0061	**92**	**0**	**0.0047**	110	**0**	**0.0047**	110	**0**
	10	**0.0054**	134	2	0.0056	134	2	0.0059	**121**	2	**0.0054**	134	**1**	**0.0054**	134	2
	15	0.0062	157	2	0.0060	154	2	**0.0056**	**143**	**0**	0.0059	157	2	0.0061	157	2
	20	0.0068	169	2	0.0064	165	**1**	**0.0051**	**138**	**1**	0.0064	167	2	0.0066	169	2
	30	0.0075	227	3	0.0069	214	3	**0.0054**	**164**	**1**	0.0069	221	3	0.0072	226	3
	100	0.0023	192	**2**	0.0021	188	**2**	**0.0020**	**182**	**2**	0.0023	192	**2**	0.0023	192	**2**
Medical 2011	5	0.0496	173	1	0.0313	101	**0**	0.0296	**66**	**0**	**0.0258**	89	**0**	0.0301	90	**0**
	10	0.0595	229	10	0.0345	126	2	0.0331	**89**	**0**	0.0314	112	9	**0.0289**	100	**0**
	15	0.0396	176	3	0.0230	96	**0**	0.0215	80	**0**	0.0261	101	19	**0.0203**	**78**	**0**
	20	0.0297	132	**1**	0.0172	67	**1**	**0.0165**	**58**	**1**	0.0265	116	23	0.0221	103	3
	30	0.0198	107	**0**	0.0115	61	**0**	**0.0113**	**59**	**0**	0.0250	130	36	0.0161	86	**0**
	100	0.0059	81	**0**	**0.0034**	43	**0**	0.0035	**40**	**0**	0.0138	155	45	0.0071	101	7
Genomics 2005	5	0.0060	64	**0**	0.0059	64	**0**	0.0065	**52**	**0**	**0.0043**	57	**0**	0.0055	62	**0**
	10	0.0066	123	**0**	0.0058	114	**0**	0.0054	**104**	**0**	**0.0046**	111	**0**	0.0057	120	**0**
	15	0.0067	96	**0**	0.0053	81	**0**	0.0051	**65**	**0**	**0.0046**	79	**0**	0.0057	86	**0**
	20	0.0070	100	**0**	0.0053	83	**0**	0.0053	**67**	**0**	**0.0048**	85	**0**	0.0059	86	**0**
	30	0.0082	139	**0**	0.0056	96	**0**	0.0055	**76**	**0**	**0.0053**	96	**0**	0.0064	107	**0**
	100	0.0071	158	**0**	0.0039	96	**0**	**0.0034**	93	**0**	0.0044	**81**	**0**	0.0043	82	**0**
Robust 2005	5	0.0209	19	3	0.0238	23	3	0.0247	16	**0**	**0.0163**	**15**	1	0.0170	16	1
	10	0.0240	20	10	0.0275	22	7	0.0281	**13**	**6**	**0.0187**	14	**6**	0.0204	16	**6**
	15	0.0265	26	9	0.0281	29	10	0.0303	**14**	5	**0.0202**	16	**4**	0.0221	19	**4**
	20	0.0288	27	9	0.0282	28	9	0.0299	**14**	**5**	**0.0209**	19	**5**	0.0236	20	**5**
	30	0.0326	27	11	0.0293	31	10	0.0302	**21**	8	**0.0225**	**21**	**7**	0.0253	**21**	**7**
	100	0.0231	36	5	0.0160	30	5	0.0159	**25**	4	0.0155	27	**1**	**0.0145**	27	**1**

Table 7.7: Summary of the results for P@n of the Reduced Pool and its four presented estimators. These are generated through a leave-one organisation-out approach using the top 75% best performing pooled runs. The dotted lines represent the point when $n \leq K$ becomes false, where K is the depth of the Depth@K strategy used to build the test collection.

C	n	Pool			BS			kNS			kLP			λTkLP		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Ad Hoc 2	5	0.0065	19	**0**	0.0071	18	**0**	**0.0043**	**15**	**0**	0.0055	19	**0**	0.0063	19	**0**
	10	0.0085	32	**0**	0.0086	27	**0**	**0.0043**	**18**	**0**	0.0071	31	**0**	0.0082	32	**0**
	15	0.0120	44	1	0.0108	32	**0**	**0.0058**	**14**	**0**	0.0090	35	**0**	0.0116	42	1
	20	0.0138	52	**0**	0.0119	48	**0**	**0.0051**	**27**	**0**	0.0102	49	**0**	0.0132	50	**0**
	30	0.0164	56	1	0.0129	42	**0**	**0.0073**	**26**	**0**	0.0121	51	**0**	0.0148	51	**0**
	100	0.0293	122	10	0.0168	62	3	**0.0091**	**29**	**0**	0.0170	81	4	0.0214	87	4
Ad Hoc 3	5	0.0028	**2**	**0**	0.0043	**2**	**0**	0.0046	**2**	**0**	**0.0027**	**2**	**0**	**0.0027**	**2**	**0**
	10	0.0026	2	**0**	0.0032	**0**	**0**	0.0041	3	**0**	**0.0025**	2	**0**	**0.0025**	2	**0**
	15	0.0035	3	**0**	0.0041	5	**0**	0.0033	**2**	**0**	**0.0032**	3	**0**	0.0033	3	**0**
	20	0.0043	5	**0**	0.0049	4	**0**	**0.0038**	**3**	**0**	0.0039	4	**0**	0.0041	4	**0**
	30	0.0050	10	**0**	0.0057	11	**0**	**0.0039**	**8**	**0**	0.0043	9	**0**	0.0045	9	**0**
	100	0.0089	19	**0**	0.0077	20	**0**	**0.0046**	**11**	**0**	0.0062	12	**0**	0.0069	14	**0**
Ad Hoc 4	5	0.0073	18	**0**	0.0070	20	**0**	0.0081	21	**0**	**0.0057**	**16**	**0**	0.0065	17	**0**
	10	0.0093	25	**0**	0.0082	21	**0**	0.0075	**18**	**0**	**0.0068**	23	**0**	0.0081	23	**0**
	15	0.0095	28	**0**	0.0088	23	**0**	**0.0068**	**16**	**0**	0.0072	22	**0**	0.0079	25	**0**
	20	0.0107	33	**0**	0.0099	32	**0**	**0.0075**	**25**	**0**	0.0080	29	**0**	0.0088	29	**0**
	30	0.0117	32	**0**	0.0100	31	**0**	**0.0077**	**23**	**0**	**0.0077**	24	**0**	0.0086	27	**0**
	100	0.0161	52	**0**	0.0117	36	1	0.0089	**22**	**0**	0.0098	24	**0**	**0.0083**	26	**0**
Ad Hoc 5	5	0.0069	39	**0**	0.0073	39	**0**	0.0096	50	**0**	**0.0064**	37	**0**	0.0067	39	**0**
	10	0.0075	50	**0**	0.0072	49	**0**	**0.0057**	**42**	**0**	0.0066	49	**0**	0.0070	50	**0**
	15	0.0077	57	**0**	0.0077	59	**0**	**0.0055**	**46**	**0**	0.0066	**46**	**0**	0.0071	52	**0**
	20	0.0079	65	**0**	0.0077	57	**0**	**0.0055**	**49**	**0**	0.0063	55	**0**	0.0068	57	**0**
	30	0.0086	74	**0**	0.0081	69	**0**	**0.0056**	59	**0**	0.0064	59	**0**	0.0069	**58**	**0**
	100	0.0106	133	**0**	0.0086	114	**0**	0.0068	97	**0**	0.0059	108	**0**	**0.0051**	**74**	**0**
Ad Hoc 6	5	0.0097	12	**0**	0.0112	14	**0**	**0.0061**	**8**	**0**	0.0082	10	**0**	0.0086	11	**0**
	10	0.0091	6	**0**	0.0112	7	**0**	**0.0048**	8	**0**	0.0071	**5**	**0**	0.0075	6	**0**
	15	0.0064	6	**0**	0.0070	9	**0**	**0.0042**	**4**	**0**	0.0053	5	**0**	0.0056	5	**0**
	20	0.0072	9	**0**	0.0074	8	**0**	**0.0047**	**1**	**0**	0.0057	5	**0**	0.0057	5	**0**
	30	0.0076	6	**0**	0.0068	11	**0**	**0.0041**	**2**	**0**	0.0052	4	**0**	0.0054	4	**0**
	100	0.0109	23	**0**	0.0071	19	**0**	**0.0043**	**11**	**0**	0.0072	16	**0**	0.0058	18	**0**
Ad Hoc 7	5	**0.0012**	**4**	**0**	0.0019	**4**	**0**	0.0060	42	**0**	0.0013	**4**	**0**	**0.0012**	**4**	**0**
	10	0.0018	8	**0**	0.0024	8	**0**	0.0028	18	**0**	**0.0017**	**7**	**0**	**0.0017**	8	**0**
	15	0.0021	**10**	**0**	0.0028	19	**0**	0.0033	28	**0**	**0.0020**	**10**	**0**	**0.0020**	**10**	**0**
	20	0.0023	**12**	**0**	0.0029	19	**0**	0.0030	28	**0**	**0.0022**	**12**	**0**	**0.0022**	**12**	**0**
	30	0.0026	27	**0**	0.0031	32	**0**	0.0027	25	**0**	**0.0022**	**20**	**0**	0.0023	25	**0**
	100	0.0038	40	**0**	0.0038	47	**0**	**0.0027**	**32**	**0**	0.0043	71	**0**	0.0030	45	**0**
Ad Hoc 8	5	0.0043	**11**	4	0.0048	**11**	4	0.0045	15	**0**	**0.0041**	**11**	4	**0.0041**	**11**	4
	10	0.0042	11	6	0.0048	11	6	0.0047	16	**0**	0.0041	**9**	4	**0.0040**	**9**	4
	15	0.0039	3	2	0.0044	**2**	**1**	0.0042	15	**1**	0.0037	**2**	**1**	**0.0036**	**2**	**1**
	20	0.0041	7	2	0.0046	13	2	0.0042	11	**1**	0.0039	7	**1**	**0.0038**	**6**	**1**
	30	0.0042	6	2	0.0046	8	**1**	0.0040	9	**1**	0.0039	8	**1**	**0.0038**	**5**	**1**
	100	0.0048	35	3	0.0053	40	3	0.0035	**17**	**2**	0.0046	28	**2**	**0.0033**	26	**2**

Table 7.8: Continuation of Table 7.7 for the rest of the test collections.

C	n	Pool			BS			kNS			kLP			λTkLP		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Web 9	5	**0.0022**	**15**	**0**	0.0030	**15**	**0**	0.0162	109	**0**	0.0024	21	**0**	**0.0022**	**15**	**0**
	10	0.0025	19	**0**	0.0028	19	**0**	0.0048	27	**0**	0.0023	**16**	**0**	**0.0022**	**16**	**0**
	15	0.0026	18	**0**	**0.0025**	**11**	**0**	0.0031	12	**0**	**0.0025**	19	**0**	**0.0025**	16	**0**
	20	0.0030	26	**0**	0.0027	21	**0**	0.0033	24	**0**	**0.0025**	**14**	**0**	0.0028	25	**0**
	30	0.0035	42	**0**	**0.0026**	**34**	**0**	0.0029	38	2	0.0027	43	**0**	0.0028	39	**0**
	100	0.0053	141	**0**	0.0036	104	**0**	**0.0028**	**67**	**0**	0.0077	190	11	0.0045	106	**0**
Web 2001	5	**0.0010**	5	**0**	0.0015	5	**0**	0.0073	54	**0**	0.0015	**4**	**0**	**0.0010**	5	**0**
	10	**0.0016**	**7**	**0**	0.0022	**7**	**0**	0.0028	14	**0**	0.0018	8	**0**	**0.0016**	**7**	**0**
	15	0.0021	**12**	**0**	0.0023	**12**	**0**	0.0030	19	**0**	0.0023	16	**0**	**0.0020**	**12**	**0**
	20	0.0019	**13**	**0**	0.0022	15	**0**	0.0029	29	**0**	0.0023	18	**0**	**0.0017**	**13**	**0**
	30	0.0022	26	**0**	0.0021	**13**	**0**	0.0027	41	**0**	0.0022	33	**0**	**0.0019**	25	**0**
	100	0.0043	83	**0**	0.0026	57	**0**	**0.0020**	**32**	**0**	0.0054	130	2	0.0044	102	2
Web 2002	5	**0.0038**	54	**0**	0.0039	54	**0**	0.0040	**52**	**0**	**0.0038**	55	**0**	0.0039	54	**0**
	10	0.0049	82	**0**	0.0039	81	**0**	**0.0024**	**52**	**0**	0.0046	88	**0**	0.0044	78	**0**
	15	0.0049	88	**0**	0.0038	68	**0**	**0.0029**	**47**	**0**	0.0055	83	2	0.0046	83	**0**
	20	0.0050	90	**0**	0.0036	76	**0**	**0.0028**	**53**	**0**	0.0067	121	4	0.0039	76	**0**
	30	0.0052	126	1	0.0034	80	1	**0.0032**	**77**	**0**	0.0092	168	34	0.0042	108	1
	100	0.0032	128	1	0.0018	75	1	**0.0015**	**67**	**0**	0.0135	380	167	0.0039	142	2
Legal 2006	5	0.1211	254	16	0.0868	222	4	**0.0493**	171	**1**	0.0612	**170**	8	0.0648	172	8
	10	0.1301	256	55	0.0831	201	32	**0.0547**	**164**	**5**	0.0992	229	40	0.0992	229	40
	15	0.0867	234	35	0.0554	184	**7**	**0.0454**	**170**	**7**	0.0823	213	36	0.0787	186	34
	20	0.0650	208	13	0.0416	174	**2**	**0.0391**	172	**2**	0.0716	199	42	0.0680	**168**	42
	30	0.0434	185	8	**0.0277**	**156**	**2**	0.0288	157	**2**	0.0554	191	40	0.0516	166	23
	100	0.0130	122	9	**0.0083**	**103**	**5**	0.0095	109	6	0.0210	158	36	0.0204	144	20
Microblog 2011	5	**0.0065**	**101**	**0**	0.0070	**101**	**0**	0.0067	137	**0**	**0.0065**	**101**	**0**	**0.0065**	**101**	**0**
	10	0.0074	152	**1**	0.0076	149	**1**	**0.0062**	**139**	**1**	0.0073	152	**1**	0.0074	152	**1**
	15	0.0077	170	**0**	0.0076	166	**0**	**0.0056**	**148**	**0**	0.0076	170	**0**	0.0076	170	**0**
	20	0.0083	167	**0**	0.0081	173	**0**	**0.0059**	**141**	**0**	0.0081	167	**0**	0.0082	167	**0**
	30	0.0093	217	**0**	0.0088	210	**0**	**0.0062**	**153**	**0**	0.0089	214	**0**	0.0091	217	**0**
	100	0.0029	194	**0**	0.0027	176	**0**	**0.0026**	**167**	**0**	0.0028	187	**0**	0.0029	194	**0**
Medical 2011	5	0.0537	156	2	0.0322	90	**0**	**0.0179**	**52**	**0**	0.0226	69	**0**	0.0291	72	**0**
	10	0.0700	223	10	0.0391	121	1	0.0238	80	**0**	**0.0226**	**59**	1	0.0238	62	1
	15	0.0467	160	1	0.0260	81	1	**0.0176**	70	**0**	0.0179	**56**	**0**	0.0233	75	**0**
	20	0.0350	118	1	0.0195	62	1	**0.0144**	**49**	**0**	0.0173	54	**0**	0.0221	74	**0**
	30	0.0233	93	**0**	0.0130	69	**0**	**0.0105**	**52**	**0**	0.0158	65	**0**	0.0168	78	**0**
	100	0.0067	90	**0**	0.0037	64	**0**	**0.0034**	**63**	**0**	0.0081	85	6	0.0068	82	**0**
Genomics 2005	5	0.0063	69	**0**	0.0057	69	**0**	0.0060	**49**	**0**	**0.0051**	61	**0**	0.0059	64	**0**
	10	0.0072	123	**0**	0.0060	111	**0**	**0.0040**	**73**	**0**	0.0054	112	**0**	0.0064	118	**0**
	15	0.0078	103	**0**	0.0057	82	**0**	**0.0033**	**36**	**0**	0.0055	89	**0**	0.0071	98	**0**
	20	0.0089	102	**0**	0.0068	79	**0**	**0.0031**	**49**	**0**	0.0056	76	**0**	0.0081	91	**0**
	30	0.0100	137	**0**	0.0069	94	**0**	**0.0032**	**56**	**0**	0.0053	89	**0**	0.0082	105	**0**
	100	0.0089	171	**0**	0.0050	92	**0**	0.0046	84	**0**	**0.0043**	81	**0**	0.0056	**80**	**0**
Robust 2005	5	0.0243	18	3	0.0310	26	3	**0.0138**	**8**	**0**	0.0177	15	1	0.0189	15	1
	10	0.0291	19	10	0.0353	25	7	**0.0156**	**10**	**2**	0.0213	14	6	0.0233	14	6
	15	0.0318	26	9	0.0350	29	10	**0.0156**	**5**	**1**	0.0227	16	4	0.0255	19	4
	20	0.0345	28	9	0.0351	31	9	**0.0149**	**8**	**0**	0.0236	16	4	0.0268	16	4
	30	0.0396	30	12	0.0363	32	10	**0.0167**	**9**	**3**	0.0240	18	5	0.0279	18	5
	100	0.0283	35	5	0.0196	32	5	0.0138	30	4	**0.0128**	**19**	**0**	0.0141	21	**0**

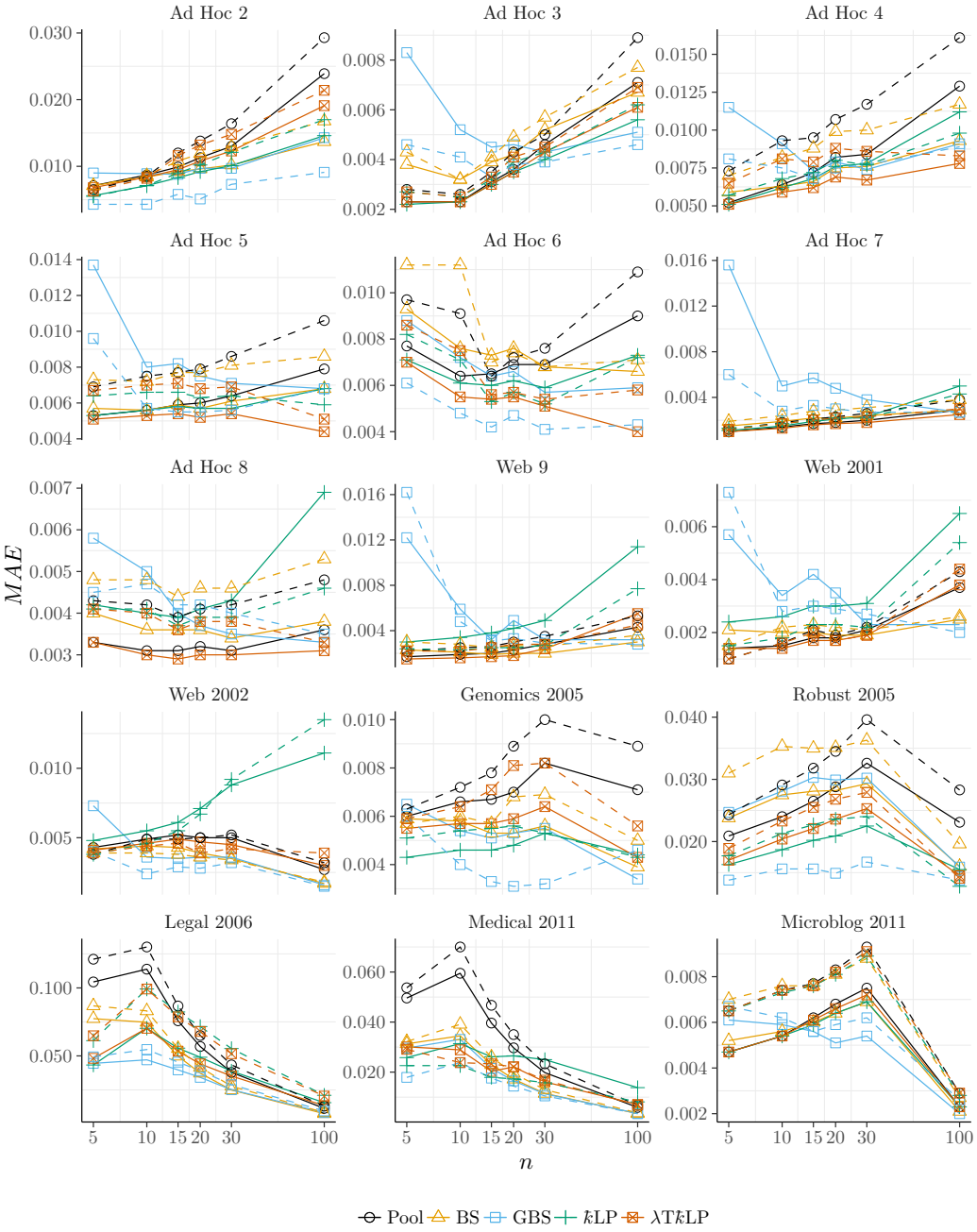

Figure 7.4: Plots per test collection of the Mean Absolute Error against the P@n of the Reduced Pool and the four presented approaches to correct pool bias. Generated using a leave-one organisation-out, using all the pooled runs for the continuous lines and only the top 75% best performing pooled runs for the dashed lines.

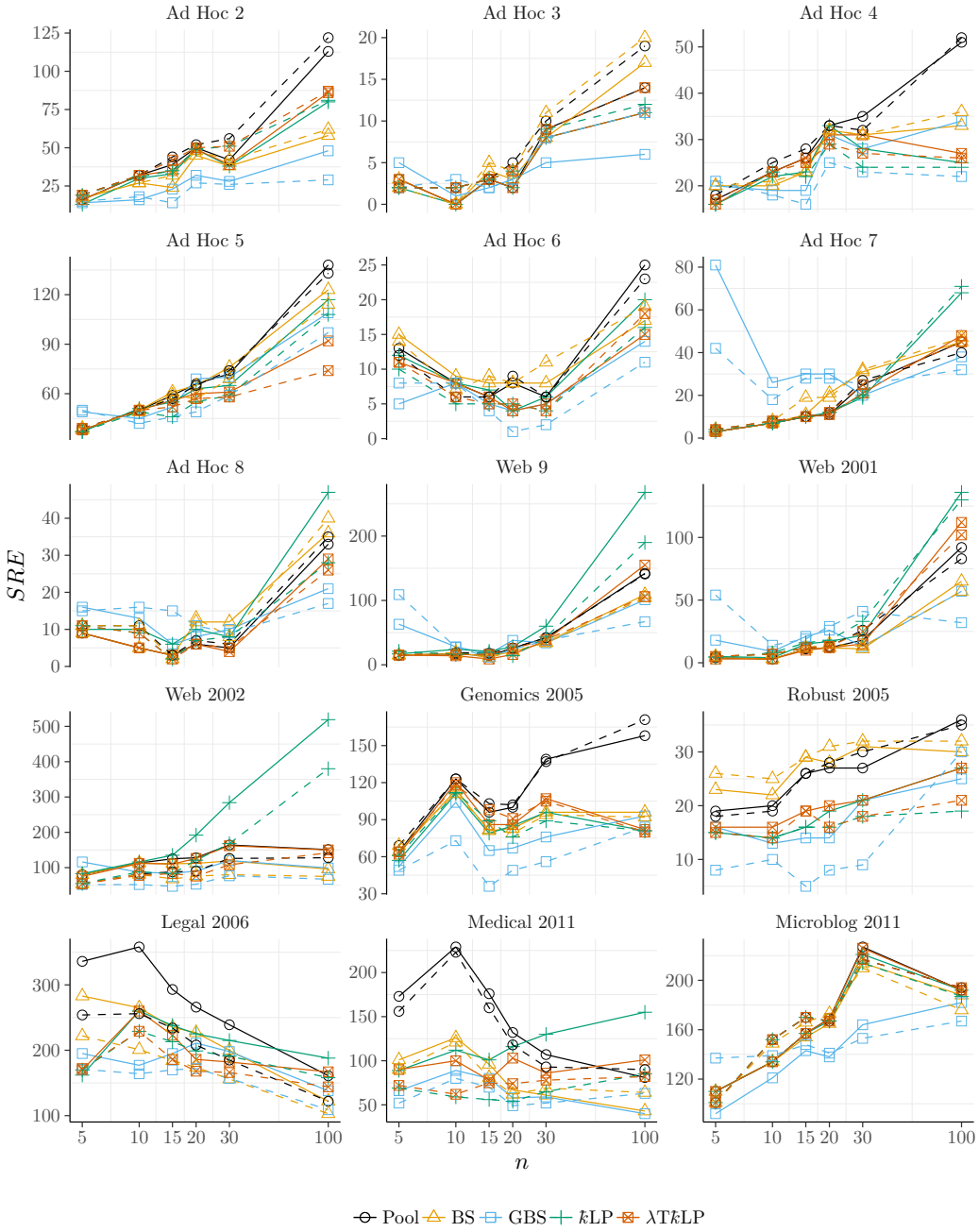

Figure 7.5: Plots per test collection of the System Rank Error against the P@n of the Reduced Pool and the four presented approaches to correct pool bias. Generated using a leave-one organisation-out, using all the pooled runs for the continuous lines and only the top 75% best performing pooled runs for the dashed lines.

Table 7.9: Summary of the results for R@n of the Reduced Pool and its three presented estimators. These are generated through a leave-one organisation-out approach using the top 75% best performing pooled runs. The dotted lines represent the point when $n \leq K$ becomes false, where K is the depth of the Depth@K strategy used to build the test collection.

C	n	Pool			BS			GS			kNS		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Ad Hoc 2	5	**0.0001**	10	**1**	**0.0001**	9	**1**	**0.0001**	**8**	**1**	**0.0001**	10	**1**
	10	**0.0003**	10	**1**	**0.0003**	10	**1**	**0.0003**	**9**	**1**	**0.0003**	10	1
	15	**0.0004**	10	**1**	0.0005	10	**1**	**0.0004**	**9**	**1**	**0.0004**	11	**1**
	20	**0.0006**	**5**	**1**	0.0007	6	**1**	**0.0006**	6	**1**	**0.0006**	**5**	**1**
	30	0.0011	14	**0**	0.0011	18	**0**	0.0010	**13**	**0**	**0.0009**	14	**0**
	100	0.0084	70	1	0.0049	32	**0**	0.0058	41	**0**	**0.0048**	**25**	**0**
Ad Hoc 3	5	**0.0002**	4	**0**	**0.0002**	**3**	**0**	**0.0002**	**3**	**0**	**0.0002**	4	**0**
	10	**0.0004**	**0**	**0**	**0.0004**	1	**0**	**0.0004**	1	**0**	**0.0004**	1	**0**
	15	**0.0005**	3	**0**	**0.0005**	**2**	**0**	**0.0005**	**2**	**0**	**0.0005**	3	**0**
	20	0.0006	**3**	**0**	**0.0005**	**3**	**0**	**0.0005**	**3**	**0**	0.0006	**3**	**0**
	30	0.0007	**1**	**0**	**0.0006**	**1**	**0**	**0.0006**	2	**0**	0.0008	3	**0**
	100	**0.0014**	**2**	**0**	0.0017	4	**0**	0.0015	**2**	**0**	0.0019	3	**0**
Ad Hoc 4	5	**0.0002**	5	**0**	**0.0002**	**4**	**0**	**0.0002**	**4**	**0**	**0.0002**	6	**0**
	10	**0.0003**	**6**	**0**	**0.0003**	7	**0**	**0.0003**	9	**0**	0.0004	**6**	**0**
	15	**0.0005**	6	**0**	**0.0005**	5	**0**	**0.0005**	6	**0**	0.0006	**4**	**0**
	20	**0.0007**	6	**0**	**0.0007**	**3**	**0**	**0.0007**	4	**0**	0.0008	4	**0**
	30	**0.0011**	**8**	**0**	**0.0011**	9	**0**	**0.0011**	9	**0**	0.0012	8	**0**
	100	0.0068	23	**0**	**0.0044**	**10**	**0**	0.0049	19	**0**	0.0054	17	**0**
Ad Hoc 5	5	**0.0001**	14	**0**	**0.0001**	14	**0**	**0.0001**	14	**0**	**0.0001**	**13**	**0**
	10	**0.0002**	8	**0**	**0.0002**	8	**0**	**0.0002**	**7**	**0**	**0.0002**	9	**0**
	15	**0.0003**	**10**	**0**	0.0004	**10**	**0**	0.0004	12	**0**	**0.0003**	11	**0**
	20	**0.0005**	14	**0**	**0.0005**	14	**0**	**0.0005**	15	**0**	**0.0005**	**13**	**0**
	30	0.0008	**17**	**0**	0.0008	**17**	**0**	0.0008	**17**	**0**	**0.0007**	18	**0**
	100	0.0038	58	**0**	**0.0032**	52	**0**	0.0033	50	**0**	0.0034	**40**	**0**
Ad Hoc 6	5	**0.0003**	3	**0**	0.0004	3	**0**	**0.0003**	3	**0**	**0.0003**	**2**	**0**
	10	**0.0004**	2	**0**	0.0005	3	**0**	**0.0004**	2	**0**	**0.0004**	**1**	**0**
	15	**0.0007**	**1**	**0**	**0.0007**	**1**	**0**	**0.0007**	**1**	**0**	**0.0007**	**1**	**0**
	20	**0.0008**	**1**	**0**	**0.0008**	**1**	**0**	**0.0008**	**1**	**0**	**0.0008**	**1**	**0**
	30	0.0011	5	**0**	0.0012	7	**0**	0.0012	6	**0**	**0.0010**	**4**	**0**
	100	0.0055	8	**0**	0.0041	5	**0**	**0.0036**	7	**0**	0.0042	**4**	1
Ad Hoc 7	5	**0.0001**	**7**	**0**	**0.0001**	8	**0**	**0.0001**	8	**0**	**0.0001**	**7**	**0**
	10	**0.0001**	**5**	**0**	**0.0001**	6	**0**	**0.0001**	6	**0**	**0.0001**	6	**0**
	15	**0.0002**	**6**	**0**	**0.0002**	7	**0**	**0.0002**	7	**0**	**0.0002**	**6**	**0**
	20	**0.0002**	**2**	**0**	**0.0002**	**2**	**0**	**0.0002**	**2**	**0**	**0.0002**	5	**0**
	30	**0.0003**	**9**	**0**	**0.0003**	11	**0**	**0.0003**	10	**0**	**0.0003**	13	**0**
	100	0.0016	23	**0**	0.0015	**17**	**0**	**0.0014**	18	**0**	0.0019	20	**0**
Ad Hoc 8	5	**0.0001**	**3**	**0**	**0.0001**	8	**0**	**0.0001**	**3**	**0**	**0.0001**	**3**	**0**
	10	**0.0002**	**2**	**0**	**0.0002**	3	**0**	**0.0002**	**2**	**0**	**0.0002**	**2**	**0**
	15	**0.0003**	2	**0**	**0.0003**	**1**	**0**	**0.0003**	**1**	**0**	**0.0003**	**1**	**0**
	20	**0.0004**	**2**	**0**	**0.0004**	3	**0**	**0.0004**	4	**0**	**0.0004**	**2**	**0**
	30	**0.0005**	2	**0**	0.0006	4	**0**	0.0006	4	**0**	**0.0005**	**1**	**0**
	100	0.0021	15	**0**	0.0021	19	**0**	**0.0018**	**13**	**0**	0.0019	**13**	**0**

Table 7.10: Continuation of Table 7.9 for the rest of the test collections.

C	n	Pool			BS			GS			kNS		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Web 9	5	**0.0005**	**14**	**0**	**0.0005**	21	**0**	**0.0005**	20	**0**	**0.0005**	15	**0**
	10	**0.0006**	**20**	**0**	**0.0006**	**20**	**0**	**0.0006**	**20**	**0**	**0.0006**	21	**0**
	15	**0.0006**	22	**0**	**0.0006**	21	**0**	**0.0006**	**20**	**0**	**0.0006**	24	**0**
	20	**0.0006**	**22**	**0**	**0.0006**	23	**0**	**0.0006**	23	**0**	0.0007	23	**0**
	30	**0.0008**	21	**0**	**0.0008**	21	**0**	**0.0008**	21	**0**	**0.0008**	**19**	**0**
	100	0.0027	36	**0**	**0.0018**	23	**0**	0.0020	24	**0**	**0.0018**	**11**	**0**
Web 2001	5	0.0002	6	**0**	**0.0001**	**4**	**0**	0.0002	**4**	**0**	0.0002	6	**0**
	10	**0.0003**	9	**0**	**0.0003**	**8**	**0**	**0.0003**	**8**	**0**	**0.0003**	11	**0**
	15	**0.0004**	**6**	**0**	**0.0004**	**6**	**0**	**0.0004**	**6**	**0**	**0.0004**	**6**	**0**
	20	**0.0005**	11	**0**	**0.0005**	11	**0**	**0.0005**	**10**	**0**	**0.0005**	11	**0**
	30	**0.0006**	**6**	**0**	**0.0006**	**6**	**0**	**0.0006**	**6**	**0**	**0.0006**	7	**0**
	100	0.0026	31	**0**	**0.0018**	**17**	**0**	0.0020	20	**0**	0.0019	18	**0**
Web 2002	5	**0.0006**	**32**	**0**	**0.0006**	**32**	**0**	**0.0006**	**32**	**0**	**0.0006**	**32**	**0**
	10	0.0013	46	**0**	0.0013	39	**0**	0.0013	46	**0**	**0.0012**	**36**	**0**
	15	0.0018	56	**0**	0.0017	51	**0**	0.0018	52	**0**	**0.0015**	**43**	**0**
	20	0.0022	50	**0**	0.0020	40	**0**	0.0021	45	**0**	**0.0016**	**35**	**0**
	30	0.0032	66	**0**	0.0025	53	**0**	0.0028	61	**0**	**0.0018**	**31**	**0**
	100	0.0054	71	**0**	0.0037	52	**0**	0.0043	59	**0**	**0.0030**	**44**	**0**
Legal 2006	5	0.0237	253	3	0.0190	223	**2**	0.0218	242	**2**	**0.0182**	**215**	**2**
	10	0.0514	274	10	0.0358	208	**0**	0.0411	234	2	**0.0310**	**179**	**0**
	15	0.0502	259	8	0.0351	190	**0**	0.0403	215	**0**	**0.0314**	**182**	**0**
	20	0.0494	226	1	0.0345	175	**0**	0.0396	196	**0**	**0.0316**	**169**	**0**
	30	0.0483	203	**3**	0.0338	151	**3**	0.0389	170	**3**	**0.0316**	**147**	**3**
	100	0.0448	153	**7**	0.0317	117	**7**	0.0363	129	**7**	**0.0305**	**114**	**7**
Microblog 2011	5	**0.0003**	21	**0**	**0.0003**	**17**	**0**	**0.0003**	31	**0**	**0.0003**	19	**0**
	10	0.0006	36	**0**	0.0006	36	**0**	0.0006	38	**0**	**0.0005**	**33**	**0**
	15	0.0011	62	**0**	0.0011	68	**0**	0.0011	63	**0**	**0.0009**	**56**	**0**
	20	0.0015	56	**0**	0.0015	57	**0**	0.0015	**52**	**0**	**0.0013**	53	**0**
	30	0.0027	95	**0**	0.0025	91	**0**	0.0025	90	**0**	**0.0019**	**65**	**0**
	100	0.0027	82	**0**	**0.0024**	74	**0**	**0.0024**	76	**0**	**0.0024**	**72**	**0**
Medical 2011	5	0.0040	77	**0**	**0.0027**	**41**	**0**	0.0031	51	**0**	**0.0027**	**41**	**0**
	10	0.0104	93	**0**	0.0062	65	**0**	0.0064	63	**0**	**0.0060**	**58**	**0**
	15	0.0099	75	**0**	0.0059	37	**0**	0.0060	39	**0**	**0.0056**	**36**	**0**
	20	0.0095	64	**0**	0.0057	40	**0**	0.0058	43	**0**	**0.0054**	**37**	**0**
	30	0.0088	55	**0**	0.0053	28	**0**	0.0054	28	**0**	**0.0051**	**21**	**0**
	100	0.0064	38	**0**	0.0040	**18**	**0**	0.0040	22	**0**	**0.0039**	**18**	**0**
Genomics 2005	5	**0.0003**	**21**	**0**	**0.0003**	23	**0**	**0.0003**	**21**	**0**	**0.0003**	23	**0**
	10	**0.0006**	**27**	**0**	**0.0006**	29	**0**	**0.0006**	**27**	**0**	**0.0006**	30	**0**
	15	**0.0010**	26	**0**	**0.0010**	24	**0**	**0.0010**	26	**0**	**0.0010**	**23**	**0**
	20	0.0011	29	**0**	0.0011	32	**0**	0.0011	34	**0**	**0.0010**	**28**	**0**
	30	0.0016	35	**0**	0.0015	32	**0**	0.0015	36	**0**	**0.0012**	**31**	**0**
	100	0.0047	64	**0**	0.0029	48	**0**	0.0034	46	**0**	**0.0025**	**35**	**0**
Robust 2005	5	**0.0008**	10	**0**	0.0009	11	**0**	0.0009	10	**0**	**0.0008**	**8**	**0**
	10	**0.0018**	**8**	**0**	0.0019	10	**0**	0.0020	11	**0**	0.0021	**8**	**0**
	15	**0.0028**	16	**0**	0.0031	15	**0**	0.0031	14	**0**	0.0031	**12**	**0**
	20	**0.0037**	12	**0**	0.0043	17	**0**	0.0039	16	**0**	0.0040	**11**	**0**
	30	0.0058	16	**0**	0.0062	15	**0**	**0.0046**	13	**0**	0.0056	**8**	**0**
	100	0.0150	20	1	0.0112	16	**0**	**0.0093**	11	**0**	0.0108	**7**	**0**

Table 7.11: Summary of the results for R@n of the Reduced Pool and its three presented estimators. These are generated through a leave-one organisation-out approach using all the pooled runs. The dotted lines represent the point when $n \leq K$ becomes false, where K is the depth of the Depth@K strategy used to build the test collection.

C	n	Pool			BS			GS			$\hbar$NS		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Ad Hoc 2	5	0.0002	13	1	**0.0001**	10	1	**0.0001**	**8**	**0**	**0.0001**	12	1
	10	**0.0003**	**11**	**1**	**0.0003**	**11**	**1**	**0.0003**	**11**	**1**	**0.0003**	12	**1**
	15	0.0006	**8**	**1**	0.0006	9	**1**	**0.0005**	9	**1**	**0.0005**	13	**1**
	20	**0.0008**	7	**1**	0.0009	**6**	**1**	0.0009	7	**1**	**0.0008**	**6**	**1**
	30	0.0014	16	**0**	0.0014	17	**0**	0.0014	15	**0**	**0.0011**	**10**	**0**
	100	0.0112	77	1	0.0063	41	**0**	0.0074	45	1	**0.0032**	**21**	**0**
Ad Hoc 3	5	**0.0003**	**3**	**0**	**0.0003**	5	**0**	**0.0003**	**3**	**0**	**0.0003**	**3**	**0**
	10	0.0006	3	**0**	0.0006	**2**	**0**	**0.0005**	**2**	**0**	0.0006	4	**0**
	15	0.0007	5	**0**	**0.0006**	**4**	**0**	**0.0006**	5	**0**	0.0008	5	**0**
	20	0.0008	**1**	**0**	0.0008	4	**0**	**0.0007**	3	**0**	0.0009	3	**0**
	30	0.0009	2	**0**	**0.0008**	1	**0**	**0.0008**	**0**	**0**	0.0010	2	**0**
	100	**0.0015**	**1**	**0**	0.0020	3	**0**	0.0017	3	**0**	0.0020	2	**0**
Ad Hoc 4	5	**0.0003**	7	**0**	**0.0003**	7	**0**	**0.0003**	**6**	**0**	**0.0003**	8	**0**
	10	**0.0005**	8	**0**	**0.0005**	8	**0**	**0.0005**	8	**0**	**0.0005**	**6**	**0**
	15	**0.0007**	5	**0**	0.0008	5	**0**	**0.0007**	5	**0**	**0.0007**	**4**	**0**
	20	**0.0009**	**7**	**0**	0.0010	8	**0**	**0.0009**	**7**	**0**	**0.0009**	9	**0**
	30	0.0015	6	**0**	0.0014	7	**0**	0.0015	6	**0**	**0.0012**	**4**	**0**
	100	0.0085	18	**0**	0.0054	**11**	**0**	0.0057	14	**0**	**0.0045**	**11**	**0**
Ad Hoc 5	5	**0.0001**	**5**	**0**	**0.0001**	**5**	**0**	0.0002	**5**	**0**	**0.0001**	7	**0**
	10	**0.0003**	7	**0**	**0.0003**	**6**	**0**	0.0004	8	**0**	**0.0003**	7	**0**
	15	**0.0005**	**15**	**0**	**0.0005**	16	**0**	**0.0005**	17	**0**	**0.0005**	**15**	**0**
	20	0.0007	13	**0**	0.0007	**12**	**0**	0.0007	13	**0**	**0.0006**	9	**0**
	30	0.0011	17	**0**	0.0011	21	**0**	0.0011	**18**	**0**	**0.0009**	**18**	**0**
	100	0.0053	63	**0**	0.0041	51	**0**	0.0044	50	**0**	**0.0033**	**39**	**0**
Ad Hoc 6	5	**0.0004**	**1**	**0**	0.0005	4	**0**	**0.0004**	2	**0**	**0.0004**	2	**0**
	10	**0.0004**	3	**0**	**0.0004**	3	**0**	**0.0004**	3	**0**	**0.0004**	**2**	**0**
	15	0.0008	**1**	**0**	0.0008	**1**	**0**	0.0008	**1**	**0**	**0.0007**	**1**	**0**
	20	**0.0008**	2	**0**	**0.0008**	2	**0**	0.0009	2	**0**	**0.0008**	**0**	**0**
	30	0.0011	**6**	**0**	0.0012	7	**0**	0.0012	8	**0**	**0.0010**	7	**0**
	100	0.0070	11	**0**	0.0047	11	**0**	0.0041	9	**0**	**0.0034**	**5**	**0**
Ad Hoc 7	5	**0.0001**	**8**	**0**	**0.0001**	10	**0**	**0.0001**	**8**	**0**	**0.0001**	9	**0**
	10	**0.0002**	**4**	**0**	**0.0002**	9	**0**	**0.0002**	8	**0**	**0.0002**	**4**	**0**
	15	**0.0003**	**8**	**0**	**0.0003**	**8**	**0**	**0.0003**	**8**	**0**	**0.0003**	10	**0**
	20	**0.0003**	7	**0**	**0.0003**	7	**0**	**0.0003**	**6**	**0**	**0.0003**	8	**0**
	30	**0.0004**	**13**	**0**	**0.0004**	14	**0**	**0.0004**	**13**	**0**	**0.0004**	14	**0**
	100	0.0023	25	**0**	0.0020	**20**	**0**	0.0018	26	**0**	**0.0017**	21	**0**
Ad Hoc 8	5	**0.0002**	**5**	**0**	**0.0002**	10	**0**	**0.0002**	6	**0**	**0.0002**	**5**	**0**
	10	**0.0003**	**3**	**0**	**0.0003**	8	**0**	**0.0003**	5	**0**	**0.0003**	**3**	**0**
	15	**0.0003**	**2**	**0**	**0.0003**	3	**0**	**0.0003**	3	**0**	**0.0003**	**2**	**0**
	20	0.0005	3	**0**	0.0005	4	**0**	**0.0004**	**2**	**0**	0.0005	3	**0**
	30	**0.0007**	2	**0**	**0.0007**	2	**0**	**0.0007**	**1**	**0**	**0.0007**	2	**0**
	100	0.0027	12	**0**	0.0030	17	**0**	0.0024	10	**0**	**0.0022**	**7**	**0**

Table 7.12: Continuation of Table 7.11, for the rest of the test collections.

C	n	Pool			BS			GS			kNS		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Web 9	5	**0.0006**	**12**	**0**	0.0007	16	**0**	**0.0006**	16	**0**	0.0007	**12**	**0**
	10	**0.0007**	18	**0**	0.0008	**17**	**0**	**0.0007**	**17**	**0**	0.0008	20	**0**
	15	**0.0007**	**18**	**0**	**0.0007**	**18**	**0**	**0.0007**	**18**	**0**	0.0008	**18**	**0**
	20	**0.0008**	**25**	**0**	**0.0008**	**25**	**0**	**0.0008**	**25**	**0**	0.0009	27	**0**
	30	**0.0009**	21	**0**	0.0010	21	**0**	0.0010	21	**0**	**0.0009**	**18**	**0**
	100	0.0031	38	**0**	0.0020	22	**0**	0.0023	25	**0**	**0.0019**	**14**	**0**
Web 2001	5	**0.0002**	**6**	**0**	**0.0002**	8	**0**	**0.0002**	7	**0**	**0.0002**	7	**0**
	10	**0.0003**	6	**0**	**0.0003**	**5**	**0**	**0.0003**	**5**	**0**	0.0004	7	**0**
	15	**0.0004**	6	**0**	**0.0004**	6	**0**	**0.0004**	**5**	**0**	0.0005	9	**0**
	20	**0.0005**	5	**0**	**0.0005**	5	**0**	**0.0005**	**4**	**0**	0.0006	9	**0**
	30	**0.0006**	**8**	**0**	**0.0006**	9	**0**	**0.0006**	9	**0**	0.0007	9	**0**
	100	0.0030	28	**0**	0.0020	**19**	**0**	0.0023	24	**0**	**0.0019**	**19**	**0**
Web 2002	5	**0.0007**	24	**0**	**0.0007**	24	**0**	**0.0007**	**22**	**0**	**0.0007**	**22**	**0**
	10	0.0013	28	**0**	0.0013	25	**0**	0.0013	27	**0**	**0.0011**	**23**	**0**
	15	0.0017	39	**0**	0.0016	35	**0**	0.0017	38	**0**	**0.0013**	**27**	**0**
	20	0.0022	36	**0**	0.0020	28	**0**	0.0021	31	**0**	**0.0014**	**17**	**0**
	30	0.0032	42	1	0.0026	37	1	0.0028	39	1	**0.0018**	**30**	**0**
	100	0.0072	63	**0**	0.0049	46	**0**	0.0058	50	**0**	**0.0039**	**37**	**0**
Legal 2006	5	0.0314	225	1	0.0239	185	1	0.0280	213	1	**0.0231**	**179**	1
	10	0.0688	219	10	0.0455	158	2	0.0527	169	3	**0.0397**	**139**	**0**
	15	0.0673	198	3	0.0467	143	**0**	0.0540	160	**0**	**0.0426**	**138**	**0**
	20	0.0634	178	1	0.0427	136	**0**	0.0499	147	**0**	**0.0396**	**129**	**0**
	30	0.0620	158	3	0.0413	114	3	0.0490	125	3	**0.0392**	**112**	3
	100	0.0563	121	**7**	0.0379	**98**	**7**	0.0456	110	**7**	**0.0371**	101	**7**
Microblog 2011	5	**0.0004**	**26**	**0**	**0.0004**	**26**	**0**	**0.0004**	**26**	**0**	**0.0004**	**26**	**0**
	10	**0.0008**	57	**0**	**0.0008**	58	**0**	**0.0008**	53	**0**	**0.0008**	**50**	**0**
	15	0.0014	82	**0**	0.0014	75	**0**	0.0014	79	**0**	**0.0013**	**68**	**0**
	20	0.0021	68	**0**	0.0021	74	**0**	0.0020	68	**0**	**0.0019**	**61**	**0**
	30	0.0037	113	**0**	0.0035	107	**0**	0.0035	105	**0**	**0.0028**	**80**	**0**
	100	0.0034	77	**0**	**0.0032**	73	**0**	**0.0032**	75	**0**	**0.0032**	**71**	**0**
Medical 2011	5	0.0054	79	**0**	0.0033	50	**0**	0.0043	64	**0**	**0.0031**	**38**	**0**
	10	0.0141	100	**0**	0.0075	65	**0**	0.0073	66	**0**	**0.0062**	**52**	**0**
	15	0.0133	76	**0**	0.0071	48	**0**	0.0070	48	**0**	**0.0057**	**31**	**0**
	20	0.0129	75	**0**	0.0071	48	**0**	0.0069	49	**0**	**0.0060**	**35**	**0**
	30	0.0118	60	**0**	0.0065	31	**0**	0.0064	31	**0**	**0.0056**	**27**	**0**
	100	0.0077	41	**0**	0.0046	21	**0**	0.0046	21	**0**	**0.0042**	**18**	**0**
Genomics 2005	5	**0.0005**	30	**0**	**0.0005**	33	**0**	**0.0005**	**29**	**0**	0.0006	31	**0**
	10	**0.0009**	41	**0**	0.0011	40	**0**	**0.0009**	**38**	**0**	0.0010	39	**0**
	15	0.0011	**31**	**0**	0.0011	**31**	**0**	0.0011	**31**	**0**	**0.0010**	**31**	**0**
	20	0.0013	24	**0**	0.0013	30	**0**	0.0013	28	**0**	**0.0011**	**23**	**0**
	30	0.0019	36	**0**	0.0020	38	**0**	0.0019	**35**	**0**	**0.0016**	**35**	**0**
	100	0.0061	87	**0**	0.0039	58	**0**	0.0046	62	**0**	**0.0032**	**53**	**0**
Robust 2005	5	**0.0011**	8	**0**	**0.0011**	**7**	**0**	0.0012	9	**0**	**0.0011**	9	**0**
	10	**0.0023**	**7**	**0**	0.0025	9	**0**	0.0026	9	**0**	0.0024	8	**0**
	15	0.0036	**11**	**0**	0.0042	15	**0**	0.0041	12	**0**	**0.0033**	**11**	**0**
	20	0.0045	11	**0**	0.0058	14	**0**	0.0053	15	**0**	**0.0039**	**8**	**0**
	30	0.0070	17	**0**	0.0084	17	**0**	0.0063	15	**0**	**0.0046**	**7**	**0**
	100	0.0185	22	1	0.0141	20	**0**	0.0114	16	**0**	**0.0094**	**9**	**0**

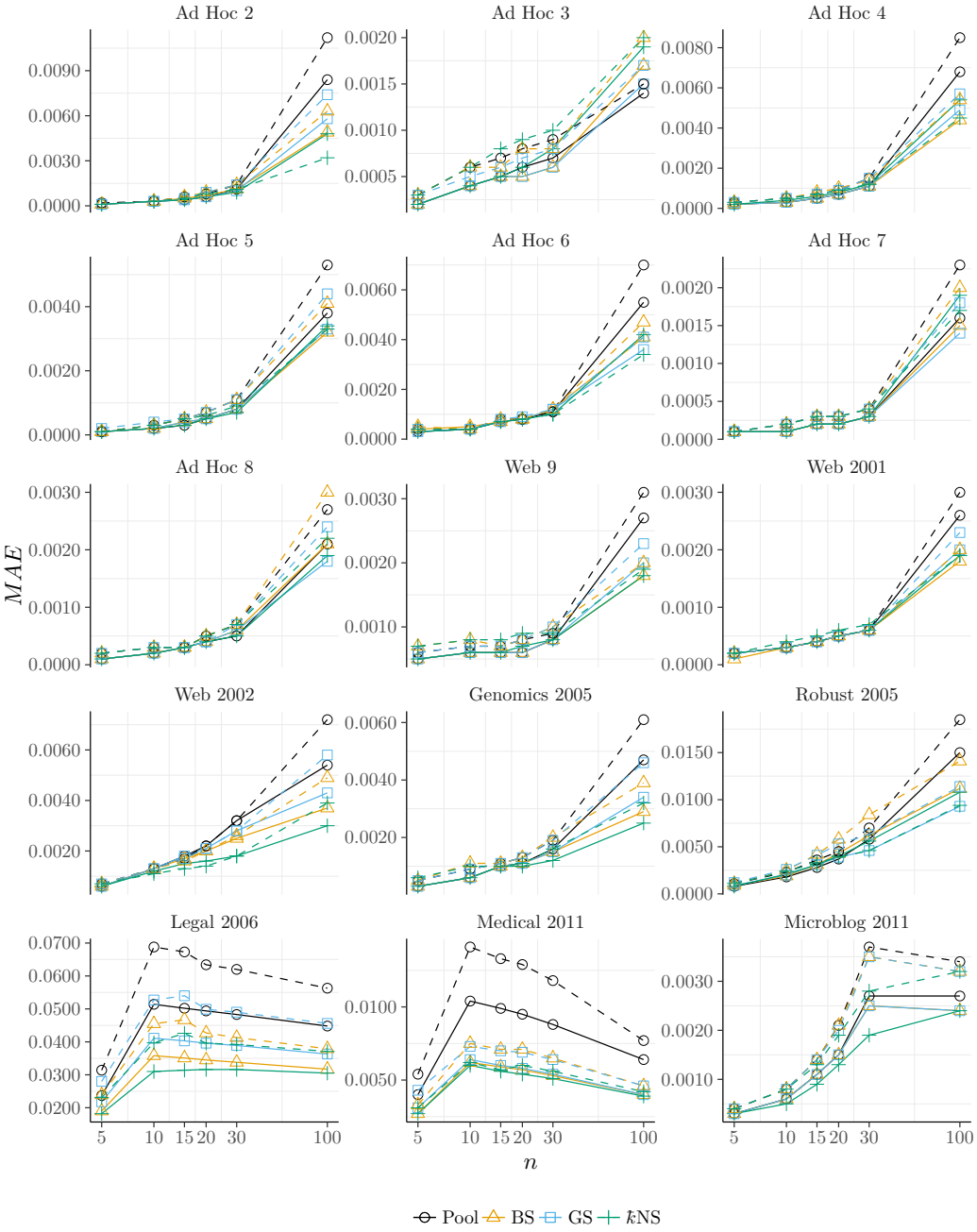

Figure 7.6: Plots per test collection of the Mean Absolute Error against the R@n of the Reduced Pool and the three presented approaches to correct pool bias. Generated using a leave-one organisation-out, using all the pooled runs for the continuous lines and only the top 75% best performing pooled runs for the dashed lines.

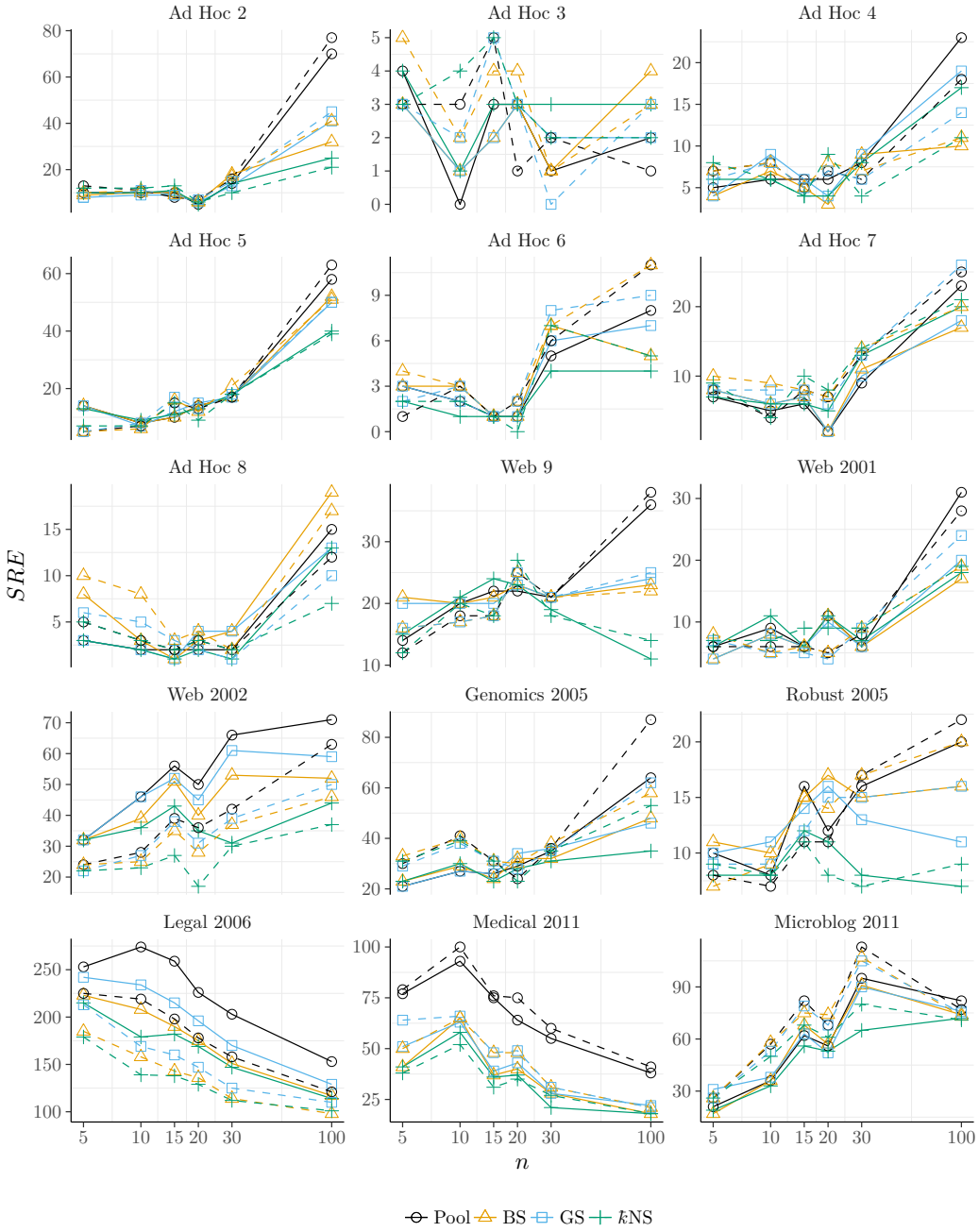

Figure 7.7: Plots per test collection of the System Rank Error against the R@n of the Reduced Pool and the three presented approaches to correct pool bias. Generated using a leave-one organisation-out, using all the pooled runs for the continuous lines and only the top 75% best performing pooled runs for the dashed lines.

only using the 75% best performing runs. These results are also presented in Figure 7.6 for MAE, and 7.7 for SRE.

Finally, for R@n estimators derived from the estimators presented for P@n, we compare, in Tables 7.13 and 7.14, the results of all R@n estimators against the baseline, the 'reduced pool'. In Table 7.15 and 7.16 we present the same but only using the 75% best performing runs. These results are also presented in Figure 7.8 for MAE, and 7.9 for SRE.

7.5 Discussion

Our empirical analysis has shown that the extent of the bias is very different across test collections. In particular we can divide these test collections into three categories based on the order of magnitude of the average bias observed in the reduced pool. The categories are: least biased, biased, and very biased. The least biased test collections are Ad Hoc 3, 5, 8 and 7, Web 9 and 2001. The biased are Ad Hoc 2, 4 and 6, Genomics 2005, Web 2002, and Microblog 2011. Very biased are Robust 2005, Legal 2006, Medical 2011. This is due to a combination of two factors: depth of pool and number of submitted runs (which we assume to be proportional to the variety of submitted runs).

In what follows, we start by discussing the bias estimators for P@n. We then move onto discuss the bias estimators for R@n. For the latter we divide the discussion into two parts: we begin with the estimators designed for R@n, then move to the estimators derived from P@n (see Eq. (7.13)). Finally we compare them against each other.

7.5.1 Bias Estimators for Precision at Cut-off

We start discussing the two simulation-based estimators, BS and kNS. We observe that on average BS performs better when considering all the pooled runs, while it is only better than the reduced pool when considering the 75% of the best performing runs. This can be observed in particular for Ad Hoc 3, 6, 7, 8, Web 2001, and Robust 2005. For kNS, it is evident that this estimator behaves extremely differently when applied to a pool built using all pooled runs with respect to a pool built using just the top 75% of best performing runs. In the former case kNS is usually worse than the reduced pool, while in the latter case it is almost always the best estimator. The best examples where we can observe this worsening behaviour are for Ad Hoc 2, 3, 4, and 5, and Robust 2005. While for Ad Hoc 6, Web 2002, Legal 2006, Medical 2011, the performance of this estimator does not degrade as much when using all the pooled runs. However, for Ad Hoc 7, Web 9 and 2001, kNS is the worst performing estimator in both cases. Moreover, kNS is not a stable estimator: when kNS is not the best performing estimator, it is likely to be the worst. This can be observed for Ad Hoc 3, 4, 5, and 7, Web 9 and 2001. In particular this happens always at low precision cut-offs, and it decreases in severity when the cut-offs are increased. This instability is accentuated when using all the pooled runs. Here, after an empirical analysis, we have two hypotheses for why this happens, and

Table 7.13: Summary of the results for R@n of the Reduced Pool and four estimators developed for P@n and used in combination with Eq. (7.15). These are generated through a leave-one organisation-out approach using all the pooled runs. The dotted lines represent the point when $n \leq K$ becomes false, where K is the depth of the Depth@K strategy used to build the test collection.

C	n	Pool			BS^P			$\hbar NS^P$			$\hbar LP^P$			$\lambda T\hbar LP^P$		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Ad Hoc 2	5	**0.0001**	10	**1**	**0.0001**	10	**1**	0.0004	23	**1**	0.0002	13	**1**	**0.0001**	**7**	**1**
	10	0.0003	10	1	0.0003	10	1	0.0011	35	2	0.0003	7	**0**	**0.0002**	**6**	**0**
	15	**0.0004**	10	1	0.0005	14	1	0.0018	41	5	**0.0004**	**5**	**0**	**0.0004**	10	**0**
	20	**0.0006**	5	1	0.0007	5	1	0.0038	69	10	**0.0006**	**2**	**0**	**0.0006**	**2**	**0**
	30	0.0011	14	**0**	0.0011	17	**0**	0.0048	57	7	**0.0009**	**12**	**0**	**0.0009**	14	**0**
	100	0.0084	70	1	0.0049	32	**0**	0.0125	49	4	0.0060	35	**0**	**0.0046**	**31**	**0**
Ad Hoc 3	5	0.0002	4	**0**	**0.0001**	**2**	**0**	0.0004	4	**0**	0.0004	4	**0**	0.0003	4	**0**
	10	0.0004	**0**	**0**	**0.0003**	**0**	**0**	0.0007	2	**0**	0.0007	4	**0**	0.0006	3	**0**
	15	0.0005	3	**0**	**0.0004**	**1**	**0**	0.0011	2	**0**	0.0010	3	**0**	0.0008	3	**0**
	20	0.0006	3	**0**	**0.0004**	**1**	**0**	0.0019	7	**0**	0.0012	10	**0**	0.0010	7	**0**
	30	0.0007	**1**	**0**	**0.0006**	**1**	**0**	0.0034	5	**0**	0.0015	3	**0**	0.0013	3	**0**
	100	**0.0014**	**2**	**0**	0.0023	4	**0**	0.0095	15	1	0.0021	3	**0**	0.0019	5	**0**
Ad Hoc 4	5	**0.0002**	5	**0**	**0.0002**	**4**	**0**	0.0010	11	**0**	0.0025	62	**0**	0.0021	56	**0**
	10	**0.0003**	**6**	**0**	**0.0003**	7	**0**	0.0017	11	**0**	0.0032	66	**0**	0.0027	60	**0**
	15	**0.0005**	6	**0**	0.0006	**5**	**0**	0.0057	39	2	0.0037	36	**0**	0.0033	38	**0**
	20	**0.0007**	6	**0**	0.0008	**4**	**0**	0.0081	53	2	0.0040	32	**0**	0.0036	35	**0**
	30	**0.0011**	**8**	**0**	0.0012	9	**0**	0.0108	41	2	0.0043	22	**0**	0.0039	24	**0**
	100	0.0068	23	**0**	**0.0045**	**10**	**0**	0.0206	48	2	0.0190	48	1	0.0054	18	**0**
Ad Hoc 5	5	**0.0001**	**14**	**0**	**0.0001**	**14**	**0**	0.0004	15	**0**	0.0048	222	2	0.0045	221	2
	10	**0.0002**	**8**	**0**	**0.0002**	9	**0**	0.0009	18	**0**	0.0062	217	1	0.0062	212	1
	15	**0.0003**	**10**	**0**	0.0004	11	**0**	0.0019	36	**0**	0.0065	183	**0**	0.0066	197	**0**
	20	**0.0005**	**14**	**0**	**0.0005**	**14**	**0**	0.0029	37	**0**	0.0067	182	**0**	0.0070	200	1
	30	**0.0008**	**17**	**0**	**0.0008**	**17**	**0**	0.0086	146	2	0.0060	135	**0**	0.0070	164	**0**
	100	0.0038	58	**0**	**0.0032**	51	**0**	0.0232	238	8	0.0268	332	5	0.0039	**44**	**0**
Ad Hoc 6	5	**0.0003**	3	**0**	0.0005	**2**	**0**	0.0015	8	**0**	0.0051	38	**0**	0.0045	34	**0**
	10	**0.0004**	**2**	**0**	0.0006	3	**0**	0.0025	8	1	0.0067	32	**0**	0.0058	28	**0**
	15	**0.0007**	**1**	**0**	0.0009	**1**	**0**	0.0046	8	2	0.0072	24	**0**	0.0065	21	**0**
	20	**0.0008**	**1**	**0**	0.0011	2	**0**	0.0053	8	3	0.0073	24	1	0.0069	23	1
	30	**0.0011**	**5**	**0**	0.0015	7	**0**	0.0088	17	3	0.0070	20	1	0.0068	20	**0**
	100	0.0055	8	**0**	**0.0047**	**7**	**0**	0.0257	32	4	0.0321	44	4	0.0078	12	1
Ad Hoc 7	5	**0.0001**	7	**0**	**0.0001**	**6**	**0**	0.0010	73	**0**	0.0009	76	**0**	0.0006	51	**0**
	10	**0.0001**	**5**	**0**	**0.0001**	6	**0**	0.0018	53	1	0.0014	70	**0**	0.0009	42	**0**
	15	**0.0002**	**6**	**0**	**0.0002**	7	**0**	0.0025	47	4	0.0017	56	**0**	0.0012	38	**0**
	20	**0.0002**	**2**	**0**	0.0003	5	**0**	0.0032	43	5	0.0019	52	**0**	0.0013	40	**0**
	30	**0.0003**	**9**	**0**	0.0004	17	**0**	0.0066	105	10	0.0019	39	**0**	0.0013	30	**0**
	100	**0.0016**	23	**0**	0.0017	**20**	**0**	0.0182	226	19	0.0101	134	8	0.0027	39	**0**
Ad Hoc 8	5	**0.0001**	**3**	**0**	**0.0001**	**3**	**0**	0.0007	27	**0**	0.0012	88	**0**	0.0009	69	**0**
	10	**0.0002**	**2**	**0**	**0.0002**	4	**0**	0.0013	26	**0**	0.0019	82	**0**	0.0014	70	**0**
	15	**0.0003**	**2**	**0**	**0.0003**	4	**0**	0.0020	25	1	0.0023	67	**0**	0.0016	53	**0**
	20	**0.0004**	**2**	**0**	**0.0004**	7	**0**	0.0038	42	2	0.0025	45	**0**	0.0018	43	**0**
	30	**0.0005**	**2**	**0**	0.0006	8	**0**	0.0046	43	3	0.0027	44	1	0.0020	40	**0**
	100	**0.0021**	**15**	**0**	0.0022	22	**0**	0.0109	102	4	0.0114	100	3	0.0032	32	**0**

Table 7.14: Continuation of Table 7.13 for the rest of the test collections.

C	n	Pool			BS^P			kNS^P			kLP^P			λTkLP^P		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Web 9	5	**0.0005**	**14**	**0**	**0.0005**	15	**0**	0.0014	46	**0**	0.0061	188	1	0.0055	168	1
	10	**0.0006**	**20**	**0**	**0.0006**	21	**0**	0.0033	59	**0**	0.0081	188	1	0.0074	182	1
	15	**0.0006**	**22**	**0**	0.0007	26	**0**	0.0048	87	**0**	0.0088	185	4	0.0079	176	4
	20	**0.0006**	**22**	**0**	0.0007	24	**0**	0.0074	102	**0**	0.0090	174	1	0.0081	165	1
	30	**0.0008**	**21**	**0**	**0.0008**	22	**0**	0.0106	126	3	0.0091	162	2	0.0081	179	2
	100	0.0027	36	**0**	**0.0016**	**20**	**0**	0.0387	297	17	0.0535	436	118	0.0135	113	**0**
Web 2001	5	**0.0002**	**6**	**0**	**0.0002**	**6**	**0**	0.0009	20	1	0.0036	120	**0**	0.0032	108	**0**
	10	**0.0003**	**9**	**0**	0.0004	14	**0**	0.0026	44	**0**	0.0052	131	**0**	0.0046	125	**0**
	15	**0.0004**	**6**	**0**	0.0005	10	**0**	0.0042	57	1	0.0057	115	**0**	0.0052	118	**0**
	20	**0.0005**	**11**	**0**	0.0006	15	**0**	0.0073	101	3	0.0056	88	1	0.0052	87	1
	30	**0.0006**	**6**	**0**	0.0007	10	**0**	0.0092	117	2	0.0052	59	1	0.0047	76	1
	100	0.0026	31	**0**	**0.0018**	**18**	**0**	0.0327	352	16	0.0251	290	13	0.0074	91	**0**
Web 2002	5	**0.0006**	32	**0**	**0.0006**	**31**	**0**	0.0036	163	**0**	0.0032	136	**0**	0.0029	124	1
	10	0.0013	46	**0**	**0.0012**	**39**	**0**	0.0073	279	20	0.0041	144	2	0.0033	107	1
	15	0.0018	56	**0**	**0.0017**	**51**	**0**	0.0127	394	42	0.0070	204	4	0.0033	106	1
	20	0.0022	50	**0**	**0.0019**	**40**	**0**	0.0222	577	112	0.0102	291	13	0.0034	89	**0**
	30	0.0032	66	**0**	**0.0024**	**47**	**0**	0.0333	678	165	0.0194	434	51	0.0033	77	**0**
	100	0.0054	71	**0**	**0.0035**	**48**	**0**	0.0502	705	192	0.0598	709	305	0.0166	235	26
Legal 2006	5	0.0237	253	3	**0.0184**	**219**	**2**	0.0383	357	80	0.0411	389	61	0.0203	236	4
	10	0.0514	274	10	0.0355	207	**0**	0.0821	421	115	0.1044	423	128	**0.0336**	**182**	**0**
	15	0.0502	259	8	0.0345	190	**0**	0.0734	413	86	0.1046	393	89	**0.0340**	**184**	**0**
	20	0.0494	226	1	0.0339	177	**0**	0.0666	343	50	0.1080	378	88	**0.0277**	**144**	**0**
	30	0.0483	203	3	0.0331	151	3	0.0606	260	25	0.1105	331	74	**0.0254**	**110**	**0**
	100	0.0448	153	7	0.0308	115	7	0.0515	178	14	0.1075	232	45	**0.0228**	**93**	**0**
Microblog 2011	5	**0.0003**	21	**0**	**0.0003**	**17**	**0**	0.0016	121	**0**	0.0005	47	**0**	0.0005	53	**0**
	10	**0.0006**	36	**0**	**0.0006**	**34**	**0**	0.0047	284	**0**	0.0007	39	**0**	0.0008	46	**0**
	15	**0.0011**	**62**	**0**	**0.0011**	69	**0**	0.0127	584	3	**0.0011**	**62**	**0**	0.0012	69	**0**
	20	**0.0015**	**56**	**0**	**0.0015**	62	**0**	0.0116	461	4	**0.0015**	**56**	**0**	0.0016	64	**0**
	30	0.0027	95	**0**	**0.0025**	**91**	**0**	0.0192	632	9	**0.0025**	97	**0**	**0.0025**	95	**0**
	100	0.0027	82	**0**	**0.0024**	**72**	**0**	0.0174	547	1	0.0028	80	**0**	0.0026	78	**0**
Medical 2011	5	0.0040	77	**0**	0.0027	39	**0**	0.0155	182	19	0.0059	70	2	**0.0022**	**47**	**0**
	10	0.0104	93	**0**	0.0062	65	**0**	0.0292	240	19	0.0149	123	12	**0.0049**	**44**	**0**
	15	0.0099	75	**0**	0.0059	37	**0**	0.0304	197	12	0.0185	109	16	**0.0048**	**31**	**0**
	20	0.0095	64	**0**	0.0056	42	**0**	0.0313	162	4	0.0226	127	20	**0.0054**	**32**	**0**
	30	0.0088	55	**0**	**0.0052**	**24**	**0**	0.0305	140	3	0.0280	140	18	0.0066	27	**0**
	100	0.0064	38	**0**	**0.0039**	**17**	**0**	0.0208	88	**0**	0.0347	134	12	0.0087	26	**0**
Genomics 2005	5	**0.0003**	**21**	**0**	**0.0003**	23	**0**	0.0012	55	**0**	0.0015	62	**0**	0.0014	60	**0**
	10	**0.0006**	**27**	**0**	0.0007	35	**0**	0.0028	101	**0**	0.0021	72	**0**	0.0018	78	**0**
	15	**0.0010**	26	**0**	**0.0010**	**23**	**0**	0.0053	116	**0**	0.0027	84	**0**	0.0021	80	**0**
	20	**0.0011**	**29**	**0**	0.0012	31	**0**	0.0076	128	4	0.0034	75	**0**	0.0023	77	**0**
	30	0.0016	35	**0**	**0.0015**	**33**	**0**	0.0108	180	3	0.0060	109	1	0.0022	59	**0**
	100	0.0047	64	**0**	**0.0028**	**48**	**0**	0.0233	278	6	0.0319	492	36	0.0079	146	**0**
Robust 2005	5	0.0008	10	**0**	0.0010	9	**0**	0.0031	20	**0**	**0.0006**	**8**	**0**	0.0007	14	**0**
	10	0.0018	8	**0**	0.0025	12	**0**	0.0078	14	1	**0.0013**	**6**	**0**	**0.0013**	9	**0**
	15	0.0028	16	**0**	0.0040	15	**0**	0.0126	23	6	0.0021	**10**	**0**	**0.0019**	12	**0**
	20	0.0037	12	**0**	0.0053	20	**0**	0.0152	18	5	**0.0028**	**9**	**0**	**0.0028**	13	**0**
	30	0.0058	16	**0**	0.0075	16	**0**	0.0212	20	5	0.0048	**5**	**0**	**0.0043**	10	**0**
	100	0.0150	20	1	0.0122	17	**0**	0.0233	16	1	0.0166	7	**0**	**0.0063**	**5**	**0**

Table 7.15: Summary of the results for R@n of the Reduced Pool and four estimators developed for P@n and used in combination with Eq. (7.15). These are generated through a leave-one organisation-out approach using the top 75% best pooled runs. The dotted lines represent the point when $n \leq K$ becomes false, where K is the depth of the Depth@K strategy used to build the test collection.

C	n	Pool			BS^P			$\not{k}NS^P$			$\not{k}LP^P$			$\lambda T\not{k}LP^P$		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Ad Hoc 2	5	0.0002	13	**1**	**0.0001**	11	**1**	0.0002	19	**1**	**0.0001**	12	**1**	**0.0001**	**7**	**1**
	10	0.0003	11	1	0.0003	11	1	0.0006	24	1	**0.0002**	9	1	**0.0002**	**7**	**0**
	15	0.0006	8	1	0.0006	11	1	0.0018	60	2	0.0005	8	**0**	**0.0003**	**7**	**0**
	20	0.0008	7	1	0.0009	10	1	0.0028	74	3	**0.0006**	7	**0**	0.0007	**4**	**0**
	30	0.0014	16	**0**	0.0014	18	**0**	0.0031	51	1	**0.0009**	**11**	**0**	0.0013	15	**0**
	100	0.0112	77	1	0.0063	40	**0**	0.0083	56	3	**0.0045**	**29**	**0**	0.0064	38	**0**
Ad Hoc 3	5	0.0003	3	**0**	**0.0002**	**2**	**0**	0.0006	4	**0**	0.0008	8	**0**	0.0005	6	**0**
	10	0.0006	3	**0**	**0.0004**	**2**	**0**	0.0008	5	**0**	0.0013	7	**0**	0.0009	4	**0**
	15	0.0007	5	**0**	**0.0005**	**4**	**0**	0.0010	6	**0**	0.0017	9	**0**	0.0012	9	**0**
	20	0.0008	**1**	**0**	**0.0006**	**1**	**0**	0.0011	6	**0**	0.0021	10	**0**	0.0015	6	**0**
	30	0.0009	2	**0**	**0.0007**	**0**	**0**	0.0017	7	**0**	0.0026	8	**0**	0.0019	6	**0**
	100	**0.0015**	**1**	**0**	0.0027	4	**0**	0.0072	14	**0**	0.0020	2	**0**	0.0028	5	**0**
Ad Hoc 4	5	**0.0003**	**7**	**0**	**0.0003**	**7**	**0**	0.0007	15	**0**	0.0039	77	1	0.0035	72	1
	10	**0.0005**	8	**0**	**0.0005**	**7**	**0**	0.0011	12	**0**	0.0046	76	**0**	0.0044	73	**0**
	15	**0.0007**	**5**	**0**	**0.0007**	**5**	**0**	0.0053	49	**0**	0.0049	43	**0**	0.0055	49	1
	20	**0.0009**	**7**	**0**	0.0010	9	**0**	0.0068	53	**0**	0.0047	36	**0**	0.0056	38	**0**
	30	**0.0015**	**6**	**0**	**0.0015**	8	**0**	0.0086	47	1	0.0040	26	**0**	0.0061	33	**0**
	100	0.0085	18	**0**	**0.0056**	**12**	**0**	0.0138	28	1	0.0182	39	1	0.0060	14	**0**
Ad Hoc 5	5	**0.0001**	5	**0**	**0.0001**	**3**	**0**	0.0012	37	**0**	0.0075	287	2	0.0070	277	2
	10	**0.0003**	7	**0**	0.0004	**6**	**0**	0.0020	44	**0**	0.0098	274	3	0.0097	260	2
	15	**0.0005**	**15**	**0**	**0.0005**	**15**	**0**	0.0022	48	**0**	0.0102	263	2	0.0103	247	2
	20	**0.0007**	13	**0**	**0.0007**	**10**	**0**	0.0023	30	**0**	0.0103	219	1	0.0112	235	3
	30	**0.0011**	**17**	**0**	**0.0011**	21	**0**	0.0040	81	**0**	0.0085	147	**0**	0.0108	180	**0**
	100	0.0053	63	**0**	**0.0041**	50	**0**	0.0216	242	1	0.0280	314	8	0.0044	**46**	**0**
Ad Hoc 6	5	**0.0004**	**1**	**0**	0.0006	4	**0**	0.0010	4	**0**	0.0075	42	**0**	0.0067	39	**0**
	10	**0.0004**	3	**0**	0.0005	**2**	**0**	0.0013	5	**0**	0.0096	38	**0**	0.0084	33	**0**
	15	**0.0008**	**1**	**0**	0.0009	**1**	**0**	0.0017	**1**	**0**	0.0101	28	**0**	0.0095	30	**0**
	20	**0.0008**	2	**0**	0.0010	**1**	**0**	0.0022	6	**0**	0.0101	28	**0**	0.0099	29	**0**
	30	**0.0011**	**6**	**0**	0.0016	9	**0**	0.0041	11	**0**	0.0090	20	**0**	0.0099	25	**0**
	100	0.0070	**11**	**0**	**0.0052**	12	**0**	0.0191	26	**0**	0.0314	46	4	0.0090	14	**0**
Ad Hoc 7	5	**0.0001**	8	**0**	**0.0001**	**7**	**0**	0.0009	64	**0**	0.0014	104	**0**	0.0010	60	**0**
	10	**0.0002**	**4**	**0**	**0.0002**	**4**	**0**	0.0022	72	**0**	0.0019	70	**0**	0.0014	55	**0**
	15	**0.0003**	**8**	**0**	**0.0003**	9	**0**	0.0024	62	**0**	0.0023	65	**0**	0.0018	47	**0**
	20	**0.0003**	**7**	**0**	0.0004	11	**0**	0.0022	53	**0**	0.0024	60	**0**	0.0020	46	**0**
	30	**0.0004**	**13**	**0**	0.0006	18	**0**	0.0029	72	**0**	0.0022	42	**0**	0.0020	41	**0**
	100	0.0023	25	**0**	**0.0021**	**24**	**0**	0.0128	184	4	0.0090	134	1	0.0032	51	**0**
Ad Hoc 8	5	**0.0002**	**5**	**0**	**0.0002**	**5**	**0**	0.0005	21	**0**	0.0017	108	**0**	0.0013	84	**0**
	10	**0.0003**	**3**	**0**	**0.0003**	**3**	**0**	0.0007	19	**0**	0.0025	100	**0**	0.0020	86	**0**
	15	**0.0003**	**2**	**0**	0.0004	**2**	**0**	0.0008	7	**0**	0.0030	93	**0**	0.0024	85	**0**
	20	**0.0005**	**3**	**0**	**0.0005**	4	**0**	0.0012	19	**0**	0.0031	70	**0**	0.0026	58	**0**
	30	**0.0007**	**2**	**0**	0.0008	8	**0**	0.0027	47	**0**	0.0030	51	**0**	0.0027	52	**0**
	100	**0.0027**	**12**	**0**	0.0032	20	**0**	0.0071	97	**0**	0.0073	76	**0**	**0.0027**	20	**0**

Table 7.16: Continuation of Table 7.15 for the rest of the test collections.

C	n	Pool			BS^P			kNS^P			kLP^P			λTkLP^P		
		MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*	MAE	SRE	SRE*
Web 9	5	**0.0006**	**12**	**0**	0.0007	**12**	**0**	0.0016	36	**0**	0.0078	191	2	0.0068	175	1
	10	**0.0007**	**18**	**0**	0.0008	20	**0**	0.0023	48	**0**	0.0105	225	3	0.0095	189	1
	15	**0.0007**	**18**	**0**	0.0009	21	**0**	0.0024	45	**0**	0.0108	201	5	0.0098	190	4
	20	**0.0008**	**25**	**0**	0.0010	28	**0**	0.0037	75	**0**	0.0106	178	1	0.0104	169	1
	30	**0.0009**	**21**	**0**	0.0010	**21**	**0**	0.0058	106	2	0.0089	158	2	0.0101	175	2
	100	0.0031	38	**0**	**0.0017**	**19**	**0**	0.0302	245	2	0.0411	329	31	0.0140	126	**0**
Web 2001	5	**0.0002**	6	**0**	**0.0002**	**5**	**0**	0.0006	16	**0**	0.0060	161	**0**	0.0057	149	**0**
	10	**0.0003**	**6**	**0**	0.0004	8	**0**	0.0018	38	**0**	0.0083	169	**0**	0.0078	156	**0**
	15	**0.0004**	**6**	**0**	0.0005	9	**0**	0.0024	36	**0**	0.0088	131	**0**	0.0084	124	**0**
	20	**0.0005**	**5**	**0**	0.0007	11	**0**	0.0041	65	1	0.0080	106	1	0.0082	104	1
	30	**0.0006**	**8**	**0**	0.0008	12	**0**	0.0059	114	1	0.0069	80	1	0.0074	95	1
	100	0.0030	28	**0**	**0.0021**	**24**	**0**	0.0237	231	**0**	0.0234	238	8	0.0090	88	**0**
Web 2002	5	**0.0007**	24	**0**	**0.0007**	**22**	**0**	0.0030	76	**0**	0.0053	138	**0**	0.0057	133	**0**
	10	0.0013	28	**0**	**0.0012**	**25**	**0**	0.0062	142	**0**	0.0055	108	**0**	0.0059	118	**0**
	15	0.0017	39	**0**	**0.0016**	**35**	**0**	0.0139	217	4	0.0072	139	**0**	0.0061	117	**0**
	20	0.0022	36	**0**	**0.0019**	**27**	**0**	0.0227	314	20	0.0094	153	**0**	0.0058	94	**0**
	30	0.0032	42	1	**0.0024**	**38**	1	0.0312	351	30	0.0187	214	17	0.0048	72	**0**
	100	0.0072	63	**0**	**0.0047**	**46**	**0**	0.0426	365	31	0.0657	501	204	0.0183	154	13
Legal 2006	5	0.0314	225	1	**0.0233**	**185**	1	0.0322	191	3	0.0450	229	4	0.0266	202	3
	10	0.0688	219	10	0.0456	157	2	0.0640	215	15	0.1096	228	26	**0.0453**	**154**	2
	15	0.0673	198	3	0.0463	145	**0**	0.0571	202	16	0.1188	226	25	**0.0402**	**135**	**0**
	20	0.0634	178	1	0.0422	133	**0**	0.0500	172	7	0.1273	228	28	**0.0335**	**106**	**0**
	30	0.0620	158	3	0.0410	113	3	0.0503	140	3	0.1347	206	22	**0.0287**	**66**	**0**
	100	0.0563	121	7	0.0373	101	7	0.0473	128	7	0.1300	155	9	**0.0274**	**61**	**1**
Microblog 2011	5	**0.0004**	26	**0**	**0.0004**	**25**	**0**	0.0022	129	**0**	0.0007	52	**0**	0.0007	59	**0**
	10	**0.0008**	**57**	**0**	**0.0008**	58	**0**	0.0038	252	**0**	0.0010	75	**0**	0.0011	80	**0**
	15	**0.0014**	82	**0**	0.0015	**76**	**0**	0.0069	378	**0**	0.0015	83	**0**	0.0016	92	**0**
	20	0.0021	68	**0**	0.0021	75	**0**	0.0084	408	**0**	**0.0020**	**66**	**0**	0.0022	74	**0**
	30	0.0037	113	**0**	**0.0035**	**107**	**0**	0.0168	560	1	**0.0035**	110	**0**	**0.0035**	108	**0**
	100	0.0034	77	**0**	**0.0032**	**73**	**0**	0.0168	432	1	0.0033	75	**0**	0.0033	74	**0**
Medical 2011	5	0.0054	79	**0**	0.0032	48	**0**	0.0112	153	**0**	0.0053	63	**0**	**0.0024**	**48**	**0**
	10	0.0141	100	**0**	0.0074	64	**0**	0.0248	218	**0**	0.0130	124	**0**	**0.0059**	**47**	**0**
	15	0.0133	76	**0**	0.0069	42	**0**	0.0269	172	**0**	0.0163	109	**0**	**0.0051**	**31**	**0**
	20	0.0129	75	**0**	0.0068	43	**0**	0.0290	151	**0**	0.0204	115	3	**0.0055**	**37**	**0**
	30	0.0118	60	**0**	**0.0062**	31	**0**	0.0275	128	**0**	0.0245	119	**0**	0.0064	**27**	**0**
	100	0.0077	41	**0**	**0.0043**	**18**	**0**	0.0199	92	**0**	0.0282	152	**0**	0.0083	30	**0**
Genomics 2005	5	**0.0005**	**30**	**0**	0.0006	34	**0**	0.0016	74	**0**	0.0022	83	**0**	0.0022	79	**0**
	10	**0.0009**	**41**	**0**	0.0012	44	**0**	0.0025	79	**0**	0.0025	85	**0**	0.0027	91	**0**
	15	**0.0011**	**31**	**0**	0.0012	32	**0**	0.0020	48	**0**	0.0025	67	**0**	0.0031	76	**0**
	20	**0.0013**	**24**	**0**	0.0014	31	**0**	0.0029	70	**0**	0.0032	72	**0**	0.0033	75	**0**
	30	**0.0019**	**36**	**0**	0.0020	39	**0**	0.0070	151	**0**	0.0048	97	**0**	0.0029	55	**0**
	100	0.0061	87	**0**	**0.0039**	**58**	**0**	0.0212	302	2	0.0372	478	17	0.0101	164	**0**
Robust 2005	5	0.0011	8	**0**	0.0013	10	**0**	0.0015	14	**0**	**0.0004**	**6**	**0**	0.0008	9	**0**
	10	0.0023	7	**0**	0.0032	12	**0**	0.0035	10	**0**	**0.0011**	**4**	**0**	0.0017	11	**0**
	15	0.0036	11	**0**	0.0051	17	**0**	0.0056	16	**0**	**0.0016**	**7**	**0**	0.0024	10	**0**
	20	0.0045	11	**0**	0.0069	21	**0**	0.0068	15	**0**	**0.0023**	**5**	**0**	0.0035	12	**0**
	30	0.0070	17	**0**	0.0096	20	**0**	0.0110	19	**0**	**0.0038**	**4**	**0**	0.0055	9	**0**
	100	0.0185	22	1	0.0147	19	**0**	0.0152	17	**0**	0.0154	13	**0**	**0.0066**	**8**	**0**

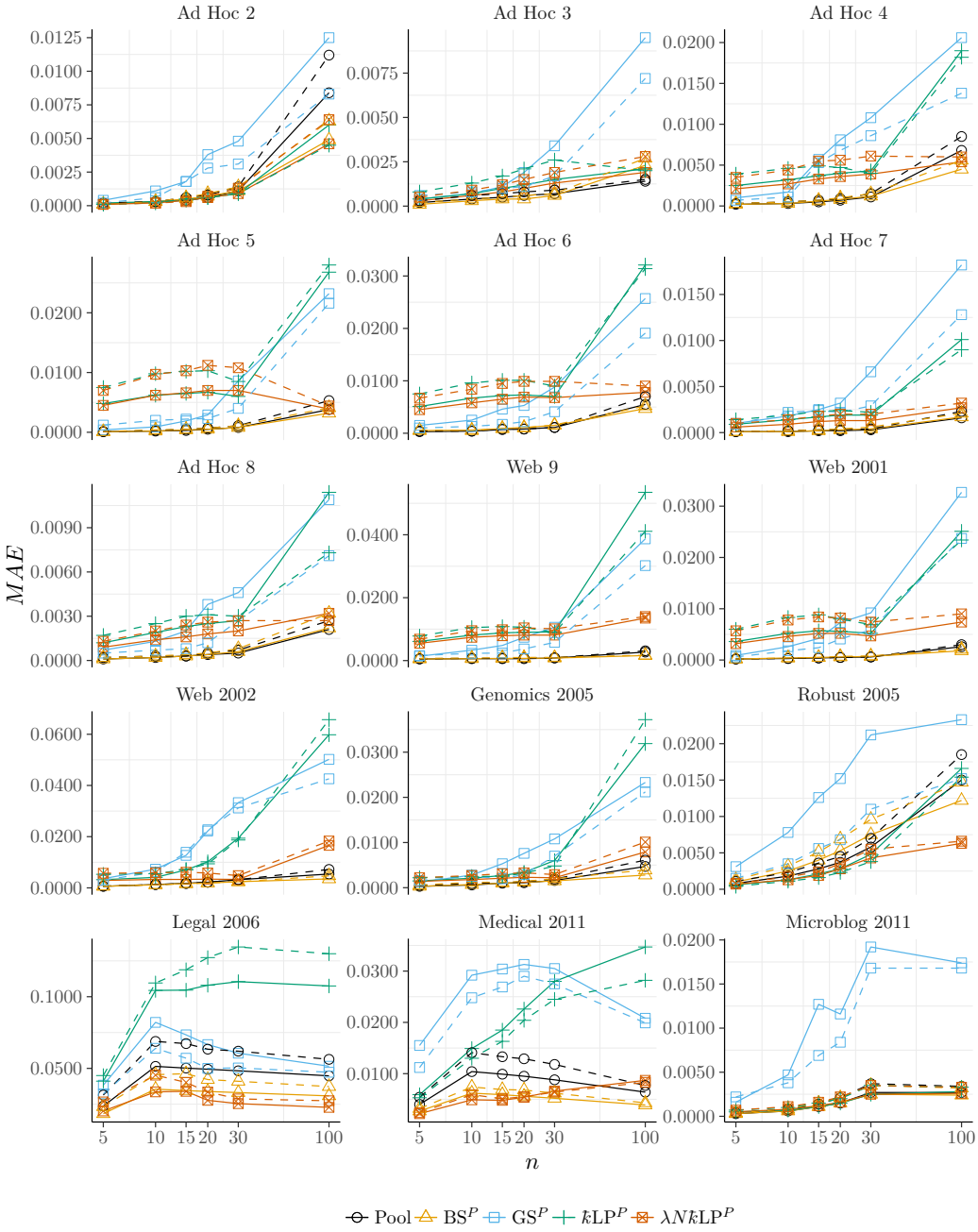

Figure 7.8: Plots per test collection of the Mean Absolute Error against the R@n of the Reduced Pool and the four presented approaches to correct pool bias for P@n. Generated using a leave-one organisation-out, using all the pooled runs for the continuous lines and only the top 75% best performing pooled runs for the dashed lines.

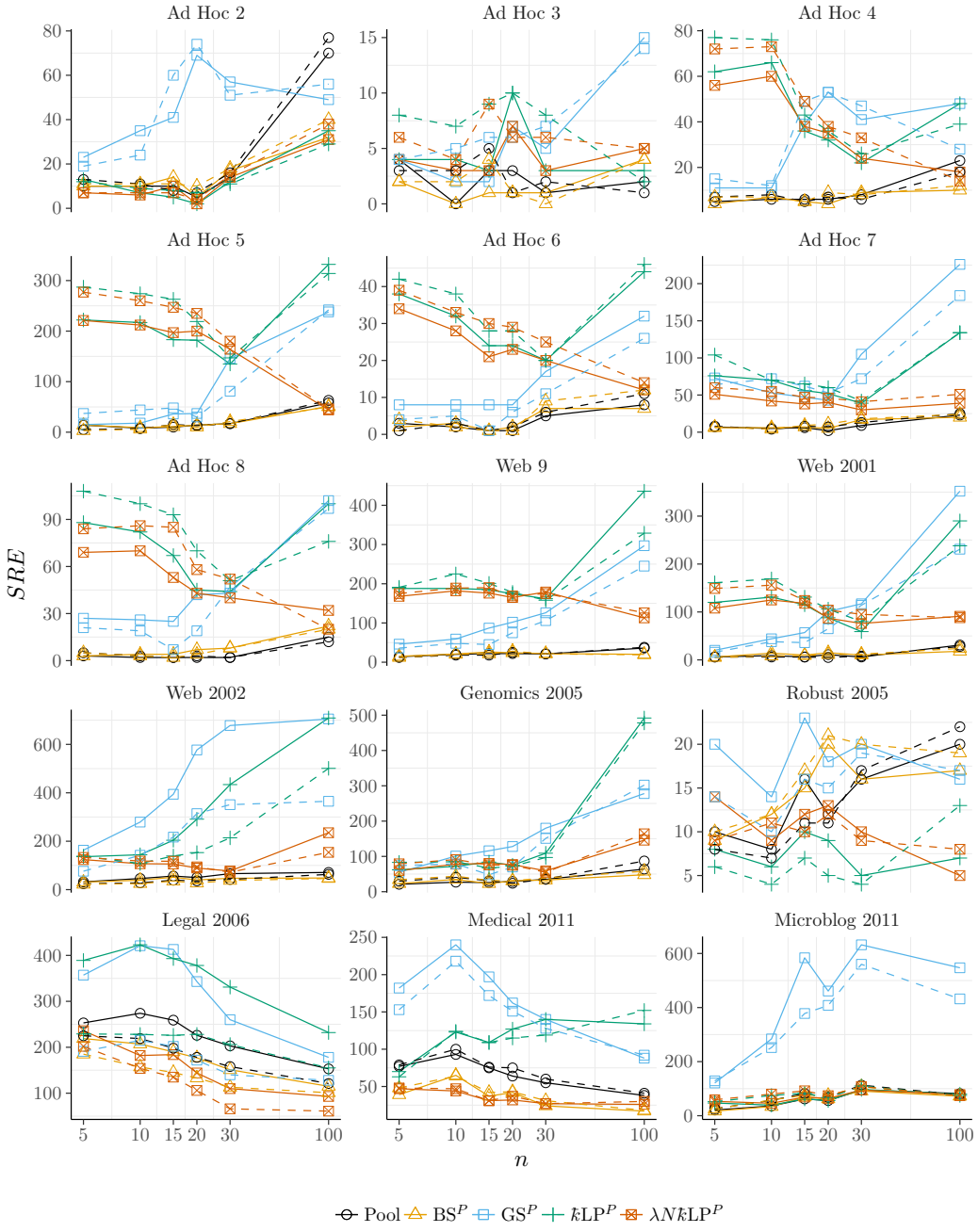

Figure 7.9: Plots per test collection of the System Rank Error against the R@n of the Reduced Pool and the four presented approaches to correct pool bias for P@n. Generated using a leave-one organisation-out, using all the pooled runs for the continuous lines and only the top 75% best performing pooled runs for the dashed lines.

we claim that it happens for a combination of both hypotheses. The first hypothesis relies on the observation that for such test collections, the ratio between the number of pooled runs and the number of organisations is much greater than 1. This means that multiple runs from the same organisation have been pooled and therefore contribute a very similar set of documents to the pool. Thereby, it nullifies the leave-one run-out approach embedded in these simulation-based estimators, as shown in Eq. (7.18) and (7.19), looking inside the expectations, when the run r', originally in the pool is removed from the pool. This leave-one run-out happens inside the estimator and is different from the leave-one organisation-out testing procedure that happens before the estimator is run. However, for Legal 2006 and Medical 2011, although the ratio is also big, we do not observe the same affect due the more shallow pool depths. The second hypothesis is that when we want to count the top number of relevant documents of a run and we have a large number of relevance judgements, there is a high likelihood that the top documents of every run have been already pooled by other runs. This means that, as for the first hypothesis, the effect of the leave-one run-out embedded in the method is nullified. We split these two hypotheses because they have a different nature, although they have the same effect that can be mitigated by the same solution.

In general, these two hypotheses cause a significant error due to the fact that the number of points collected in order to compute a meaningful estimation of the expectations $|\{r' \in \mathcal{R}_p : P@n(r', J_{\mathcal{R}_p}) - P@n(r', J_{\mathcal{R}_p \setminus \{r'\}}) \neq 0 \wedge \hbar@n(r', J_{\mathcal{R}_p}) \neq 0\}|$ for $\hbar$NS and, $|\{r' \in \mathcal{R}_p : P@n(r', J_{\mathcal{R}_p}) - P@n(r', J_{\mathcal{R}_p \setminus \{r'\}}) \neq 0\}|$ for BS, is insufficient. When these sets are small, it means that either there are no unjudged documents, or the ones that exist bring no new information in terms of being or not being in the pool. Thereby, the error introduced is bigger than the one we would have observed if we had not corrected the run at all. In fact we can observe that the second best result, excluding the simulation-based approaches, is obtained by the reduced pool. Therefore to mitigate this effect, it is necessary to introduce a trigger that checks if the number of data points collected are sufficient, to compute the estimate and perform the correction, and if not, to fall back to the reduced pool error.

We now move to the analysis of the two perturbation-based estimators, $\hbar$LP and λT$\hbar$LP. $\hbar$LP is on average better than λT$\hbar$LP when applied to the pool built using the 75% of best performing runs. However, this estimator performance is worse than the reduced pool for Ad Hoc 7, and 8, Web 9, Web 2001, Web 2002, while λT$\hbar$LP performs at least better than the reduced pool. We observe that λT$\hbar$LP outperforms the reduced pool and the simulation-based estimators. This happens in the majority of the cases when using all the pooled runs, less often when using only the top 75% of best performing pooled runs. Moreover λT$\hbar$LP is shown to be stable.

In summary, the best simulation-based estimator is $\hbar$NS, and perturbation-based estimator λT$\hbar$LP. We have also seen that the perturbation-based estimators are more stable than the simulation-based ones, therefore these should be preferred. This because they rarely get worse than the reduced pool, although sometimes they are not the absolute best.

7.5.2 Bias Estimators for Recall at Cut-off

Here we present the bias estimators for R@n. This section is divided into two subsections. In the first subsection, we discuss the performance of the estimators designed for R@n. In the second subsection, we discuss the performance of the P@n estimators when used to estimate R@n, as shown in Eq. (7.13). This is possible thanks to a property of the analysed test collections (built using a Depth@K pooling strategy) and to a mathematical property of P@n.

R@n Estimators

We first juxtapose the results obtained by first including all the pooled runs, and then only the top 75% best performing runs. Here, we observe that on average there is no difference in quality among the estimators, as previously seen for P@n. On average, the best performing estimators are kNS and BS. However, between these kNS is much more effective than BS when only the 75% of pooled runs is used.

The relative good quality of the Ad Hoc test collections produces pools that are difficult to beat for R@n. This happens for Ad Hoc 2 to 8, but also for Web 9 and 2001, and Robust 2005. For these test collections all the estimators perform similarly to the reduced pool. However, for the rest of the test collections we can observe a clear distinction in performance among the estimators. We hypothesise that this is linked to the large pool depths ($\gg 50$) of the Depth@K used to build the abovementioned test collections. We observe that for Web 2002, Genomics 2005, Legal 2006, Medical 2011, and Microblog 2011, kNS is the best performing estimator, while BS and GS are still better than the pool but not as much as kNS.

R@n Derived Estimators from P@n Estimators

We start by observing that the best estimator is BS^P. Juxtaposing the two results, the first one obtained using all the pooled runs, and the second one using only the top 75% best performing runs, there is no important difference among the estimators. Although, on average, it is clear that the BS^P is less affected by this additional absence of relevant information.

On a per test collection basis, we observe that for Ad Hoc 2, Legal 2006, and Medical 2011, the $\lambda\text{T}k\text{LP}^P$ estimator is the best estimator, and for Robust 2005 it performs better than the reduced pool. We hypothesise that the good performance observed for Ad Hoc 2 is related to the over-estimation of the P@100 case when predicting P@n, which makes this estimator produce smaller bias corrections, therefore being a more conservative estimator. This is also confirmed by the fact that, for low recall cut-off values, this estimator is as good as the reduced pool. In the case of Legal 2006 and Medical 2011, we cannot say much due to the shallow pool depths (10) of the Depth@K strategy used to build these test collections. However, it is surprising that the $\lambda\text{T}k\text{LP}^P$ estimator performs better than the other estimators.

The BS^P estimator is the best performing estimator for Ad Hoc 3, Web 2002. Moreover, we also observe that this estimator is better than the reduced pool for Legal 2006 and Medical 2011, and it demonstrates a stable behaviour even when the reduced pool is not among the worst estimations, like for Ad Hoc 4, 5, 6, 7 and 8, Web 9 and 2001, Genomics 2005 and Microblog 2011. This makes this estimator the best of this category. However, it fails for Robust 2005. We hypothesise that the reason for this failure is due to the low number of pooled runs and shallow pool depth of this test collection, which does not allow this estimator to have enough samples to generate good estimates.

Summary

In the previous subsections, we have discussed the performances of the estimators originally designed for R@n, and derived from the P@n estimators. As already pointed out, while the former estimators are independent of the pooling strategy used to build the test collection, the latter need to be built using a Depth@K pooling strategy (see Eq. (7.13)). Juxtaposing the two classes of estimators we observe that the best estimator belongs to the former class, $\hbar$NS.

7.6 Summary

The primary focus of this chapter is an insight that information about the quality of an unpooled run can be obtained by analysing how the pool has been built. We have presented here a large array of bias estimators for P@n and R@n. We started presenting how the pool bias manifests in these IR evaluation measures and how to estimate it. We then continued formalizing the presented estimators in two big categories: simulation-based estimators and the newly introduced perturbation-based estimators. While the first infer the performance by simulating the absence of other runs, the latter extracts this information by measuring the effect of the run on existing, pooled runs.

In this chapter we have improved over the baseline, the reduced pool, for both IR evaluation measures, P@n and R@n. We have observed that for P@n, the best estimator is the perturbation-based estimator, λT$\hbar$LP; and for R@n the best estimator is the simulation-based estimator $\hbar$NS. However, for the latter case the results appear less clear with respect to the former case. Moreover, we have observed that estimating recall using precision based estimators provided promising results.

This chapter addresses a significant concern coming from research but also from practice: the necessity to have valid, yet understandable measures, which we can communicate to partners outside of our community. This last condition significantly restricts our possible choices. P@n and R@n are by far the most easily understood quantities to communicate and with this study we have shown that we can correct pool bias when considering a run that has not participated in the creation of the pool.

CHAPTER 8

Conclusion

In this thesis we explored some of the biases observed in IR, and learnt how to model, quantify, and exploit model biases to improve retrieval effectiveness; how to model and analytically quantify a particular model bias, the retrievability bias, for accessibility evaluation; and how to model, quantify, and mitigate a selection bias, the pool bias, for a more reliable test collection-based evaluation.

The exploitation of the analysed model biases has led, on the one hand to the improvement of retrieval effectiveness. In Chapter 4, we improved retrieval effectiveness by quantifying the model biases observed on the document verboseness and length, and embedded these factors into several probabilistic IR models. On the other hand, it has led to the development of a new theoretical perspective on the accessibility evaluation. In Chapter 5, we analytically quantified the retrievability bias on Boolean models, making this quantification much faster than it would have been with the standard empiric methods.

The mitigation of the selection bias, the pool bias, has led to the building of less biased test collections and less biased evaluations of new IR systems on existing test collections. In Chapters 6 and 7, we mitigated the pool bias, which manifests when building test collections using the pooling method. In particular, in Chapter 6 we developed new pooling strategies and then identified the least biased one. In Chapter 7 we developed bias estimators to correct IR evaluation measures when evaluating new systems on existing test collections.

The remainder of this chapter goes as follows. We first comment on the interaction of model and selection biases in IR. Next, we discuss the effort made to unify the mathematical framework across the IR topics treated in this thesis. Then, before concluding, we summarise the main findings and limitations of the chapters mentioned above.

8.1 Model and Selection Biases Interaction

We now make the interaction of the two studied types of bias explicit, *i.e.*, model bias and selection bias in the test collection-based evaluation of IR systems. In particular we observe (a) instances of model bias interacting in test collection-based evaluation, which we have observed to suffer from pool bias, a selection bias; and (b) instances of selection bias interacting when evaluating IR models, which we have observed to suffer from several model biases.

In this thesis we have seen in multiple occasions that a big advantage of the test collection-based evaluation is the possibility to test newly developed IR systems without the need to involve users into the evaluation loop. However, as we have seen, this evaluation suffers from pool bias due to the way these test collections have been built, using the pooling method. To understand where the pool bias (a selection bias) interacts with model biases, we need to recall on what the pooling method relies, the output of a set of retrieval systems. These systems, as we have seen, can potentially suffer from several model biases. In other words, the pool bias observed when evaluating these newly developed IR systems on an existing test collection is actually affected in turn by the model biases of the set of the IR systems used to build this test collection. In this way, causing a model-selection bias interaction. On the contrary, if this newly developed IR model is affected by model biases, its test collection-based evalution would be an instance of selection-model bias interaction, since this evaluation would suffer from pool bias (a selection bias). Considering now both interactions, if we imagine that these newly developed IR systems were used to build a new test collection using the pooling method, we could in order concatenate these interactions forming a selection-model-selection bias interaction, or going backward, a model-selection-model bias interaction. Moreover, by reiterating the same reasoning we could generate longer bias interaction chains.

The quantification of the individual contributions of these two bias interactions is an interesting research question that deserves to be explored. In fact, these long chains of bias interaction seem to suggest that if the IR experimentation had been left in isolation – *i.e.*, no external factor participating into the building of test collections (*e.g.* introducing manual runs) and the development of IR models (*e.g.* embedding into them insights developed through users' studies and language analysis) – this would have guided this experimentation to reward more a specific set of pooling results and IR models. Nonetheless, some instances of this effect have been observed in the literature: no discernible upward trend has been observed in IR models in Ad Hoc tasks in IR papers dating between 1998 and 2008 [Arm+09], and pooling results have been observed to be biased towards longer documents [LAB08].

A trivial and unrealistic solution to analyse these interactions would be to fully judge a collection of documents. This would, in fact, eliminate the selection bias caused by the pooling method, and since no pooling method is involved, the selection-model bias interaction mentioned above also disappears. However, even in this epic effort, other selection biases, which we have neglected so far, could play an important role. For

example, the topic selection bias and assessor selection bias, which are due to how topics and assessors are selected. A more realistic solution could instead come from an hybrid evaluation approach, *i.e.*, an evaluation at the intersection between the on-line and the test collection-based evaluation. If we were to imagine a test collection not as a static entity but rather as a dynamic one – a test collection that changes overtime by requiring document judging when needed – we would, like for the unrealistic solution presented above, eliminate the selection bias and therefore also its selection-model bias interaction. Moreover, if this test collection were to be on-line, *e.g.* usable by users, this dynamic test collection would be free of topic selection bias and assessor selection bias. The former because the topics would be provided by the users. The latter because the assessors would be the actual users of the test collection. However, there are many other problems with such an hybrid evaluation, most notably the use of indirected evidence for relevance assessments.

8.2 Model Bias: Term Frequency Normalisation

In Chapter 4, we empirically demonstrated that normalisations based on the document verboseness together with the document length provide higher retrieval quality than the standard normalisations based only on the document length. Through an exhaustive study of normalisation factors in several IR probabilistic models: several TF-IDF variants, BM25 and D-LM, we made the case that different domains, having different text statistics, can be directly factored into these existing IR models. This is done by embedding various *document and term statistics* into one factor that balances multiple prior probabilities that all these IR models, more or less explicitly, rely on.

We here studied a model bias that was causing retrieval models to retrieve more repetitive documents. Moreover, the model bias observed in these retrieval models shown to not only be language specific but also domain specific. The different use of modern English in these studied domains *i.e.*, News, Web, Legal, and Medical, affects retrieval effectiveness. However, we saw that these domain differences can be compensated by statistics that at first glance can come across as simple, but which are also very effective as demonstrated by our experiments.

8.3 Model Bias: Retrievability

In Chapter 5, we demonstrated that the retrievability of Boolean models can be computed analytically. Here, we learnt that this quantifies the a-priori probability of a document to be retrieved, and as expected, we analytically observed that documents that contain more terms are more likely to be retrieved. Moreover, we proved that the retrievability of a Boolean model is the upper-bound of any best-match model.

The quantification of the retrievability bias with the Gini coefficient has led to the discovery that: disjunctive queries are less biased than conjunctive queries as well as when increasing the term-size of the queries. In particular, when increasing the term-size

of the queries, for disjunctive queries, we observe a marginal decrement in bias, while for conjunctive queries, we observe a rapid increase in bias. This behaviour has been demonstrated to be the same also when changing the shape of the distribution of the document-term size.

However, this analytical framework presents a major limitation. When developing the formulae, no realistic assumptions about the likelihood of a query to be submitted to the IR system are made. It has been, in fact, assumed that all the queries have equal likelihood to be submitted. This assumption does not respect the behaviour of actual users. For example, it is well-known that users of the Web search by keywords, which define a subset of the queries used in our analysis. However, even if this assumption is not realistic, this is commonly made when computing retrievability empirically by running large experiments, making this analytical approach not dissimilar to the empirical one. Nonetheless, we believe that this analytical perspective on the retrievability bias should be extended in order to include more sophisticated assumptions.

8.4 Selection Bias: Pooling Method

In Chapter 6, we saw that the standard pooling method can be improved upon. We did this by evaluating 22 pooling strategies on a large scale experiment including 9 test collections sampled from multiple domains: News, Web, Genomics, Legal, Blog and Microblog. We evaluated the selection bias by using three bias measures: MAE, SRE and SRE* and observed it on three IR evaluation measures: AP, NDCG, and P@10. Every strategy was evaluated at different numbers of judged documents N.

Under these empirical restrictions, we discovered that the best pooling strategy is MABMaxMeanTake@N based on a multi-armed bandit approach. This strategy resulted to be the best performing strategy over all the tested measures of bias and IR measures. Moreover, it resulted also to be the strategy discovering the largest number of relevant documents. MABMaxMeanTake@N consists in pooling the first document from a run selected based on the judgements observed on the previous selections. Thus, it requires the judgement of a document at every selection step, making it less operationable. Furthermore, the need to make the assessors judge documents in order of relevance can potentially introduce additional cognitive biases into the evaluation process. Hence, if the lower operationability and the potential cognitive biases introduced by this strategy are not a good trade-off for a less biased test collection, CombMAXTake@N is an alternative strategy that does not have these issues. This, in fact, pools the documents in order of a ranking formed based on the maximum normalised score assigned by the runs across the runs. However, this is a less effective strategy than MABMaxMeanTake@N, but is still better than the standard pooling strategy.

8.5 Selection Bias: Evaluation measures

In Chapter 7 we demonstrated that it is possible to mitigate the pool bias of existing test collections for two IR evaluation measures, P@n and R@n. To do this we developed several pool bias estimators that output a value to be added to the biased scores of non pooled runs. These estimators take as input: (1) the run to be corrected and (2) information consumed and generated by the pooling method, which is the set of pooled runs and relevance assessments. Based on the formalisation of the estimators, we classify these estimators into two categories: simulation-based estimators and perturbation-based estimators. The simulation-based ones estimate by simulating the absence of pooled runs from the pool. These estimators exploit how the scores of pooled runs change when not pooled. The idea behind simulation-based estimators is to quantify how much a pooled run would have been biased if it had not been pooled. The perturbation-based ones estimate by perturbing the pooled runs with the unpooled run. These estimators exploit how the scores of pooled runs change when perturbed by the unpooled run. The idea behind perturbation-based estimators is to quantify how important is the information contained in the new run by measuring how much it would have contributed if their document preference had been merged together with pooled runs.

We evaluated these estimators by running a large scale experiment including 15 test collections selected from multiple domains: News, Web, Genomics, Legal, Medical, and Microblog. We assessed the bias via three measures of bias, MAE, SRE, and SRE*. Under these experimental constraints, we discovered that for P@n, the best estimator is among the perturbation-based estimators. Named as λTkLP, this estimator consists in measuring the variation in precision and anti-precision of the score of the pooled runs when perturbed by the unpooled run, where anti-precision is the proportion of irrelevant documents among the retrieved ones. The averages of these quantities across the pooled runs are used: (1) calculating the correction, and (2) an indicator function used to trigger, if required, the application of this correction. For R@n , the best estimator is kNS. This simulation-based estimator consists in computing the average difference between the R@n of the run when pooled and not pooled, then normalises these values by the margin of improvement each of these runs has if every unjudged document had been judged as relevant.

The classification provided about the estimators suggests a naturally third category of estimators yet unexplored, the hybrid class: *perturbation simulation-based estimators*. Combining the ideas behind the two classes of estimators could potentially provide better performing estimators.

8.6 Future Work

The research presented in this thesis opens different opportunities, which have been left for future work. These are here listed per topic:

Model Bias: Term Frequency Normalisation. A theoretical analysis of term frequency normalisation components based on more complex statistics measured with word similarities techniques, like: word embeddings via neural networks and dimensionality reduction techniques on the word co-occurrence matrix;

Model Bias: Retrievability. A further development of the retrievability theory, which is still on its infancy;

Selection Bias: Pooling Method and Evaluation Measures. The exploration of pooling strategies and evaluation measures that work in synergy to mitigate the pool bias.

8.7 Final Remarks

The analysis of biases in IR has led to new insights about the performance of IR systems. We have found that IR models can be improved by embedding a document verboseness component together with an (already present in most IR models) document length component. We have also seen that document length affects the accessibility of collection of documents, and potentially this is also true for the document verboseness, since both can be interpreted as prior probabilities of a document to be retrieved. Also, when evaluating IR systems on a test collection we should be aware of its pool bias, which means that a more accurate analysis of the results is necessary, *e.g.* by using pool bias estimators, if existing, to correct the IR measure used to evaluate the system, or by building less biased test collections using less biased pooling strategies.

With this thesis, we have shown that verboseness plays a role in IR models, that retrievability can be computed analytically, and that the pool bias can be mitigated at test collection build time, and also for existing test collections at evaluation time. These findings are important for different members of the IR community. Practitioners will benefit from this work because, as pointed out in the introduction, they are interested in having a reliable IR evaluation that is easy to interpret and translate into questions that have a direct impact on their non-IR colleagues, *e.g.* how many relevant documents are they reading? (precision) How many relevant documents are they missing? (recall) Is the system making the documents accessible? (retrievability). The empirical IR researchers will benefit from less biased IR evaluation measures, and in particular, among them, the test collection builders will benefit from less biased pooling strategies. Finally, the more theoretical IR researchers will benefit from a theoretical perspective on all the topics treated in this thesis, but in particular on the initial theoretical exploration of retrievability.

APPENDIX A

Selection Bias: Pooling Method

A.1 Borda vs. Condorcet

In this section we analyse the relationships between the Borda and Condorcet counting method employed in this paper. Here, we prove that a relaxation of the Condorcet condition in the Copeland's method leads to the Borda method. We start substituting to the Copeland's method function, as defined in Eq. 6.8, the definition of the Condorcet criteria in Eq. 6.7 obtaining:

$$s(d, \mathcal{R}_p) = \sum_{d' \in \mathcal{D}} \begin{cases} 1 & \sum_{r \in \mathcal{R}_p} \text{sign}(\rho(d', r) - \rho(d, r)) > 0 \\ 0 & \text{otherwise} \end{cases} + \epsilon \cdot \mu(0, 1)$$

This equation is characterised by a condition that is calculated over all the pooled runs. We now relax this condition counting the individual run contributions by inverting the two summations as follows:

$$\sum_{d' \in \mathcal{D}} \begin{cases} 1 & \sum_{r \in \mathcal{R}_p} \text{sign}(\rho(d', r) - \rho(d, r)) > 0 \\ 0 & \text{otherwise} \end{cases} \approx$$

$$\approx \sum_{r \in \mathcal{R}_p} \sum_{d' \in \mathcal{D}} \begin{cases} 1 & \text{sign}(\rho(d', r) - \rho(d, r)) > 0 \\ 0 & \text{otherwise} \end{cases}$$

It is now possible to simplify the relaxation by observing that the condition is true for every document d' below document d and therefore can be written as a function of the size of r and the rank of d as follows:

$$\sum_{r \in \mathcal{R}_p} \sum_{d' \in \mathcal{D}} \begin{cases} 1 & \text{sign}(\rho(d', r) - \rho(d, r)) > 0 \\ 0 & \text{otherwise} \end{cases} =$$

$$= \sum_{r \in \mathcal{R}_p} (|\mathcal{D}| - \rho(d, r)) = |\mathcal{R}_p|\,|\mathcal{D}| - \sum_{r \in \mathcal{R}_p} \rho(d, r)$$

The outcome of such a relaxation is a Borda counting shifted by a constant $C = |\mathcal{R}_p||\mathcal{D}|$. Since C is a constant, it does not affect the order in which the documents are selected. Indeed, by rank equivalence:

$$C + \sum_{r \in \mathcal{R}_p} -\rho(d, r) \simeq \sum_{r \in \mathcal{R}_p} -\rho(d, r)$$

With the above, we have demonstrated that relaxing the Condorcet condition, the Condorcet Copeland's counting is equivalent to Borda counting. However this is not exactly the Borda counting implemented in the paper because, while in this strategy when d is not retrieved by r, $\rho(d, r)$ returns $-|\mathcal{D}|$ (see the definition of ρ in Section 3.1), in our presented version when d is not retrieved by r, the function B returns $-(|\mathcal{D}| + |\mathcal{D}_r| + 1)/2$.

A.2 Hedge Strategy's Behaviour at its Extremes

In this section we show that the behaviour of the Hedge strategy when setting β to the extremes 0, 1, and $+\infty$ can be assimilated to: multi-armed bandit-based pooling strategies when $\beta = 0$ and $\beta = +\infty$, and to an IR evaluation measure-based pooling strategy when $\beta = 1$.

A.2.1 Case $\beta = 0$

We now explore how the Hedge pooling strategy simplifies when $\beta = 0$. To do it we study the limit of $\beta \to 0$ for Eq. 6.24:

$$\lim_{\beta \to 0} \overline{L}(r, \mathcal{J}_{n-1}) = \lim_{\beta \to 0} \frac{\beta^{L(r, \mathcal{J}_{n-1})}}{\sum_{r' \in \mathcal{R}_p} \beta^{L(r', \mathcal{J}_{n-1})}}$$

To solve this limit we distinguish two cases, when the r at the numerator scores the minimum $L(r, \mathcal{J}_{n-1})$ among the runs in $\mathcal{R}_p$, and when it is not. To do it we first define $r_{\min} = \arg\min_{r \in \mathcal{R}_p} (L(r, \mathcal{J}_{n-1}))$. In the first case we have:

$$\lim_{\beta \to 0} \frac{\beta^{L(r_{\min}, \mathcal{J}_{n-1})}}{\sum_{r' \in \mathcal{R}_p} \beta^{L(r', \mathcal{J}_{n-1})}} = \lim_{\beta \to 0} \frac{1}{\sum_{r' \in \mathcal{R}_p} \beta^{L(r', \mathcal{J}_{n-1}) - L(r_{\min}, \mathcal{J}_{n-1})}} =$$
$$= \lim_{\beta \to 0} \frac{1}{\sum_{r' \in \mathcal{R}_p \setminus \{r_{\min}\}} \beta^{L(r', \mathcal{J}_{n-1}) - L(r_{\min}, \mathcal{J}_{n-1})} + \beta^0} = \frac{1}{0 + 1} = 1$$

In the second case, dividing and multiplying by $\beta^{L(r_{\min}, \mathcal{J}_{n-1})}$ we obtain:

$$\lim_{\beta \to 0} \frac{\beta^{L(r, \mathcal{J}_{n-1})}}{\sum_{r' \in \mathcal{R}_p} \beta^{L(r', \mathcal{J}_{n-1})}} = \lim_{\beta \to 0} \frac{\beta^{L(r', \mathcal{J}_{n-1}) - L(r_{\min}, \mathcal{J}_{n-1})}}{\sum_{r' \in \mathcal{R}_p} \beta^{L(r', \mathcal{J}_{n-1}) - L(r_{\min}, \mathcal{J}_{n-1})}} =$$
$$= \lim_{\beta \to 0} \frac{\beta^{L(r', \mathcal{J}_{n-1}) - L(r_{\min}, \mathcal{J}_{n-1})}}{\sum_{r' \in \mathcal{R}_p \setminus \{r_{\min}\}} \beta^{L(r', \mathcal{J}_{n-1}) - L(r_{\min}, \mathcal{J}_{n-1})} + \beta^0} = \frac{0}{0 + 1} = 0$$

With these two limits we see that this normalisation when $\beta = 0$ is 0 for every run but 1 for the run that scores the lowest loss $L(r, \mathcal{J}_{n-1})$. We substitute this result to Eq. 6.25 and simplify it as follows:

$$
\begin{aligned}
s^{\mathcal{J}_{n-1}}(d, \mathcal{R}_p) &= \sum_{r \in \mathcal{R}_p} \left(\overline{L}(r, \mathcal{J}_{n-1}) \cdot G^*(d, r) \right) = \\
&= \overline{L}(r_{\min}, \mathcal{J}_{n-1}) \cdot G^*(d, r_{\min}) + \sum_{r \in \mathcal{R}_p \setminus \{r_{\min}\}} \left(\overline{L}(r, \mathcal{J}_{n-1}) \cdot G^*(d, r) \right) = \\
&= G^*(d, r_{\min}) \simeq -\rho(d, r_{\min})
\end{aligned}
$$

This simplification shows that s is now rank equivalent to the definition of s for the run allocation strategies (Eq. 6.22). With run allocation function defined as:

$$
r_n = r_{\min} = \arg\max_{r \in \mathcal{R}_p} \left(-L(r, \mathcal{J}_{n-1}) \right)
$$

We now simplify the function $-L(r, \mathcal{J}_{n-1})$ as follows:

$$
\begin{aligned}
-L(r, \mathcal{J}_{n-1}) &= \frac{1}{2} \sum_{d \in \mathcal{J}^+_{n-1}} G^*(d, r) - \frac{1}{2} \sum_{d \in \mathcal{J}^-_{n-1}} G^*(d, r) = \\
&\frac{1}{2} \sum_{d \in \mathcal{J}^+_{n-1} : d \in \mathcal{D}_r} \log\left(\frac{|\mathcal{D}|}{\rho(d, r)} \right) + \frac{1}{2} \sum_{d \in \mathcal{J}^+_{n-1} : d \notin \mathcal{D}_r} \left(\frac{1}{|\mathcal{D}| - |\mathcal{D}_r|} \sum_{i=|r|}^{|\mathcal{D}|} \log\left(\frac{|\mathcal{D}|}{\rho(d, r)} \right) \right) - \\
&- \frac{1}{2} \sum_{d \in \mathcal{J}^-_{n-1} : d \in \mathcal{D}_r} \log\left(\frac{|\mathcal{D}|}{\rho(d, r)} \right) - \frac{1}{2} \sum_{d \in \mathcal{J}^-_{n-1} : d \notin \mathcal{D}_r} \left(\frac{1}{|\mathcal{D}| - |\mathcal{D}_r|} \sum_{i=|r|}^{|\mathcal{D}|} \log\left(\frac{|\mathcal{D}|}{\rho(d, r)} \right) \right) = \\
&= \frac{1}{2} (|\mathcal{J}^+_{n-1}| - |\mathcal{J}^-_{n-1}|) \log(|\mathcal{D}|) + \\
&+ \frac{1}{2} \sum_{d \in \mathcal{J}^+_{n-1} : d \in \mathcal{D}_r} \log\left(\frac{1}{\rho(d, r)} \right) + \frac{1}{2} \sum_{d \in \mathcal{J}^+_{n-1} : d \notin \mathcal{D}_r} \left(\frac{1}{|\mathcal{D}| - |\mathcal{D}_r|} \sum_{i=|\mathcal{D}_r|}^{|\mathcal{D}|} \log\left(\frac{1}{\rho(d, r)} \right) \right) - \\
&- \frac{1}{2} \sum_{d \in \mathcal{J}^-_{n-1} : d \in \mathcal{D}_r} \log\left(\frac{1}{\rho(d, r)} \right) - \frac{1}{2} \sum_{d \in \mathcal{J}^-_{n-1} : d \notin \mathcal{D}_r} \left(\frac{1}{|\mathcal{D}| - |\mathcal{D}_r|} \sum_{i=|r|}^{|\mathcal{D}|} \log\left(\frac{1}{\rho(d, r)} \right) \right) = \\
&= \frac{1}{2} (|\mathcal{J}^+_{n-1}| - |\mathcal{J}^-_{n-1}|) \log(|\mathcal{D}|) + \\
&+ \frac{1}{2} \sum_{d \in \mathcal{J}^+_{n-1} : d \in \mathcal{D}_r} \log\left(\frac{1}{\rho(d, r)} \right) - \frac{1}{2} \sum_{d \in \mathcal{J}^-_{n-1} : d \in \mathcal{D}_r} \log\left(\frac{1}{\rho(d, r)} \right) + \\
&+ \frac{1}{2} \frac{\log(|\mathcal{D}|! / |\mathcal{D}_r|!)}{|\mathcal{D}| - |\mathcal{D}_r|} (|\mathcal{J}^+_{n-1} \setminus \mathcal{D}_r| - |\mathcal{J}^-_{n-1} \setminus \mathcal{D}_r|)
\end{aligned}
$$

We observe that the right inside of the last equation is rank equivalent to:

$$-\sum_{d\in\mathcal{J}^+_{n-1}:d\in\mathcal{D}_r}\log\left(\rho(d,r)\right)+\sum_{d\in\mathcal{J}^-_{n-1}:d\in\mathcal{D}_r}\log\left(\rho(d,r)\right)+ \\ +\frac{\log(|\mathcal{D}|!/|\mathcal{D}_r|!)}{|\mathcal{D}|-|\mathcal{D}_r|}(|\mathcal{J}^+_{n-1}\setminus r|-|\mathcal{J}^-_{n-1}\setminus r|)$$

For the sake of clarity, let us define a constant $C_r=\log(|\mathcal{D}|!/|\mathcal{D}_r|!)/(|\mathcal{D}|-|\mathcal{D}_r|)$. This constant when $|\mathcal{D}|\gg|\mathcal{D}_r|$ can be approximated to $C_r\approx\log(|\mathcal{D}|!)/|\mathcal{D}|=C$. Thereby obtaining the following allocation strategy:

$$r_n=\arg\max_{r\in\mathcal{R}_p}\left(C\cdot(|\mathcal{J}^+_{n-1}\setminus\mathcal{D}_r|-|\mathcal{J}^-_{n-1}\setminus\mathcal{D}_r|)+\sum_{d\in\mathcal{J}_{n-1}\cap\mathcal{D}_r}\begin{cases}-\log\left(\rho(d,r)\right) & d\in\mathcal{J}^+_{n-1}\\ +\log\left(\rho(d,r)\right) & d\in\mathcal{J}^-_{n-1}\end{cases}\right)$$

Following a more compact rank equivalent form of the same strategy:

$$r_n=\arg\max_{r\in\mathcal{R}_p}\left(\sum_{d\in\mathcal{J}_{n-1}\cap\mathcal{D}_r}\left[\left(\log(\rho(d,r))+\frac{\log(|\mathcal{D}|!)}{|\mathcal{D}|}\right)\cdot\begin{cases}-1 & d\in\mathcal{J}^+_{n-1}\\ +1 & d\in\mathcal{J}^-_{n-1}\end{cases}\right]\right)$$

In this run allocation strategy we can distinguish two addenda. On the left inside one about documents in the relevance assessments but not in the run, and on the right inside one about documents in the relevance assessments and in the run. With the former addend r gains a positive gain if more non-relevant documents than relevant ones have been discovered by r. With the latter addend r gains a gain for every non-relevant document discovered by r, and negative otherwise.

A.2.2 Case $\beta=1$

When $\beta=1$, Eq. 6.24 can be simplified as follows:

$$\overline{L}(r,\mathcal{J}_{n-1})=\frac{1}{|\mathcal{R}_p|}$$

and Eq. 6.25 becomes:

$$s^{\mathcal{J}_{n-1}}(d,\mathcal{R}_p)=\sum_{r\in\mathcal{R}_p}\left(\frac{1}{|\mathcal{R}_p|}\cdot G^*(d,r)\right)=\frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p}G^*(d,r)$$

We observe that, since there is no condition on n on the right inside of the equation, the definition of $s^{\mathcal{J}_{n-1}}$ is equivalent to its non sequential definition s. Therefore the strategy is now a non-adaptive strategy defined by the following s:

$$s(d,\mathcal{R}_p)=\frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p}G^*(d,r)$$

To compare this strategy with the other pooling strategy presented in this paper we perform the following simplifications.

$$s(d,\mathcal{R}_p) = \frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p:d\in\mathcal{D}_r} G(\rho(d,r)) + \frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p:d\notin\mathcal{D}_r}\frac{1}{|\mathcal{D}|-|\mathcal{D}_r|}\sum_{n=|\mathcal{D}_r|+1}^{|\mathcal{D}|} G(i)$$

We now subtract from the right inside of this formula a constant. This can be done because to sum a constant quantity is still rank equivalent to the previous one:

$$\begin{aligned}
&\frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p:d\in\mathcal{D}_r} G(\rho(d,r)) + \frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p:d\notin\mathcal{D}_r}\frac{1}{|\mathcal{D}|-|\mathcal{D}_r|}\sum_{n=|\mathcal{D}_r|+1}^{|\mathcal{D}|} G(i) - \\
&\qquad - \frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p}\frac{1}{|\mathcal{D}|-|\mathcal{D}_r|}\sum_{n=|\mathcal{D}_r|+1}^{|\mathcal{D}|} G(i) = \\
&\qquad = \frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p:d\in\mathcal{D}_r}\left(G(\rho(d,r)) - \frac{1}{|\mathcal{D}|-|\mathcal{D}_r|}\sum_{n=|\mathcal{D}_r|+1}^{|\mathcal{D}|} G(i)\right) = \\
&= \frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p:d\in\mathcal{D}_r}\left(\log\left(\frac{|\mathcal{D}|}{\rho(d,r)}\right) - \frac{1}{|\mathcal{D}|-|\mathcal{D}_r|}\sum_{n=|\mathcal{D}_r|+1}^{|\mathcal{D}|}\log\left(\frac{|\mathcal{D}|}{n}\right)\right) = \\
&= \frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p:d\in\mathcal{D}_r}\left(\log\left(\frac{1}{\rho(d,r)}\right) + \frac{1}{|\mathcal{D}|-|\mathcal{D}_r|}\sum_{n=|\mathcal{D}_r|+1}^{|\mathcal{D}|}\log(n)\right) = \\
&\qquad = \frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p:d\in\mathcal{D}_r}\left(\log\left(\frac{1}{\rho(d,r)}\right) + \frac{\log(|\mathcal{D}|!/|\mathcal{D}_r|!)}{|\mathcal{D}|-|\mathcal{D}_r|}\right)
\end{aligned}$$

We now define a constant, as done in the previous case, $C_r = \frac{\log(|\mathcal{D}|!/|\mathcal{D}_r|!)}{|\mathcal{D}|-|\mathcal{D}_r|}$, and for $|\mathcal{D}| \gg |\mathcal{D}_r|$, we approximate C_r to $C = \frac{\log(|\mathcal{D}|!)}{|\mathcal{D}|}$. We then substitute this to the previous formula:

$$s(d,\mathcal{R}_p) = \frac{1}{|\mathcal{R}_p|}\sum_{r\in\mathcal{R}_p:d\in\mathcal{D}_r}\left(\log\left(\frac{1}{\rho(d,r)}\right) + C\right)$$

Finally, by rank equivalence we obtain:

$$s(d,\mathcal{R}_p) = \sum_{r\in\mathcal{R}_p:d\in\mathcal{D}_r}\left(\log\left(\frac{1}{\rho(d,r)}\right) + C\right)$$

We now observe that this run allocation is equivalent to an evaluation measure-based strategy with function gain G defined as:

$$G(\rho) = \log\left(\frac{1}{\rho}\right) + C$$

The derivative of this function is shown in Figure 6.2 for comparison with the other evaluation measure based-strategies.

A.2.3 Case $\beta = +\infty$

When $\beta = +\infty$ we obtain a behaviour opposite to when $\beta = 0$. With run allocation function equal to:

$$r_n = \arg\max_{r \in \mathcal{R}_p} \left(L(r, \mathcal{J}_{n-1}) \right)$$

Performing similar simplifications to the case $\beta = 0$ we obtain:

$$r_n = \arg\max_{r \in \mathcal{R}_p} \left(C \cdot (|\mathcal{J}_{n-1}^- \setminus \mathcal{D}_r| - |\mathcal{J}_{n-1}^+ \setminus \mathcal{D}_r|) + \sum_{d \in \mathcal{J}_{n-1} \cap \mathcal{D}_r} \begin{cases} +\log\left(\rho(d,r)\right) & d \in \mathcal{J}_{n-1}^+ \\ -\log\left(\rho(d,r)\right) & d \in \mathcal{J}_{n-1}^- \end{cases} \right)$$

Following a more compact rank equivalent form of the same strategy:

$$r_n = \arg\max_{r \in \mathcal{R}_p} \left(\sum_{d \in \mathcal{J}_{n-1} \cap \mathcal{D}_r} \left[\left(\log(\rho(d,r)) + \frac{\log(|\mathcal{D}|!)}{|\mathcal{D}|} \right) \cdot \begin{cases} +1 & d \in \mathcal{J}_{n-1}^+ \\ -1 & d \in \mathcal{J}_{n-1}^- \end{cases} \right] \right)$$

List of Figures

List of Tables

Bibliography

[AK92] G Amati and S Kerpedjiev. *An Information Retrieval Logic Model: Implementation and Experiments*. Tech. rep. REL 5b04892. Fondazione Ugo Bordoni, Rome, Italy, Feb. 1992.

[AV02] Gianni Amati and Cornelis Joost Van Rijsbergen. "Probabilistic Models of Information Retrieval Based on Measuring the Divergence from Randomness". In: *ACM Trans. Inf. Syst.* 20.4 (Oct. 2002), pp. 357–389.

[Arm+09] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. "Improvements That Don'T Add Up: Ad-hoc Retrieval Results Since 1998". In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM '09. Hong Kong, China: ACM, 2009, pp. 601–610.

[AM01] Javed A. Aslam and Mark Montague. "Models for Metasearch". In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '01. New Orleans, Louisiana, USA: ACM, 2001, pp. 276–284.

[APS03] Javed A. Aslam, Virgiliu Pavlu, and Robert Savell. "A Unified Model for Metasearch, Pooling, and System Evaluation". In: *Proceedings of the 12th International Conference on Information and Knowledge Management*. CIKM '03. New Orleans, LA, USA: ACM, 2003, pp. 484–491.

[ACF02] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. "Finite-time Analysis of the Multiarmed Bandit Problem". In: *Mach. Learn.* 47.2-3 (May 2002), pp. 235–256.

[AO09] Leif Azzopardi and Ciaran Owens. "Search Engine Predilection Towards News Media Providers". In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: ACM, 2009, pp. 774–775.

[AV08a] Leif Azzopardi and Vishwa Vinay. "Accessibility in Information Retrieval". In: *Proceedings of the 30th European Conference on IR Research*. ECIR '08. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 482–489.

[AV08b] Leif Azzopardi and Vishwa Vinay. “Retrievability: An Evaluation Measure for Higher Order Information Access Tasks”. In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. CIKM ’08. Napa Valley, California, USA: ACM, 2008, pp. 561–570.

[Bac11] Richard Bache. “Measuring and Improving Access to the Corpus”. In: *Current Challenges in Patent Information Retrieval*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 147–165.

[BR11] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. 2nd. USA: Addison-Wesley Publishing Company, 2011.

[BLO06] Jason R Baron, David D Lewis, and Douglas W Oard. “TREC 2006 Legal Track Overview”. In: *Proceedings of the 15th Text REtrieval Conference*. TREC ’06. Gaithersburg, Maryland (USA): NIST, 2006.

[BR09] Shariq Bashir and Andreas Rauber. “Improving Retrievability of Patents with Cluster-based Pseudo-relevance Feedback Documents Selection”. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM ’09. Hong Kong, China: ACM, 2009, pp. 1863–1866.

[BR10] Shariq Bashir and Andreas Rauber. “Improving Retrievability of Patents in Prior-Art Search”. In: *Proceedings of the 32nd European Conference on IR Research*. ECIR ’10. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 457–470.

[BH95] Yoav Benjamini and Yosef Hochberg. “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1 (1995), pp. 289–300.

[BL07] David Bodoff and Pu Li. “Test Theory for Assessing IR Test Collections”. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’07. Amsterdam, The Netherlands: ACM, 2007, pp. 367–374.

[Buc+07] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen Voorhees. “Bias and the limits of pooling for large collections”. In: *Information Retrieval* 10.6 (2007), pp. 491–508.

[BV00] Chris Buckley and Ellen M. Voorhees. “Evaluating Evaluation Measure Stability”. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’00. Athens, Greece: ACM, 2000, pp. 33–40.

[BV04] Chris Buckley and Ellen M. Voorhees. “Retrieval Evaluation with Incomplete Information”. In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’04. Sheffield, United Kingdom: ACM, 2004, pp. 25–32.

[Büt+07] Stefan Büttcher, Charles L. A. Clarke, Peter C. K. Yeung, and Ian Soboroff. "Reliable Information Retrieval Evaluation with Incomplete and Biased Judgements". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '07. Amsterdam, The Netherlands: ACM, 2007, pp. 63–70.

[Car11] Ben Carterette. "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. Beijing, China: ACM, 2011, pp. 903–912.

[CG99] K. Church and W. Gale. "Inverse Document Frequency (IDF): A Measure of Deviations from Poisson". In: *Natural Language Processing Using Very Large Corpora*. Dordrecht: Springer Netherlands, 1999, pp. 283–295.

[CS14] Charles L. A. Clarke and Mark D. Smucker. "Time Well Spent". In: *Proceedings of the 5th Information Interaction in Context Symposium*. IIiX '14. Regensburg, Germany: ACM, 2014, pp. 205–214.

[Col+15] Kevyn Collins-Thompson, Craig Macdonald, Paul Bennett, Fernando Diaz, and Ellen M Voorhees. "TREC 2014 Web Track Overview". In: *Proceedings of the 23rd Text REtrieval Conference*. TREC '14. Gaithersburg, Maryland (USA): NIST, 2015.

[CCB09] Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. "Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods". In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: ACM, 2009, pp. 758–759.

[CPC98] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. "Efficient Construction of Large Test Collections". In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '98. Melbourne, Australia: ACM, 1998, pp. 282–289.

[Cro00] W. Bruce Croft. "Combining Approaches to Information Retrieval". In: *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*. Ed. by W. Bruce Croft. Boston, MA: Springer US, 2000, pp. 1–36.

[CO12] Ronan Cummins and Colm O'Riordan. "A Constraint to Automatically Regulate Document-length Normalisation". In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. CIKM '12. Maui, Hawaii, USA: ACM, 2012, pp. 2443–2446.

[Dum+02] Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. "Web Question Answering: Is More Always Better?" In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '02. Tampere, Finland: ACM, 2002, pp. 291–298.

[FTZ04] Hui Fang, Tao Tao, and ChengXiang Zhai. "A Formal Study of Information Retrieval Heuristics". In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '04. Sheffield, United Kingdom: ACM, 2004, pp. 49–56.

[Gas72] Joseph L Gastwirth. "The estimation of the Lorenz curve and Gini index". In: *The Review of Economics and Statistics* 54.3 (1972), pp. 306–316.

[GG05] Cyril Goutte and Eric Gaussier. "A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation". In: *Proceedings of the 27th European Conference on IR Research*. ECIR '05. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 345–359.

[HL13] Allan Hanbury and Mihai Lupu. "Toward a Model of Domain-specific Search". In: *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*. OAIR '13. Lisbon, Portugal: CID, 2013, pp. 33–36.

[Har93] Donna Harman. "Overview of the First TREC Conference". In: *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '93. Pittsburgh, Pennsylvania, USA: ACM, 1993, pp. 36–47.

[Har94] Donna Harman. "Overview of the Third Text Retrieval Conference (TREC-3)". In: *Proceedings of the 3rd Text REtrieval Conference*. TREC '94. Gaithersburg, Maryland (USA): NIST, 1994.

[Har95] Donna Harman. "Overview of the fourth text retrieval conference (TREC-4)". In: *Proceedings of the 4th Text REtrieval Conference*. TREC '04 4. Gaithersburg, Maryland (USA): NIST, 1995, pp. 1–24.

[HJ10] Claudia Hauff and Franciska de Jong. "Retrieval System Evaluation: Automatic Evaluation Versus Incomplete Judgments". In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '10. Geneva, Switzerland: ACM, 2010, pp. 863–864.

[Haw00] David Hawking. "Overview of the TREC-9 Web Track". In: *Proceedings of the 9th Text REtrieval Conference*. TREC '00. Gaithersburg, Maryland (USA): NIST, 2000.

[HO03] Ben HE and Iadh Ounis. "A Study of Parameter Tuning for Term Frequency Normalization". In: *Proceedings of the 12th International Conference on Information and Knowledge Management*. CIKM '03. New Orleans, LA, USA: ACM, 2003, pp. 10–16.

[HO05a] Ben He and Iadh Ounis. “A Study of the Dirichlet Priors for Term Frequency Normalisation”. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’05. Salvador, Brazil: ACM, 2005, pp. 465–471.

[HO05b] Ben He and Iadh Ounis. “Term Frequency Normalisation Tuning for BM25 and DFR Models”. In: *Proceedings of the 27th European Conference on IR Research*. ECIR ’05. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 200–214.

[Her+05] William Hersh, Aaron Cohen, Jianji Yang, Ravi Teja Bhupatiraju, Phoebe Roberts, and Marti Hearst. “TREC 2005 Genomics Track Overview”. In: *Proceedings of the 14th Text REtrieval Conference*. TREC ’05. Gaithersburg, Maryland (USA): NIST, 2005.

[HLR16] Katja Hofmann, Lihong Li, and Filip Radlinski. “Online Evaluation for Information Retrieval”. In: *Foundations and Trendső in Information Retrieval* 10.1 (2016), pp. 1–117.

[JK02] Kalervo Järvelin and Jaana Kekäläinen. “Cumulated Gain-based Evaluation of IR Techniques”. In: *ACM Trans. Inf. Syst.* 20.4 (Oct. 2002), pp. 422–446.

[JS16] Thorsten Joachims and Adith Swaminathan. “Counterfactual Evaluation and Learning for Search, Recommendation and Ad Placement”. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’16. Pisa, Italy: ACM, 2016, pp. 1199–1201.

[KMS94] Daniel Knaus, Elke Mittendorf, and Peter Schauble. “Improving a basic retrieval method by links and passage level evidence”. In: *Proceedings of the 3rd Text REtrieval Conference*. TREC ’94. Gaithersburg, Maryland (USA): NIST, 1994, pp. 241–241.

[KZ14] Bevan Koopman and Guido Zuccon. “Why Assessing Relevance in Medical IR is Demanding”. In: *Proceedings of the 1st Medical Information Retrieval Workshop at SIGIR*. MedIR ’14. ACM, 2014, pp. 16–19.

[Lee97] Joon Ho Lee. “Analyses of Multiple Evidence Combination”. In: *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’97. Philadelphia, Pennsylvania, USA: ACM, 1997, pp. 267–276.

[LH05] Wei-Hao Lin and Alexander Hauptmann. “Revisiting the Effect of Topic Set Size on Retrieval Error”. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’05. Salvador, Brazil: ACM, 2005, pp. 637–638.

[Lip16] Aldo Lipani. “Fairness in Information Retrieval”. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’16. Pisa, Italy: ACM, 2016, pp. 1171–1171.

[Lip+14a] Aldo Lipani, Linda Andersson, Florina Piroi, Mihai Lupu, and Allan Hanbury. "TUW-IMP at the NTCIR-11 Math-2". In: *Proceedings of the 11th NII Test Collection for IR Systems*. NTCIR '14. Tokyo, Japan: NII, 2014.

[Lip+15a] Aldo Lipani, Mihai Lupu, Akiko Aizawa, and Allan Hanbury. "An Initial Analytical Exploration of Retrievability". In: *Proceedings of the 2015 ACM International Conference on The Theory of Information Retrieval*. ICTIR '15. Northampton, Massachusetts, USA: ACM, 2015, pp. 329–332.

[LLH15] Aldo Lipani, Mihai Lupu, and Allan Hanbury. "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. Santiago, Chile: ACM, 2015, pp. 103–112.

[LLH16] Aldo Lipani, Mihai Lupu, and Allan Hanbury. "The Curious Incidence of Bias Corrections in the Pool". In: *Proceedings of the 38th European Conference on IR Research*. ECIR '16. Cham: Springer International Publishing, 2016, pp. 267–279.

[LLH17] Aldo Lipani, Mihai Lupu, and Allan Hanbury. "Visual Pool: A Tool to Visualize and Interact with the Pooling Method". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '17. Shinjuku, Tokyo, Japan: ACM, 2017, pp. 1321–1324.

[Lip+15b] Aldo Lipani, Mihai Lupu, Allan Hanbury, and Akiko Aizawa. "Verboseness Fission for BM25 Document Length Normalization". In: *Proceedings of the 1st ACM International Conference on The Theory of Information Retrieval*. ICTIR '15. Northampton, Massachusetts, USA: ACM, 2015, pp. 385–388.

[Lip+16a] Aldo Lipani, Mihai Lupu, Evangelos Kanoulas, and Allan Hanbury. "The Solitude of Relevant Documents in the Pool". In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. CIKM '16. Indianapolis, Indiana, USA: ACM, 2016, pp. 1989–1992.

[Lip+17a] Aldo Lipani, Mihai Lupu, Joao Palotti, Guido Zuccon, and Allan Hanbury. "Fixed Budget Pooling Strategies Based on Fusion Methods". In: *Proceedings of the 32nd ACM SIGAPP Symposium On Applied Computing*. SAC '17. Marrakech, Morocco: ACM, 2017, pp. 919–924.

[Lip+17b] Aldo Lipani, Joao Palotti, Mihai Lupu, Florina Piroi, Guido Zuccon, and Allan Hanbury. "Fixed-Cost Pooling Strategies Based on IR Evaluation Measures". In: *Proceedings of the 39th European Conference on IR Research*. ECIR '17. Cham: Springer International Publishing, 2017, pp. 357–368.

[Lip+14b] Aldo Lipani, Florina Piroi, Linda Andersson, and Allan Hanbury. "An Information Retrieval Ontology for Information Retrieval Nanopublications". In: *Proceedings of the 5th International Conference of the CLEF Initiative*. CLEF '14. Cham: Springer International Publishing, 2014, pp. 44–49.

[Lip+14c] Aldo Lipani, Florina Piroi, Linda Andersson, and Allan Hanbury. “Extracting Nanopublications from IR Papers”. In: *Proceedings of the 7th Information Retrieval Facility Conference*. IRFC ’14. Cham: Springer International Publishing, 2014, pp. 53–62.

[Lip+18] Aldo Lipani, Thomas Roelleke, Mihai Lupu, and Allan Hanbury. “A Systematic Approach to Normalization in Probabilistic Models”. In: *Information Retrieval Journal* (June 2018).

[Lip+16b] Aldo Lipani, Guido Zuccon, Mihai Lupu, Bevan Koopman, and Allan Hanbury. “The Impact of Fixed-Cost Pooling Strategies on Test Collection Bias”. In: *Proceedings of the 2nd ACM International Conference on the Theory of Information Retrieval*. ICTIR ’16. Newark, Delaware, USA: ACM, 2016, pp. 105–108.

[LAB08] David E. Losada, Leif Azzopardi, and Mark Baillie. “Revisiting the Relationship Between Document Length and Relevance”. In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. CIKM ’08. Napa Valley, California, USA: ACM, 2008, pp. 419–428.

[LPB16] David E. Losada, Javier Parapar, and Álvaro Barreiro. “Feeling Lucky?: Multi-armed Bandits for Ordering Judgements in Pooling-based Evaluation”. In: *Proceedings of the 31st ACM SIGAPP Symposium On Applied Computing*. SAC ’16. Pisa, Italy: ACM, 2016, pp. 1027–1034.

[LZ11a] Yuanhua Lv and ChengXiang Zhai. “Lower-bounding Term Frequency Normalization”. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. CIKM ’11. Glasgow, Scotland, UK: ACM, 2011, pp. 7–16.

[LZ11b] Yuanhua Lv and ChengXiang Zhai. “When Documents Are Very Long, BM25 Fails!” In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’11. Beijing, China: ACM, 2011, pp. 1103–1104.

[Mac09] Craig Macdonald. “The Voting Model for People Search”. In: *SIGIR Forum* 43.1 (June 2009), pp. 73–73.

[MRS08] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[AS10] Azzah Al-Maskari and Mark Sanderson. “A review of factors influencing user satisfaction in information retrieval”. In: *Journal of the American Society for Information Science and Technology* 61.5 (2010), pp. 859–868.

[MWZ07] Alistair Moffat, William Webber, and Justin Zobel. “Strategic System Comparisons via Targeted Relevance Judgments”. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’07. Amsterdam, The Netherlands: ACM, 2007, pp. 375–382.

[MZ08] Alistair Moffat and Justin Zobel. “Rank-biased Precision for Measurement of Retrieval Effectiveness”. In: *ACM Trans. Inf. Syst.* 27.1 (Dec. 2008), 2:1–2:27.

[MA01] Mark Montague and Javed A. Aslam. “Relevance Score Normalization for Metasearch”. In: *Proceedings of the 10th International Conference on Information and Knowledge Management*. CIKM ’01. Atlanta, Georgia, USA: ACM, 2001, pp. 427–433.

[MA02] Mark Montague and Javed A. Aslam. “Condorcet Fusion for Improved Retrieval”. In: *Proceedings of the 11th International Conference on Information and Knowledge Management*. CIKM ’02. McLean, Virginia, USA: ACM, 2002, pp. 538–548.

[NKL08] Seung-Hoon Na, In-Su Kang, and Jong-Hyeok Lee. “Improving Term Frequency Normalization for Multi-topical Documents and Application to Language Modeling Approaches”. In: *Proceedings of the 30th European Conference on IR Research*. ECIR ’08. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 382–393.

[Oun+11] Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. “Overview of the TREC-2011 microblog track”. In: *Proceedings of the 20th Text REtrieval Conference*. TREC ’11. Gaithersburg, Maryland (USA): NIST, 2011.

[Oun+06] Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. “Overview of the TREC-2006 Blog Track”. In: *Proceedings of the 15th Text REtrieval Conference*. TREC ’06. Gaithersburg, Maryland (USA): NIST, 2006.

[PZ07] Laurence AF Park and Yuye Zhang. “On the distribution of user persistence for rank-biased precision”. In: *Proceedings of the 12th Australasian Document Computing Symposium*. ADCS ’07. 2007, pp. 17–24.

[PCG10] Jeremy Pickens, Matthew Cooper, and Gene Golovchinsky. “Reverted Indexing for Feedback and Expansion”. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. CIKM ’10. Toronto, ON, Canada: ACM, 2010, pp. 1049–1058.

[Pir+15] F. Piroi, A. Lipani, M. Lupu, and A. Hanbury. “DASyR(IR) - document analysis system for systematic reviews (in Information Retrieval)”. In: *Proceedings of the 13th International Conference on Document Analysis and Recognition*. ICDAR ’13. Tunis, Tunisia: IEEE, Aug. 2015, pp. 591–595.

[PC98] Jay M. Ponte and W. Bruce Croft. “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. Melbourne, Australia: ACM, 1998, pp. 275–281.

[Rob08] Stephen Robertson. “On the history of evaluation in IR”. In: *Journal of Information Science* 34.4 (2008), pp. 439–456.

[Rob+93] Stephen E. Robertson, Stephen Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. “Okapi at TREC-2”. In: *Proceedings of the 2nd Text REtrieval Conference*. TREC ’93. Gaithersburg, Maryland (USA): NIST, Jan. 1993, pp. 21–34.

[Rob+94] Stephen E Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. “Okapi at TREC-3”. In: *Proceedings of the 3rd Text REtrieval Conference*. Vol. 3. TREC ’94. Gaithersburg, Maryland (USA): NIST, 1994, pp. 109–126.

[RZ09] Stephen Robertson and Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond”. In: *Found. Trends Inf. Retr.* 3.4 (Apr. 2009), pp. 333–389.

[Roe13] Thomas Roelleke. “Information Retrieval Models: Foundations and Relationships”. In: *Synthesis Lectures on Information Concepts, Retrieval, and Services* 5.3 (2013), pp. 1–163.

[RKB15] Thomas Roelleke, Andreas Kaltenbrunner, and Ricardo Baeza-Yates. “Harmony Assumptions in Information Retrieval and Social Networks”. In: *The Computer Journal* 58.11 (2015), p. 2982.

[RW08] Thomas Roelleke and Jun Wang. “TF-IDF Uncovered: A Study of Theories and Probabilities”. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’08. Singapore, Singapore: ACM, 2008, pp. 435–442.

[RV13] François Rousseau and Michalis Vazirgiannis. “Composition of TF Normalizations: New Insights on Scoring Functions for Ad Hoc IR”. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’13. Dublin, Ireland: ACM, 2013, pp. 917–920.

[Sak04] Tetsuya Sakai. “New Performance Metrics Based on Multigrade Relevance: Their Application to Question Answering.” In: *Proceedings of the 4th NII Test Collection for IR Systems*. NTCIR ’04. Tokyo, Japan: NII, 2004.

[Sak07] Tetsuya Sakai. “Alternatives to Bpref”. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’07. Amsterdam, The Netherlands: ACM, 2007, pp. 71–78.

[SK08] Tetsuya Sakai and Noriko Kando. “On information retrieval metrics designed for evaluation with incomplete relevance assessments”. In: *Information Retrieval* 11.5 (Oct. 2008), pp. 447–470.

[San10] Mark Sanderson. “Test Collection Based Evaluation of Information Retrieval Systems”. In: *Foundations and Trendső in Information Retrieval* 4.4 (2010), pp. 247–375.

[SZ05] Mark Sanderson and Justin Zobel. “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. Salvador, Brazil: ACM, 2005, pp. 162–169.

[SBM96] Amit Singhal, Chris Buckley, and Mandar Mitra. “Pivoted Document Length Normalization”. In: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’96. Zurich, Switzerland: ACM, 1996, pp. 21–29.

[SR75] K. Spärck Jones and C. J. van Rijsbergen. “Report on the need for and provision of an ’ideal’ information retrieval test collection”. In: *British Library Research and Development Report No. 5266* (1975), p. 44.

[Spä03] Karen Spärck Jones. “Letter to the editor”. In: *Information Processing & Management* 39.1 (2003), pp. 156–159.

[SB98] R. S. Sutton and A. G. Barto. “Reinforcement Learning: An Introduction”. In: *IEEE Transactions on Neural Networks* 9.5 (Sept. 1998), pp. 1054–1054.

[Tho33] William R. Thompson. “On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples”. In: *Biometrika* 25.3/4 (1933), pp. 285–294.

[UMM13] Julián Urbano, Mónica Marrero, and Diego Martn. “On the Measurement of Test Collection Reliability”. In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’13. Dublin, Ireland: ACM, 2013, pp. 393–402.

[Voo05] Ellen M. Voorhees. “Overview of the TREC 2005 robust retrieval track”. In: *Proceedings of the 14th Text REtrieval Conference*. TREC ’05. Gaithersburg, Maryland (USA): NIST, 2005.

[Voo09] Ellen M. Voorhees. “Topic Set Size Redux”. In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’09. Boston, MA, USA: ACM, 2009, pp. 806–807.

[Voo14] Ellen M. Voorhees. “The Effect of Sampling Strategy on Inferred Measures”. In: *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’14. Gold Coast, Queensland, Australia: ACM, 2014, pp. 1119–1122.

[VB02] Ellen M. Voorhees and Chris Buckley. “The Effect of Topic Set Size on Retrieval Experiment Error”. In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’02. Tampere, Finland: ACM, 2002, pp. 316–323.

[VH99a] Ellen M. Voorhees and Donna Harman. “Overview of the Eight Text Retrieval Conference”. In: *Proceedings of the 8th Text REtrieval Conference*. TREC ’99. Gaithersburg, Maryland (USA): NIST, 1999.

[VH05] Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. The MIT Press, 2005.

[VH99b] Ellen Voorhes and Donna Harman. "Overview of the Eighth Text Retrieval Conference". In: *Proceedings of the 8th Text REtrieval Conference*. TREC '99. Gaithersburg, Maryland (USA): NIST, 1999.

[WP09] William Webber and Laurence A. F. Park. "Score Adjustment for Correction of Pooling Bias". In: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA: ACM, 2009, pp. 444–451.

[WA13] Colin Wilkie and Leif Azzopardi. "Relating Retrievability, Performance and Length". In: *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '13. Dublin, Ireland: ACM, 2013, pp. 937–940.

[WA14] Colin Wilkie and Leif Azzopardi. "A Retrievability Analysis: Exploring the Relationship Between Retrieval Bias and Retrieval Performance". In: *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. CIKM '14. Shanghai, China: ACM, 2014, pp. 81–90.

[WA15] Colin Wilkie and Leif Azzopardi. "Retrievability and Retrieval Bias: A Comparison of Inequality Measures". In: *Proceedings of the 37th European Conference on IR Research*. ECIR '05. Cham: Springer International Publishing, 2015, pp. 209–214.

[YA06] Emine Yilmaz and Javed A. Aslam. "Estimating Average Precision with Incomplete and Imperfect Judgments". In: *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*. CIKM '06. Arlington, Virginia, USA: ACM, 2006, pp. 102–111.

[YKA08] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. "A Simple and Efficient Sampling Method for Estimating AP and NDCG". In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '08. Singapore, Singapore: ACM, 2008, pp. 603–610.

[ZL01] Chengxiang Zhai and John Lafferty. "A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval". In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '01. New Orleans, Louisiana, USA: ACM, 2001, pp. 334–342.

[ZPM10] Yuye Zhang, Laurence A. F. Park, and Alistair Moffat. "Click-based evidence for decaying weight distributions in search effectiveness metrics". In: *Information Retrieval* 13.1 (2010), pp. 46–69.

[ZC09] L. Zheng and I. J. Cox. “Document-Oriented Pruning of the Inverted Index in Information Retrieval Systems”. In: *Proceedings of the 23rd International Conference on Advanced Information Networking and Applications Workshops*. WAINA ’09. Bradford, UK: IEEE, May 2009, pp. 697–702.

[Zob98] Justin Zobel. “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. Melbourne, Australia: ACM, 1998, pp. 307–314.