# Predicting the Effectiveness of Queries and Retrieval Systems

Claudia Hauff

**PhD dissertation committee:**

Chairman and Secretary:

Prof. dr. ir. A. J. Mouthaan, Universiteit Twente, NL

Promotor:

Prof. dr. F. M. G. de Jong, Universiteit Twente, NL

Assistant-promotor:

Dr. ir. D. Hiemstra, Universiteit Twente, NL

Members:

Prof. dr. P. Apers, Universiteit Twente, NL
Dr. L. Azzopardi, University of Glasgow, UK
Prof. dr. ir. W. Kraaij, Radboud Unversiteit Nijmegen/TNO, NL
Dr. V. Murdock, Yahoo! Research Barcelona, Spain
Prof. dr. ir. A. Nijholt, Universiteit Twente, NL
Prof. dr. ir. A. P. de Vries, TU Delft/CWI, NL

CTIT Dissertation Series No. 09-161
Center for Telematics and Information Technology (CTIT)
P.O. Box 217 – 7500AE Enschede – the Netherlands
ISSN: 1381-3617

MultimediaN: http://www.multimedian.nl
The research reported in this thesis was supported by MultimediaN, a program financed by the Dutch government under contract BSIK 03031.

Human Media Interaction: http://hmi.ewi.utwente.nl
The research reported in this thesis was carried out at the Human Media Interaction research group of the University of Twente.

SIKS Dissertation Series No. 2010-05
The research reported in this thesis was carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.



ISBN: 978-90-365-2953-2
ISSN: 1381-3617, No. 09-161

# PREDICTING THE EFFECTIVENESS OF QUERIES AND RETRIEVAL SYSTEMS

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
prof. dr. H. Brinksma,
on account of the decision of the graduation committee
to be publicly defended
on Friday, January 29, 2010 at 15.00

by

Claudia Hauff

born on June 10, 1980

in Forst (Lausitz), Germany

Promotor: Prof. dr. F. M. G. de Jong

Assistant-promotor: Dr. ir. D. Hiemstra


ISBN: 978-90-365-2953-2

# Acknowledgements

After four years and three months this book is finally seeing the light of day and I am finally seeing the end of my PhD road. An achievement only made possible with the help of family members, friends and colleagues all of whom I am deeply grateful to.

I would like to thank my supervisors Franciska de Jong and Djoerd Hiemstra who gave me the opportunity to pursue this PhD in the first place and offered their help and advice whenever I needed it. They kept encouraging me and consistently insisted that everything will work out in the end while I was consistently less optimistic. Fortunately, they were right and I was wrong.

Arjen de Vries deserves a big thanks for putting me in contact with Vanessa Murdock who in turn gave me the chance to spend three wonderful months at Yahoo! Research Barcelona. I learned a lot from her and met many kind people. I would also like to thank Georgina, Börkur, Lluís, Simon and Gleb with whom I shared an office (Simon and Gleb were also my flatmates) and many lunches during that time.

After Barcelona and a few months of Enschede, I packed again and moved for another internship to Glasgow. Leif Azzopardi hosted my visit to the IR Group at Glasgow University where I encountered numerous interesting people. Leif always had time for one more IR-related discussion (no matter how obscure the subject) and one more paper writing session.

Vishwa Vinay, who is thoroughly critical of my research topic, helped me a lot with long e-mail discussions that routinely made me re-evaluate my ideas. At the same time, Vinay also managed to keep my spirits up in my final PhD year when he replied to my “it is Sunday and i am working” e-mails with “me too!”. Somehow that made me feel a little better.

Throughout the four years that I worked at HMI, Dolf Trieschnigg was the prime suspect for talks about life and IR - the many, many discussions, coffees and whiteboard drawings were of paramount importance to finishing this thesis.

In the early days of my PhD, collaborations with Thijs and Roeland from HMI as well as Henning and Robin from the Database group helped me to find my way around IR.

The group of PhD students at HMI seemed ever increasing, which made for many discussions at various coffee/lunch/cake/borrel/Christmas-dinner breaks. Andreea, Nataša, Olga and Yujia were always up for chats on important non-work related topics. Olga was also often up for free time activities in and around Enschede. My office mates Marijn and Frans managed to survive my moods when things were not going so well. From Wim I learned that underwater hockey is not an imaginary

activity, from Ronald I gained insights into the Dutch psyche, while thanks to Dolf I now know that sailing is probably not my sport.

Working at HMI was made a lot easier by Alice and Charlotte who dealt with all the bureaucracy that comes along with being a PhD student. They never failed to remain helpful and patient even at times when I discovered nearly a year late that some administrative procedures had changed.

Lynn not only corrected my English (I finally learned the difference between “if” and “whether”), she also regularly provided HMI with great cakes and cookies!

My family, and in particular my parents and grandparents, deserve a special thank-you as they always remained very supportive of my endeavours no matter what country I decided to move to next. I will stay in Europe (at least for the next couple of years), I promise! Despite me having to miss numerous family festivities in the past two years, they never got irritated. I hope I can make it up to them.

Friends in various regions of Germany, Britain and the Netherlands also remained curious about my progress here. They kept me distracted at times when I needed it, especially Lena, Jan, Anett, Christian, Stefan, Kodo, Cath, Alistair and Gordon.

The person that certainly had to suffer the most through my PhD-related ups and downs was Stelios, who always managed to convince me that all of this is worth it and that the outcome will be a positive one.

The work reported in this thesis was supported by the BSIK-program MultimediaN which I would also like to thank for funding my research and providing a platform for interaction with other researchers.

Claudia

# Contents

# Chapter 1

# Introduction

The ability to make accurate predictions of the outcome of an event or a process is highly desirable in several contexts of human activity. For instance, when considering financial benefit, a very desirable prediction would be that of the successful anticipation of numbers appearing in an upcoming lottery. A successful attempt, in this context, can only be labeled a lucky guess, as previous lottery results or other factors such as the number of lottery tickets sold have no impact on the outcome (assuming a fair draw). Similarly, in terms of financial gain, a prediction on the stock market's behavior would also be very desirable. However, in this case, as opposed to lottery draws, the outcome can be predicted to some extent based on the available historical data and current economical and political events [28, 86, 172]. Notably, in both previous instances a rational agent may be highly motivated by the prospect of financial gain to make a successful guess on a future outcome but only in the latter are predictions to some extent possible.

In this work, the investigation theme is that of predictions and the factors that allow a measurable and consistent degree of success in anticipating a certain outcome. Specifically, two types of predictions in the context of information retrieval are set in focus. First, we consider users' attempts to express their information needs through queries, or search requests and try to predict whether those requests will be of high or low quality. Intuitively, the query's quality is determined by the outcome of the query, that is, whether the results meet the user's expectations. Depending on the predicted outcome, action can be taken by the search system in view of improving overall user satisfaction. The second type of predictions under investigation are those which attempt to predict the quality of search systems themselves. So, given a number of search systems to consider, these predictive methods attempt to estimate how well or how poorly they will perform in comparison to each other.

## 1.1 Motivation

Predicting the quality of a query is a worthwhile and important research endeavor, as evidenced by the significant amount of related research activity in recent years. Notably, if a technique allows for a quality estimate of queries in advance of, or

during the retrieval stage, specific measures can be taken to improve the overall performance of the system. For instance, if the performance of a query is considered to be poor, remedial action by the system can ensure that the users' information needs are satisfied by alerting them to the unsuitability of the query and asking for refinement or by providing a number of different term expansion possibilities.

An intuition of the above may be provided with the following simple example. Consider the query "jaguar", which, given a general corpus like the World Wide Web, is substantially ambiguous. If a user issues this query to a search engine such as A9[1], Yahoo![2] or Google[3], it is not possible for the search engine to determine the user's information need without knowledge of the user's search history or profile. So only a random guess may be attempted on whether the user expects search results on the Jaguar car, the animal, the Atari video console, the guitar, the football team or even Apple's Mac OS X 10.2 (also referred to as Jaguar). When submitting the query "jaguar" to the Yahoo! search engine on August 31, 2009, of the ten top ranked returned results, seven were about Jaguar cars and three about the animal. In fact, most results that followed up to rank 500 also dealt with Jaguar cars; the first result concerning Mac OS X 10.2 could be found at rank 409. So a user needing information on the operating system would likely have been unable to acquire it. It is important to note that an algorithm predicting the extent of this ambiguity could have pointed out the unsuitability of the query and suggested additional terms for the user to choose from as some search engines do.

A query predicted to perform poorly, such as the one above, may not necessarily be ambiguous but may just not be covered in the corpus to which it is submitted [31, 167]. Also, identifying difficult queries related to a particular topic can be a valuable asset for collection keepers who can determine what kind of documents are expected by users and missing in the collection. Another important factor for collection keepers is the findability of documents, that is how easy is it for searchers to retrieve documents of interest [10, 30].

Predictions are also important in the case of well-performing queries. When deriving search results from different search engines and corpora, the predictions of the query with respect to each corpus can be used to select the best corpus or to merge the results across all corpora with weights according to the predicted query effectiveness score [160, 167]. Also, consider that the cost of searching can be decreased given a multiple partitioned corpus, as is common practice for very large corpora. If the documents are partitioned by, for instance, language or by topic, predicting to which partition to send the query saves time and bandwidth, as not all partitions need to be searched [12, 51]. Moreover, should the performance of a query appear to be sufficiently good, the query can be improved by some affirmative action such as automatic query expansion with pseudo-relevance feedback. In pseudo-relevance feedback it is assumed that the top $K$ retrieved documents are relevant and so for a query with low effectiveness most or all of the top $K$ documents would be irrelevant. Notably, expanding a poorly performing query leads to

[1] http://www.a9.com/
[2] http://www.yahoo.com/
[3] http://www.google.com/

query drift and possibly to an even lower effectiveness while expanding queries with a reasonable performance and thus a number of relevant documents among the top $K$ retrieved documents is more likely to lead to a gain in effectiveness. Another recently proposed application of prediction methods is to shorten long queries by filtering out predicted extraneous terms [94], in view of improving their effectiveness.

Cost considerations are also the prevalent driving force behind the research in predicting the ranking of retrieval systems according to their retrieval effectiveness without relying on manually derived relevance judgments. The creation of test collections, coupled with more and larger collections becoming available, can be very expensive. Consider that in a typical benchmark setting, the number of documents to judge depends on the number of retrieval systems participating. In the data sets used throughout this thesis, for example, the number of documents judged varies between a minimum of $31984$ documents and a maximum of $86830$ documents. If we assume that a document can be judged for its relevance within $30$ seconds [150], this means that between $267$ and $724$ assessor hours are necessary to create the relevance judgments of one data set – a substantial amount.

Moreover, in a dynamic environment such as the World Wide Web, where the collection and user search behavior change over time, regular evaluation of search engines with manual assessments is not feasible [133]. If it were possible, however, to determine the relative effectiveness of a set of retrieval systems, reliably and accurately, without the need for relevance judgments, the cost of evaluation would be greatly reduced.

Correctly identifying the ranking of retrieval systems can also be advantageous in a more practical setting when relying on different retrieval approaches (such as Okapi [125] and Language Modeling [121]) and a single corpus. Intuitively, different types of queries benefit from different retrieval approaches. If it is possible to predict which of the available retrieval approaches will perform well for a particular query, the best predicted retrieval strategy can then be selected. Overall, this would lead to an improvement in effectiveness.

The motivation for this work is to improve user satisfaction in retrieval, by enabling the automatic identification of well performing retrieval systems as well as allowing retrieval systems to identify queries as either performing well or poorly and reacting accordingly. This thesis includes a thorough evaluation of existing prediction methods in the literature and proposes an honest appraisal of their effectiveness. We carefully enumerate the limitations of contemporary work in this field, propose enhancements to existing proposals and clearly outline their scope of use. Ultimately, there is considerable scope for improvement in existing retrieval systems if predictive methods are evaluated in a consistent and objective manner; this work, we believe, contributes substantially in accomplishing this goal.

## 1.2 Prediction Aspects

Evaluation of new ideas, such as new retrieval approaches, improved algorithms of pseudo-relevance feedback and others, is of great importance in information retrieval research. The development of a new model is not useful if the model does not substantially reflect reality and does not lead to improved results in a practical setting. For this reason, there are a number of yearly benchmarking events, where different retrieval tasks are used to compare retrieval models and approaches on common data sets. In settings such as TREC[4], TRECVID[5], FIRE[6], INEX[7], CLEF[8], and NTCIR[9], a set of topics $t_1$ to $t_m$ is released for different tasks, and the participating research groups submit runs, that are ranked lists of results for each topic $t_i$ produced by their retrieval systems $s_1$ to $s_n$. The performance of each system is determined by so-called relevance judgments, that is manually created judgments of results that determine a result's relevance or irrelevance to the topic. The retrieval tasks and corpora are manifold - they include the classic adhoc task [62], the entry page finding task [89], question answering [147], entity ranking [47] and others on corpora of text documents, images and videos.

The results returned thus depend on the task and corpus - a valid result might be a text document, a passage or paragraph of text, an image or a short video sequence. In this thesis, we restrict ourselves to collections of text documents and mostly the classical adhoc task.

For each pairing $(t_i, s_j)$ of topic and system, one can determine a retrieval effectiveness value $e_{ij}$, which can be a measure such as average precision, precision at 10 documents, reciprocal rank and others [13]. The decision as to which measure to use is task dependent. This setup can be represented by an $m \times n$ matrix as shown in Figure 1.1. When relevance judgments are *not* available, it is evident from Figure 1.1 that the performances of four different aspects can be predicted. In previous work, all four aspects have been investigated by various authors and are outlined below. We also include in the list a fifth aspect, which can be considered as an aggregate of evaluation aspects EA1 to EA4.

**(EA1) How difficult is a topic in general?** Given a set of $m$ topics and a corpus of documents, the goal is to predict the retrieval effectiveness or difficulty ranking of the topics *independent* of a particular retrieval system [7, 30], thus the topics are evaluated for their inherent difficulty with respect to the corpus.

[4] Text REtrieval Conference (TREC), http://trec.nist.gov/

[5] TREC Video Retrieval Evaluation (TRECVID), http://trecvid.nist.gov/

[6] Forum for Information Retrieval Evaluation (FIRE), http://www.isical.ac.in/~fire/

[7] INitiative for the Evaluation of XML Retrieval (INEX), http://www.inex.otago.ac.nz/

[8] Cross Language Evaluation Forum (CLEF), http://clef.iei.pi.cnr.it/

[9] NII Test Collection for Information Retrieval Systems (NTCIR), http://research.nii.ac.jp/ntcir/

**(EA2) How difficult is a topic for a particular system?** Given a set of $m$ topics, a retrieval system $s_j$ and a corpus of documents, the aim is to estimate the effectiveness of the topics given $s_j$. A topic with low effectiveness is considered to be difficult for the system. This is the most common evaluation strategy which has been investigated for instance in [45, 71, 167, 175].

**(EA3) How well does a system perform for a particular topic?** Given a topic $t_i$, $n$ retrieval systems and a corpus of documents, the systems are ranked according to their performance on $t_i$. This approach is somewhat similar to aspect EA4, although here, the evaluation is performed on a per topic basis rather than across a set of topics [50].

**(EA4) How well does a system perform in general?** Given a set of $n$ retrieval systems and a corpus of documents, the aim is to estimate a performance ranking of systems independent of a particular topic [9, 114, 133, 135, 161].

**(EA5) How hard is this benchmark for all systems participating?** This evaluation aspect can be considered as an aggregate of the evaluation aspects EA1 to EA4.

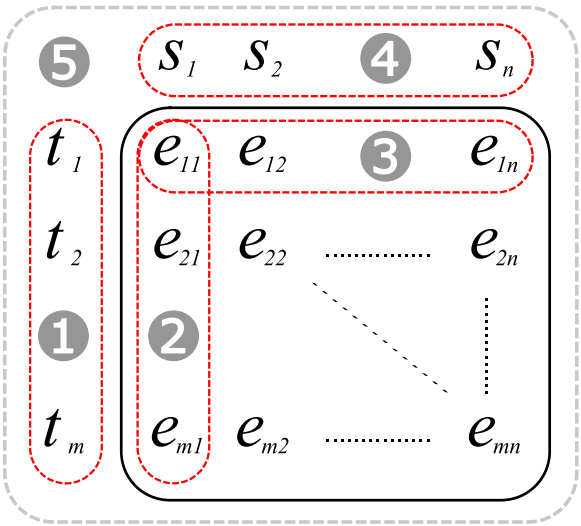



Figure 1.1: A matrix of retrieval effectiveness values; $e_{ij}$ is the retrieval effectiveness, system $s_j$ achieves for topic $t_i$ on a particular corpus. The numbered labels refer to the different aspects (label 1 corresponds to EA1, etcetera).

A topic is an expression of an information need – in the benchmark setting of TREC it usually consists of a title, a description and a narrative. A query is a formulation of the topic that is used in a particular retrieval system. Often, only the title part of the topic is used and a formulation is derived by, for instance, stemming and stopword removal or the combination of query terms by Boolean operators. As the input to a retrieval system is the formulation of an information need, that is a query, this concept is often expressed as query performance or query effectiveness prediction.

Search engines, even if they perform well on average, suffer from a great variance in retrieval effectiveness [63, 151, 152], that is, not all queries can be answered with equal accuracy. Predicting whether a query will lead to low quality results is a challenging task, even for information retrieval experts. In an experiment described by Voorhees and Harman [148], a number of researchers were asked to classify a set of queries as either easy, medium or difficult for a corpus of newswire articles they were familiar with. The researchers were not given ranked lists of results, just the queries themselves. It was found that the experts were unable to predict the query types correctly and, somewhat surprisingly, they could not even agree among themselves how to classify the queries[10]. Inspired by the aforementioned experiment, we performed a similar one with Web queries and a Web corpus; specifically, we relied on the newly released ClueWeb09 corpus and the 50 queries of the TREC 2009 Web adhoc task. Nowadays (as opposed to the late 90's time frame of [148]), Web search engines are used daily, and instead of information retrieval experts, we relied on members of the Database and the Human Media Interaction group of the University of Twente, who could be considered expert users. Thirty-three people were recruited and asked to judge each of the provided queries for their expected result quality. The users were asked to choose for each query one of four levels of quality: *low*, *medium* and *high* quality as well as *unknown*. The quality score of each query is derived by summing up the scores across all users that did not choose the option *unknown* where the scores $1$, $2$ and $3$ are assigned to *low*, *medium* and *high* quality respectively. Note that a higher quality score denotes a greater expectation by the users that the query will perform well on a Web search engine. Named entities such as "volvo" and "orange county convention center" as well as seemingly concrete search request such as "wedding budget calculator" received the highest scores. The lowest scores were given to unspecific queries such as "map" and "the current". The correlation between the averaged quality scores of the users and the retrieval effectiveness scores of the queries evaluated to $r = 0.46$. This moderate correlation indicates, that users can, to some extent, predict the quality of search results, though not with a very high accuracy, which denotes the difficulty of the task[11].

In recent years many different kinds of predictions in information retrieval have been investigated. This includes, for instance, the prediction of a Web summary's quality [83] and of a Q&A pair's answer quality [22, 81], as well as the prediction of the usefulness of involving the user or user profile in query expansion [92, 93, 137]. Further examples in this vain include predicting the effect of labeling images [85], predictions on the amount of external feedback [53] and predicting clicks on Web advertisements [18] and news results [88].

In this thesis we focus specifically on predicting the effectiveness of informational queries and retrieval systems, as we believe that these two aspects will bring about the most tangible benefits in retrieval effectiveness and in improvement of user satisfaction, considering that around $80\%$ of the queries submitted to the Web are

[10] The highest linear correlation coefficient between an expert's predictions and the ground truth was $r = 0.26$, the highest correlation between any two experts' predictions was $r = 0.39$.

[11] The user study is described in more detail in Appendix A.

informational in nature [78]. Furthermore, the works cited above depend partially on large-scale samples of query logs or interaction logs [92, 137] which cannot be assumed to be available to all search systems.

## 1.3 Definition of Terms

As of yet there is no widely accepted standard terminology in this research area. Depending on the publication forum and the particular author different phrases are used to refer to specific evaluation aspects. As can be expected the same term can also have different meanings in works by different authors. In this section, we explicitly state our interpretation of ambiguous terms in the literature and we use them consistently throughout.

Firstly, while generally - and also in this thesis - *to predict* and *to estimate* a query's quality are used interchangeably, in a number of works in the literature a distinction is made between the two. *Predicting* the quality of a query is used when the algorithms do *not* rely on the ranked lists of results, while the quality of a query is *estimated* if the calculations are based on the ranked list of results. In this work the meaning becomes clear in the context of each topic under investigation, it is always explicitly stated whether a ranked list of results used.

Throughout, the term *query quality* means the *retrieval effectiveness* that a query achieves with respect to a particular retrieval system, which is also referred to as *query performance*. When we investigate *query difficulty* we are indirectly also interested in predicting the retrieval effectiveness of a query, however we are only interested whether the effectiveness will be low or high. We thus expect a binary outcome - the query is either classified as easy or it is classified as difficult. In contrast, when investigating *query performance prediction* we are interested in the predicted effectiveness score and thus expect a non-binary outcome such as an estimate of average precision.

| | |
|---|---|
| **EA1** | collection query hardness [7], topic difficulty [30] |
| **EA2** | query difficulty [167], topic difficulty [30], query performance prediction [175], precision prediction [52], system query hardness [7], search result quality estimation [40], search effectiveness estimation [145] |
| **EA3** | performance prediction of "retrievals" [50] |
| **EA4** | automatic evaluation of retrieval systems [114], ranking retrieval systems without relevance judgments [133], retrieval system ranking estimation |
| **EA5** | - |

Table 1.1: Overview of commonly used terminology of the evaluation aspects of Figure 1.1.

In Table 1.1 we have summarized the expressions for each evaluation aspect as they occur in the literature. Evaluation aspect EA2 has the most diverse set of labels, as it is the most widely evaluated aspect. Most commonly, it is referred to as *query*

*performance prediction*, *query difficulty* as well as *query effectiveness estimation*. Evaluation aspect EA3 on the other hand has only been considered in one publication so far [50], where the pair $(t_i, s_j)$ of topic and system is referred to as "retrieval".

Aspect EA4 of Figure 1.1 was originally referred to as *ranking retrieval systems without relevance judgments*, but has also come to be known as *automatic evaluation of retrieval systems*. We refer to it as *retrieval system ranking estimation* as in this setup we attempt to estimate a ranking of retrieval systems.

## 1.4 Research Themes

The analysis presented in this thesis aims to provide a comprehensive picture of research efforts in the prediction of query and retrieval system effectiveness. By organizing previous works according to evaluation aspects we methodically clarify and categorise the different dimensions of this research area. The thesis focuses on two evaluation aspects (enumerated in full in Section 1.2), in particular, EA2 and EA4, as their analysis has value in practical settings as well as for evaluation purposes.

The other aspects are not directly considered. Evaluation aspect EA1, which assumes the difficulty of a topic to be inherent to a corpus, is mainly of interest in the creation of benchmarks, so as to, for instance, choose the right set of topics. In an adaptive retrieval system, where the system cannot choose which queries to answer it is less useful. The same argumentation applies intuitively to aspect EA5. As part of the work on evaluation aspect EA4 in Chapter 5 we will briefly discuss EA3.

Four main research themes are covered in this work and will now be explicitly stated. The first three (**RT1**, **RT2** and **RT3**) are concerned with evaluation aspect EA2, while the last one (**RT4**) is concerned with evaluation aspect EA4 (and partly with EA3). The backbone of all results reported and observations made in this work form two large-scale empirical studies. In Chapter 2 and Chapter 3 a total of twenty-eight prediction methods are evaluated on three different test corpora. The second study, discussed in detail in Chapter 5 puts emphasis on the influence of the diversity of data sets: therein five system ranking estimation approaches are evaluated on sixteen highly diverse data sets.

**RT1: Quality of pre-retrieval predictors** Pre-retrieval query effectiveness prediction methods are so termed because they predict a query's performance before the retrieval step. They are thus independent of the ranked list of results. Such predictors base their predictions solely on query terms, the collection statistics and possibly external sources such as WordNet [57] or Wikipedia[12]. In this work we analyze and evaluate a large subset of the main approaches and answer the following questions: on what heuristics are the prediction algorithms based? Can the algorithms be categorized in a meaningful way? How similar are different approaches to

[12] http://www.wikipedia.org/

each other? How sensitive are the algorithms to a change in the retrieval approach? What gain can be achieved by combining different approaches?

**RT2: The case of the post-retrieval predictor Clarity Score** The class of post-retrieval approaches estimates a query's effectiveness based on the ranked list of results. The approaches in this class are usually more complex than pre-retrieval predictors, as more information (the list of results) is available to form an estimate of the query's effectiveness. Focusing on one characteristic approach, namely Clarity Score [45], the questions we explore are: how sensitive is this post-retrieval predictor to the retrieval algorithm? How does the algorithm's performance change over different test collections? Is it possible to improve upon the prediction accuracy of existing approaches?

**RT3: The relationship between correlation and application** The quality of query effectiveness prediction methods is commonly evaluated by reporting correlation coefficients, such as Kendall's Tau [84] and the linear correlation coefficient. These measures denote how well the methods perform at predicting the retrieval performance of a given set of queries. The following essential questions have so far remained unexplored: what is the relationship between the correlation coefficient as an evaluation measure for query performance prediction and the effect of such a method on retrieval effectiveness? At what levels of correlation can we be reasonably sure that a query performance prediction method will be useful in a practical setting?

**RT4: System ranking estimation** Substantial research work has also been undertaken in estimating the effectiveness of retrieval systems. However, most of the evaluations have been performed on a small number of older corpora. Current work in this area lacks a broad evaluation scope which gives rise to the following questions: is the performance of system ranking estimation approaches as reported in previous studies comparable with their performance on more recent and diverse data sets? What factors influence the accuracy of system ranking estimation? Can the accuracy be improved when selecting a subset of topics to rank retrieval systems?

## 1.5 Thesis Overview

The organization of the thesis follows the order of the research themes. In Chapter 2, we turn our attention to pre-retrieval prediction algorithms and provide a comprehensive overview of existing methods. We examine their similarities and differences analytically and then verify our findings empirically. A categorization of algorithms is proposed and the change in predictor performance when combining different approaches is investigated. The major results of this chapter have previously been published in [65, 69].

In Chapter 3, post-retrieval approaches are introduced, with a focus on Clarity Score for which an improved variation is proposed and an explanation is offered as to why some test collections are more difficult for query effectiveness estimation than others. Part of this work can also be found in [70].

The connection between a common evaluation measure (Kendall's Tau) in query performance prediction approaches and the performance of retrieval systems relying on those predictions is evaluated in Chapter 4. Insights in the level of correlation required in order to ensure that an application of a predictor in an operational setting is likely to lead to an overall improvement in the retrieval system are reported. Parts of this chapter have been described in [64, 67].

Chapter 5 then focuses on system ranking estimation approaches. A number of algorithms are compared and the hypothesis that subsets of topics lead to a better performance of the approaches is evaluated. The work of this chapter was initially presented in [66, 68].

The thesis concludes with Chapter 6 where a summary of the conclusions is included and suggestions for future research are offered.

# Chapter 2

# Pre-Retrieval Predictors

## 2.1 Introduction

Pre-retrieval prediction algorithms predict the effectiveness of a query before the retrieval stage is reached and are, thus, independent of the ranked list of results; essentially, they are search-independent. Such methods base their predictions solely on query terms, the collection statistics and possibly an external source such as WordNet [57], which provides information on the query terms' semantic relationships. Since pre-retrieval predictors rely on information that is available at indexing time, they can be calculated more efficiently than methods relying on the result list, causing less overhead to the search system. In this chapter we provide a comprehensive overview of pre-retrieval query performance prediction methods.

Specifically, this chapter contains the following contributions:

- the introduction of a predictor taxonomy and a clarification of evaluation goals and evaluation measures,
- an analytical and empirical evaluation of a wide range of prediction methods over a range of corpora and retrieval approaches, and,
- an investigation into the utility of combining different prediction methods in a principled way.

The organization of the chapter is set up accordingly. First, in Section 2.2 we present our predictor taxonomy, then in Section 2.3 we discuss the goals of query effectiveness prediction and subsequently lay out what evaluations exist and when they are applicable. A brief overview of the notation and the data sets used in the evaluations (Sections 2.4 and 2.5) follows. In Sections 2.6, 2.7, 2.8 and 2.9 we cover the four different classes of predictor heuristics. While these sections give a very detailed view on each method, in Section 2.10 we discuss the results of an evaluation that has so far been neglected in query performance prediction: the evaluation whether two predictors perform differently from each other in a statistically significant way. How diverse retrieval approaches influence the quality of various prediction methods is evaluated in Section 2.11. A final matter of investigation is the utility of combining prediction methods in a principled way, which is described in Section 2.12. The conclusions in Section 2.13 round off the chapter.

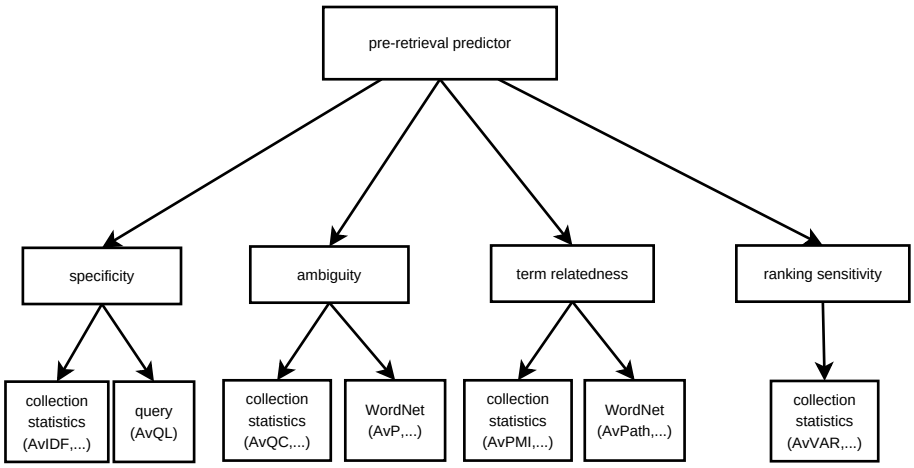



Figure 2.1: Categories of pre-retrieval predictors.

## 2.2 A Pre-Retrieval Predictor Taxonomy

In general, pre-retrieval predictors can be divided into four different groups according to the heuristics they exploit (Figure 2.1). First, *specificity* based predictors predict a query to perform better with increased specificity. How the specificity is determined further divides these predictors into collection statistics based and query based predictors.

Other predictors exploit the query terms' *ambiguity* to predict the query's quality; in those cases, high ambiguity is likely to result in poor performance. In such a scheme, if a term always appears in the same or similar contexts, the term is considered to be unambiguous. However, if the term appears in many different contexts it is considered to be ambiguous. For instance, consider that the term "tennis" will mainly appear in the context of sports and will rarely be mentioned in documents discussing finances or politics. The term "field", however, is more ambiguous and can easily occur in sports articles, agriculture articles or even politics (e.g. "field of Democratic candidates"). Intuitively, ambiguity is somewhat related to specificity, as an ambiguous term can have a high document frequency, but there are exceptions - consider that the term "tennis" might not be specific in a corpus containing many sports-related documents, but it is unambiguous and while specificity based predictors would predict it to be a poor query, ambiguity based predictors would not. The ambiguity of a term may be derived from collection statistics, additionally it can also be determined by relying on an external source such as WordNet.

The drawback of predictors in the first two categories (specificity and ambiguity) stems from their lack of consideration of the relationship between terms. To illustrate this point, consider that the query "political field" is actually unambiguous due to the relationship between the two terms, but an ambiguity based predictor is likely to predict a poor effectiveness, since "field" can appear in many contexts. Similarly for a specificity based predictor, the term "field" will likely occur often in a general corpus. To offset this weakness, a third category of predictors makes use of *term relatedness* in an attempt to exploit the relationship between query terms.

Specifically, if there is a strong relationship between terms, the query is predicted to be of good quality.

Finally, the *ranking sensitivity* can also be utilized as source of information for a query's effectiveness. In such a case, a query is predicted to be ineffective, if the documents containing the query terms appear similar to each other, making them indistinguishable for a retrieval system and thus difficult to rank. In contrast to post-retrieval methods, which work directly on the rankings produced by the retrieval algorithms, these predictors attempt to predict how easy it is to return a stable ranking. Moreover, they rely exclusively on collection statistics, and more specifically the distribution of query terms within the corpus.

## 2.3 Evaluation Framework

Query effectiveness prediction methods are usually evaluated by reporting the correlation they achieve with the ground truth, which is the effectiveness of queries derived for a retrieval approach with the help of relevance judgments. The commonly reported correlations coefficients are Kendall's Tau $\tau$ [84], Spearman's Rho $\rho$, and the linear correlation coefficient $r$ (also known as Pearson's $r$). In general, the choice of correlation coefficient should depend on the goals of the prediction algorithm. As often prediction algorithms are evaluated but not applied in practice, a mix of correlation coefficients is usually reported as will become evident in Chapter 3 (in particular in Table 3.1).

### 2.3.1 Evaluation Goals

Evaluation goals can be for instance the determination whether a query can be answered by a corpus or an estimation of the retrieval effectiveness of a query. We now present three categories of evaluation goals which apply both to pre- and post-retrieval algorithms.

#### Query Difficulty

The query difficulty criterion can be defined as follows: given a query $\mathbf{q}$, a corpus of documents $C$, external knowledge sources $E$ and a ranking function $R$ (which returns a ranked list of documents), we can estimate whether $\mathbf{q}$ is difficult as follows:

$$f_{diff}(\mathbf{q}, C, E, R) \rightarrow \{0, 1\}. \tag{2.1}$$

Here, $f_{diff} = 0$ is an indicator of the class of difficult queries which exhibit unsatisfactory retrieval effectiveness and $f_{diff} = 1$ represents the class of well performing queries. When $R = \emptyset$ we are dealing with pre-retrieval prediction methods. A number of algorithms involve external sources $E$ such as Wikipedia or WordNet. The majority of methods however, rely on $C$ and $R$ only. Evaluation measures that are

in particular applicable to $f_{diff}$ emphasize the correct identification of the worst performing queries and largely ignore the particular performance ranking and the best performing queries [145, 151].

#### Query Performance

Determining whether a query will perform well or poorly is not always sufficient. Consider for example a number of alternative query formulations for an information need. In order to select the best performing query, a more general approach is needed; such an approach is *query performance prediction*. Using the notation above, we express this as follows:

$$f_{perf}(\mathbf{q}, C, E, R) \rightarrow \mathbb{R} \tag{2.2}$$

The query with the largest score according to $f_{perf}$ is deemed to be the best formulation of the information need. In this scenario, we are not interested in the particular scores, but in correctly ranking the queries according to their predicted effectiveness. In such a setup, evaluating the agreement between the predicted query ranking and the actual query effectiveness ranking is a sound evaluation strategy. The alignment of these two rankings is usually reported in terms of rank correlation coefficients such as Kendall's $\tau$ and Spearman's $\rho$.

#### Normalized Query Performance

In a number of instances, absolute estimation scores as returned by $f_{perf}$ cannot be utilized to locate the best query from a pool of queries. Consider a query being submitted to a number of collections and the ranked list that is estimated to best fit the query is to be selected, or alternatively the ranked lists are to be merged with weights according to the estimated query quality. Absolute scores as given by $f_{perf}$ will fail, as they usually depend on collection statistics and are, thus, not comparable across corpora. The evaluation should thus emphasize, how well the algorithms estimate the effectiveness of a query according to a particular effectiveness measure such as average precision. Again, using the usual notation:

$$f_{norm}(\mathbf{q}, C, E, R) \rightarrow [0, 1]. \tag{2.3}$$

By estimating a *normalized* score, scores can be compared across different collections. The standard evaluation measure in this setting is the linear correlation coefficient $r$.

### 2.3.2 Evaluation Measures

As described in the previous section, different evaluation methodologies are applicable to different application scenarios. The standard correlation based approach to evaluation is as follows. Let $Q$ be the set of queries $\{\mathbf{q}\}^i$ and let $R_{\mathbf{q_i}}$ be the ranked list returned by the ranking function $R$ for $\mathbf{q_i}$. For each $\mathbf{q_i} \in Q$, the predicted score $s_i$ is obtained from a given predictor; additionally the retrieval effectiveness of $R$ is

determined (based on the relevance judgments). Commonly, the average precision $ap_i$ of $R_{\mathbf{q_i}}$ is calculated as ground truth effectiveness. Then, given all pairs $(s_i, ap_i)$, the correlation coefficient is determined.

### Ranking Based Approaches

Rank correlations make no assumptions about the type of relationship between the two lists of scores (predictor scores and retrieval effectiveness scores). Both score lists are converted to lists of ranks where the highest score is assigned rank $1$ and so on. Then, the correlation of the ranks is measured. In this case, the ranks give an indication of each query's effectiveness relative to the other queries in the list but no quantitative prediction is made about the retrieval score of the query.

The TREC Robust Retrieval track [151, 152], where query effectiveness prediction was first proposed as part of the adhoc retrieval task, aimed at distinguishing the poorly performing queries from the successful ones. The participants were asked to rank the given set of queries according to their estimated performance. As measure of agreement between the predicted ranking and the actual ranking, Kendall's $\tau$ was proposed.

A common approach to comparing two predictors is to compare their point estimates and to view a higher correlation coefficient as proof of a better predictor method. However, to be able to say with confidence that one predictor outperforms another, it is necessary to perform a test of statistical significance of the difference between the two [39]. Additionally, we can give an indication of how confident we are in the result by providing the confidence interval (CI) of the correlation coefficient. Currently, predictors are only tested for their significance against a correlation of zero.

While Kendall's $\tau$ is suitable for the setup given by $f_{pref}$, it is sensitive to all differences in ranking. If we are only interested in identifying the poorly performing queries ($f_{diff}$), ranking differences at the top of the ranking are of no importance and can be ignored. The *area between the MAP curves*, proposed by Voorhees [151], is an evaluation measure for this scenario. The mean average precision (MAP) is computed over the best performing $b$ queries and $b$ ranges from the full query set to successively fewer queries, leading to a MAP curve. Two such MAP curves are generated: one based on the actual ranking of queries according to retrieval effectiveness and one based on the predicted ranking of queries. If the predicted ranking conforms to the actual ranking, the two curves are identical and the area between the curves is zero. The more the predicted ranking deviates from the actual ranking, the more the two curves will diverge and, thus, the larger the area between them. It follows, that the larger the area between the curves, the worse the accuracy of the predictor. A simpler evaluation measure that is also geared towards query difficulty, was proposed by Vinay et al. [145]. Here, the bottom ranked $10\%$ or $20\%$ of predicted and actual queries are compared and the overlap is computed; the larger the overlap, the better the predictor.

### Linear Correlation Coefficient

Ranking based approaches are not suitable to evaluate the scenario $f_{norm}$, as they disregard differences between the particular predicted and actual scores. In such a case, the linear correlation coefficient $r$ can be used instead. This coefficient is defined as the covariance, normalized by the product of the standard deviations of the predicted scores and the actual scores.

The value of $r^2$ is known as the *coefficient of determination*. A number of tests for the significance of difference between overlapping correlations have been proposed in the literature [77, 103, 158]. In our evaluation, we employed the test proposed by Meng et al. [103].

In the case of multiple linear regression, $r$ increases due to the increase in regressors. To account for that, the value of the *adjusted* $r^2$ can be reported, which takes the number $p$ of regressors into account ($n$ is the sample size):

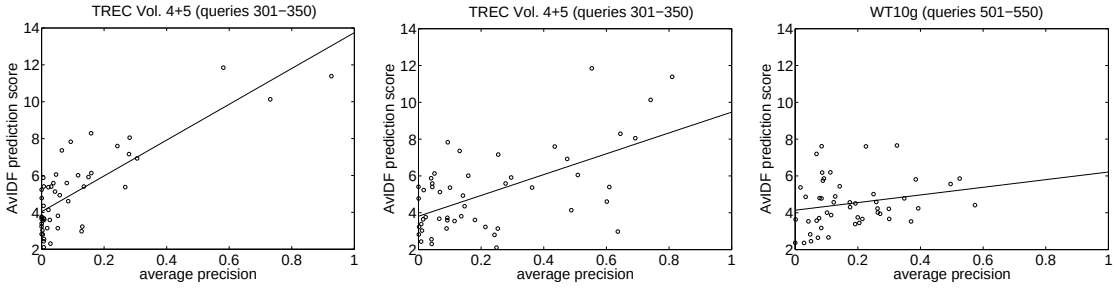
$$r^2_{adj} = 1 - (1 - r^2)\frac{n-1}{n-p-1}. \tag{2.4}$$

### Limitations of Correlation Coefficients

Correlation coefficients compress a considerable amount of information into a single number, which can lead to problems of interpretation. To illustrate this point, consider the cases depicted in Figure 2.2 for the linear correlation coefficient $r$ and Kendall's $\tau$. Each point represents a query with its corresponding retrieval effectiveness value (given in average precision) on the x-axis and its predicted score on the y-axis. The three plots are examples of high, moderate and low correlation coefficients; for the sake of $r$, the best linear fit is also shown. Further, the MAP as average measure of retrieval effectiveness is provided as well. These plots are derived from data that reflects existing predictors and retrieval approaches. In the case of Figure 2.2a, the predictor scores are plotted against a very basic retrieval approach with a low retrieval effectiveness (MAP of $0.11$). The high correlations of $r = 0.81$ and $\tau = 0.48$ respectively highlight a possible problem: the correlation coefficient of a predictor can be improved by correlating the prediction scores with the "right" retrieval method instead of improving the quality of the prediction method itself.

To aid understanding, consider Figures 2.2b and 2.2c, which were generated from the same predictor for different query sets and a better performing retrieval approach. They show the difference between a medium and a low correlation. Note that, in general, the value of Kendall's $\tau$ is lower than $r$, but the trend is similar.

In Section 2.12, we evaluate the utility of combining predictors in a principled way. The evaluation is performed according to $f_{norm}$, which is usually reported in the literature in terms of $r$. However, when combining predictors, a drawback of $r$ is the increase in correlation if multiple predictors are linearly combined. Independent of the quality of the predictors, $r$ increases as more predictors are added to the model. An extreme example of this, is shown in Figure 2.3 where the average precision scores of a query set were correlated with randomly generated predictors numbering between $1$ and $75$. Note that at $75$ predictors, $r > 0.9$. Figure 2.3 also contains




(a) *MAP* $= 0.11, r = 0.81$, $\tau = 0.48$ (b) *MAP* $= 0.23$, $r = 0.59$, $\tau = 0.32$ (c) *MAP* $= 0.18, r = 0.22$, $\tau = 0.18$

Figure 2.2: Scatter plots of retrieval effectiveness scores versus predicted scores.

the trend of $r_{adj}$, which takes the number of predictors in the model into account, but despite this adaptation we observe $r_{adj} > 0.6$ when $75$ random predictors are combined.

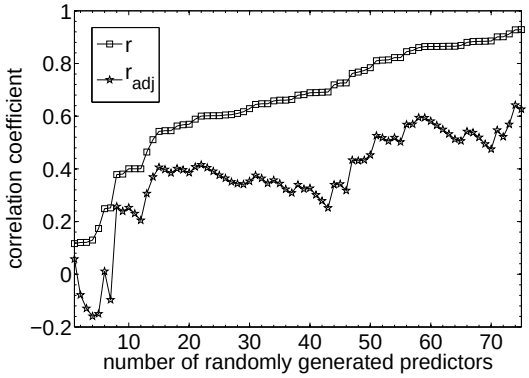



Figure 2.3: Development of $r$ and $r_{adj}$ with increasing number of random predictors.

## 2.4 Notation

We now briefly present definitions and notations as used for the remainder of this chapter. A query $\mathbf{q} = \{q_1, q_2, .., q_m\}$ is composed of query terms $q_i$ and has length $|\mathbf{q}| = m$. A term $t_i$ occurs $tf(t_i, d_j)$ times in document $d_j$. Further, a term $t_i$ occurs $tf(t_i) = \sum_j tf(t_i, d_j)$ times in the collection and in $df(t_i)$ documents. The document length $|d_j|$ is equal to the number of terms in the document. The total number of terms in the collection is denoted by $termcount$ and $doccount$ marks the total number of documents. $N_{\mathbf{q}}$ is the set of all documents containing at least one of the query terms in $\mathbf{q}$.

The maximum likelihood estimate of term $t_i$ in document $d_j$ is given by

$$P_{ml}(t_i|d_j) = \frac{tf(t_i, d_j)}{|d_j|}. \tag{2.5}$$

The probability of term $t_i$ occurring in the collection is $P_{ml}(t_i) = \frac{tf(t_i)}{termcount}$. Finally, $P_{ml}(t_i, t_j)$ is the maximum likelihood probability of $t_i$ and $t_j$ occurring in the same document.

## 2.5 Materials and Methods

The evaluations of the methods outlined in the current and the following chapters are performed on a range of query sets and corpora. In this section, we briefly describe the corpora, the query sets and the retrieval approaches utilized. A more comprehensive overview of the data sets can be found in Appendix B.

### 2.5.1 Test Corpora

To perform the experiments, the adhoc retrieval task is evaluated on three different TREC corpora, namely, TREC Volumes 4 and 5 minus the Congressional Records [148] (TREC Vol. 4+5), WT10g [132] and GOV2 [38]. The corpora differ in size as well as content. TREC Vol. 4+5 is the smallest, containing newswire articles, WT10g is derived from a crawl of the Web and GOV2, the largest corpus with more than 25 million documents, was created from a crawl of the .gov domain. The corpora were stemmed with the Krovetz stemmer [90] and stopwords were removed[1]. All experiments in this thesis are performed with the Lemur Toolkit for Language Modeling and Information Retrieval[2], version 4.3.2.

The queries are derived from the TREC title topics of the adhoc tasks, available for each corpus. We focus on title topics as we consider them to be more realistic than the longer description and narrative components of a TREC topic. Please note again, that we distinguish the concepts of *topic* and *query*: whereas a topic is a textual expression of an information need, we consider a query to be the string of characters that is submitted to the retrieval system. In our experiments we turn a TREC title topic into a query by removing stopwords and applying the Krovetz stemmer.

Table 2.1 contains the list of query sets under consideration, the corpus they belong to and the average number of query terms. In query set 451-500, we manually identified and corrected three spelling errors. Our focus is on investigating query effectiveness predictors and we assume the ideal case of error-free queries. In practical applications, spelling error correction would be a preprocessing step.

### 2.5.2 Retrieval Approaches

The goal of prediction algorithms is to predict the (relative) retrieval effectiveness of a query as well as possible. Since there are many retrieval approaches with various degrees of retrieval effectiveness, an immediate concern lies in determining

[1] stopword list: http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words
[2] http://www.lemurproject.org/

| Corpus | Queries | Av. Query Length |
|---|---|---|
| TREC Vol. 4+5 | 301-350 | 2.54 |
| | 351-400 | 2.50 |
| | 401-450 | 2.40 |
| WT10g | 451-500 | 2.43 |
| | 501-550 | 2.84 |
| GOV2 | 701-750 | 3.10 |
| | 751-800 | 2.94 |
| | 801-850 | 2.86 |

Table 2.1: Overview of query sets.

for which retrieval approach and for which effectiveness measure the prediction method should be evaluated. In this chapter, we rely on average precision as the effectiveness measure as it is most widely used in the literature and available for all retrieval experiments we perform.

We chose to address the question of which retrieval approach to utilize in two ways. First, we investigate three common retrieval approaches, namely Language Modeling with Dirichlet Smoothing [170], Okapi [125] with its default parameter settings, and TF.IDF [13], the most basic retrieval approach based on term and document frequencies. Although this setup allows us to investigate the influence of a change in parameter setting for one particular retrieval approach, the results cannot be further generalized. In order to gain an understanding of predictor performances over a wider variety of retrieval approaches, we also rely on the retrieval runs submitted to TREC for each title topic set and their achieved retrieval effectiveness as ground truth.

Table 2.2 lists the retrieval effectiveness of the three retrieval approaches in MAP over all query sets. The level of smoothing in the Language Modeling approach is varied between $\mu = \{100, 500, 1000, 1500, 2000, 2500\}$. Larger values of $\mu$ show no further improvements in retrieval effectiveness (see Appendix B, Figure B.2). As expected, TF.IDF performs poorly, consistently degrading in performance as the collection size increases. Notably, while it reaches a MAP up to $0.11$ on the query sets of TREC Vol. 4+5, for the query sets of the GOV2 collection, the MAP degrades to $0.04$ at best. In contrast, Okapi outperforms the Language Modeling approach with $\mu = 100$ for all but one query set (401-450). In all other settings of $\mu$, Okapi performs (slightly) worse. The highest effectiveness in the Language Modeling approach is achieved for a smoothing level $\mu$ between $500$ and $2000$, depending on the individual query set.

## 2.6 Specificity

Query performance predictors in this category estimate the effectiveness of a query by the query terms' specificity. Consequently, a query consisting of common (collection) terms is deemed hard to answer as the retrieval algorithm is unable to

| Corpus | Queries | TF.IDF | Okapi | Language Modeling with Dirichlet Smoothing | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | $\mu = 100$ | $\mu = 500$ | $\mu = 1000$ | $\mu = 1500$ | $\mu = 2000$ | $\mu = 2500$ |
| TREC | 301-350 | 0.109 | 0.218 | 0.216 | **0.227** | 0.226 | 0.224 | 0.220 | 0.218 |
| Vol. 4+5 | 351-400 | 0.073 | 0.176 | 0.169 | 0.182 | 0.187 | 0.189 | **0.190** | 0.189 |
| | 401-450 | 0.088 | 0.223 | 0.229 | 0.242 | **0.245** | 0.244 | 0.241 | 0.239 |
| WT10g | 451-500 | 0.055 | 0.183 | 0.154 | 0.195 | **0.207** | 0.206 | 0.201 | 0.203 |
| | 501-550 | 0.061 | 0.163 | 0.137 | 0.168 | 0.180 | 0.185 | **0.189** | 0.189 |
| GOV2 | 701-750 | 0.029 | 0.230 | 0.212 | 0.262 | **0.269** | 0.266 | 0.261 | 0.256 |
| | 751-800 | 0.036 | 0.296 | 0.279 | 0.317 | **0.324** | 0.324 | 0.321 | 0.318 |
| | 801-850 | 0.023 | 0.250 | 0.247 | 0.293 | **0.297** | 0.292 | 0.284 | 0.275 |

Table 2.2: Overview of mean average precision over different retrieval approaches. Shown in bold is the most effective retrieval approach for each query set.

distinguish relevant and non-relevant documents based on term frequencies. The following is a list of predictors in the literature that exploit the specificity heuristic:

- Averaged Query Length (*AvQL*) [111],
- Averaged Inverse Document Frequency (*AvIDF*) [45],
- Maximum Inverse Document Frequency (*MaxIDF*) [128],
- Standard Deviation of IDF (*DevIDF*) [71],
- Averaged Inverse Collection Term Frequency (*AvICTF*) [71],
- Simplified Clarity Score (*SCS*) [71],
- Summed Collection Query Similarity (*SumSCQ*) [174],
- Averaged Collection Query Similarity *AvSCQ* [174],
- Maximum Collection Query Similarity *MaxSCQ* [174], and,
- Query Scope (*QS*) [71].

### 2.6.1 Query Based Specificity

The specificity of a query can be estimated to some extent without considering any other sources apart from the query itself. The average number *AvQL* of characters in the query terms is such a predictor: the higher the average length of a query, the more specific the query is assumed to be. For instance, TREC title topic 348 "Agoraphobia" has an average query length of 11, whilst TREC title topic 344 "Abuses of E-Mail" has an average length of $AvQL = 4.67$. Hence, "Agoraphobia" is considered to be more specific and therefore would be predicted to perform better than "Abuses of E-Mail".

Intuitively, making a prediction without taking the collection into account will often go wrong. Consider, for instance, that "BM25", which would be a very specific term in a corpus of newswire articles, contains few characters and hence, is erroneously considered to be non-specific according to the previous scheme. The success of predictors of this type also depends on the language of the collection. Text collections in languages that allow compounding such as Dutch and German might benefit more from predictors of this type than corpora consisting of English documents.

An alternative interpretation of query length is to consider the number of query terms in the search request as an indicator of specificity. We do not cover this interpretation here, as TREC title topics have very little variation in the number of terms, while TREC description and narrative topics on the other hand, often do not resemble realistic search requests. We note though, that Phan et al. [119] performed a user study where participants were asked to judge search requests of differing length, on a four point scale according to how narrow or broad they judge the underlying information need to be. A significant correlation was found between the number of terms in the search requests and the information need's specificity.

### 2.6.2 Collection Based Specificity

The specificity of a term $q_i$ can be approximated by either the document frequency $df(q_i)$ or the term frequency $tf(q_i)$. Both measures are closely related as a term that occurs in many documents can be expected to have a high term frequency in the collection. The opposite is also normally true: when a term occurs in very few documents then its term frequency will be low, if we assume that all documents in the collection are reasonable and no corner cases exist.

The most basic predictor in this context is *AvIDF* which determines the specificity of a query by relying on the average of the inverse document frequency ($idf$) of the query terms:

$$\begin{aligned} \textit{AvIDF} &= \frac{1}{m}\sum_{i=1}^{m}\left[\log\frac{doccount}{df(q_i)}\right] \\ &= \frac{1}{m}\sum_{i=1}^{m}\left[\log(doccount)-\log(df(q_i))\right] \qquad (2.6) \\ &= \log(doccount)-\frac{1}{m}\log\left[\prod_{i=1}^{m}df(q_i)\right] \qquad (2.7) \end{aligned}$$

*MaxIDF* is the maximum $idf$ value over all query terms. As an alternative metric, instead of averaging or maximizing the $idf$ values of all query terms, the predictor *DevIDF* relies on the standard deviation of the $idf$ values:

$$\textit{DevIDF} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\log\frac{doccount}{df(q_i)}-\textit{AvIDF}\right)^2} \qquad (2.8)$$

Note that a query with a high *DevIDF* score has at least one specific term and one general term, otherwise the standard deviation would be small. A shortcoming of this predictor lies in the fact that single term queries or queries containing only specific terms are assigned a score of $0$ and a low prediction score respectively. Thus, *DevIDF* can be expected to perform worse as predictor, on average, than *AvIDF* or *MaxIDF*.

In previous work [71], INQUERY's $idf$ formulation has been used in the predictors. In contrast to *AvIDF*, this approach normalizes the values to the $[0,1]$ interval.

Since such normalization makes no difference to the predictor performance it is not considered here. Among the pre-retrieval predictors proposed by He and Ounis [71] are the *AvICTF* and *SCS* predictors. *AvICTF* is defined as follows:

$$\begin{aligned} \textit{AvICTF} &= \frac{\log_2 \prod_{i=1}^{m} \left[\frac{termcount}{tf(q_i)}\right]}{m} \\ &= \frac{1}{m}\sum_{i=1}^{m} \log_2 \left[\frac{termcount}{tf(q_i)}\right] \\ &= \frac{1}{m}\sum_{i=1}^{m} \left[\log_2(termcount) - \log_2(tf(q_i))\right] \end{aligned} \tag{2.9}$$

When comparing Equations 2.6 and 2.9, the similarity between *AvICTF* and *AvIDF* becomes clear; instead of document frequencies, *AvICTF* relies on term frequencies.

The Simplified Clarity Score is, as the name implies, a simplification of the post-retrieval method *Clarity Score* which will be introduced in detail in Chapter 3. Instead of applying Clarity Score to the ranked list of results however, it is applied to the query itself, as follows:

$$\begin{aligned} \textit{SCS} &= \sum_{i=1}^{m} P_{ml}(q_i|\mathbf{q}) \log_2 \frac{P_{ml}(q_i|\mathbf{q})}{P(q_i)} \\ &\approx \sum_{i=1}^{m} \frac{1}{m} \log_2 \frac{\frac{1}{m}}{\frac{tf(q_i)}{termcount}} \\ &\approx \log_2 \frac{1}{m} + \frac{1}{m}\sum_{i=1}^{m} \left[\log_2(termcount) - \log_2(tf(q_i))\right]. \end{aligned} \tag{2.10}$$

$P_{ml}(q_i|\mathbf{q})$ is the maximum likelihood estimate of $q_i$ occurring in query $\mathbf{q}$. If we assume that each query term occurs exactly once in a query, then $P_{ml}(q_i|\mathbf{q}) = \frac{1}{m}$ and $\textit{SCS} = \log_2 \frac{1}{m} + \textit{AvICTF}$ (consider the similarity of Equations 2.9 and 2.10).

Importantly, if two queries have the same *AvICTF* score, *SCS* will give the query containing fewer query terms a higher score. The assumption of each term occurring only once in the query is a reasonable one, when one considers short queries such as those derived from TREC title topics. In the case of short queries, we can expect that the *SCS* and *AvICTF* scores for a set of queries will have a correlation close to 1, as the query length does not vary significantly. Longer queries such as those derived from TREC description topics that often include repetitions of terms will result in a larger margin.

Combining the collection term frequency and inverse document frequency was proposed by Zhao et al. [174]. The collection query similarity summed over all query terms is defined as:

$$\textit{SumSCQ} = \sum_{i=1}^{m} (1 + \ln(cf(q_i))) \times \ln\left(1 + \frac{doccount}{df(q_i)}\right). \tag{2.11}$$

*AvSCQ* is the average similarity over all query terms: $AvSCQ = \frac{1}{m} \times SumSCQ$, whereas the maximum query collection similarity *MaxSCQ* relies on the maximum collection query similarity score over all query terms. The authors argue that a query, which is similar to the collection as a whole is easier to retrieve documents for, since the similarity is an indicator of whether documents answering the information need are contained in the collection. As the score increases with increased collection term frequency and increased inverse document frequency, terms that appear in few documents many times are favored. Those terms can be seen as highly specific, as they occur in relatively few documents, while at the same time they occur often enough to be important to the query.

Query Scope is a measure that makes use of the document frequencies. In this instance, the number of documents containing at least one of the query terms is used as an indicator of query quality; the more documents contained in this set, the lower the predicted effectiveness of the query:

$$QS = -\log \frac{N_{\mathbf{q}}}{doccount}. \tag{2.12}$$

Finally, we observe that for queries consisting of a single term, the predictors *QS*, *MaxIDF* and *AvIDF* will return exactly the same score.

### 2.6.3 Experimental Evaluation

The evaluation of the introduced prediction methods is performed in two steps. First, to support the mathematical derivation, we present the correlations, as given by Kendall's $\tau$, between the different predictors. A high correlation coefficient indicates a strong relationship. Then, we evaluate the predictors according to their ability to predict the performance of different query sets across different corpora and retrieval approaches. This evaluation is presented in terms of Kendall's $\tau$ and the linear correlation coefficient.

#### Predictor-Predictor Correlations

The correlations between the predictor scores are shown in Table 2.3 aggregated over the query sets of TREC Vol. 4+5 and over the query sets of GOV2. Let us first consider the results over the queries 301-450. The three predictors *AvIDF*, *SCS* and *AvICTF* are highly correlated, with a minimum $\tau = 0.88$ (the same evaluation with the linear correlation coefficient yields $r = 0.98$). The predictors *QS* and *MaxIDF* can also be considered in this group to some extent as they correlate with all three predictors with $\tau \geq 0.65$ ($r \geq 0.75$). Most predictors have a moderate to strong relationship to each other. Only *AvQL*, *DevIDF* and *SumSCQ* consistently behave differently.

The similarity between the prediction methods is different for the queries of the GOV2 corpus. While *AvICTF*, *AvIDF*, *MaxIDF*, *SCS*, *QS* and *DevIDF* exhibit similar though somewhat lower correlations to each other, *AvQL* and the query collection similarity based predictors, on the other hand, behave differently. The query length based predictor is now consistently uncorrelated to any of the other predictors.

| | *AvIDF* | *MaxIDF* | *DevIDF* | *SCS* | *QS* | *AvICTF* | *AvQL* | *SumSCQ* | *AvSCQ* | *MaxSCQ* |
|---|---|---|---|---|---|---|---|---|---|---|
| *AvIDF* | | 0.721 | 0.164 | 0.875 | 0.683 | 0.933 | 0.292 | 0.155 | 0.710 | 0.517 |
| *MaxIDF* | | | 0.429 | 0.651 | 0.417 | 0.694 | 0.249 | 0.165 | 0.453 | 0.625 |
| *DevIDF* | | | | 0.119 | -0.137 | 0.142 | -0.002 | 0.203 | 0.019 | 0.397 |
| *SCS* | | | | | 0.723 | 0.915 | 0.310 | 0.053 | 0.662 | 0.439 |
| *QS* | | | | | | 0.693 | 0.268 | 0.076 | 0.722 | 0.290 |
| *AvICTF* | | | | | | | 0.295 | 0.138 | 0.683 | 0.469 |
| *AvQL* | | | | | | | | -0.063 | 0.196 | 0.111 |
| *SumSCQ* | | | | | | | | | 0.297 | 0.236 |
| *AvSCQ* | | | | | | | | | | 0.524 |

(a) Queries 301-450 (TREC Vol. 4+5)

| | *AvIDF* | *MaxIDF* | *DevIDF* | *SCS* | *QS* | *AvICTF* | *AvQL* | *SumSCQ* | *AvSCQ* | *MaxSCQ* |
|---|---|---|---|---|---|---|---|---|---|---|
| *AvIDF* | | 0.598 | 0.139 | 0.833 | 0.615 | 0.894 | 0.052 | 0.127 | 0.835 | 0.515 |
| *MaxIDF* | | | 0.513 | 0.521 | 0.238 | 0.585 | 0.032 | 0.192 | 0.450 | 0.777 |
| *DevIDF* | | | | 0.095 | -0.220 | 0.133 | 0.036 | 0.157 | -0.010 | 0.445 |
| *SCS* | | | | | 0.665 | 0.895 | 0.086 | -0.023 | 0.742 | 0.419 |
| *QS* | | | | | | 0.613 | 0.073 | -0.043 | 0.714 | 0.201 |
| *AvICTF* | | | | | | | 0.068 | 0.083 | 0.767 | 0.476 |
| *AvQL* | | | | | | | | -0.085 | 0.037 | 0.148 |
| *SumSCQ* | | | | | | | | | 0.184 | 0.253 |
| *AvSCQ* | | | | | | | | | | 0.435 |

(b) Queries 701-850 (GOV2)

Table 2.3: Kendall's $\tau$ between scores of specificity based predictors.

### Predictor Evaluation

While the relationship between the predictors is certainly important (for instance, it is not necessary to report both *AvICTF* and *AvIDF*), the more important question that arises is how well the predictors perform in predicting the retrieval effectiveness of queries. The retrieval effectiveness can be measured in various ways, including average precision, precision at 10 documents, reciprocal rank, and other measures. In this frame of inquiry, average precision is utilized as the measure of true retrieval performance of each query. The predictors were evaluated for their prediction capabilities of TF.IDF, Okapi and Language Modeling with Dirichlet smoothing. For the latter, the level of smoothing $\mu$ was fixed to the best performing retrieval setting as observed in Table 2.2. In Table 2.4 the linear correlation coefficient $r$ is reported, in Table 2.5 the results of Kendall's $\tau$ are listed. The query sets are evaluated individually, as well as combined for all query sets of a particular corpus.

We observe that the predictor performance is influenced considerably by the particular query set under consideration. This observation holds even within the scope of a single collection. *MaxSCQ* for instance can be considered as the best predictor overall, but for one particular query set, 301-350, it breaks down completely, achieving no significant correlation. A contrasting example is *DevIDF*, which generally does not result in meaningful correlations, however for two query sets (401-450, 501-550) it is among the best performing predictors with respect to $r$. The group of *AvIDF*, *AvICTF*, *SCS*, *QS* and *MaxIDF* predictors achieve their highest correlations in the TF.IDF setting for TREC Vol. 4+5 and the WT10g collection.

When comparing the results of the Okapi and Language Modeling approach across all predictors, considerable differences in predictor performances are only visible for a single query set (701-750). In most other instances the predictors can

| Queries | | *AvIDF* | *MaxIDF* | *DevIDF* | *SCS* | *QS* | *AvICTF* | *AvQL* | *SumSCQ* | *AvSCQ* | *MaxSCQ* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 301- | **TFIDF** | 0.809 | 0.687 | -0.068 | **0.822** | 0.796 | 0.813 | 0.458 | -0.340 | -0.033 | -0.085 |
| 350 | **Okapi** | **0.625** | 0.609 | 0.127 | 0.611 | 0.557 | 0.619 | 0.326 | -0.126 | 0.040 | 0.110 |
| | $\mu = 500$ | **0.591** | 0.574 | 0.119 | 0.578 | 0.531 | 0.582 | 0.310 | -0.123 | 0.074 | 0.122 |
| 351- | **TFIDF** | **0.604** | 0.422 | -0.068 | 0.584 | 0.603 | 0.578 | 0.014 | -0.150 | 0.442 | 0.350 |
| 400 | **Okapi** | 0.330 | 0.346 | 0.133 | 0.265 | 0.252 | 0.301 | -0.210 | 0.189 | 0.360 | **0.465** |
| | $\mu = 2000$ | 0.374 | 0.383 | 0.166 | 0.319 | 0.284 | 0.348 | -0.172 | 0.123 | 0.412 | **0.507** |
| 401- | **TFIDF** | **0.541** | 0.492 | 0.176 | 0.540 | 0.493 | 0.494 | 0.465 | -0.188 | 0.333 | 0.403 |
| 450 | **Okapi** | 0.502 | **0.587** | 0.448 | 0.444 | 0.302 | 0.444 | 0.177 | 0.039 | 0.347 | 0.507 |
| | $\mu = 1000$ | 0.576 | **0.649** | 0.450 | 0.518 | 0.381 | 0.516 | 0.193 | 0.046 | 0.408 | 0.524 |
| 301- | **TFIDF** | 0.693 | 0.565 | -0.001 | **0.696** | 0.673 | 0.680 | 0.345 | -0.244 | 0.176 | 0.150 |
| 450 | **Okapi** | 0.508 | **0.523** | 0.226 | 0.469 | 0.400 | 0.483 | 0.129 | 0.009 | 0.214 | 0.322 |
| | $\mu = 1000$ | 0.516 | **0.532** | 0.239 | 0.480 | 0.407 | 0.490 | 0.133 | -0.002 | 0.256 | 0.341 |
| 451- | **TFIDF** | 0.641 | 0.408 | -0.369 | 0.658 | **0.699** | 0.634 | 0.130 | -0.391 | 0.332 | 0.092 |
| 500 | **Okapi** | 0.204 | 0.280 | 0.158 | 0.146 | 0.134 | 0.193 | -0.280 | 0.105 | 0.242 | **0.284** |
| | $\mu = 1000$ | 0.153 | 0.214 | 0.139 | 0.087 | 0.092 | 0.141 | -0.262 | 0.176 | 0.384 | **0.429** |
| 501- | **TFIDF** | 0.441 | 0.398 | 0.146 | 0.400 | 0.318 | 0.415 | 0.122 | -0.346 | 0.345 | **0.442** |
| 550 | **Okapi** | 0.143 | 0.383 | **0.415** | 0.168 | -0.092 | 0.111 | 0.068 | 0.160 | 0.089 | 0.373 |
| | $\mu = 2000$ | 0.221 | 0.469 | **0.450** | 0.189 | -0.061 | 0.200 | 0.052 | 0.192 | 0.154 | 0.393 |
| 451- | **TFIDF** | **0.525** | 0.386 | -0.108 | 0.523 | 0.513 | 0.511 | 0.127 | -0.365 | 0.332 | 0.260 |
| 550 | **Okapi** | 0.195 | 0.315 | 0.245 | 0.160 | 0.075 | 0.179 | -0.147 | 0.116 | 0.191 | **0.309** |
| | $\mu = 1000$ | 0.182 | 0.292 | 0.233 | 0.126 | 0.062 | 0.167 | -0.135 | 0.183 | 0.307 | **0.400** |
| 701- | **TFIDF** | 0.247 | 0.312 | 0.290 | 0.207 | 0.146 | 0.191 | -0.134 | -0.041 | 0.282 | **0.388** |
| 750 | **Okapi** | 0.202 | 0.263 | 0.121 | 0.128 | 0.150 | 0.154 | -0.202 | 0.199 | 0.290 | **0.382** |
| | $\mu = 1000$ | 0.393 | 0.425 | 0.160 | 0.325 | 0.334 | 0.354 | -0.150 | 0.151 | 0.444 | **0.473** |
| 751- | **TFIDF** | 0.008 | 0.017 | 0.019 | 0.035 | 0.146 | 0.031 | 0.149 | -0.125 | 0.073 | **0.253** |
| 800 | **Okapi** | 0.304 | 0.244 | 0.052 | 0.274 | 0.267 | 0.297 | 0.049 | 0.200 | **0.332** | 0.283 |
| | $\mu = 1000$ | 0.315 | 0.232 | 0.061 | 0.278 | 0.252 | 0.304 | 0.122 | 0.258 | **0.393** | 0.371 |
| 801- | **TFIDF** | **0.581** | 0.435 | -0.076 | 0.534 | 0.533 | 0.567 | 0.042 | 0.213 | 0.359 | 0.225 |
| 850 | **Okapi** | 0.309 | 0.309 | 0.147 | 0.220 | 0.162 | 0.263 | 0.019 | 0.333 | **0.368** | 0.345 |
| | $\mu = 1000$ | 0.223 | 0.337 | 0.317 | 0.137 | 0.009 | 0.185 | 0.043 | 0.323 | 0.248 | **0.362** |
| 701- | **TFIDF** | 0.270 | 0.228 | 0.052 | 0.250 | 0.247 | 0.251 | 0.045 | 0.017 | 0.220 | **0.272** |
| 850 | **Okapi** | 0.278 | 0.283 | 0.121 | 0.215 | 0.187 | 0.245 | -0.040 | 0.229 | 0.324 | **0.341** |
| | $\mu = 1000$ | 0.309 | 0.331 | 0.185 | 0.248 | 0.179 | 0.281 | 0.007 | 0.235 | 0.352 | **0.403** |

Table 2.4: Linear correlation coefficients $r$ of specificity-based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

be considered to predict equally well for both retrieval approaches, only *MaxSCQ* performs consequently worse on Okapi. We pointed out earlier, that the only difference between *AvIDF* and *AvICTF* is the reliance on $doccount$ versus $termcount$. Across all collections, *AvIDF* is slightly better than *AvICTF*, hence we can conclude that $doccount$ is somewhat more reliable. The performance of *SCS* is comparable to *AvICTF*, but always slightly worse than *AvIDF*. The predictors *AvQL*, *DevIDF* and *SumSCQ* consistently perform poorly, at best they result in moderate correlations for one or two query sets. In the case of *AvQL* the reasons for failure are the lack of term length distribution. For instance, consider query set 701-750, where $31$ out of $50$ queries have an average term length between $5$ and $6$, rendering the predictor unusable. Note that the spread is considerably larger for queries 301-350, where *AvQL* results in a small positive correlation.

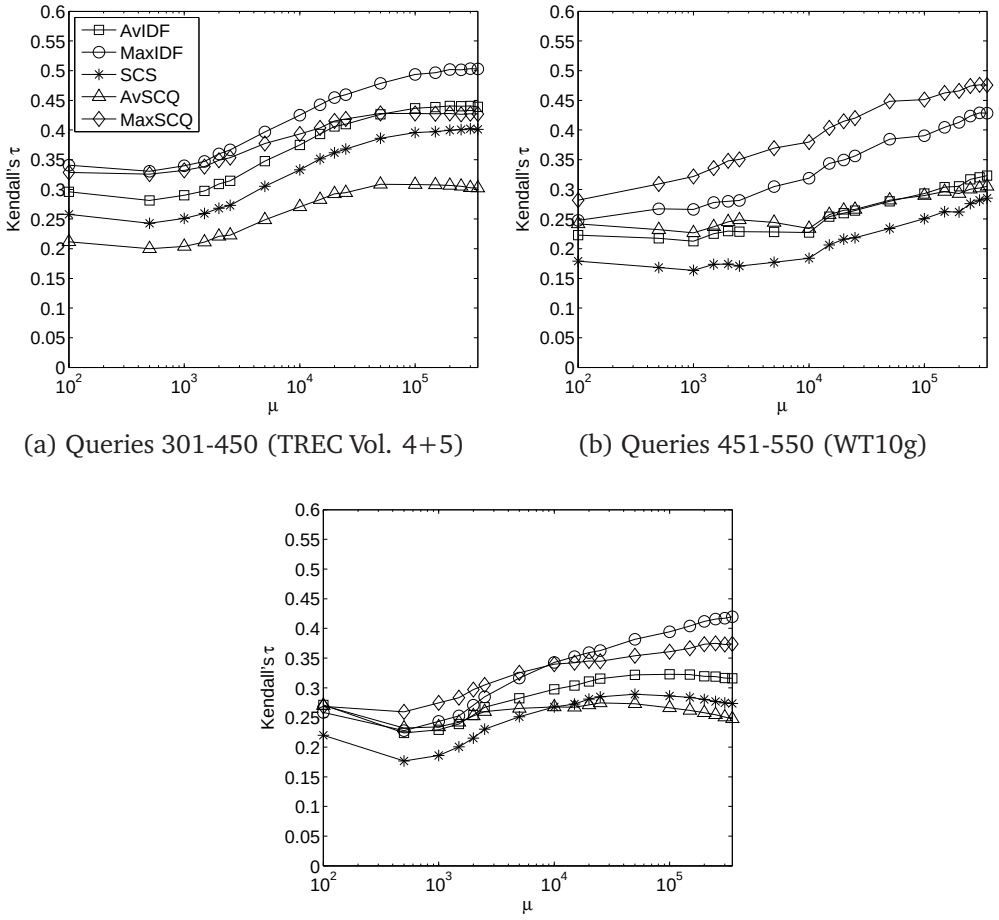
Overall, the predictor *MaxSCQ* performs best, however due to its drastic failure on query set 301-350, a safer choice would be the slightly worse performing *MaxIDF*. If we focus on the corpora, we observe that TREC Vol. 4+5 is easiest to predict for, whereas the WT10g and GOV2 corpora pose significant difficulties to the predictors. Although our observations hold for both the linear correlation coeffi-

| Queries | | *AvIDF* | *MaxIDF* | *DevIDF* | *SCS* | *QS* | *AvICTF* | *AvQL* | *SumSCQ* | *AvSCQ* | *MaxSCQ* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 301-350 | **TFIDF** | **0.480** | 0.474 | 0.093 | 0.439 | 0.356 | 0.465 | 0.281 | 0.045 | 0.225 | 0.286 |
| | **Okapi** | 0.348 | **0.409** | 0.115 | 0.304 | 0.220 | 0.327 | 0.171 | 0.067 | 0.093 | 0.162 |
| | $\mu = 500$ | 0.314 | **0.368** | 0.086 | 0.286 | 0.219 | 0.289 | 0.165 | 0.087 | 0.095 | 0.181 |
| 351-400 | **TFIDF** | 0.355 | 0.336 | 0.045 | 0.328 | 0.333 | 0.336 | -0.043 | 0.042 | 0.368 | **0.413** |
| | **Okapi** | 0.244 | 0.287 | 0.116 | 0.180 | 0.200 | 0.202 | -0.097 | 0.146 | 0.275 | **0.398** |
| | $\mu = 2000$ | 0.271 | 0.307 | 0.153 | 0.227 | 0.227 | 0.238 | -0.095 | 0.126 | 0.315 | **0.422** |
| 401-450 | **TFIDF** | 0.310 | 0.320 | 0.146 | 0.300 | 0.197 | 0.293 | 0.250 | -0.009 | 0.275 | **0.439** |
| | **Okapi** | 0.275 | 0.354 | 0.276 | 0.252 | 0.146 | 0.249 | 0.046 | 0.033 | 0.228 | **0.424** |
| | $\mu = 1000$ | 0.313 | 0.402 | 0.314 | 0.277 | 0.161 | 0.273 | 0.048 | 0.058 | 0.265 | **0.474** |
| 301-450 | **TFIDF** | **0.400** | 0.390 | 0.113 | 0.375 | 0.299 | 0.383 | 0.169 | 0.019 | 0.292 | 0.373 |
| | **Okapi** | 0.287 | **0.340** | 0.177 | 0.248 | 0.188 | 0.265 | 0.042 | 0.089 | 0.195 | 0.330 |
| | $\mu = 1000$ | 0.290 | **0.340** | 0.180 | 0.251 | 0.190 | 0.266 | 0.039 | 0.080 | 0.204 | 0.332 |
| 451-500 | **TFIDF** | 0.480 | 0.316 | -0.182 | 0.480 | **0.494** | 0.470 | 0.100 | -0.169 | 0.448 | 0.364 |
| | **Okapi** | 0.261 | **0.361** | 0.144 | 0.203 | 0.151 | 0.254 | -0.115 | 0.079 | 0.188 | 0.336 |
| | $\mu = 1000$ | 0.249 | 0.281 | 0.137 | 0.174 | 0.135 | 0.236 | -0.076 | 0.147 | 0.321 | **0.435** |
| 501-550 | **TFIDF** | 0.364 | 0.355 | 0.017 | 0.349 | 0.236 | 0.338 | 0.202 | -0.246 | 0.337 | **0.391** |
| | **Okapi** | 0.139 | 0.233 | 0.184 | 0.156 | 0.005 | 0.099 | 0.109 | 0.087 | 0.102 | **0.240** |
| | $\mu = 2000$ | 0.187 | **0.277** | 0.174 | 0.136 | 0.046 | 0.143 | 0.087 | 0.111 | 0.160 | 0.270 |
| 451-550 | **TFIDF** | **0.403** | 0.319 | -0.085 | 0.401 | 0.355 | 0.393 | 0.155 | -0.210 | 0.371 | 0.354 |
| | **Okapi** | 0.192 | **0.274** | 0.165 | 0.175 | 0.069 | 0.177 | -0.019 | 0.081 | 0.132 | 0.262 |
| | $\mu = 1000$ | 0.213 | 0.266 | 0.157 | 0.163 | 0.079 | 0.192 | -0.005 | 0.138 | 0.227 | **0.322** |
| 701-750 | **TFIDF** | 0.186 | 0.258 | 0.257 | 0.186 | 0.084 | 0.173 | 0.017 | -0.045 | 0.188 | **0.297** |
| | **Okapi** | 0.151 | 0.189 | 0.050 | 0.099 | 0.124 | 0.112 | -0.111 | 0.160 | 0.184 | **0.247** |
| | $\mu = 1000$ | 0.277 | 0.304 | 0.108 | 0.211 | 0.218 | 0.248 | -0.065 | 0.161 | 0.300 | **0.331** |
| 751-800 | **TFIDF** | -0.016 | 0.034 | 0.021 | -0.006 | 0.084 | -0.034 | 0.082 | -0.002 | 0.012 | **0.173** |
| | **Okapi** | 0.207 | 0.169 | 0.011 | 0.192 | 0.193 | 0.205 | 0.041 | 0.151 | **0.224** | 0.174 |
| | $\mu = 1000$ | 0.253 | 0.204 | 0.059 | 0.240 | 0.217 | 0.260 | 0.117 | 0.165 | 0.274 | **0.291** |
| 801-850 | **TFIDF** | 0.255 | **0.267** | 0.144 | 0.193 | 0.094 | 0.232 | 0.104 | 0.219 | 0.221 | 0.250 |
| | **Okapi** | 0.246 | 0.218 | 0.144 | 0.166 | 0.118 | 0.205 | 0.045 | **0.277** | 0.256 | 0.241 |
| | $\mu = 1000$ | 0.193 | 0.228 | **0.255** | 0.130 | 0.004 | 0.166 | 0.057 | 0.238 | 0.171 | 0.241 |
| 701-850 | **TFIDF** | 0.120 | 0.176 | 0.127 | 0.096 | 0.040 | 0.103 | 0.077 | 0.045 | 0.122 | **0.229** |
| | **Okapi** | 0.199 | 0.195 | 0.076 | 0.151 | 0.142 | 0.172 | -0.011 | 0.182 | 0.216 | **0.221** |
| | $\mu = 1000$ | 0.229 | 0.243 | 0.143 | 0.186 | 0.137 | 0.209 | 0.028 | 0.179 | 0.234 | **0.274** |

Table 2.5: Kendall's $\tau$ coefficients of specificity-based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

cient $r$ and Kendall's $\tau$, there are also differences visible when comparing Tables 2.4 and 2.5. Comparing the performance of *MaxIDF* and *MaxSCQ* for queries 301-450 yields hardly any differences in performance when reporting $\tau$ ($\tau_{MaxIDF} = 0.34$, $\tau_{MaxSCQ} = 0.33$); the linear correlation coefficient on the other hand indicates a considerable performance gap, namely, $r_{MaxIDF} = 0.52$ versus $r_{MaxSCQ} = 0.34$. Thus, if query performance prediction should be applied in a practical setup, where the average precision score is of importance, *MaxIDF* is a better predictor than *MaxSCQ*, while the reverse is true if the application relies on the effectiveness ranking of the queries.

Due to the nature of most specificity based prediction methods, it is expected that the amount of smoothing in the Language Modeling approach will have a considerable influence on their quality as increased smoothing results in an increasing influence of collection statistics. To investigate the influence of high levels of smoothing, $\mu$ is evaluated for levels ranging from $\mu = 5 \times 10^3$ to $\mu = 3.5 \times 10^5$ (more specifics are given in Appendix B, Figure B.2). We report the prediction accuracy of *AvIDF*, *MaxIDF*, *SCS*, *AvSCQ* and *MaxSCQ*, the remaining predictors were excluded either due to poor performance or their similarity to one of the reported predictors.




(a) Queries 301-450 (TREC Vol. 4+5)

(b) Queries 451-550 (WT10g)

(c) Queries 701-850 (GOV2)

Figure 2.4: The influence of the level $\mu$ of smoothing on the accuracy of various predictors.

The results shown in Figure 2.4 are reported in terms of Kendall's $\tau$ (the results were similar for the linear correlation coefficient). They confirm the hypothesis, that increasing levels of $\mu$ generally lead to a positive change in correlation for the specificity-based predictors. The relative predictor performance s remain largely the same, the correlation increases occur to similar degrees. Depending on the corpus and the predictor, the performance difference can be large, for instance at low levels of smoothing *MaxIDF* has a correlation of $\tau_{\mu=500} = 0.33$, whereas it reaches $\tau_{\mu=3\times10^5} = 0.5$ when the amount of smoothing is increased. Changing $\mu$ has the least effect on *AvSCQ*; although its correlation also rises with the rise of $\mu$, the improvements are small and they trail off after $\tau$ reaches $5 \times 10^4$ for TREC Vol. 4+5 and GOV2.

## 2.7 Ranking Sensitivity

Although pre-retrieval predictors do not consider the ranked list of results returned by the retrieval system for a given query, they can still rely on collection statistics to infer how difficult it will be for the system to rank the documents according to the query. The three predictors in this category are all variations of the same principle, and are presented below:

- Summed Term Weight Variability (*SumVAR*) [174],
- Averaged Term Weight Variability (*AvVAR*) [174], and,
- Maximum Term Weight Variability (*MaxVAR*) [174].

### 2.7.1 Collection Based Sensitivity

This family of predictors exploits the distribution of term weights across the collection. If the term weights across all documents containing query term $q_i$ are similar, there is little evidence for a retrieval system on how to rank those documents given $q_i$, and thus different retrieval algorithms are likely to produce widely different rankings. Conversely, if the term weights differ widely across the collection, ranking becomes easier and different retrieval algorithms are expected to produce similar rankings. Here we assume that the retrieval system relies solely on collection statistics, without considering external sources or additional information.

In [174], the term weight $w(q_i, d)$ is based on TF.IDF, the average term weight $\overline{w}_{q_i}$ is the average weight over all documents containing $q_i$. *SumVAR* is the sum of the query term weight deviations:

$$SumVAR = \sum_{i=1}^{m} \sqrt{\frac{1}{df(q_i)} \sum_{d \in N_{q_i}} (w(q_i, d) - \overline{w}_{q_i})^2}. \tag{2.13}$$

In contrast to *SumVAR* which is not normalized according to the query length, $AvVAR = \frac{1}{m} \times SumVAR$ is normalized. Finally, the maximum variability score over all query terms is used as prediction score for the predictor *MaxVAR*. Note, that the three predictors in this category are more complex than for example *MaxIDF*, as they rely on TF.IDF weights and require additional pre-processing.

### 2.7.2 Experimental Evaluation

Analogous to the specificity based predictors, the algorithms in this category are first evaluated with respect to their similarity to each other. Then, their ability to predict retrieval effectiveness will be evaluated.

#### Predictor-Predictor Correlations

In Table 2.6 the correlations between the predictor scores are shown. While the results of the query sets of TREC Vol. 4+5 and GOV2 are similar, with *AvVAR* and

*MaxVAR* being more closeley related to each other than to *SumVAR*, in the WT10g collection, *SumVAR* is hardly related to the other two predictor variations. Since *SumVAR* is not normalized with regard to query length, we expect it to perform rather poorly as predictor.

| | *SumVAR* | *AvVAR* | *MaxVAR* |
|---|---|---|---|
| *SumVAR* | | 0.546 | 0.561 |
| *AvVAR* | | | 0.721 |

(a) Queries 301-450 (TREC Vol. 4+5)

| | *SumVAR* | *AvVAR* | *MaxVAR* |
|---|---|---|---|
| *SumVAR* | | 0.075 | 0.210 |
| *AvVAR* | | | 0.669 |

(b) Queries 451-550 (WT10g)

| | *SumVAR* | *AvVAR* | *MaxVAR* |
|---|---|---|---|
| *SumVAR* | | 0.397 | 0.478 |
| *AvVAR* | | | 0.616 |

(c) Queries 701-850 (GOV2)

Table 2.6: Kendall's $\tau$ between scores of ranking sensitivity based predictors.

### Predictor Evaluation

Table 2.7 contains the correlation coefficients the predictors achieve across all query sets and across the standard retrieval approaches. Of the three predictor variations, *SumVAR* is the most erratic. This is not surprising, as it is not normalized with respect to the number of terms in the queries. *MaxVAR* is the best predictor of this category, with a surprisingly good performance on query set 501-550 of the WT10g collection, which provided the most difficulties to the specificity based predictors. *AvVAR*'s overall performance is slightly worse than *MaxVAR*'s. There are two query sets which yield somewhat unexpected results: for one, query set 301-350, which has shown to be the easiest for *AvIDF* and related predictors (leading to the highest observed correlation), is the most difficult for the ranking sensitivity based predictors. Secondly, query set 801-850 shows hardly any variation for the performance of the three predictors, unlike the other query sets.

Similar to the observations made for the specificity based predictors, increasing the level of smoothing in the Language Modeling approach increases the correlation coefficients of *AvVAR* and *MaxVAR* across the three corpora. The largest improvements are recorded for the query sets of the WT10g corpus; the correlation of *MaxVAR* ranges from $\tau_{\mu=100} = 0.29$ to $\tau_{\mu=3.5\times10^5} = 0.44$ at the highest level of smoothing. Smaller improvements up to $\tau = 0.1$ are also achieved for TREC Vol. 4+5 and *MaxVAR*, where $\tau$ peaks at $\mu = 2.5 \times 10^4$. Relatively unaffected is the GOV2 corpus, where the trend is positive, but the changes in correlation are minor. The *SumVAR* predictor, on the other hand, continuously degrades when the level of smoothing is improved; it achieves its highest correlation at $\mu = 100$.

| | | *SumVAR* | *AvVAR* | *MaxVAR* |
|---|---|---|---|---|
| 301-350 | **TF.IDF** | −0.035 | **0.306** | 0.203 |
| | **Okapi** | 0.151 | **0.371** | 0.359 |
| | $\mu = 500$ | 0.163 | **0.403** | 0.369 |
| 351-400 | **TF.IDF** | 0.149 | **0.583** | 0.455 |
| | **Okapi** | 0.318 | 0.400 | **0.426** |
| | $\mu = 2000$ | 0.288 | 0.431 | **0.445** |
| 401-450 | **TF.IDF** | 0.262 | **0.706** | 0.631 |
| | **Okapi** | 0.517 | 0.699 | **0.723** |
| | $\mu = 1000$ | 0.552 | 0.758 | **0.764** |
| 301-450 | **TF.IDF** | 0.089 | **0.487** | 0.388 |
| | **Okapi** | 0.293 | 0.476 | **0.491** |
| | $\mu = 1000$ | 0.297 | 0.510 | **0.513** |
| 451-500 | **TF.IDF** | −0.266 | **0.336** | 0.181 |
| | **Okapi** | 0.173 | 0.197 | **0.253** |
| | $\mu = 1000$ | 0.259 | 0.324 | **0.381** |
| 501-550 | **TF.IDF** | −0.168 | 0.489 | **0.566** |
| | **Okapi** | 0.336 | 0.201 | **0.513** |
| | $\mu = 2000$ | 0.366 | 0.233 | **0.533** |
| 451-550 | **TF.IDF** | −0.219 | **0.401** | 0.366 |
| | **Okapi** | 0.221 | 0.198 | **0.337** |
| | $\mu = 1000$ | 0.300 | 0.291 | **0.411** |
| 701-750 | **TF.IDF** | 0.160 | 0.442 | **0.479** |
| | **Okapi** | 0.360 | 0.392 | **0.437** |
| | $\mu = 1000$ | 0.293 | **0.464** | 0.435 |
| 751-800 | **TF.IDF** | −0.062 | 0.119 | **0.167** |
| | **Okapi** | 0.295 | **0.406** | 0.371 |
| | $\mu = 1000$ | 0.363 | **0.438** | 0.434 |
| 801-850 | **TF.IDF** | 0.357 | **0.495** | 0.355 |
| | **Okapi** | 0.401 | **0.430** | 0.420 |
| | $\mu = 1000$ | 0.380 | 0.314 | **0.389** |
| 701-850 | **TF.IDF** | 0.143 | **0.323** | 0.300 |
| | **Okapi** | 0.336 | 0.397 | **0.402** |
| | $\mu = 1000$ | 0.337 | 0.392 | **0.412** |

(a) Linear correlation coefficient $r$

| | | *SumVAR* | *AvVAR* | *MaxVAR* |
|---|---|---|---|---|
| 301-350 | **TF.IDF** | 0.166 | 0.383 | **0.390** |
| | **Okapi** | 0.201 | 0.302 | **0.367** |
| | $\mu = 500$ | 0.203 | 0.291 | **0.353** |
| 351-400 | **TF.IDF** | 0.218 | **0.434** | 0.410 |
| | **Okapi** | 0.334 | 0.339 | **0.415** |
| | $\mu = 2000$ | 0.317 | 0.382 | **0.434** |
| 401-450 | **TF.IDF** | 0.252 | 0.413 | **0.437** |
| | **Okapi** | 0.304 | 0.432 | **0.443** |
| | $\mu = 1000$ | 0.352 | 0.460 | **0.494** |
| 301-450 | **TF.IDF** | 0.220 | 0.403 | **0.417** |
| | **Okapi** | 0.285 | 0.356 | **0.407** |
| | $\mu = 1000$ | 0.283 | 0.356 | **0.411** |
| 451-500 | **TF.IDF** | −0.078 | **0.424** | 0.330 |
| | **Okapi** | 0.118 | 0.188 | **0.241** |
| | $\mu = 1000$ | 0.203 | 0.300 | **0.339** |
| 501-550 | **TF.IDF** | −0.154 | 0.400 | **0.451** |
| | **Okapi** | 0.189 | 0.189 | **0.323** |
| | $\mu = 2000$ | 0.189 | 0.233 | **0.327** |
| 451-550 | **TF.IDF** | −0.121 | **0.394** | 0.385 |
| | **Okapi** | 0.145 | 0.168 | **0.262** |
| | $\mu = 1000$ | 0.213 | 0.249 | **0.321** |
| 701-750 | **TF.IDF** | 0.093 | 0.287 | **0.336** |
| | **Okapi** | 0.245 | 0.261 | **0.276** |
| | $\mu = 1000$ | 0.250 | **0.330** | 0.288 |
| 751-800 | **TF.IDF** | 0.012 | 0.089 | **0.172** |
| | **Okapi** | 0.197 | **0.259** | 0.247 |
| | $\mu = 1000$ | 0.230 | 0.292 | **0.318** |
| 801-850 | **TF.IDF** | 0.204 | 0.242 | **0.272** |
| | **Okapi** | 0.303 | **0.314** | 0.306 |
| | $\mu = 1000$ | **0.280** | 0.233 | 0.274 |
| 701-850 | **TF.IDF** | 0.107 | 0.198 | **0.241** |
| | **Okapi** | 0.237 | **0.268** | 0.267 |
| | $\mu = 1000$ | 0.241 | 0.269 | **0.280** |

(b) Kendall's $\tau$

Table 2.7: Correlation coefficients of ranking sensitivity based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

## 2.8 Ambiguity

Predictors that infer the quality of a query from the ambiguity of the query terms include the following:

- Averaged Query Term Coherence (*AvQC*) [73],
- Averaged Query Term Coherence with Global Constraint (*AvQCG*)[73],
- Averaged Polysemy (*AvP*) [111], and,
- Averaged Noun Polysemy (*AvNP*).

The first two predictors rely on the collection and, specifically, on all documents containing any of the query terms, to determine the amount of ambiguity. The latter two predictors exploit WordNet, an external source which provides the number of senses a term has, thus making further calculations on the corpus unnecessary.

### 2.8.1 Collection Based Ambiguity

He et al. [73] derive the ambiguity of a query term $q_i$ by calculating the similarity between all documents that contain $q_i$. The set of all those documents is $N_{q_i}$, with $|N_{q_i}| = n$. The *set coherence* of $N_{q_i}$ is then defined as:

$$\textit{SetCoherence}(N_{q_i}) = \frac{\sum_{i \neq j \in 1,\ldots,n} \sigma(d_i, d_j)}{n(n-1)}$$

where $\sigma(d_i, d_j)$ is a similarity function that returns $1$ if the similarity between $d_i$ and $d_j$ exceeds a threshold $\theta$; otherwise $\sigma = 0$. The *SetCoherence* is defined in the interval $[0, 1]$: *SetCoherence* $= 1$ if all documents in $N_{q_i}$ are similar to each other, and *SetCoherence* $= 0$ if none are.

When viewed as a clustering task, the documents in $N_{q_i}$ are clustered agglomeratively. Initially, each document is assigned its own cluster and iteratively the two closest clusters are merged. The distance between two clusters is given by the distance of the two farthest points in the clusters (complete linkage clustering). The merging process stops, if the merged clusters have a similarity less than $\theta$. *SetCoherence* is then the number of links between nodes within a cluster, divided by the number of links between all nodes independent of the cluster. In the ideal case, all documents are clustered into a single cluster. The *SetCoherence* score is mainly influenced by the size of the largest cluster: the larger the dominant cluster, the larger the score. Equally sized clusters receive a lower *SetCoherence* score.

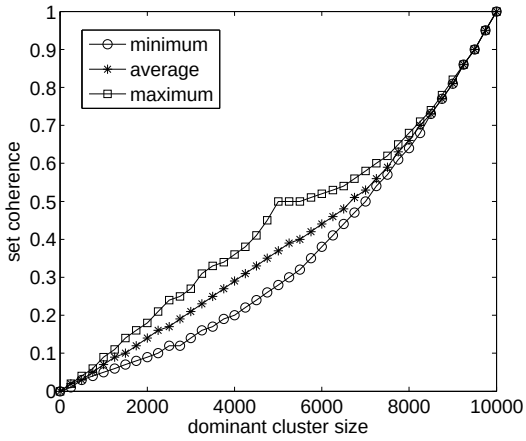



Figure 2.5: The development of the *SetCoherence* score with increased size of the dominant cluster.

We investigated the influence of the dominant cluster size with a small simulation experiment. The size of the document set to cluster was fixed to $10000$ documents, while the size of the dominant cluster was varied between $1$ and $10000$ with a step size of $250$. Once the dominant cluster is fixed, cluster sizes, with the restriction of being smaller than the dominant cluster, are randomly generated until the number of $10000$ documents is reached. This process is repeated $10000$ times for each dominant cluster size. Figure 2.5 contains the minimum, average and maximum

*SetCoherence* for each dominant cluster size. When $50\%$ of all documents belong to the dominant cluster, *SetCoherence* $= 0.37$ on average, although one might expect a higher score as half of all possible documents belong to a single cluster.

In the work by He et al. [73], the documents are vectors and $\sigma$ is the cosine similarity. The similarity threshold $\theta$ is set heuristically by averaging the top $5\%$ of similarity scores from randomly sampled sets of documents. $AvQC$ is the average set coherence over all query terms.

Additionally, the following constraint is added to *AvQCG*:

$$AvQCG = SetCoherence(N_{\mathbf{q}}) \times AvQC.$$

Here, *AvQC* is multiplied by the global set coherence, that is the coherence of the set of documents that contains any of the query terms: $N_{\mathbf{q}} = \cup_{i=1}^{m} N_{q_i}$. If $N_{\mathbf{q}}$ is large (in [73] the limit of 10000 documents is given for the AP88 & 89 corpus), the global set coherence is approximated by the threshold $\theta$. In particular, for longer queries with a high number of general terms it can be expected that the global set coherence is close to constant for a set of queries, as in almost all cases $\theta$ will be used as global set coherence. A similar result is expected for a query set from a large corpus. The GOV2 corpus contains $25$ million documents and even specific terms will often appear in more than $10000$ documents. We implemented the proposed method to the highest precision degree possible, as the original publication did not disclose all details. For the newspaper corpus, the limit was set to $10000$ documents, for the WT10g corpus the limit was increased to $20000$ documents and for the GOV2 collection it was set to $50000$ documents.

*AvQC* and *AvQCG* both require a great amount of computation; determining the document similarity between all document pairs of a collection for example is not feasible, samples have to be drawn instead.

### 2.8.2 Ambiguity as Covered by WordNet

WordNet [57] is an online lexical database developed at Princeton University, inspired by psycholinguistic theories. It is continuously enlarged and updated by human experts and can be viewed as a general domain knowledge base. WordNet's building blocks are sets of synonymous terms[3], called *synsets*, each representing a lexical concept and each connected to others through a range of semantic relationships. Relations between terms instead of synsets exist as well but are not very frequent. WordNet also provides glosses, which are example sentences and definitions for the synsets. Relationships exist mainly between synsets of the same word type; there are separate structures for nouns, verbs, adjectives and adverbs. Notably, nouns make up by far the largest fraction of WordNet.

The number of WordNet senses of a term is an indicator of its ambiguity - the more senses a term has, the more ambiguous it is. For example, the term *"go"* has a total of thirty-five senses in WordNet[4]. These are, four noun senses, one adjective

[3] A term can be a single word, a compound or a phrase.

[4] All figures are based on WordNet version 3.0.

sense and thirty verb senses. On the other hand, *"Agoraphobia"* has a single noun sense and thus is considered to be unambiguous. A limiting factor of WordNet is the fact that it is a general knowledge semantic dictionary and therefore it is only useful for a general collection of documents. Additionally, WordNet also contains rare senses of terms, which may not appear at all in a corpus, while many proper nouns that do appear in a corpus may not be a part of WordNet.

For each synset, WordNet provides a gloss of varying length. Take for example the concepts "viral hepatitis" and "aspirin" from TREC description queries. The gloss of the former is: "hepatitis caused by a virus" whereas "aspirin" is described as follows: "the acetylated derivative of salicylic acid; used as an analgesic anti-inflammatory drug (trade names Bayer, Empirin, and St. Joseph) usually taken in tablet form; used as an antipyretic; slows clotting of the blood by poisoning platelets".

The *AvP* [111] value is derived from WordNet in the following way. Initially, each query is tokenized and mapped to WordNet terms. Since WordNet contains phrases such as "organized crime", the matching is first performed based on a window of five terms, then four terms and so on, with morphological variations also being tested. Then the number of senses of each phrase found is recorded - a term that is not found in WordNet and is not part of a WordNet phrase, is assigned a single sense. Finally, the average number of senses over all found phrases/terms is calculated. For example, TREC title topic "black bear attacks" is WordNet tokenized into {*black bear, attack*}. The phrase "black bear" has two senses, while "attack" has fifteen senses, and therefore $AvP = 8.5$. For comparison purposes, we also evaluate *AvNP*, which is similar to *AvP* but it only considers the noun senses instead of the senses over all word types.

### 2.8.3 Experimental Evaluation

This segment contains the results of the evaluation of the presented ambiguity based predictors. The presentation of numerical findings is accompanied by a discussion on the causes of discovered differences.

#### Predictor-Predictor Correlations

The correlation between the *AvQC* and *AvQCG* predictors is high across all three corpora. With increased collection size, the correlation approaches one, specifically $\tau = 0.87$ for the queries of TREC Vol. 4+5 and $\tau = 0.98$ for the queries of the GOV2 corpus. The two WordNet based prediction methods are less highly correlated, reaching $\tau = 0.8$ at best. The correlation between the WordNet based and the collection based predictors is moderately negative for TREC Vol. 4+5 and approximately zero for the queries of WT10g and GOV2. The negative correlation can be attributed to the fact that the more WordNet senses the query terms have, the lower the quality of the query, whereas the collection based predictors predict a higher quality with increased score.

### Predictor Evaluation

The results in Table 2.8 show that the two WordNet based predictors (*AvP* and *AvNP*) generally perform very poorly; only for query sets 301-350 and 451-500 do they exhibit meaningful negative correlations across the range of retrieval methods. The reason for this failure can be attributed, in part, to the fact that the TREC title topics of the WT10g and the GOV2 corpus contain a significant number of proper nouns such as "Chevrolet", "Skoda", "Peer Gynt", "Nirvana", "John Edwards" and "TMJ" which are not part of WordNet. As these terms and phrases often make up the most important or even the sole part of a title topic, the results become unusable. A second reason for the discrepancy is rooted in the collection size and makeup. Arguably, the newswire corpus (TREC Vol. 4+5) employs a limited vocabulary and reasonably structured prose, while the newer Web and Terabyte corpora contain a more diverse vocabulary with more noise (frequent use of esoteric or non-sensical words and phrases). In such cases, WordNet does not provide an accurate sense count.

| | | *AvP* | *AvNP* | *AvQC* | *AvQCG* |
|---|---|---|---|---|---|
| 301-350 | **TF.IDF** | −0.283 | −0.354 | 0.545 | **0.584** |
| | **Okapi** | −0.360 | −0.467 | **0.487** | 0.436 |
| | $\mu = 500$ | −0.347 | −0.445 | **0.449** | 0.404 |
| 351-400 | **TF.IDF** | −0.329 | −0.290 | 0.525 | **0.611** |
| | **Okapi** | −0.104 | −0.160 | 0.192 | **0.245** |
| | $\mu = 2000$ | −0.131 | −0.153 | 0.238 | **0.273** |
| 401-450 | **TF.IDF** | 0.009 | −0.110 | **0.732** | 0.551 |
| | **Okapi** | 0.125 | 0.022 | **0.611** | 0.365 |
| | $\mu = 1000$ | 0.076 | −0.069 | **0.627** | 0.390 |
| 301-450 | **TF.IDF** | −0.187 | −0.248 | **0.591** | 0.475 |
| | **Okapi** | −0.115 | −0.211 | **0.456** | 0.316 |
| | $\mu = 1000$ | −0.121 | −0.220 | **0.457** | 0.330 |
| 451-500 | **TF.IDF** | −0.305 | −0.253 | **0.657** | 0.544 |
| | **Okapi** | **−0.303** | −0.262 | 0.208 | 0.091 |
| | $\mu = 1000$ | **−0.305** | −0.246 | 0.138 | −0.047 |
| 501-550 | **TF.IDF** | −0.282 | −0.247 | **0.512** | 0.394 |
| | **Okapi** | −0.064 | 0.063 | **0.154** | 0.056 |
| | $\mu = 2000$ | 0.015 | 0.109 | **0.210** | 0.052 |
| 451-550 | **TF.IDF** | −0.289 | −0.247 | **0.579** | 0.460 |
| | **Okapi** | **−0.210** | −0.141 | 0.195 | 0.088 |
| | $\mu = 1000$ | **−0.183** | −0.108 | 0.162 | −0.021 |
| 701-750 | **TF.IDF** | 0.075 | 0.027 | **0.221** | 0.020 |
| | **Okapi** | 0.047 | −0.068 | **0.177** | 0.131 |
| | $\mu = 1000$ | 0.050 | 0.007 | **0.253** | 0.104 |
| 751-800 | **TF.IDF** | −0.184 | −0.170 | 0.145 | **0.206** |
| | **Okapi** | 0.130 | −0.042 | **0.371** | 0.182 |
| | $\mu = 1000$ | 0.014 | −0.040 | **0.410** | 0.164 |
| 801-850 | **TF.IDF** | −0.188 | −0.224 | 0.384 | **0.427** |
| | **Okapi** | −0.014 | −0.119 | **0.274** | 0.010 |
| | $\mu = 1000$ | −0.038 | −0.111 | **0.265** | 0.006 |
| 701-850 | **TF.IDF** | −0.124 | −0.124 | **0.252** | 0.214 |
| | **Okapi** | 0.075 | −0.049 | **0.263** | 0.060 |
| | $\mu = 1000$ | 0.017 | −0.035 | **0.298** | 0.048 |

(a) Linear correlation coefficient $r$

| | | *AvP* | *AvNP* | *AvQC* | *AvQCG* |
|---|---|---|---|---|---|
| 301-305 | **TF.IDF** | −0.283 | −0.329 | 0.483 | **0.503** |
| | **Okapi** | −0.314 | **−0.375** | 0.370 | 0.374 |
| | $\mu = 500$ | −0.334 | **−0.371** | 0.350 | 0.347 |
| 351-400 | **TF.IDF** | −0.208 | −0.186 | 0.297 | **0.318** |
| | **Okapi** | −0.039 | −0.094 | 0.181 | **0.209** |
| | $\mu = 2000$ | −0.046 | −0.065 | 0.213 | **0.241** |
| 401-450 | **TF.IDF** | −0.129 | −0.128 | 0.382 | **0.412** |
| | **Okapi** | 0.029 | 0.032 | **0.341** | 0.311 |
| | $\mu = 1000$ | 0.007 | −0.014 | **0.385** | 0.352 |
| 301-450 | **TF.IDF** | −0.210 | −0.227 | 0.397 | **0.421** |
| | **Okapi** | −0.107 | 0.152 | 0.295 | **0.298** |
| | $\mu = 1000$ | −0.118 | −0.157 | 0.296 | **0.301** |
| 451-500 | **TF.IDF** | −0.292 | −0.185 | **0.540** | 0.529 |
| | **Okapi** | −0.271 | −0.226 | **0.293** | 0.276 |
| | $\mu = 1000$ | −0.195 | −0.143 | **0.260** | 0.246 |
| 501-550 | **TF.IDF** | −0.247 | −0.176 | 0.418 | **0.423** |
| | **Okapi** | −0.053 | 0.053 | **0.098** | 0.093 |
| | $\mu = 2000$ | −0.044 | 0.084 | **0.152** | 0.147 |
| 451-550 | **TF.IDF** | −0.261 | −0.194 | **0.458** | 0.454 |
| | **Okapi** | −0.153 | 0.088 | **0.195** | 0.178 |
| | $\mu = 1000$ | −0.114 | 0.044 | **0.210** | 0.195 |
| 701-750 | **TF.IDF** | 0.111 | 0.118 | **0.264** | **0.264** |
| | **Okapi** | 0.029 | −0.027 | **0.163** | **0.163** |
| | $\mu = 1000$ | 0.031 | 0.010 | 0.279 | **0.287** |
| 751-800 | **TF.IDF** | −0.110 | −0.045 | 0.288 | **0.298** |
| | **Okapi** | −0.031 | −0.046 | **0.268** | 0.258 |
| | $\mu = 1000$ | −0.015 | 0.007 | **0.280** | 0.267 |
| 801-850 | **TF.IDF** | −0.171 | −0.208 | 0.272 | **0.298** |
| | **Okapi** | 0.005 | −0.093 | 0.247 | **0.251** |
| | $\mu = 1000$ | 0.027 | −0.038 | **0.268** | 0.249 |
| 701-850 | **TF.IDF** | −0.059 | −0.029 | 0.269 | **0.282** |
| | **Okapi** | 0.017 | −0.026 | **0.232** | 0.226 |
| | $\mu = 1000$ | 0.027 | 0.014 | **0.273** | 0.265 |

(b) Kendall's $\tau$

Table 2.8: Correlation coefficients of ambiguity based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

The predictor performances of *AvQC* and *AvQCG* are mixed and greatly depend on the particular collection. The best performance is achieved for the query sets of TREC Vol. 4+5. For the query sets of WT10g and GOV2, in many instances only insignificant correlation coefficients are achieved. This might be due to the fact that *AvQC* is geared towards smaller collections or that our parameter settings were not optimal. Note that thorough calibration of parameters has not been conducted during this work, given time constraints and the great computational requirement of this method.

With respect to the level of smoothing in the Language Modeling approach, the two clustering based predictors *AvQC* and *AvQCG* show considerable performance increases over the three corpora when the amount of smoothing is increased. On query set 701-850 for instance, *AvQC* reaches $\tau_{\mu=100} = 0.27$ for low amounts of smoothing; however, when $\mu$ is raised to the maximum, $\tau$ reaches $0.39$. The two WordNet based predictors on the other hand show hardly any change in correlation with changing amounts of smoothing.

## 2.9 Term Relatedness

The previously introduced specificity and ambiguity based predictors ignore an important aspect of the query, namely the relationship between the query terms. Consider for example the two queries $\mathbf{q_1} = \{American, football\}$ and $\mathbf{q_2} = \{foot, porch\}$. Specificity based predictors might predict $\mathbf{q_2}$ to be an easier query because the terms *foot* and *porch* might occur less frequently than *American* and *football*. However, in a general corpus one would expect $\mathbf{q_1}$ to be an easier query for a retrieval system than $\mathbf{q_2}$ due to the strong relationship between the two query terms. Term relatedness measures predict a query to perform well, if there is a measurable relationship between query terms. The degree of relationship can either be derived from co-occurrence statistics of the collection or from WordNet based measures that determine the degree of *semantic* relatedness.

The predictors surveyed in this section are:

- Averaged Pointwise Mutual Information (*AvPMI*),
- Maximum Pointwise Mutual Information (*MaxPMI*),
- Averaged Path Length (*AvPath*) [124],
- Averaged Lesk Relatedness (*AvLesk*) [15], and,
- Averaged Vector Pair Relatedness (*AvVP*) [116].

A drawback of these predictors is, that queries consisting of a single term will be assigned a score of zero, as in such cases no relatedness value can be derived. If a significant number of queries in the query set used for evaluation are single term queries, the correlation will be lower than what the actual quality of the predictor implies.

### 2.9.1 Collection Based Relatedness

Predictors that exploit co-occurrence statistics of the collection are more precise than those based on standard deviations such as *DevIDF*. *AvPMI* and *MaxPMI* both rely on the concept of pointwise mutual information, which for two terms $q_i$ and $q_j$ is defined by:

$$PMI(q_i, q_j) = \log_2 \frac{P_{ml}(q_i, q_j)}{P_{ml}(q_i)P_{ml}(q_j)}.$$

The nominator is the probability that the two terms occur together in a document; the denominator is the probability of them occurring together by chance. If $P_{ml}(q_i, q_j) \approx P_{ml}(q_i)P_{ml}(q_j)$, the terms are independent and $PMI \approx 0$. Query terms that co-occur significantly more often than by chance lead to a high *PMI* value. *AvPMI* is the average over all *PMI* scores across all query term pairs, while *MaxPMI* is the maximum $PMI$ score across all query term pairs.

### 2.9.2 WordNet Based Relatedness

As an alternative to the collection statistics based methods, the degree of relatedness of query terms can also be determined by exploiting the graph structure of WordNet. In general, the closer two terms are in the WordNet graph, the higher their semantic similarity. Diverse WordNet based measures exist; in this work, we evaluate three measures as pre-retrieval predictors.

*AvPath*, initially proposed by Rada et al. [124], determines the relatedness between two terms by the reciprocal of the number of nodes on the shortest path of the IS-A hierarchy between the two corresponding synset nodes. Since the IS-A relationship is defined on the noun graph, the measure ignores all non-noun query terms. The maximum relatedness score is one (two identical synsets) and the minimum is zero (no path between two synsets). The average over all query term pair scores is then utilized as *AvPath* score[5].

*AvLesk* [14] is a relatedness measure that exploits the gloss overlap between two synsets, as well as the glosses of their related synsets. Generally, the more terms the glosses have in common, the more related the two synsets are.

Finally, *AvVP*, introduced by Patwardhan and Pedersen [116], is a measure where each synset is represented as a second-order co-occurrence vector of glosses, including the glosses of related synsets. Relatedness, in this case, is the cosine similarity between the gloss vectors.

The three measures just described rely on synsets instead of terms. In a practical applications, it would be necessary to first disambiguate the query terms and then to locate the correct synset in WordNet. Since in this experiment were are interested in the general feasibility of WordNet based relatedness measures, in a preprocessing step we manually disambiguated the query terms and identified the correct synset. A number of proper nouns in the queries could not be matched and had to be ignored in the relatedness calculations.

[5] All WordNet based predictors were calculated with the WordNet::Similarity package available at http://wn-similarity.sourceforge.net/.

### 2.9.3 Experimental Evaluation

#### Predictor-Predictor Correlations

The two collection based predictors are naturally highly correlated as one relies on the average and the other on the maximum of *PMI* scores. Moreover, in instances where a query consists of one or two terms only, both predictors produce exactly the same score. Intuitively, larger differences between the two predictors can be expected for queries derived from TREC description topics. With respect to the different corpora, a clear trend can be discerned: the larger the corpus, the more the correlation between *AvPMI* and *MaxPMI* degrades. While for the queries 301-450 of TREC Vol. 4+5 the correlation reaches $\tau = 0.80$ ($r = 0.91$), for the queries 701-850 of the GOV2 corpus, the correlation degrades to $\tau = 0.63$ ($r = 0.74$). The correlations between the WordNet and the corpus based measures are low, yet significant, for the queries of TREC Vol. 4+5 and WT10g; the correlation is close to zero for the queries of the GOV2 corpus. A comparison of the WordNet based measures with each other, yields eratic results, none of them are consistently highly correlated to each other.

#### Predictor Evaluation

Table 2.9 shows the quality of the five algorithms as query effectiveness predictors. *AvPMI* and *MaxPMI* exhibit significant correlations across all collections for the Okapi and Language Modeling approaches, although with respect to the best performing specificity and ambiguity based predictors the correlations are relatively low. The WordNet based predictors have a significant linear correlation for queries 301-350. However, for the same corpus the query set 401-450 leads to negative correlations, which, due to their unreliability, renders these WordNet based predictors unusable, even for the smallest of the evaluated corpora.

The influence of the smoothing parameter $\mu$ is corpus dependent, but not particularly pronounced. For TREC Vol. 4+5, increasing the amount of smoothing also increases the correlation coefficients. In the case of queries 301-450, for instance, consider $\tau_{\mu=100} = 0.22$ for *AvPMI*, which, when the level of smoothing is increased, becomes $\tau_{\mu=2\times10^5} = 0.29$. In contrast, for the WT10g corpus, both *AvPMI* and *MaxPMI* show consistent degradation in correlation with increased smoothing. Lastly, for the GOV2 corpus increasing $\mu$ leads to slightly increased correlation coefficients. The development of the WordNet based measures is similarly mixed – slight improvements and degradations depending on the query set. However, apart from the query sets of TREC Vol. 4+5, the correlation coefficients are not significantly different from zero.

## 2.10 Significant Results

The previous sections have provided a comprehensive and detailed overview of a number of pre-retrieval predictors. The extensive evaluation that followed each

| | | *AvPMI* | *MaxPMI* | *AvPath* | *AvLesk* | *AvVP* |
|---|---|---|---|---|---|---|
| 301-350 | **TF.IDF** | 0.297 | 0.295 | 0.253 | **0.327** | 0.318 |
| | **Okapi** | 0.314 | 0.315 | 0.312 | **0.413** | 0.411 |
| | $\mu = 500$ | 0.316 | 0.298 | 0.294 | 0.374 | **0.411** |
| 351-400 | **TF.IDF** | **0.326** | 0.196 | 0.120 | 0.142 | −0.004 |
| | **Okapi** | **0.331** | 0.203 | 0.050 | 0.210 | 0.192 |
| | $\mu = 2000$ | **0.376** | 0.234 | 0.005 | 0.219 | 0.254 |
| 401-450 | **TF.IDF** | 0.163 | 0.108 | **−0.246** | −0.127 | −0.165 |
| | **Okapi** | **0.401** | 0.371 | −0.247 | −0.014 | −0.238 |
| | $\mu = 1000$ | **0.438** | 0.398 | −0.240 | −0.014 | −0.214 |
| 301-450 | **TF.IDF** | **0.275** | 0.230 | 0.155 | 0.252 | 0.170 |
| | **Okapi** | **0.336** | 0.292 | 0.151 | 0.278 | 0.237 |
| | $\mu = 1000$ | **0.353** | 0.295 | 0.135 | 0.252 | 0.253 |
| 451-500 | **TF.IDF** | **−0.258** | −0.224 | −0.084 | 0.076 | −0.037 |
| | **Okapi** | **0.199** | 0.152 | 0.004 | −0.034 | −0.087 |
| | $\mu = 1000$ | **0.288** | 0.199 | −0.033 | −0.022 | −0.073 |
| 501-550 | **TF.IDF** | −0.150 | −0.178 | **−0.242** | −0.133 | −0.049 |
| | **Okapi** | 0.176 | **0.292** | 0.124 | 0.243 | 0.005 |
| | $\mu = 2000$ | 0.235 | **0.403** | 0.150 | 0.104 | −0.057 |
| 451-550 | **TF.IDF** | **−0.212** | −0.201 | −0.151 | 0.032 | −0.032 |
| | **Okapi** | **0.196** | 0.195 | 0.041 | 0.001 | 0.068 |
| | $\mu = 1000$ | **0.285** | 0.269 | 0.033 | 0.006 | −0.058 |
| 701-750 | **TF.IDF** | 0.250 | **0.262** | 0.010 | −0.106 | −0.059 |
| | **Okapi** | 0.276 | **0.333** | −0.101 | 0.066 | 0.064 |
| | $\mu = 1000$ | 0.431 | **0.436** | −0.118 | 0.035 | 0.057 |
| 751-800 | **TF.IDF** | −0.044 | −0.072 | −0.020 | −0.069 | **−0.112** |
| | **Okapi** | **0.425** | 0.296 | −0.116 | −0.039 | −0.089 |
| | $\mu = 1000$ | **0.456** | 0.353 | −0.089 | 0.019 | 0.016 |
| 801-850 | **TF.IDF** | 0.661 | 0.545 | 0.875 | **0.916** | 0.885 |
| | **Okapi** | 0.116 | **0.188** | 0.091 | 0.112 | 0.127 |
| | $\mu = 1000$ | 0.076 | **0.203** | 0.097 | 0.098 | 0.102 |
| 701-850 | **TF.IDF** | 0.320 | 0.225 | 0.481 | **0.509** | 0.355 |
| | **Okapi** | 0.247 | **0.257** | 0.009 | 0.064 | 0.043 |
| | $\mu = 1000$ | 0.277 | **0.302** | 0.025 | 0.069 | 0.071 |

(a) Linear correlation coefficient $r$

| | | *AvPMI* | *MaxPMI* | *AvPath* | *AvLesk* | *AvVP* |
|---|---|---|---|---|---|---|
| 301-350 | **TF.IDF** | 0.288 | 0.295 | 0.022 | 0.169 | 0.024 |
| | **Okapi** | 0.191 | 0.236 | −0.029 | 0.185 | 0.059 |
| | $\mu = 500$ | 0.176 | 0.218 | −0.037 | 0.191 | 0.039 |
| 351-400 | **TF.IDF** | 0.221 | 0.199 | 0.014 | 0.141 | 0.074 |
| | **Okapi** | 0.247 | 0.252 | 0.070 | 0.202 | 0.089 |
| | $\mu = 2000$ | 0.290 | 0.287 | 0.054 | 0.188 | 0.088 |
| 401-450 | **TF.IDF** | 0.250 | 0.164 | −0.138 | −0.144 | −0.140 |
| | **Okapi** | 0.234 | 0.206 | −0.097 | −0.025 | −0.210 |
| | $\mu = 1000$ | 0.232 | 0.195 | −0.100 | −0.046 | −0.219 |
| 301-450 | **TF.IDF** | 0.219 | 0.219 | −0.033 | 0.062 | −0.021 |
| | **Okapi** | 0.228 | 0.229 | −0.015 | 0.120 | −0.025 |
| | $\mu = 1000$ | 0.223 | 0.217 | −0.027 | 0.114 | −0.036 |
| 451-500 | **TF.IDF** | 0.018 | −0.037 | −0.174 | −0.144 | −0.176 |
| | **Okapi** | 0.140 | 0.163 | −0.051 | −0.062 | −0.065 |
| | $\mu = 1000$ | 0.208 | 0.213 | −0.030 | −0.071 | −0.027 |
| 501-550 | **TF.IDF** | −0.074 | −0.083 | −0.230 | −0.100 | −0.139 |
| | **Okapi** | 0.191 | 0.239 | 0.020 | 0.103 | −0.049 |
| | $\mu = 2000$ | 0.212 | 0.263 | 0.072 | 0.045 | −0.085 |
| 451-550 | **TF.IDF** | −0.039 | −0.066 | −0.192 | −0.123 | −0.134 |
| | **Okapi** | 0.149 | 0.179 | 0.001 | 0.000 | −0.045 |
| | $\mu = 1000$ | 0.204 | 0.225 | 0.029 | −0.026 | −0.040 |
| 701-750 | **TF.IDF** | 0.204 | 0.205 | 0.055 | 0.002 | −0.013 |
| | **Okapi** | 0.215 | 0.206 | −0.081 | 0.112 | 0.114 |
| | $\mu = 1000$ | 0.301 | 0.339 | −0.105 | 0.057 | 0.053 |
| 751-800 | **TF.IDF** | 0.034 | 0.076 | −0.011 | −0.036 | −0.141 |
| | **Okapi** | 0.270 | 0.259 | −0.121 | −0.078 | −0.044 |
| | $\mu = 1000$ | 0.314 | 0.302 | −0.117 | −0.027 | −0.040 |
| 801-850 | **TF.IDF** | 0.164 | 0.177 | 0.170 | 0.174 | 0.159 |
| | **Okapi** | 0.078 | 0.158 | 0.008 | 0.010 | 0.098 |
| | $\mu = 1000$ | 0.069 | 0.155 | 0.035 | 0.049 | 0.078 |
| 701-850 | **TF.IDF** | 0.118 | 0.146 | 0.077 | 0.045 | 0.009 |
| | **Okapi** | 0.189 | 0.186 | −0.050 | 0.059 | 0.077 |
| | $\mu = 1000$ | 0.215 | 0.227 | −0.046 | 0.050 | 0.050 |

(b) Kendall's $\tau$

Table 2.9: Correlation coefficients of term relatedness based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

category of predictors aimed to emphasize the strong dependency of the predictors on the retrieval approach, the collection and the particular query set.

What we have largely neglected so far, are a discussion of the significance of the correlations and the comparison of predictor performances across all categories. In this section, we address both issues. Testing the significance of a correlation coefficient can be considered from two angles. On the one hand, we need to test, whether a recorded correlation coefficient is significantly different from zero. While this test is performed and acknowledged in publications, it is usually neglected to test, whether a predictor is significantly different from the best performing predictor. As in retrieval experiments, where we routinely evaluate the significance of the difference between two retrieval approaches, we should do the same in the evaluation of query performance prediction.

In Table 2.10, we summarize the results of the significance tests. For each collection, all predictors, that result in a correlation significantly different from zero for both the linear correlation and Kendall's $\tau$ coefficient, are listed. Additionally, we report the confidence intervals ($\alpha = 0.95$) of the coefficients. As the retrieval

approach to predict for, Language Modeling with Dirichlet smoothing and the best setting of $\mu$ was used (Table 2.2). Presented in bold, is the best predictor for each collection. Given the best performing predictor, all other predictors were tested for their statistical difference; underlined are those predictors for which no significant difference ($\alpha = 0.95$) was found.

With respect to the linear correlation coefficient $r$, *MaxIDF* is the best predictor for the query sets of TREC Vol. 4+5. When determining the significance of the difference, six other prediction methods show no significant difference, including *AvICTF*, *AvQC* and *MaxVAR*. Thus, apart from the relatedness predictors, all categories provide useful predictors. When considering Kendall's $\tau$, *MaxVAR* is the best performing predictor. It is, however, not significantly different from *MaxIDF*. Note, that although *AvVAR* exhibits a higher Kendall's $\tau$ than *MaxIDF*, its correlation is significantly worse than *MaxVAR*'s correlation. This is due to the fact, that the correlation between *AvVAR* and *MaxVAR* is larger ($\tau = 0.72$) than between *MaxIDF* and *MaxVAR* ($\tau = 0.53$).

For the query sets of WT10g, *MaxVAR* also reports the highest linear correlation with $r = 0.41$, but again the significance test shows, most predictors which achieve a significant correlation (different from zero) are not significantly different from the best performing predictor. It is notable that in this corpus, none of the ambiguity based predictors are significantly different from zero. The situation is similar for Kendall's $\tau$; by absolute correlation scores *MaxSCQ* performs best, but apart from *DevIDF* all other predictors exhibit no significantly worse performance. When comparing the confidence intervals of TREC Vol. 4+5 and the WT10g corpus, it becomes apparent that the intervals are wider for WT10g. The reason is, that we only deal with a query set of size $100$ in this corpus, whereas TREC Vol. 4+5 are evaluated for $150$ queries. Hence, the more queries exist for evaluation purposes, the more reliable the correlation coefficient and thus the smaller the confidence interval.

Finally, Table 2.10 also shows that the GOV2 corpus is easier to predict for than the WT10g corpus; more predictors are significantly different from zero. The most accurate predictor is once more *MaxVAR*, although again, it can be shown that a variety of predictors are similarly useful.

## 2.11 Predictor Robustness

In the beginning of this chapter we stated that, ideally, since the pre-retrieval predictors are search independent, a robust predictor should be indifferent to the particular retrieval algorithm. Since the commonly relied upon retrieval models such as Okapi [125], Language Modeling [76, 97, 121, 170, 171], the Markov Random Field model [104] and the Divergence from Randomness model [5], are based exclusively on term and document frequencies, one might expect similar predictor performances across all of them. However, as seen in the previous sections, prediction methods are indeed sensitive to the retrieval approach as well as the specific parameter settings such as the level of smoothing $\mu$.

In the current section, we expand considerably on the variety of retrieval ap-

| | r | CI | $\tau$ | CI |
|---|---|---|---|---|
| *AvICTF* | 0.490 | [0.358,0.603] | 0.266 | [0.161,0.371] |
| *AvIDF* | 0.516 | [0.388,0.625] | 0.290 | [0.186,0.394] |
| *AvPMI* | 0.352 | [0.203,0.485] | 0.222 | [0.112,0.333] |
| *AvQC* | 0.457 | [0.320,0.575] | 0.292 | [0.193,0.391] |
| *AvQCG* | 0.330 | [0.179,0.465] | 0.297 | [0.199,0.395] |
| *AvSCQ* | 0.256 | [0.100,0.400] | 0.204 | [0.094,0.314] |
| *AvVAR* | 0.510 | [0.381,0.620] | 0.356 | [0.260,0.452] |
| *DevIDF* | 0.239 | [0.082,0.384] | 0.180 | [0.066,0.293] |
| *MaxIDF* | **0.532** | [0.407,0.638] | 0.339 | [0.237,0.442] |
| *MaxPMI* | 0.295 | [0.142,0.435] | 0.216 | [0.102,0.330] |
| *MaxSCQ* | 0.341 | [0.191,0.475] | 0.332 | [0.230,0.433] |
| *MaxVAR* | 0.513 | [0.384,0.622] | **0.411** | [0.316,0.505] |
| *QS* | 0.407 | [0.264,0.533] | 0.190 | [0.077,0.303] |
| *SCS* | 0.480 | [0.347,0.595] | 0.251 | [0.145,0.357] |
| *SumVAR* | 0.297 | [0.144,0.437] | 0.283 | [0.182,0.384] |
| *AvNP* | -0.220 | [-0.367,-0.062] | -0.145 | [-0.261,-0.029] |

(a) Queries 301-450 (TREC Vol. 4+5)

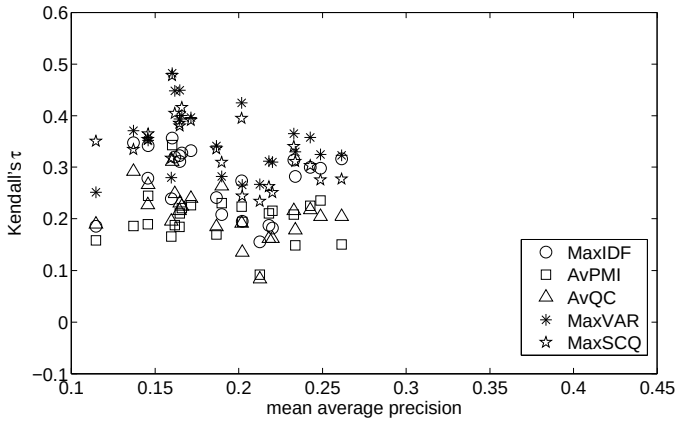
| | r | CI | $\tau$ | CI |
|---|---|---|---|---|
| *AvPMI* | 0.282 | [0.092,0.458] | 0.201 | [0.063,0.338] |
| *AvSCQ* | 0.307 | [0.116,0.477] | 0.227 | [0.098,0.356] |
| *AvVAR* | 0.292 | [0.099,0.463] | 0.249 | [0.123,0.375] |
| *DevIDF* | 0.233 | [0.037,0.413] | 0.154 | [0.016,0.292] |
| *MaxIDF* | 0.292 | [0.099,0.463] | 0.266 | [0.122,0.411] |
| *MaxPMI* | 0.269 | [0.075,0.444] | 0.221 | [0.096,0.393] |
| *MaxSCQ* | 0.400 | [0.218,0.554] | **0.322** | [0.195,0.448] |
| *MaxVAR* | **0.411** | [0.231,0.563] | 0.321 | [0.197,0.445] |
| *SumVAR* | 0.300 | [0.108,0.470] | 0.213 | [0.084,0.341] |

(b) Queries 451-550 (WT10g)

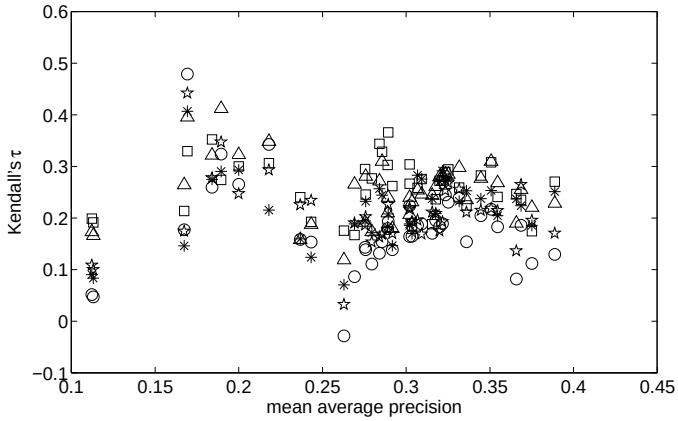
| | r | CI | $\tau$ | CI |
|---|---|---|---|---|
| *AvICTF* | 0.281 | [0.125,0.423] | 0.209 | [0.118,0.299] |
| *AvIDF* | 0.309 | [0.155,0.448] | 0.229 | [0.139,0.320] |
| *AvPMI* | 0.277 | [0.121,0.419] | 0.215 | [0.113,0.317] |
| *AvQC* | 0.298 | [0.144,0.439] | 0.240 | [0.154,0.325] |
| *AvSCQ* | 0.352 | [0.202,0.520] | 0.234 | [0.137,0.331] |
| *AvVAR* | 0.392 | [0.246,0.520] | 0.269 | [0.179,0.359] |
| *DevIDF* | 0.185 | [0.024,0.336] | 0.143 | [0.038,0.249] |
| *MaxIDF* | 0.331 | [0.179,0.468] | 0.243 | [0.147,0.340] |
| *MaxPMI* | 0.302 | [0.148,0.442] | 0.227 | [0.114,0.341] |
| *MaxSCQ* | 0.403 | [0.259,0.530] | 0.274 | [0.175,0.374] |
| *MaxVAR* | **0.412** | [0.269,0.538] | **0.280** | [0.183,0.376] |
| *QS* | 0.179 | [0.019,0.331] | 0.137 | [0.030,0.245] |
| *SCS* | 0.248 | [0.090,0.394] | 0.186 | [0.094,0.278] |
| *SumSCQ* | 0.235 | [0.076,0.382] | 0.179 | [0.061,0.297] |
| *SumVAR* | 0.337 | [0.186,0.472] | 0.241 | [0.127,0.355] |

(c) Queries 701-850 (GOV2)

Table 2.10: Prediction quality of all predictors significantly different from $0$ for both $r$ and $\tau$. The best performing predictor for each corpus with respect to $r$ or $\tau$ are in bold. Underlined are all predictors that are not significantly different from the best performing predictor.




(a) Title topics 351-400




(b) Title topics 751-800

Figure 2.6: Robustness behavior of pre-retrieval predictors on the automatic TREC title runs.

proaches under investigation. Specifically, we evaluate the predictor performances against retrieval runs submitted to TREC. The runs are restricted to those with a MAP above 0.1 and they are required to be automatic title runs (Appendix B.3.2). Since pre-retrieval predictors rely on the query terms to predict a query's quality, it would be an unfair comparison to include runs based on TREC topic description or narratives. For the same reason, we also exclude manual runs, as they have not necessarily a strong overlap with the query terms. Based on our experimental results in the earlier sections, we selected the five best performing predictors: *MaxIDF*, *AvPMI*, *AvQC*, *MaxVAR* and *MaxSCQ*. Incidentally, this means that a predictor of each category is evaluated. For these predictors, their correlations with all selected TREC runs are determined.

To aid understanding, consider Figure 2.6, where exemplary the results of title topics 351-400 and 751-800 are shown in the form of scatter plots. Each point in a plot indicates a particular correlation coefficient between a TREC run and a pre-retrieval predictor. Therein, the wide spread in predictor performance is clearly visible. For title topics 751-800 (Figure 2.6a) for instance, *MaxIDF* exhibits both the

highest ($\tau = 0.48$) and the lowest ($\tau = -0.03$) correlation, depending on the TREC run. In general, the highest correlations are achieved for rather poorly performing TREC runs, while the predictors' capabilities continuously degrade with increasing retrieval effectiveness of the runs. We speculate that this development is due to the more advanced retrieval approaches of the well performing runs, which do not only rely on term and document frequencies but possibly among others take into account n-grams and the hyperlink structure (for WT10g and GOV2). There are differences between the predictors though. *AvPMI*, for instance, is somewhat less susceptible to the above effects as its performance does not change considerably over the different TREC runs; however compared to other predictors the achieved correlations are lower.

Although not shown, we note that when considering the results over all title topic sets *MaxVAR* and *MaxSCQ* can be considered as the most stable; over most topic sets they exhibit the highest *minimum* correlation with all TREC runs. Note, though, that this stability is relative and the correlation range is still considerable, for example between $\tau = 0.27$ and $\tau = 0.49$ for *MaxSCQ* and title topics 401-450. Across all topic sets, the maximum correlation achieved by a predictor and TREC run is $\tau = 0.50$ (*MaxVAR*, title topics 401-450) and $r = 0.74$ (*MaxSCQ*, title topics 401-450) respectively. This implies that high correlations can be achieved, if pre-retrieval predictors are used with the "right" retrieval approach.

In our experiments, we determined the predicted scores from stemmed and stopworded indices. As such, if stemming and/or stopword removal did not occur in the submitted TREC runs, the results will not reflect the quality of the predictors accurately. In order to determine the influence of those two preprocessing steps on the reported predictor performances, six indices of TREC Vol. 4+5 were created, each with a different combination of stemming (Krovetz stemmer, Porter stemmer [122] and no stemmer) and stopwording (removal, no removal). Three observations could be made. First of all, the type of stemmer employed, that is Krovetz or Porter, is not of importance, the reported correlations change only slightly in both directions, depending on the prediction method. Secondly, stopword removal across all three stemming options leads to somewhat higher correlations than no stopword removal, though again the changes are minor. Finally, switching off stemming has the largest effect, the correlations degrade across all pre-retrieval predictors to a considerable degree, for instance *MaxVAR* degrades from $\tau = 0.41$ with Krovetz stemming and stopwording to $\tau = 0.34$ without stemming and no stopword removal. We conclude that our indices and the pre-retrieval scores derived from them are sufficiently good to draw conclusions about the methods' robustness. One influence that we have not tested though is influence of tokenization. There might still be some differences in performance.

## 2.12 Combining Pre-Retrieval Predictors

Despite the numerous methods proposed, little research has been performed on combining predictors in a principled way. In [174] the proposed predictors are lin-

early combined, and the best performing combination is reported. In this section, we explore if predictor combination with penalized regression methods, which have shown to perform well in analogous prediction scenarios in microarray data analysis [49, 129, 178], lead to better performing prediction methods.

To measure the algorithms' performance we make use of the $f_{norm}$ setup, which is commonly evaluated by reporting $r$. While this approach is reasonable for parameter-free methods, such as the predictors introduced earlier, it is problematic when combining different predictors. Combination methods have a higher degree of freedom and can thus fit the set of predictor/retrieval effectiveness values very well (see Section 2.3.2). Overfitting leads to a high value of $r$, while at the same time lowering the prediction accuracy, which is the accuracy of the predictor when predicting values of unseen queries. To avoid this issue, we adopt the methodology applied in machine learning [20] and report the *root mean squared error* (*RMSE*) derived from training a linear model on a training set and evaluating it on a separate test set.

### 2.12.1 Evaluating Predictor Combinations

Let $\hat{\mathbf{y}}$ be the predictions of $m$ queries and let $\mathbf{y}$ be the true effectiveness values, then the *RMSE* is given by:

$$RMSE = \sqrt{\frac{1}{m}\sum_i (y_i - \hat{y}_i)^2}. \tag{2.14}$$

Since $RMSE^2$ is the function minimized in linear regression, in effect, the pre-retrieval predictor with the highest linear correlation coefficient will have the lowest *RMSE*. This approach mixes training and test data - what we are evaluating is the fit of the predictor with the training data, while we are interested in the evaluation of the predictor given novel queries. Ideally, we would perform regression on the training data to determine the model parameters and then use the model to predict the query performance on separate test queries. However, due to the very limited query set size, this is not feasible, and cross-validation is utilized instead: the query set is split into $k$ partitions, where the model is tuned on $k-1$ partitions and the $k^{th}$ partition functions as test set. This process is repeated for all $k$ partitions and the overall *RMSE* is reported.

### 2.12.2 Penalized Regression Approaches

Modeling a continuous dependent variable $\mathbf{y}$, which in our case is a vector of average precision values, as a function of $p$ independent predictor variables $\mathbf{x}_i$ is referred to as multiple regression. If we also assume a linear relationship between the variables, we refer to it as multiple linear regression. Given the data $(\mathbf{x}^i, y_i)$, $i = 1, 2, .., m$ and $\mathbf{x}^i = (x_{i1}, ..., x_{ip})^T$, the parameters $\beta = (\beta_1, ..., \beta_p)$ of the model $\mathbf{y} = \mathbf{X}\beta + \epsilon$ are to be estimated. $\mathbf{X}$ is the $m \times p$ matrix of predictors and $\epsilon$ is the vector of errors, which are assumed to be normally distributed.

The ordinary least squares (OLS) estimates of $\beta$ are derived by minimizing the squared error of the residuals: $\sum_i (y_i - \hat{y}_i)^2$, where $\hat{y}_i = \sum_j \beta_j x_{ij}$. The two drawbacks

of OLS are the low prediction accuracy due to overfitting and the difficulty of model interpretation. All predictors remain in the model and very similar predictors might occur with very different coefficients. If we have a large number of predictors, methods are preferred that perform automatic model selection, thereby only introducing the most important subset of predictors into the model. While this has not yet been explored in the context of query effectiveness prediction, it has received considerable attention among others in microarray data analysis [49, 129, 178] where good results have been reported with penalized regression approaches. As the problems in both areas are similar (very small data sets, possibly many predictors) it appears sensible to attempt to apply those methods to query performance prediction.

Penalized regression approaches place penalties on the regression coefficients $\beta$ to keep the coefficients small or exactly zero which essentially removes a number of predictors from the model. The least absolute shrinkage and selection operator (LASSO) [138] is such a method:

$$LASSO(\hat{\beta}) = \arg\min\Big\{\sum_{i=1}^{m}\big(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\big)^2\Big\} \text{ subject to } \sum_{j=1}^{p}|\beta_j| \leq t. \qquad (2.15)$$

The total weight of the coefficients is restricted by the tuning parameter $t \geq 0$. If a number of predictors are very similar, LASSO tends to include only one of them in the final model whereas the Elastic Net [179] has a grouping effect such that highly correlated predictors acquire similar coefficients. It relies on a penalty combination of the squared and absolute sum of beta coefficients.

LASSO is a special case of the later developed least angle regression (LARS) [54]. LARS determines the full regularization path: in each step, LARS selects the predictor that is most highly correlated with the residuals $\mathbf{y} - \hat{\mathbf{y}}$ of the current model, resulting in a $p \times p$ matrix of beta coefficients. In our experiments, such regularization paths were derived for LASSO, LARS and the Elastic Net. The question remains, which vector of beta coefficients from the matrix to choose as model coefficients. Several stopping criteria exist. *Traps* are randomly generated predictors that are added to the set of predictors. The regularization is stopped, as soon as one of the random predictors is picked to enter the model. An alternative is cross-validation: the beta coefficients are learned from $k-1$ partitions of the training data and the $k^{th}$ partition is used to calculate the error; the vector of beta coefficients with the smallest error is then chosen. A third possibility is the recently proposed bootstrapped LASSO (BOLASSO) [11], where a number of bootstrap samples are generated from the training data, the matrix of beta coefficients of LASSO are determined for each sample and in the end, only those predictors with non-zero coefficients in all bootstrap samples are utilized in the final model.

We investigate four variations of these approaches: LARS with traps as stopping criterion (LARS-Traps), LARS with cross-validation to determine the beta coefficients (LARS-CV), BOLASSO and the Elastic Net.

### 2.12.3 Experiments and Results

All predictors described in the previous sections were utilized, with the exception of the WordNet based term relatedness predictors, as they exhibited no predictive power over any of the corpora. As retrieval approach to predict for, we relied on Language Modeling with Dirichlet smoothing and the best performing $\mu$ (Table 2.2). The parameter settings of the Elastic Net were taken from [179]. LARS-Traps was tested with 6 randomly generated traps while LARS-CV was set up with 10-fold cross validation. BOLASSO was used with 10 bootstrapped samples and each sample was cross-validated to retrieve the best beta coefficient vector.

| | **TREC Vol. 4+5** | **WT10g** | **GOV2** |
|---|---|---|---|
| *AvPMI* | 0.207 | 0.191 | 0.187 |
| *AvQC* | 0.191 | 0.196 | 0.184 |
| *AvQL* | 0.215 | 0.197 | 0.194 |
| *MaxIDF* | 0.181 | 0.187 | 0.181 |
| *MaxSCQ* | 0.205 | 0.184 | 0.178 |
| *MaxVAR* | 0.182 | 0.184 | 0.176 |
| *AvP* | 0.214 | 0.195 | 0.194 |

Table 2.11: Performance of selected pre-retrieval predictors given in *RMSE*.

For evaluation purposes, the *RMSE* of all methods was determined by leave-one-out cross validation, where each query is once assigned as test set and the model is trained on all other queries. This setting is sensible due to the small query set size with a maximum of 150. To emphasize the cross validation *RMSE* approach being different from $r/CI$ established on the training set only, we write $r_{train}$ and $CI_{train}$.

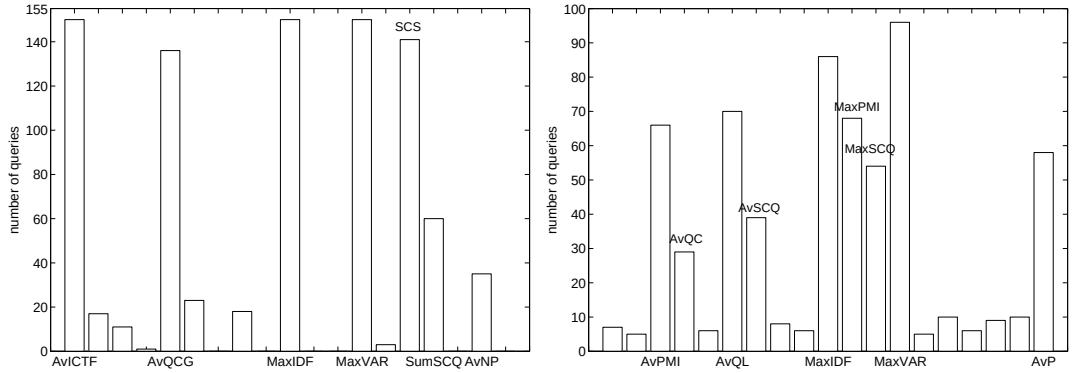
In order to provide a comparison between the combination methods and the constituent predictors, for a number of best and worst (*AvQL*, *AvP*) performing predictors, we list their *RMSE* in Table 2.11. In this set of experiments, we summarized all queries of each corpus.

The penalized regression results are reported in Table 2.12 along with $r_{train}$ and $CI_{train}$. For illustration, the predictors selected for LARS-Traps and LARS-CV are shown in the form of histograms in Figure 2.7. The bars indicate in how many of the $m$ times the algorithm run each predictor was selected to be in the model.

| | **TREC Vol. 4+5** | | | **WT10g** | | | **GOV2** | | |
|---|---|---|---|---|---|---|---|---|---|
| | $r_{train}$ | $CI_{train}$ | *RMSE* | $r_{train}$ | $CI_{train}$ | *RMSE* | $r_{train}$ | $CI_{train}$ | *RMSE* |
| *OLS* | **0.69** | [ 0.60, 0.77] | 0.188 | **0.64** | [0.51, 0.74] | 0.208 | **0.52** | [ 0.39, 0.63] | 0.190 |
| *LARS-Traps* | **0.59** | [ 0.47, 0.68] | **0.179** | **0.52** | [0.36, 0.65] | 0.187 | **0.44** | [ 0.30, 0.56] | 0.178 |
| *LARS-CV* | **0.68** | [ 0.59, 0.76] | 0.183 | **0.53** | [0.38, 0.66] | **0.178** | **0.46** | [ 0.33, 0.58] | 0.184 |
| *BOLASSO* | **0.59** | [ 0.47, 0.68] | **0.181** | **0.43** | [0.25, 0.58] | 0.198 | **0.43** | [ 0.28, 0.55] | 0.180 |
| *Elastic Net* | **0.69** | [ 0.60, 0.77] | 0.182 | **0.52** | [0.35, 0.65] | **0.182** | **0.46** | [ 0.32, 0.57] | 0.178 |

Table 2.12: Results of the penalized regression approaches. In bold the improvements over the best single predictor per collection are shown.

While the correlation coefficient $r$ suggests that the combined methods perform better than the single predictors, when we examine the results of the stronger *RMSE* based evaluation methodology, different conclusions can be drawn. There is a relatively small difference in error of predicted average precision in cases were the correlation coefficients appear quite distinct, e.g. $r_1 = 0.18$ and $r_2 = 0.41$ lead to $RMSE_1 = 0.184$ and $RMSE_2 = 0.194$ respectively. Although the penalized regression approaches have a lower error than the OLS baseline as expected, the decrease in error compared to the single predictors is smaller than one might expect. In fact, on the GOV2 corpus the error increased.




(a) Queries 301-450 (TREC Vol. 4+5), LARS-Traps

(b) Queries 451-550 (WT10g), LARS-CV

Figure 2.7: Penalized regression: predictor selection.

The histograms show that as desired, not all predictors are selected. For TREC Vol. 4+5 (Figure 2.7a) in most instances the same five predictors appear in the models. Notably is the absence of any term relatedness based predictor. The results are less clear for WT10g (Figure 2.7b): *MaxVAR* appears in most instances, the remaining included predictors fluctuate to a greater degree. The majority of single predictors fail to capture any more variance within the data and so are not used. This is due to two reasons: the poor accuracy of a number predictors which might not be better than random, and, as evident from the scatter plots in Section 2.3.2 (Figure 2.2), the training data is over-represented at the lower end of the average precision values (0.0-0.2) while very few queries exist in the middle and high range. The problems caused by poor predictors is further exemplified in Figure 2.7b where a large variety of predictors are added to the model.

## 2.13 Conclusions

This chapter has provided a detailed overview of a large number of pre-retrieval prediction methods. A taxonomy of predictors was introduced and in turn the predictors in each class were analyzed and empirically evaluated. The evaluation was performed on three diverse test collections: a corpus of newswire documents, a

Web corpus and an Intranet-like corpus. We were able to show that the prediction method accuracy – independent of the particular evaluation goal ($f_{diff}$, $f_{pred}$, $f_{norm}$) and thus independent of the particular correlation type used for evaluation – is dependent on the retrieval approach, the collection and the query set under investigation. First, this was shown on three standard retrieval approaches, and later confirmed when we investigated the predictor performances on widely differing TREC runs. For most prediction methods, the Web corpus proved to be the most difficult corpus to predict the effectiveness for.

Moreover, we showed that when using significance tests, significant differences in performance between the prediction methods are difficult to obtain. This is mainly because the query set sizes available to us are so small and thus the correlation coefficients need to have a large difference to point to potential significant improvements. Despite this lack of significant differences, we can use the correlation coefficients and the observations of the predictors performances on diverse TREC runs together to conclude that the ranking sensitivity based *MaxVAR* and the specificity based *MaxSCQ* predictors are overall the best performing predictors: they are among the top performing and they are the most stable across all TREC runs. Term relatedness based predictors on the other hand only perform well on specific query sets and can be considered unreliable. WordNet based measures failed to achieve meaningful predictor accuracies for most query sets.

Experimenting with combining predictors in a principled way through penalized regression which has the advantage of predictor sparseness, lead to reporting the cross-validated *RMSE* as a measure of the prediction accuracy to achieve a more reliable indicator of a method's quality. We showed that under the previous evaluation methodology the combination methods would be considered better in terms of $r$, though they are in fact not considerably better than single predictors.

# Chapter 3

# Post-Retrieval Prediction: Clarity Score Adaptations

## 3.1 Introduction

The previous chapter focused on pre-retrieval prediction methods which are mostly based on the query terms' collection statistics. We now turn our attention to post-retrieval methods which derive their performance estimations from ranked lists of results retrieved in response to a query. Post-retrieval predictors are computationally more expensive than pre-retrieval approaches since at least one, or possibly more, retrieval round has to be performed before a prediction can be made. However, the drawback of increased complexity is considered worthwhile, as an increased accuracy in the predictions is expected due to the greater amount of information available. As evident in the the previous chapter, *TF.IDF*, for instance, is a very poor retrieval approach for our available corpora, whereas Okapi achieves adequate retrieval effectiveness. A pre-retrieval predictor, however, assigns the same predicted score in both cases to a given query, while a post-retrieval predictor derives two distinct predictive scores as it is based on the ranked list of results.

In this chapter, we first present an overview of post-retrieval prediction methods. Then, we will focus on one post-retrieval approach in particular, namely Clarity Score, which was proposed by Cronen-Townsend et al. [45]. Clarity Score is based on the intuition that the top ranked results of an unambiguous and thus well performing query are topically cohesive, whereas the result list of an ambiguous and thus poorly performing query will cover a variety of topics. The degree of topical cohesiveness is derived from the term distribution of the top ranked documents: a homogeneous result list will lead to a distribution where terms particular to the topic appear with high frequency, while topically not homogeneous results with documents covering a variety of topics are assumed to be more similar to the collection distribution. Consider, for instance, the example query "jaguar" and its results, described in Chapter 1. The result list of the first 500 documents would be predicted to be of high quality by Clarity Score, as by far most results are concerned with the topic of cars. If, however, we were only to consider the result list up to rank ten, the query would be predicted to be somewhat ambiguous, as three documents are

concerned with the animal, while seven documents are concerned with cars. This sensitivity to the number of documents to rely on in the Clarity Score calculation is problematic. Currently, it is necessary to search exhaustively through the parameter space in order to find a reasonable setting.

In this work, we propose *Adaptive Clarity* which addresses two perceived shortcomings of Clarity Score. First, Clarity Score uses a fixed number of top-retrieved documents from which to derive the term distribution - we show that this is not optimal and propose an easy solution for a query adaptive automatic setting of this parameter. Second, the predicted performance score is the Kullback-Leibler (KL) divergence [91] between the term distribution of the top ranked results and the term distribution of the entire collection. Although all terms of the vocabulary participate in this equation, terms that have a high document frequency in the collection (and which for some reason occur uncharacteristically rarely in the top retrieved documents) add nothing to distinguish a set of homogeneous documents from the collection and thus we use a threshold of maximum document frequency to exclude those terms.

We will show in this chapter the following:

- Clarity Score in its original form is very sensitive to its parameter setting, and,
- Adaptive Clarity improves over Clarity Score and other state-of-the-art pre- and post-retrieval prediction approaches.

The work we present is organized as follows: first, in Section 3.2 we will provide an overview of related work. In order to offer some guidance to the levels of accuracy different methods achieve, this section also includes a summary in table form of the correlations found in various publications. Clarity Score is then introduced formally in Section 3.3. Section 3.4 contains an analysis of Clarity Score's sensitivity to its parameter settings. A similar analysis is also reported for Query Feedback, a post-retrieval method introduced by Zhou and Croft [177]. In Section 3.5, the two proposed changes to the Clarity Score algorithm are outlined. The experimental results, where Clarity Score and our adaptations to it are compared to a number of pre- and post-retrieval predictors are detailed in Section 3.6 and a discussion of the results follows in Section 3.7. The chapter concludes in Section 3.8.

## 3.2 Related Work

Post-retrieval prediction algorithms can be categorized into different classes, depending on the basic approach they take in order to estimate a query's result list quality. In this overview, we are going to distinguish the approaches by the type of information they exploit, that is information derived from

- perturbing the query and considering the differences in the respective ranked list of results (Section 3.2.1),
- perturbing the documents of the initially retrieved result list and considering the stability of the ranking (Section 3.2.2),

- perturbing the retrieval approach and considering the diversity of the ranked list of results (Section 3.2.3),
- analysing the ranked list of results of the original query (Section 3.2.4), and lastly,
- Web resources (Section 3.2.5).

Each of the different classes and proposed approaches are outlined below. If not explicitly stated otherwise, the approaches belong to evaluation aspect EA2 of Figure 1.1. Note that some approaches may fall into more than one class. We present those methods in the class of approaches where their most important effect lays.

### 3.2.1 Query Perturbation

Slightly altering a query and determining the similarity of the result lists of the original and perturbed query indicates the amount of *query drift*, which has been found to be a good indicator of result quality. Query drift refers to the change of focus of a query due to faulty query expansion [107]. A query is considered to be of high quality if slight variations of it do not result in a large change in retrieval result (the query drift is low), while a query whose ranked list of results changes considerably with a small change in the query exhibits a high amount of query drift - a change of focus implying that the originally retrieved ranked list contains a diverse and at least partially non-relevant set of documents.

A direct exploitation of the query drift concept is the *Query Feedback* method, introduced by Zhou and Croft [177]. Query Feedback frames query effectiveness estimation as a communication channel problem. The input is query $\mathbf{q}$, the channel is the retrieval system and the ranked list $L$ is the noisy output of the channel. From the ranked list $L$, a new, perturbed query $\mathbf{q}'$ is generated and a second ranking $L'$ is retrieved with $\mathbf{q}'$ as input. The overlap between the lists $L$ and $L'$ is used as query quality score. The lower the overlap between the two rankings, the higher the query drift and thus the lower the predicted effectiveness.

In the *Weighted Information Gain* [177] approach, a number of query variations are derived from a given query and the difference in the probability of occurrence of those variations in the top retrieved documents and in the corpus is used as estimate of retrieval effectiveness. This approach was developed for retrieval models that exploit term dependencies such as the Markov Random Field model [104] and thus the query variations include single term queries, exact phrase queries and unordered window queries, the latter being queries whose constituent terms need to occur in a document within a specified term range. The more the top ranked documents and the corpus as a whole differ with respect to those query variations, the better the estimated quality of the result list. This approach could have also been included in Section 3.2.4, but due to its derivation of different queries we chose to include it here. The experiments reported by Zhou and Croft [177] show that Weighted Information Gain outperforms Query Feedback and that a combination of both in turn overall performs best. The results though are not applicable to our

experiments, as we rely on unigram language models. We leave experimentation on term dependency based retrieval approaches for future work.

Yom-Tov et al. [167] also present an estimator based on query variations. In contrast to Weighted Information Gain though, their query variations are derived from the queries' constituent terms. Those "sub-queries" are then used in retrieval and the result lists of the original query and the sub-queries are compared to each other. The higher the overlap between them, the higher the estimated result list quality. The idea behind this approach is that for well performing queries the result list does not change considerably if only a subset of query terms is used. The two proposed estimators are based on machine learning approaches, and apart from the overlap of the result lists also exploit features such as the retrieval status value of the top ranked document and the number of query terms. The reported experiments suggest that the best predictor performance can be achieved on queries derived from TREC description topics (long queries). It is difficult though to put the results in context, as the predictor baselines in [167] (such as the standard deviation of IDF) are not very strong.

Finally, Vinay et al. [145] propose to perturb the weights that are assigned to each query term, and to consider a result list of high quality, if slight changes of term weights do not lead to a drastically different result list. Among the four approaches they propose this is the weakest performing one. This idea can also be viewed as a generalization of Yom-Tov et al. [167]'s estimators, where each sub-query is formed by setting the weights of all other query terms to zero.

### 3.2.2 Document Perturbation

The notion of estimating the quality of a result list by its ability to withstand the introduction of noise is based on observations made in retrieval on noisy text corpora. Transforming audio and images to text with automatic speech and optical character recognition leads to corrupted text documents, as the recognition process is imperfect. Experiments on such text corpora have shown that retrieval approaches which are robust in the presence of errors also exhibit a higher retrieval effectiveness on noise free corpora [131]. Translating this observation to query effectiveness prediction leads to the following heuristic: a result list which is stable in the presence of introduced noise is considered to be of high quality, while a result list which is unstable when noise is added to documents (the documents are perturbed) is considered to be of low retrieval effectiveness.

The *Ranking Robustness* approach by Zhou and Croft [176] exploits this heuristic by retrieving a result list for a given query, perturbing the documents by adding or removing terms and then ranking those perturbed documents based on the original query and retrieval approach. The similarity between the original result list and the result list derived from the perturbed documents indicates the robustness. In particular, perturbed are the term frequencies of the query terms occurring in each document by sampling a new frequency value from a Poisson distribution. The similarity between two result lists is determined by Spearman's rank correlation coefficient. Finally this process is repeated a number of times and the average rank correlation

constitutes the robustness score. The higher the score, the better the estimated retrieval effectiveness of the original ranked list of results. A comparison between Clarity Score and Ranking Robustness showed a slightly better performance of the latter. Zhou and Croft [177] also propose a variation of Ranking Robustness, the so-called *First Rank Change* approach, which is modified to be applicable to navigational queries [23]. Instead of comparing the original and perturbed result list, now it is only of interest in how many trials the top ranked document of the original list is also returned at the top of the perturbed result list.

Among the prediction methods proposed by Vinay et al. [145], the best performing one is based on document perturbations. To estimate the retrieval performance of a query, each document in the result list is perturbed by varying degrees of noise and then the perturbed document in turn is used as query. Of interest is the rank, the unperturbed document is retrieved at in response to such a query. When no noise is added to a document, the original (unperturbed) document can be expected to be retrieved at rank one. With increasing levels of noise, the rank of the original document is expected to drop. This rate of change between the level of noise and the drop in rank of the unperturbed document is the measure query quality. The more quickly the rank drops in response to noise, the lower the estimated retrieval performance of a query. The experiments on TREC Vol. 4+5 show the validity of the approach, a Kendall's $\tau$ of $0.52$ is reported. However, since the retrieval approach in these experiments is TF.IDF and the queries are derived from TREC topic descriptions (instead of TREC topic titles as in most other experiments), it is not possible to directly compare the results to other works.

As opposed to directly altering the terms or term weights of a document as done in the previous two approaches, Diaz [50] proposes a method based on spatial autocorrelation. In this approach a document's retrieval score is replaced by the weighted sum of retrieval scores of its most similar documents in the result list as determined by TF.IDF. The linear correlation coefficient between the original document scores and the perturbed document scores is then used as estimate of result list quality. This method is based on the notion that the result lists of well performing queries are likely to fulfill the cluster hypothesis [143], while poorly performing queries are not. If the cluster hypothesis is fullfilled, we expect the most similar documents to also receive similar retrieval scores by the retrieval system, while in the opposite case, high document similarity is not expressed in similar retrieval scores and the perturbed scores will be very different from the original ones. Note that in [50] this method is referred to simply as $\rho(\tilde{\mathbf{y}}, \mathbf{y})$, in our experiments (and in Table 3.1) we denote it with *ACSim* for autocorrelation based on document similarity. The results reported in [50] show that this approach outperforms both Clarity Score and Ranking Robustness on a range of query sets and corpora. An adaptation of this method, where the retrieval scores are modelled by Gaussian random variables is described by Vinay et al. [146].

### 3.2.3 Retrieval System Perturbation

Different retrieval systems applied to a single corpus return different result lists for a topic, depending on the particular retrieval approach, the approach's parameter settings, the pre-processing steps applied as well as the corpus content relied upon, such as for instance document content, document titles, anchor text and hyperlink structure. In a controlled setting such as TREC, a very limited number of relevant documents exist per topic (see Table 3.2) and retrieval approaches achieving a high retrieval effectiveness for a topic necessarily have a large amount of overlap among their top retrieved documents. Together with the observation that retrieval systems do not return the same non-relevant documents [134], the following heuristic arises: a topic is easy, that is, it will result in a high retrieval effectiveness, if the document overlap among different retrieval approaches is high. Conversely, a small amount of document overlap indicates a topic whose retrieval effectiveness will be low. In general, approaches in this category evaluate the effectiveness of a topic without considering a particular retrieval system (evaluation aspect EA1 in Figure 1.1) and we speak of *topic* as opposed to *query* since the retrieval approaches may rely on different (TREC) topic parts or different instantiations of the same topic part (different stemming algorithms for instance). The ground truth in this evaluation setup is commonly derived by considering the retrieval effectiveness of each topic across all participating retrieval approaches. The average, median or majority average precision is then utilized as ground truth effectiveness.

The first approach in this direction is *AnchorMap*, proposed by Buckley [25]. In his work, the document overlap between two rankings is equivalent to the mean average precision a ranking achieves if the documents of a second ranking are considered to be the relevant ones (the documents are "anchored"). The reported correlation with the ground truth is significant at $r = 0.61$. However, due to the small scale of the study – $30$ topics and $8$ retrieval systems – it is not possible to draw further conclusions from the result.

Counting the number of unique documents among the result lists of different retrieval approaches was suggested by Takaku et al. [136]. The larger the count, the more diverse the result lists and thus the more difficult the topic is estimated to be. The reported correlation of $r = -0.34$ suggests that there is some relationship between topic difficulty and the number of unique documents. However, due to the size of the experiment ($14$ retrieval systems) and the the type of topics (navigational topics from NTCIR) it remains unclear if these results will hold for informational queries.

Aslam and Pavlu [7] propose to determine the document overlap between retrieval systems by the Jensen-Shannon divergence [100] of their result lists. The authors experiment with a wide range of TREC data sets and report consistently high correlations, which indicates that the topic difficulty inherent to a collection is easier to estimate than the query effectiveness for a particular retrieval approach.

### 3.2.4 Result List Analysis

The ranked list of results can either be evaluated with respect to the corpus or by comparing the documents in the result list to each other without any further frame of reference. Comparing the retrieved result list to the collection as a whole is a measure of the result list's *ambiguity*. If the top retrieved results appear similar to the collection, the query is estimated to be difficult, as the results are not distinct from the entire corpus which covers many topics. On the other hand, if the top retrieved results are homogeneous and different from the corpus, the query's result list is estimated to be of high quality.

Clarity Score [45], which will be covered in more detail in Section 3.3, is based on the intuition that the top ranked results of an unambiguous query will be topically cohesive and terms particular to the topic will appear with high frequency. The term distribution of the top ranked results of such a query will be different from the general term distribution, which is derived from the entire corpus of documents. In contrast, a retrieval system will return results belonging to number of different topics when the query is ambiguous, and thus the term distribution will be less distinguishable from the corpus distribution. The higher the Clarity Score, the more distinct the top ranked results are from the corpus. In this case, the results are estimated to be unambiguous and therefore the estimated quality of the query is high. While Clarity Score is based on the Language Modeling framework, the same idea of comparing the term distribution in the top retrieved documents to the corpus as a query quality measure has been introduced for the Divergence From Randomness retrieval model [5] by Amati et al. [4]. In the work of Diaz and Jones [52], it is proposed to linearly combine Clarity Score with temporal features derived from the top ranked results. In particular in news corpora, distinctive temporal profiles exist for certain terms, such as *Christmas* which will occur mostly in articles published around December of each year, while a term such as *explosion* is likely to occur in bursts across the year depending on current events. This combination of article content and article publication time based query quality measures proved to lead to considerably higher correlations with average precision than Clarity Score alone. The two corpora used in the experiments were derived from TREC collections to only include newspaper articles, making them particularly suitable for the task. The same approach is unlikely to lead to improvements on WT10g and GOV2 though.

Clarity Score has also been applied to tasks such as selective query expansion [46] and the automatic identification of extraneous terms in long queries [94].

Carmel et al. [30] hypothesize that query difficulty is positively correlated with the distances between the query, the corpus and the set of relevant documents. The motivational experiments show that as expected the distance between the set of relevant documents and the set of all documents (the corpus) exhibits a positive correlation with retrieval effectiveness, an observation that is similar to the motivation for Clarity Score. Since in the topic difficulty setting the set of relevant documents is unknown, it is proposed to approximate this set by performing an initial retrieval and then selecting those documents from the result list that lead to the smallest distance to the query. The distances derived from the query, corpus and the approxi-

mation of the set of relevant documents are then used as features in a support vector machine [44, 144]. The approach is evaluated on GOV2 and queries derived from title topics 701-800. The reported linear correlation coefficient ($r = 0.36$) though lacks behind other reported approaches.

The *Clustering Tendency* method developed by Vinay et al. [145] deems a result list to be of high quality if the documents are tightly clustered, whereas the lack of clusters in the result list indicates poor retrieval effectiveness. In contrast to the previously described approaches, this method does not compare the result list to the corpus, instead, the amount of clustering is derived from the top retrieved documents alone. In this approach, documents are points in a high-dimensional space (vector space model). Then, random points are generated and two distances are recorded for each generated sample: the distance between the random point and the nearest document point $d_D$ and the distance between $d_D$ and its nearest document point neighbor. A large difference in those two distances indicates a high quality result list: the documents are highly clustered as their distances to each other are lower than their distances to random points. The advantage of such an approach is that we do not require collection statistics, on the other hand this might also make us miss vital information. A corpus that contains sports documents only, and whose top 100 results are about sports are not really tightly clustered, whereas they are appear clustered when we deal with a general news corpus. The reported results show a good estimation performance ($\tau = 0.44$). It should be noted though, that the distances between documents are determined based on query terms only, which works well for longer queries (in the experiments TREC description topics were used), the effect on short queries is not known.

Instead of considering the content of the top retrieved documents, recent studies have also investigated the possibility of deriving quality measures directly based on the retrieval scores of the top ranked documents. The most basic possibility is to estimate the quality of a result list by the retrieval score assigned to the top retrieved document as proposed by Tomlinson [139]: the higher the retrieval score, the better the result list is estimated to be. The performance of this approach is naturally dependent on the retrieval approach (which in turn determines what retrieval scores to assign to each document) and the query set. Depending on the retrieval model settings, the results reported in [139] for the Hummingbird SearchServer vary between $\tau = 0.23$ and $\tau = 0.35$ for queries derived from TREC title topics and between $\tau = 0.26$ and $\tau = 0.43$ for queries derived from TREC description topics. The observation that this approach is better suited for longer queries was also confirmed by Yom-Tov et al. [167] who evaluated the Juru search engine.

Shtok et al. [130] and Perez-Iglesias and Araujo [118] experiment with estimating the coverage of query aspects in the ranked list of results by deriving the retrieval scores' standard deviation, possibly normalized by a query dependent corpus statistic. It is hypothesized, that a high standard deviation indicates a high "query-commitment" [130] and the absence of aspects unrelated to the query. This indicates a result list of high quality. Conversely, if the retrieval scores of the top ranked documents exhibit a low score diversity, the result list is estimated to be dominated by aspects unrelated to the query and it is therefore considered to be of

poor quality. The work by Lang et al. [96] is also similar in spirit. Here, each query term is considered as a separate concept and the more concepts are covered in the result list, the better the estimated result quality. The *coverage score* of a query is defined as the sum of the term coverage scores weighted by the terms' importance. The term coverage scores in turn are estimated based on the retrieval scores of the top ranked documents. The advantage of retrieval score based approaches is the very low complexity, compared to approaches relying on document content, or document and query perturbations. At the same time the reliance on retrieval scores can also be considered a drawback, as such approaches require collaborating search systems that make the retrieval status values available. The results reported in [96, 130] show the potential of these approaches: retrieval score based methods achieve similar or higher correlations than the evaluated document content based approaches (including Clarity Score).

### 3.2.5 Web Resources

A number of recent studies take advantages of resources from Web search engines such as interaction logs, query logs and the Web graph ($\mathcal{W}$). We describe these approaches here as they offer valuable insights. However, we admit that we cannot apply any of the insights directly to our own work due to the unavailability of such resources to us. A second type of studies we describe in this section relies on freely available Web resources such as the Open Directory Project[1] (ODP) to infer the quality of search results.

Jensen et al. [80] infer the difficulty of a query on the Web by submitting it to different search engines, collecting the presented snippets of the top retrieved results and extracting thirty-one "visual clues" from these snippets such as the percentage of character n-grams of the query appearing in the snippet, the snippet title and the URL. An SVM regression approach is then applied to train a model. The reported results confirm the validity of the approach, the Spearman rank correlation between average precision at 10 documents (averaged over all search engines) and the quality estimate is $\rho = 0.57$. Since all baseline approaches are pre-retrieval predictors and this approach amounts to a post-retrieval approach it remains to be seen how its performance will compare against other approaches relying on the result list.

Leskovec et al. [99] propose the utilization of search result dependent subgraphs of $\mathcal{W}$ (on the URL and domain level) to train search result quality classifiers. The nodes of the top ranked Web pages retrieved in response to a query are located in $\mathcal{W}$; together with their connecting edges they form the *query projection graph*. A second subgraph, the *query connection graph* is generated by adding nodes and edges to the query projection graph until all nodes of the top ranked results are in a single connected component. A total of fifty-five features, most of them topological in nature, are derived from these two subgraphs, the query and the search result list. Together they are used in a Bayesian network classifier. The approach, evaluated on nearly $30000$ queries, was found to achieve a high degree of classification accuracy.

[1] http://www.dmoz.org/

The drawback of such a reliance on the Web graph is the complexity of the approach, in particular in finding the query connection graph.

A step further goes the reseach reported by White et al. [157], where "supported search engine switching" is investigated. Here, the aim is to predict which search engine of a given set of engines is going to provide the best results for a particular query. A large interaction log, containing search sessions of more than five million users across Google, Yahoo! and MS Live, was analyzed to find a set of useful features for the task of determining which of two rankings is of higher quality. Three types of features, used in a neural network based classifier, were found to lead to a high prediction accuracy: features derived from the two result rankings, features based on the query and features based on the similarity between query and result ranking. The evaluation on a data set of $17000$ queries showed that automatic engine switching can considerably improve result precision.

Predicting when to switch between states is also explored by Teevan et al. [137], whose goal is to predict whether to switch query personalization on or off. The motivation for this work stems from the observation, that not all queries perform equally well when user dependent factors, such as search history and user profile, are taken into account. While ambiguous queries benefit from personalization, the result quality of non-ambiguous queries can deteriorate. In order to predict query ambiguity, the authors rely on a query log of more than 1.5 million users and $44000$ distinct queries. A Bayesian dependency network is learned with forty features extracted from the query, the result list and the query log, including the average number of results, the click position, the time a query is issued and the amount of seconds passed until a result is clicked. Click entropy, which is the variability in the clicked results across users, is one of the evaluated query ambiguity measures. The results show that the trained model predicts the click entropy based ambiguity with high accuracy. Notably, Clarity Score is also listed as one of the features. Its correlation with click entropy is reported as approximately zero, similarly to most other features that are not based on query log information. The reason of this result remains unclear, though one possible explanation is that Clarity Score determined on the title and summary of the top $20$ retrieved results (as done here) is not as effective as on the full document content of possibly hundreds of documents. Furthermore, the ranking produced by a Web search engine is derived from a number of sources of evidence, instead of document content only, which might also influence the performance of Clarity Score.

Finally, Collins-Thompson and Bennett [40] and Qiu et al. [123] propose to exploit the ODP to measure a query's ambiguity. Although assigning precomputed ODP categories to each document [40] and relying on them to calculate Clarity Score and related approaches has the advantage of a low computational overhead, the reported results are not convincing: the maximum correlations achieved are $\tau = 0.09$ and $\tau = 0.13$ on the WT10g and GOV2 corpora respectively. The ODP also provides a search service, which given a query as input, returns not only a list of result pages but also a list of ODP categories. In [123], an ODP category list is determined for each term of a query and the overlap between the different lists of a query is used as an indicator of query ambiguity. A problem of the approach is that it

cannot assign ambiguity scores to queries consisting of a single term only. The high correlations reported (up to $\rho = 0.81$) were achieved on queries derived from the topics of the TREC 2003/04 Novelty track. In this track, only $25$ documents exist per topic and thus the entire document collection consists of merely $2500$ documents. It is not known how the reported results will translate to larger corpus sizes.

### 3.2.6 Literature Based Result Overview

While outlining the various approaches in the previous sections, we have largely refrained from comparing the effectiveness of different algorithms. The reason for this is the diversity of the test corpora, retrieval approaches, topic and query sets and evaluation measures, that have been employed to evaluate these algorithms. It is only possible to draw conclusions from evaluations performed on exactly the same setup. As was shown in Chapter 2, a small change in the parameter settings of a retrieval approach can already influence the correlation a query performance prediction method achieves. This observation also holds for post-retrieval methods as will become clear in the result section of this chapter.

Due to the complexity of most approaches, it is not possible to perform an evaluation across all methods. In order though to give some indication of the success of selected methods, we provide an overview of the correlations they exhibit on TREC and NTCIR data sets in Table 3.1. All correlations are taken from the cited publications. If a publication contains several proposed prediction methods, we include the best performing ones.

For each method included in Table 3.1, the overview contains the evaluation aspect that is investigated, the effectiveness measure relied upon as ground truth, the topic set and were applicable the part of the TREC/NTCIR topic the queries are derived from, either title (T), description (D), any part including the narrative (-) or unknown (?). The last three columns list the correlation coefficients.

The respective evaluation aspect – EA1 or EA2 – determines how the ground truth is derived. The ground truth of methods that predict the effectiveness of a set of queries for a particular retrieval approach (EA2) is most often the average precision (AP), some results also exist for precision at 10 documents (P@10) and reciprocal rank (RR). The latter measure is used in instances were navigational query sets are evaluated. The column *Models/#Runs* lists the retrieval approach the ground truth effectiveness is derived from: either Language Modeling (LM), TF.IDF, BM25, the Markov Random Field model (MRF) or the Divergence From Randomness model (DFR).

The ground truth of methods that aim to predict the topic difficulty inherent to the corpus (EA1) is derived from a set of retrieval approaches. When diverse retrieval approaches achieve a low retrieval effectiveness a topic is deemed difficult for a corpus. In this setting, the median, the average or the majority AP value across the retrieval runs participating in the experiment form the ground truth. The column *Models/#Runs* specifically indicates how many TREC or NTCIR runs are relied upon.

Table 3.1: Overview of selected post-retrieval query and topic effectiveness predictors. **Models/#Runs** describes the number of runs or the retrieval model used, depending on the evaluation aspect **Eval. Asp.**: EA1 (How difficult is a topic in general?) and EA2 (How difficult is a topic for a particular system?). **T-N** indicates which topic part is used: title (T), description (D), any part including the narrative (-) or unknown (?). **Evaluation Measure** lists the effectiveness measure relied upon as ground truth: average precision (AP), precision at 10 documents (P@10), reciprocal rank (RR) and the average and median AP value over all evaluated runs (TREC av. AP and TREC med. AP respectively). The correlations reported in each publication are shown in the last three columns: the linear correlation coefficient $r$, Kendall's $\tau$ and Spearman's $\rho$.

| | **Topics** | **Models/#Runs** | **T-N** | **Eval. Asp.** | **Evaluation Measure** | **Correlations** | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | **r** | $\tau$ | $\rho$ |
| Clarity Score[45]: divergence between the query language model and the collection language model | 201-250 | LM | D | EA2 | AP | | | 0.490 |
| | 251-300 | LM | T | EA2 | AP | | | 0.459 |
| | 351-400 | LM | T | EA2 | AP | | | 0.577 |
| | 401-450 | LM | T | EA2 | AP | | | 0.494 |
| | 351-450 | LM | T | EA2 | AP | | | 0.536 |
| $Info_{Bo2}$[4]: divergence between query term frequencies of the collection and result list | 100 topics (TREC Robust 2003) | DFR | D | EA2 | AP | 0.52 | | |
| Temporal Clarity [52]: linear regression with Clarity Score and two features of temporal profiles based on the creation dates of the top retrieved documents | $\approx$ 100 topics from AP corpus | LM | ? | EA2 | AP | 0.52 | | |
| | $\approx$ 100 topics from WSJ corpus | LM | ? | EA2 | AP | 0.60 | | |
| AnchorMap[25]: document overlap between ranked lists measured by mean average precision | 30 topics from 301-450 | 8 auto. TREC runs | D | EA1 | majority AP | 0.608 | | |
| Top Score[139]: retrieval score of the top retrieved document | 301-450,601-700 | Hummingbird | T | EA2 | AP | | 0.35 | |
| | 301-450,601-700 | Hummingbird | D | EA2 | AP | | 0.43 | |
| Decision Tree[167]: document overlap between the result lists of the full query and its sub-queries; decision tree based machine learning approach | 301-450,601-650 | Juru | T | EA2 | AP | | 0.305 | |
| | 301-450,601-650 | Juru | T | EA2 | P@10 | | 0.268 | |
| | 451-550 | Juru | D | EA2 | AP | | 0.202 | |
| | 451-550 | Juru | T | EA2 | P@10 | | 0.175 | |
| Histogram[167]: document overlap between the result lists of the full query and its sub-queries; feature histogram based machine learning approach | 301-450,601-650 | Juru | D | EA2 | AP | | 0.439 | |
| | 301-450,601-650 | Juru | D | EA2 | P@10 | | 0.360 | |
| | 451-550 | Juru | T | EA2 | AP | | 0.143 | |
| | 451-550 | Juru | T | EA2 | P@10 | | 0.187 | |
| Pool size[136]: number of unique documents in the pool of top 100 retrieved documents | 269 NP NTCIR-5 topics | 14 NTCIR runs | T | EA1 | av. RR | $-0.342$ | | |
| Document clustering tendency[145] | 301-450,601-650 | TF.IDF | D | EA2 | AP | | 0.441 | |
| Document perturbation[145] | 301-450,601-650 | TF.IDF | D | EA2 | AP | | 0.521 | |

| | Topics | Models/#Runs | T-N | Eval. Asp. | Evaluation Measure | Correlations r | $\tau$ | $\rho$ |
|---|---|---|---|---|---|---|---|---|
| Combination of topic distances[30]: SVM based machine learning approach | 701-800 | Juru | T | EA2 | AP | 0.362 | | |
| Robustness score[176]: stability of the result list in the presence of perturbed documents | 201-250 | LM | D | EA2 | AP | 0.613 | 0.548 | |
| | 251-300 | LM | T | EA2 | AP | 0.454 | 0.328 | |
| | 301-450,601-700 | LM | T | EA2 | AP | 0.550 | 0.392 | |
| | 701-750 | LM | T | EA2 | AP | 0.341 | 0.213 | |
| | 751-800 | LM | T | EA2 | AP | 0.301 | 0.208 | |
| JS divergence[7]: diversity between the result lists of multiple retrieval systems | 251-300 | all TREC runs | - | EA1 | TREC av. AP | 0.623 | 0.469 | |
| | 301-350 | all TREC runs | - | EA1 | TREC av. AP | 0.698 | 0.491 | |
| | 351-400 | all TREC runs | - | EA1 | TREC av. AP | 0.722 | 0.623 | |
| | 401-450 | all TREC runs | - | EA1 | TREC av. AP | 0.770 | 0.615 | |
| | 301-450,601-700 | all TREC runs | - | EA1 | TREC av. AP | 0.695 | 0.530 | |
| | 701-750 | all TREC runs | - | EA1 | TREC av. AP | 0.682 | 0.502 | |
| | 751-800 | all TREC runs | - | EA1 | TREC av. AP | 0.581 | 0.440 | |
| Weighted Information Gain[177]: change of information from average retrieval state to observed retrieval result state | 301-450,601-700 | MRF | T | EA2 | AP | 0.468 | | |
| | 701-800 | MRF | T | EA2 | AP | 0.574 | | |
| | 801-850 | MRF | T | EA2 | AP | 0.464 | | |
| | 252 NP-05 topics | MRF | - | EA2 | RR | 0.458 | | |
| | 181 NP-06 topics | MRF | - | EA2 | RR | 0.478 | | |
| Query Feedback[177]: overlap between the original result list $\ell_O$ and the result list derived by constructing a query from $\ell_O$ | 301-450,601-700 | MRF | T | EA2 | AP | 0.464 | | |
| | 701-800 | MRF | T | EA2 | AP | 0.480 | | |
| | 801-850 | MRF | T | EA2 | AP | 0.422 | | |
| First Rank Change[177]: stability of top result when perturbing the documents of the result list | 252 NP-05 topics | MRF | - | EA2 | RR | 0.440 | | |
| | 181 NP-06 topics | MRF | - | EA2 | RR | 0.386 | | |
| Linear combination of Weighted Information Gain and First Rank Change[177] | 252 NP-05 topics | MRF | - | EA2 | RR | 0.525 | | |
| | 181 NP-06 topics | MRF | - | EA2 | RR | 0.515 | | |
| Linear combination of Weighted Information Gain and Query Feedback[177] | 701-800 | MRF | T | EA2 | AP | 0.637 | | |
| | 801-850 | MRF | T | EA2 | AP | 0.511 | | |
| ODP based query ambiguity[123]: ambiguity based on the number of different topics assigned to query terms in an ODP search | 100 topics (TREC Novelty 2003/04) | LM | T | EA2 | AP | | | 0.597 |
| | 70 topics (TREC Novelty 2003/04) | LM | T | EA2 | AP | | | 0.808 |

| | Topics | Models/#Runs | T-N | Eval. Asp. | Evaluation Measure | Correlations r | $\tau$ | $\rho$ |
|---|---|---|---|---|---|---|---|---|
| *ACSim* ($\rho(\mathbf{y}, \tilde{\mathbf{y}})$)[50]: spatial autocorrelation based on document similarity | 201-250 | LM | D | EA2 | AP | 0.650 | 0.513 | |
| | 251-300 | LM | T | EA2 | AP | 0.486 | 0.357 | |
| | 301-450,601-700 | LM | T | EA2 | AP | 0.527 | 0.373 | |
| | 701-750 | LM | T | EA2 | AP | 0.540 | 0.454 | |
| | 751-800 | LM | T | EA2 | AP | 0.439 | 0.383 | |
| Covering Topic Score (IM2/CD2)[96]: coverage of topic concepts in the result list | 201-250 | LM | D | EA2 | AP | | 0.615 | |
| | 251-300 | LM | T | EA2 | AP | | 0.422 | |
| | 301-450, 601-700 | LM | T | EA2 | AP | | 0.454 | |
| | 701-750 | LM | T | EA2 | AP | | 0.313 | |
| | 751-800 | LM | T | EA2 | AP | | 0.356 | |
| AP Scoring[146] | 301-450,601-650 | BM25 | T | EA2 | AP | | 0.328 | |
| | 301-450,601-650 | BM25 | D | EA2 | AP | | 0.345 | |
| Topic Prediction $\frac{\Delta_{QR1}}{\Delta_{QG}}$ [40] | 451-550 | LM | T | EA2 | AP | | 0.091 | |
| | 701-850 | LM | T | EA2 | AP | | 0.130 | |
| Ranking Dispersion[118] | 301-450,601-700 | BM25 | T | EA2 | AP | 0.55 | 0.41 | |
| Normalized Query-Commitment (NQC)[130]: standard deviation of retrieval scores | 201-250 | LM | D | EA2 | AP | 0.556 | 0.414 | |
| | 251-300 | LM | T | EA2 | AP | 0.431 | 0.300 | |
| | 301-450,601-700 | LM | T | EA2 | AP | 0.563 | 0.419 | |
| | 451-550 | LM | T | EA2 | AP | 0.527 | 0.303 | |

## 3.3 Clarity Score

In this section, the Clarity Score query effectiveness estimator will be explained in more detail. To compute Clarity Score, the ranked list of documents returned for a given query is used to create a query language model [97] where terms that often co-occur in documents with query terms receive higher probabilities:

$$P_{qm}(w) = \sum_{D \in R} P(w|D)P(D|Q). \tag{3.1}$$

$R$ is the set of retrieved documents, $w$ is a term in the vocabulary, $D$ is a document, and $Q$ is a query. In the query model, $P(D|Q)$ is estimated using Bayesian inversion:

$$P(D|Q) = P(Q|D)P(D) \tag{3.2}$$

where the prior probability of a document $P(D)$ is zero for documents containing no query terms.

Typically, the probability estimations are smoothed to give non-zero probability to terms not appearing the query, by redistributing some of the collection probability mass:

$$\begin{aligned} P(D|Q) &= P(Q|D)P(D) \\ &= P(D)\prod_i P(q_i|D) \\ &\approx P(D)\prod_i \lambda P(q_i|D) + (1-\lambda)P(q_i|C) \end{aligned} \tag{3.3}$$

where $P(q_i|C)$ is the probability of the $i$th term in the query, given the collection, and $\lambda$ is a smoothing parameter. The parameter $\lambda$ is constant for all query terms, and is typically determined empirically on a separate test collection.

Clarity Score is the Kullback-Leibler (KL) divergence between the query language model $P_{qm}$ and the collection language model $P_{coll}$:

$$D_{KL}(P_{qm}||P_{coll}) = \sum_{w \in V} P_{qm}(w) \log \frac{P_{qm}(w)}{P_{coll}(w)}. \tag{3.4}$$

The larger the KL divergence, the more distinct is the query language model from the collection language model. If the documents of the ranked list are very similar to each other, Clarity Score assigns a relatively high score, as the divergence between the query language model and the collection language model will be large. Ambiguous queries on the other hand are hypothesized to result in ranked lists that are not topically homogeneous, which leads to a lower divergence between the two language models.

### 3.3.1 Example Distributions of Clarity Score

In this section, we experimentally assess whether the homogeneity assumption described above holds. For each query of our query sets, we calculate the Clarity Scores of three ranked lists, namely the lists of:

- the $x$ relevant documents,
- a random sample of $x$ documents from the pool of non-relevant documents, and,
- a random sample of $x$ documents from the pool of all documents in the collection containing at least one title topic term.

To derive the ranked lists for the relevant and the pool of non-relevant documents, we rely on the relevance judgments (the so-called *qrels*), available for each TREC test collection. For each topic, the qrels contain the relevant as well as the judged non-relevant documents. The number $x$ of documents to use is topic dependent and equal to the total number of relevant documents available for a topic. Table 3.2 shows the minimum, average and maximum number of of relevant documents $x$ across all topics of a corpus. While in all test corpora topics occur with very few relevant documents, the average number of relevant documents for the GOV2 collection is significantly higher than for the topics of TREC Vol. 4+5 and WT10g.

We distinguish between two random samples: the *non-relevant* random sample and the *collection-wide* random sample. The non-relevant random sample is derived from documents judged as non-relevant in the qrels. As TREC assessments are made of a pool of documents which have been returned as the top ranked documents by participating systems, it can be expected that those non-relevant documents are somewhat close in spirit to the relevant documents from the point of view of keyword based retrieval. As the number of judged non-relevant documents is always larger than the number of relevant documents, $x$ samples are drawn. This sampling process is repeated five times and the average Clarity Score of those five iterations is reported. Note that only documents containing at least $50$ terms were considered. For the collection-wide sample, we rely on our stopped and stemmed indices. All documents in the collection, that contain at least one of the title topic terms and have a length of $50$ or more terms (including stopwords), are used as sample space and as before sampling is performed five times and the average is reported.

| **Corpus** | **Topics** | **#Relevant Documents** | | |
|---|---|---|---|---|
| | | **Min.** | **Average** | **Max.** |
| TREC Vol. 4+5 | 301-450 | 3 | 93 | 474 |
| WT10g | 451-550 | 1 | 60 | 519 |
| GOV2 | 701-850 | 4 | 181 | 617 |

Table 3.2: Minimum, average and maximum number of relevant documents in the relevance judgments.

If the homogeneity assumption holds, we can expect a noticeable difference between the *relevant*, *non-relevant* and *collection-wide* Clarity Scores. Ideally, we would expect the scores of the *relevant* lists to lie in a narrow band, as no non-relevant document enters the language model and the ranked lists of results are unambiguous. The Clarity Scores of the *non-relevant* lists are expected to be somewhat lower, but still higher than those of the *collection-wide* lists as the non-relevant documents were

mistaken to be relevant by at least one retrieval system, whereas the *collection-wide* lists are generally created from a very large pool of random documents. In cases, where the title topic consists of a single very specific term, the pool of random documents will become very small and no large difference in scores can be expected. This effect is observed rarely though.

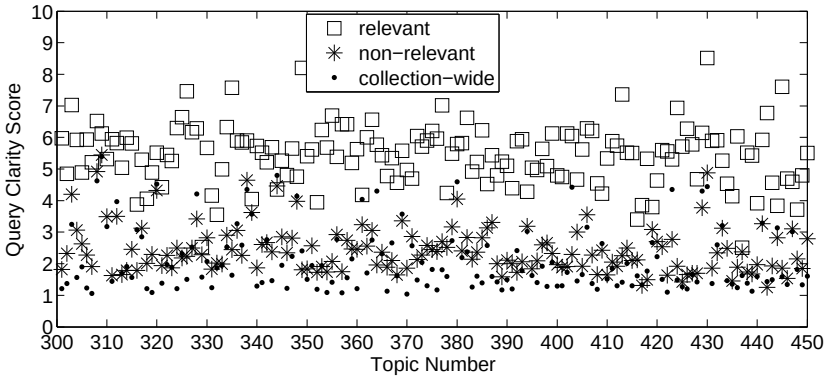
The results of this experiment are shown in the form of scatter plots in Figure 3.1. Each point marks the Clarity Score of a query for a particular type of ranked list, either *relevant*, *non-relevant* or *collection-wide*. In general, the results are as expected, that is the Clarity Scores of the lists of relevant documents are higher than those of the non-relevant and the collection-wide ones. However, there are differences visible in the quality of separation between the three plots. Figure 3.1c contains the results of the GOV2 collection. Here, in all instances, the scores of the lists of relevant documents are higher than those of the two random samples and furthermore, there are only $22$ cases (out of $150$) where the collection-wide samples achieve a higher score than the non-relevant samples. Slightly less well separated are the queries of TREC Volumes 4+5 (Figure 3.1a); for one query[2], the list of relevant documents has a slightly lower score than the two random samples and in $33$ additional cases the collection-wide samples are considered more homogeneous than the non-relevant samples. The results of the WT10g collection in Figure 3.1b are worst with respect to the separability of the different list types. There exist nine queries where the Clarity Scores of the lists of relevant documents are lower than the scores of one or both random samples. For a further $33$ queries, the scores of the collection-wide samples are higher than the non-relevant random samples. These results indicate, that Clarity Score is likely to perform better on GOV2 and TREC Vol. 4+5 than on WT10g, since the separation between the relevant and random samples is considerably clearer for them.

## 3.4 Sensitivity Analysis

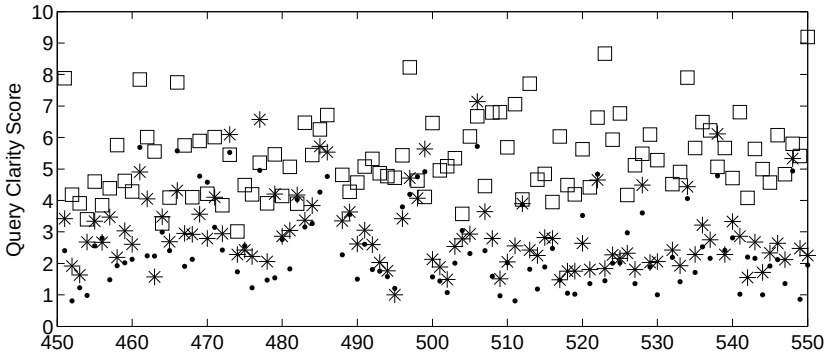
During the analysis of Clarity Score's homogeneity we assumed the number of feedback documents $x$ to be query dependent, equaling the number of relevant documents existing for a topic. This knowledge over the number of relevant documents is, of course, not available in practical applications and the original Clarity Score algorithm utilizes a uniform setting of $x$ across all queries, with $x = 500$ being reported to be a good value [45].

In this section, we investigate the influence of different factors affecting effectiveness prediction quality by giving examples of the behavior of Clarity Score and Query Feedback [177] as their (i) parameters, (ii) the retrieval setting, (iii) the collections and (iv) the query sets vary. In particular we are interested how sensitive Clarity Score actually is to the setting of $x$ and how well it can perform for the WT10g collection, having in mind the homogeneity analysis of Section 3.3.1. Additionally, we perform a similar analysis for the Query Feedback algorithm, as it

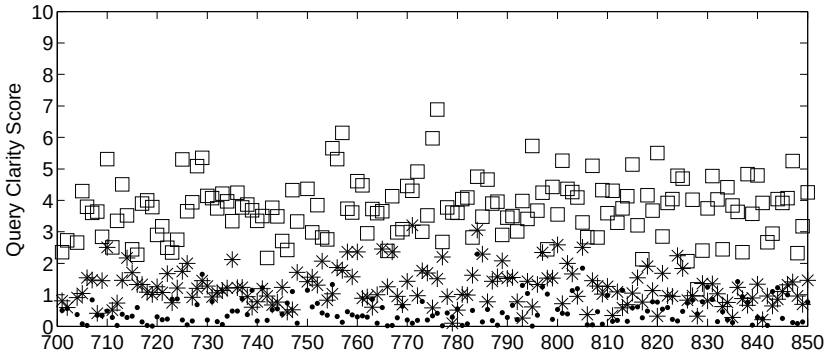
[2]Title topic 344: "Abuses of E-Mail"; the corresponding stemmed and stopword-free query is "abuse e mail"




(a) Queries 301-450 (TREC Vol. 4+5)




(b) Queries 451-550 (WT10g)




(c) Queries 701-850 (GOV2)

Figure 3.1: Distribution of Clarity Scores of the lists of relevant documents, sampled lists of non-relevant documents and sampled lists of collection-wide documents.

has been shown to achieve a good prediction performance across various TREC test collections. The parameters of Query Feedback are the number $t = |\mathbf{q}'|$ of terms $\mathbf{q}'$ consists of and the number of top documents $s$ for which the overlap between the two rankings $L$ and $L'$ is considered.

In Chapter 2 we already observed that the retrieval approach relied upon has a considerable influence on the accuracy of prediction algorithms. As will become evident shortly, the same observation holds for post-retrieval algorithms. For this reason, we evaluate Clarity Score and Query Feedback for a number of parame-

ters of the Language Modeling with Dirichlet smoothing approach, in particular $\mu = \{100, 500, 1000, 1500, 2000, 2500\}$. As in all experiments, we derive the queries from the TREC title topics.

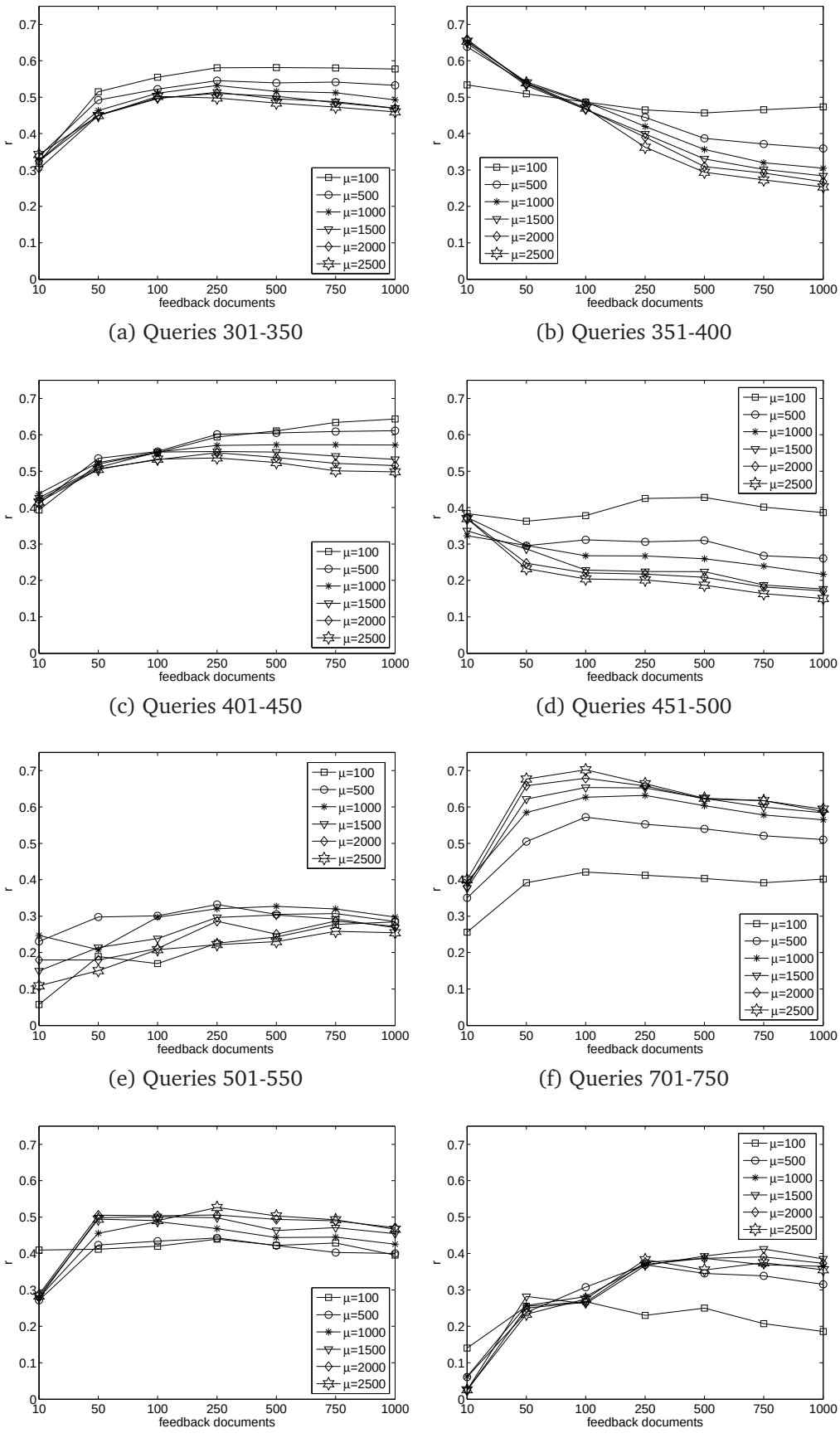
### 3.4.1 Sensitivity of Clarity Score

Figure 3.2 shows the development of Clarity Score's performance in terms of the linear correlation coefficient (the trends are similar for Kendall's $\tau$). The number $x$ of feedback documents is evaluated for the range of $x = \{10, 50, 100, 250, 500, 750, 1000\}$. Figures 3.2a, 3.2b and 3.2c display the behavior of the three different query sets of TREC Vol. 4+5. While queries 301-350 are relatively insensitive to the specific number of feedback documents and do not show much change in performance once $250$ feedback documents are reached, queries 351-400 exhibit a very different behavior. At $10$ feedback documents and $\mu = 2000$, the linear correlation coefficient is as high as $r = 0.66$, while at $1000$ feedback documents the correlation has degraded to $r = 0.27$. Finally, queries 401-450 show a continuous increase in $r$ for the lower levels of smoothing, while for $\mu = 1500$ and above, Clarity Score's performance peaks at $250$ feedback documents. In more general terms, for this collection the observation holds that low levels of smoothing favor a good Clarity Score performance: across most settings of $x$, the lowest level of smoothing ($\mu = 100$) leads to the highest correlation.

In Section 3.3.1, we hypothesized that Clarity Score's performance will be worst for WT10g, due to the insufficient separation between the scores of the relevant, the non-relevant nd the collection-wide randomly drawn documents. This hypothesis is now empirically confirmed when considering Figures 3.2d and 3.2e. They contain the results of the two query sets of the WT10g collection. Compared to the other query sets, the prediction performance is considerably lower, achieving in the most favorable setting $r = 0.43$. The influence of the level of smoothing is visible, but less clear: while for queries 451-500 $\mu = 100$ gives the highest correlation, the same level of smoothing leads to a low performance when considering queries 501-550. The influence of the number of feedback documents also varies; for queries 451-500, at $x = 10$ the Clarity Score's performance peaks for all but one smoothing level ($\mu = 100$). In contrast, for queries 501-550 the highest performance is achieved when $x$ is set to between $250$ and $1000$, depending on $\mu$.

Finally, the results for the query sets of the GOV2 corpus are shown in Figures 3.2f, 3.2g and 3.2h. Here, overall a greater amount of smoothing leads to a better performance and the optimal setting of $x$ varies between $100$ and $250$.

The setting of $\mu$ that leads to the highest correlation, often does not result in the best retrieval performance as measured in MAP. Consider Table 2.2, which contains the overview of the retrieval effectiveness for all query sets and various settings of $\mu$. In all but one case $\mu \geq 1000$ results in the highest retrieval effectiveness. However, for queries 451-500 for instance, the highest linear correlation coefficient ($r = 0.43$) is achieved for the setting of $\mu = 100$ and $x = 500$ feedback documents. The MAP of this retrieval run is only $0.15$ though. This is significantly worse than the MAP of the best performing run ($0.21$), which in turn leads to a maximum predictor correlation




(a) Queries 301-350

(b) Queries 351-400

(c) Queries 401-450

(d) Queries 451-500

(e) Queries 501-550

(f) Queries 701-750

(g) Queries 751-800

(h) Queries 801-850

Figure 3.2: Sensitivity of Clarity Score towards the corpus, the smoothing parameter $\mu$ and the number of feedback documents.

| Corpus | Queries | Best | Standard | Worst |
|---|---|---|---|---|
| TREC Vol. 4+5 | 301-350 | 0.545 | 0.539 | 0.338 |
| | 351-400 | 0.659 | 0.311 | 0.268 |
| | 401-450 | 0.573 | 0.573 | 0.438 |
| WT10g | 451-500 | 0.323 | 0.260 | 0.217 |
| | 501-550 | 0.287 | 0.251 | 0.180 |
| GOV2 | 701-750 | 0.635 | 0.603 | 0.406 |
| | 751-800 | 0.488 | 0.444 | 0.279 |
| | 801-850 | 0.387 | 0.387 | 0.062 |

Table 3.3: Linear correlation coefficient $r$ of the best, standard (500 feedback documents) and worst performing Clarity Score with respect to the retrieval run with the highest retrieval effectiveness as given in Table 2.2.

| Corpus | Queries | Best | Standard | Worst |
|---|---|---|---|---|
| TREC Vol. 4+5 | 301-350 | 0.436 | 0.420 | 0.302 |
| | 351-400 | 0.503 | 0.217 | 0.155 |
| | 401-450 | 0.367 | 0.305 | 0.305 |
| WT10g | 451-500 | 0.300 | 0.129 | 0.118 |
| | 501-550 | 0.243 | 0.223 | 0.053 |
| GOV2 | 701-750 | 0.475 | 0.415 | 0.257 |
| | 751-800 | 0.377 | 0.330 | 0.243 |
| | 801-850† | 0.247 | 0.235 | 0.061 |

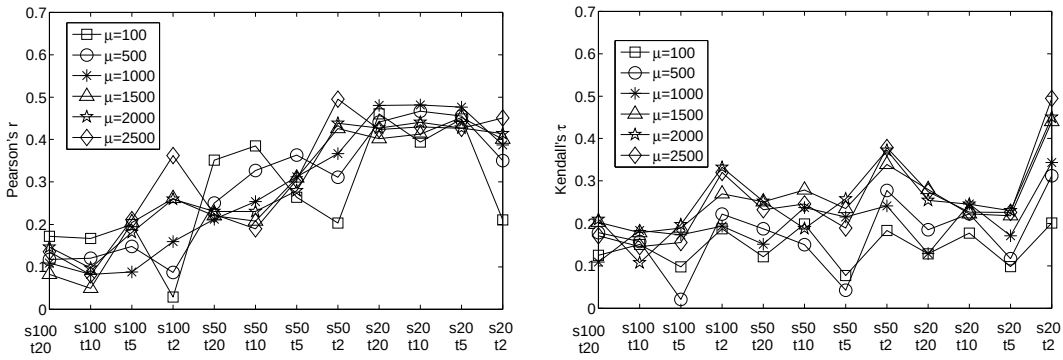
Table 3.4: Kendall's $\tau$ of the best, standard (500 feedback documents) and worst performing Clarity Score with respect to the retrieval run with the highest retrieval effectiveness as given in Table 2.2.

of $r = 0.32$.

To stress the point that the standard setting of 500 feedback documents may not always be adequate for Clarity Score, we present in Tables 3.3 and 3.4 the linear correlation coefficient $r$ and Kendall's $\tau$ that Clarity Score achieves with 500 feedback documents (standard) as well as the correlations of the best and worst performing feedback document setting. It is evident, that the feedback parameter is important for the accuracy of the Clarity Score algorithm and a wrong setting of this parameter can lead to poor results.

### 3.4.2 Sensitivity of Query Feedback

Figure 3.3 shows Query Feedback's sensitivity to changes in its parameter settings exemplary for queries 351-400 (reported as linear correlation coefficient $r$) and queries 451-500 (reported as Kendall's $\tau$ for comparison). The parameters $s$ and $t$ are evaluated for the range of $s = \{20, 50, 100\}$ and $t = \{2, 5, 10, 20\}$ respectively. Noticeable, as for Clarity Score, is the dependency on the correct parameter setting. The correlations achieved fluctuate widely, depending on $s$ and $t$ but also depending on the query set. For instance, for queries 351-400, the setting of $s = 20$ results




(a) Queries 351-400 (b) Queries 451-500

Figure 3.3: Sensitivity of Query Feedback towards its parameters and the smoothing parameter $\mu$ of the retrieval approach (language modeling with Dirichlet smoothing).

in a very stable and good performance across all $t$ except for $t = 2$, whereas for queries 451-500, $t = 2$ performs best across all settings of $s$. Finally, the effect of the level of smoothing on the algorithm's quality is generally reversed compared to Clarity Score: the lower the level of smoothing $\mu$, the less well the Query Feedback algorithm performs.

The conclusion to be drawn from this analysis is that both Clarity Score and Query Feedback can be very sensitive to both the initial retrieval parameter tuning, as well as their own parameters. Furthermore, parameters tuned to one query set do not produce reliable results for other query sets. Even when the query set and the collection are fixed, the performance of the predictors vary depending on the parameter settings of the retrieval approach.

## 3.5 Clarity Score Adaptations

In this section we introduce our proposed adaptations to Clarity Score. First the approach to setting the number of feedback documents automatically is described, followed by the frequency dependent term selection.

### 3.5.1 Setting the Number of Feedback Documents Automatically

In the literature, setting the number of feedback documents to a fixed value for all queries is the standard approach. Cronen-Townsend et al. [45] suggest that the exact number of feedback documents used is of no particular importance and $500$ feedback documents are proposed to be sufficient. In Section 3.4 experimental results showed that the performance of Clarity Score indeed depends on the number of feedback documents.

In real-world situations, such a dependence on the tuning of the parameter in order to achieve meaningful performance can have adverse effects if training on one query set does not translate to another query set. Preferably, it should be possible

to set parameters automatically such that performance on the evaluation set is close to or better than the best performing parameter setting.

When computing Clarity Score, if the query language model is created from a mixture of topically relevant and off-topic documents, its score will be lower compared to a query language model that is made up only of topically relevant documents, due to the increase in vocabulary size of the language model and the added noise.

Whereas Clarity Score sets the prior to zero for documents not containing at least one query term, *Adapted Clarity* sets the prior to zero for documents not containing all $m$ query terms, independent of the rank of the document in the result list. This effectively sets the number of feedback documents in the Clarity Score automatically; for each query, the number of feedback documents utilized in the generation of the query language model is equal to the number of documents in the collection containing all query terms.

As an example, consider TREC title topic 476: "*Jennifer Aniston*". Among the top $1000$ retrieved documents for the respective query there are $214$ documents that contain both terms, $780$ contain only the term *Jennifer* and $6$ documents contain only the term *Aniston*. Including all documents in the query language model that do not contain both query terms adds noise to the query language model. Although documents containing only the term *Aniston* are likely to be on topic as well, the method works well as an automatic threshold. In practice, in cases where there are fewer than $10$ documents fulfilling the requirement, documents with $m - 1$ query terms are included. Note that a document returned at rank $i$ that does not contain all query terms is ignored, while a document returned at rank $j > i$ is included in the query language model if it contains all query terms.

### 3.5.2 Frequency-Dependent Term Selection

In Section 3.4.1 we observed that the performance of Clarity Score depends on the initial retrieval run. In the Language Modeling approach Clarity Score often performs better with retrieval algorithms relying on a small amount of smoothing. Since increased smoothing in many instances though increases the retrieval effectiveness (Table 2.2), retrieval with greater smoothing is preferred. Hence, our goal is to improve Clarity Score for retrieval runs with greater smoothing. Increased smoothing also increases the influence of high frequency terms on the KL divergence calculation (Equation 3.4), despite the fact that terms with a high document frequency do not aid in retrieval and therefore should not have a strong influence on Clarity Score. Thus, we would like to minimize the contribution of terms that have a high document frequency in the collection.

The situation is similar in a retrieval setting where we estimate a query model using feedback documents. One proposed solution by Zhai and Lafferty [169], uses expectation maximization (EM) to learn a separate weight for each of the terms in the set of feedback documents. In doing this they reduce noise from terms that are frequent in the collection, as they have less power to distinguish relevant from nonrelevant documents.

A similar approach is proposed by Hiemstra et al. [76]. The effect of both approaches is to select the terms that are frequent in the set of feedback documents, but infrequent in the collection as a whole.

Generally, a notable requirement of web retrieval is speed. Running EM to convergence, although principled, would be computationally impractical. As a remedy, to approximate the effect of selecting terms frequent in the query model, but infrequent in the collection, we select the terms from the set of feedback documents that appear in $N$% of the collection, where $N = \{1, 10, 100\}$. We leave the comparison of a fixed document frequency-based threshold and a variable EM-based threshold to future work.

## 3.6 Experiments

We tested *Adapted Clarity*, that is our adaptations on Clarity Score, on the TREC corpora and query sets already employed in Chapter 2. Apart from Clarity Score, for reasons of comparison we include a number of the best performing pre-retrieval predictor scores as already presented in the previous chapter. We also implemented four post-retrieval prediction methods, described in Section 3.2, which base their predictions on different types of information:

- *Ranking Robustness* [176] (based on document perturbation),
- *Query Feedback* [177] (based on query perturbation),
- *Normalized Query-Commitment* (*NQC*) [130] (based on retrieval scores), and,
- *Autocorrelation* $\rho(\mathbf{y}, \tilde{\mathbf{y}})$ (*ACSim*) [50] (based on document content).

The two parameters of the Robustness approach are the number of top ranked documents to include in the perturbation and the number of trials. We settled on $50$ top documents and $100$ perturbation trials; varying the parameters yielded no great changes in performance in line with the observations in [176].

The parameter settings of the Query Feedback approach were determined by training $s$ and $t$ on one query set and evaluating it on another. That is, the best setting of $s$ and $t$ on queries 301-350 was used to evaluate query sets 351-400 and 401-450; similarly for the query sets of WT10g and GOV2.

The single parameter of *NQC* is the number of top ranked documents to include in the calculation of the standard deviation. Our results are based on the top $100$ documents as recommended in [130]. Similarly to the Robustness approach, small changes in the parameter do not affect the performance of this approach.

The most complex of the implemented post-retrieval predictors is *ACSim* whose parameters are the number of top ranked documents to include in the calculations, the number of most similar documents to derive the weightest sum of scores from and the similarity measure. Due to time constraints we did not train this model and instead chose the parameter settings recommended by Diaz [50] for our data sets.

In all reported experiments that follow, the smoothing parameter $\mu$ is set individually for each query set, according to the best performing retrieval effectiveness

| | | TREC Vol. 4+5 | | | WT10g | | GOV2 | | | Av. |
|---|---|---|---|---|---|---|---|---|---|---|
| **Approach** | **N** | **301-350** | **351-400** | **401-450** | **451-500** | **501-550** | **701-750** | **751-800** | **801-850** | |
| **AvIDF** | | 0.591† | 0.374† | 0.576† | 0.153 | 0.221 | 0.393† | 0.315† | 0.172 | 0.361 |
| **SCS** | | 0.578† | 0.319† | 0.518† | 0.087 | 0.189 | 0.325† | 0.278 | 0.096 | 0.310 |
| **MaxSCQ** | | 0.122 | 0.507† | 0.524† | 0.429† | 0.393† | 0.473† | 0.371† | 0.306† | 0.397 |
| **MaxVAR** | | 0.369† | 0.445† | **0.764**† | 0.381† | **0.533**† | 0.435† | 0.434† | 0.345† | 0.477 |
| **AvPMI** | | 0.316† | 0.376† | 0.438† | 0.288† | 0.235 | 0.431† | 0.456† | 0.037 | 0.327 |
| **Query Feedback** | | 0.318† | 0.427† | 0.382† | 0.290† | 0.216 | 0.602† | 0.535† | **0.490**† | 0.415 |
| ***ACSim*** | | 0.330† | **0.536**† | 0.525† | 0.379† | 0.353† | 0.550† | 0.469† | 0.488† | 0.457 |
| **Robustness** | | 0.526† | 0.424† | 0.581† | 0.312† | 0.489† | 0.340† | 0.307† | 0.415† | 0.429 |
| ***NQC*** | | 0.545† | 0.472† | 0.678† | 0.569† | 0.385† | 0.314† | 0.297† | 0.413† | 0.469 |
| **Clarity Score** | 100% | 0.539† | 0.310† | 0.573† | 0.260 | 0.251 | 0.603† | 0.444† | 0.387† | 0.430 |
| **Adapted Clarity** | 10% | *0.656*† | *0.409*† | 0.572† | *0.348*† | *0.253* | 0.527† | *0.467*† | *0.470*† | 0.471 |
| **(Fixed)** | 1% | **0.664**† | *0.443*† | *0.674*† | *0.545*† | 0.199 | 0.527† | 0.426† | 0.386† | 0.495 |
| **Adapted Clarity** | 100% | *0.549*† | *0.485*† | *0.666*† | *0.426*† | *0.397*† | **0.619**† | **0.603**† | 0.335† | 0.519 |
| **(Automatic)** | 10% | *0.629*† | *0.529*† | *0.639*† | *0.428*† | *0.366*† | 0.577† | *0.602*† | 0.356† | 0.524 |
| | 1% | *0.633*† | *0.511*† | *0.706*† | **0.592**† | *0.281* | 0.542† | *0.550*† | 0.370† | **0.535** |

Table 3.5: Linear correlation coefficient $r$ with respect to the retrieval run with the best mean average precision as given in Table 2.2. Given in bold is the best performing predictor for each query set. The Adapted Clarity variations that outperform Clarity Score are given in italics. Correlations significantly different from zero are marked with †($\alpha = 0.95$).

| | | TREC Vol. 4+5 | | | WT10g | | GOV2 | | | Av. |
|---|---|---|---|---|---|---|---|---|---|---|
| **approach** | **N** | **301-350** | **351-400** | **401-450** | **451-500** | **501-550** | **701-750** | **751-800** | **801-850** | |
| **AvIDF** | | 0.314† | 0.271† | 0.313† | 0.249† | 0.187 | 0.277† | 0.253† | 0.160 | 0.253 |
| **SCS** | | 0.286† | 0.227† | 0.277† | 0.174 | 0.136 | 0.211† | 0.240† | 0.095 | 0.206 |
| **MaxSCQ** | | 0.181 | 0.422† | 0.474† | **0.435**† | 0.270† | 0.331† | 0.291† | 0.209† | 0.327 |
| **MaxVAR** | | 0.353† | **0.434**† | 0.494† | 0.339† | **0.327**† | 0.288† | 0.318† | 0.243† | 0.350 |
| **AvPMI** | | 0.176 | 0.290† | 0.232† | 0.208† | 0.212† | 0.301† | 0.314† | 0.034 | 0.221 |
| **Query Feedback** | | 0.294† | 0.274† | 0.224† | 0.237† | 0.160 | **0.432**† | 0.420† | 0.275† | 0.290 |
| ***ACSim*** | | 0.332† | 0.358† | 0.471† | 0.363† | 0.265† | 0.377† | 0.359† | 0.248† | 0.347 |
| **Robustness** | | 0.423† | 0.323† | 0.424† | 0.208† | 0.315† | 0.216† | 0.199† | **0.308**† | 0.302 |
| ***NQC*** | | 0.377† | 0.371† | 0.381† | 0.409† | 0.315† | 0.147 | 0.240† | 0.255† | 0.312 |
| **Clarity Score** | 100% | 0.420† | 0.217† | 0.305† | 0.129 | 0.223† | 0.415† | 0.330† | 0.235† | 0.284 |
| **Adapted Clarity** | 10% | *0.474*† | *0.304*† | *0.398*† | *0.225*† | *0.225*† | 0.348† | *0.359*† | *0.291*† | 0.328 |
| **(Fixed)** | 1% | *0.485*† | *0.345*† | *0.497*† | *0.345*† | 0.160 | 0.351† | 0.310† | *0.272*† | 0.346 |
| **Adapted Clarity** | 100% | *0.423*† | *0.376*† | *0.448*† | *0.217*† | *0.286*† | *0.420*† | *0.441*† | 0.214† | 0.353 |
| **(Automatic)** | 10% | *0.461*† | *0.397*† | *0.465*† | *0.260*† | *0.277*† | 0.397† | **0.457**† | 0.221† | 0.367 |
| | 1% | **0.500**† | *0.400*† | **0.562**† | *0.374*† | 0.184 | 0.372† | *0.418*† | *0.250*† | **0.383** |

Table 3.6: Kendall's $\tau$ with respect to the retrieval run with the best mean average precision as given in Table 2.2. Given in bold is the best performing predictor for each query set. The Adapted Clarity variations that outperform Clarity Score are given in italics. Correlations significantly different from zero are marked with †($\alpha = 0.95$).

setting (Table 2.2). Tables 3.5 and 3.6 contain the linear correlation coefficient $r$ and Kendall's $\tau$ respectively of the baselines, the original Clarity Score and the Adapted Clarity variations. The rows marked with *Fixed* have the same fixed number of feedback documents for all queries as well as frequency-dependent term selection. To make the results comparable with the original Clarity Score, the reported numbers are the correlation coefficients achieved with the standard setting of $500$ feedback documents. The rows marked *Automatic* have their number of feedback documents set automatically as described in Section 3.5.1. The parameter $N$ determines the amount of frequency-dependent term selection. At $N = 100\%$, all terms independent of their document frequency are included in the KL divergence calculation, at $N = 10\%$ ($N = 1\%$) only terms occurring in less than $\frac{1}{10}$th ($\frac{1}{100}$th) of the documents in collection are included.

The final column of Table 3.5 contains the average linear correlation coefficient over all data sets. Since $r$ is not additive due to its skewed distribution, the average correlation does not exactly correspond to the arithmetic mean [112]. In Table 3.6 the arithmetic mean of the Kendall's $\tau$ values are reported.

When testing the significance of the difference between the original Clarity Score and the variations of Adapted Clarity, the outcome is corpus dependent. In the case of the linear correlation coefficient $r$ and TREC Vol. 4+5 all Adapted Clarity variations apart from one (automatic Adapted Clarity with $N = 100\%$) perform significantly better than the original Clarity Score. For the queries of the WT10g corpus only two variations significantly outperform the baseline, namely automatic Adapted Clarity with $N = 1\%$ and $N = 100\%$ respectively. No such observations can be made about the queries of GOV2: none of the Adapted Clarity variations result in a significantly higher correlation than the original Clarity Score.

The results of the significance tests for Kendall's $\tau$ are similar. For the queries of TREC Vol. 4+5 all variations of Adapted Clarity perform significantly better than the Clarity baseline, whereas for the queries of the WT10g and GOV2 corpora none of the proposed adaptations results in a significantly better performance.

In the reported experiments in Tables 3.5 and 3.6, query 803 was removed in the evaluation of query set 801-850, as it was an extreme outlier. Due to stopword removal, the title topic "may day" is converted to the query "day". One would expect the retrieval effectiveness of the document content based predictors of this query to be very low. The term "day" is not specific and occurs in a large number of documents. However, while the retrieval effectiveness is low (AP is $0.0$) as expected, the document content based predictors assign it very high scores, in fact, Clarity Score assigns it the highest score among the $50$ queries in the set by a wide margin. This surprising result can be explained when considering the makeup of the result list. We manually assessed the top $50$ retrieved documents and found that the documents either contain very large HTML forms with a hundreds of different "day" options or large lists and tables, mostly filled with numbers and the term "day". Duplicates and near duplicates lead to the outcome that 40 out of the 50 documents fall into four groups of near-duplicates, severely misleading the document content based predictors. Since the resulting correlations are then dominated by this extreme outlier we decided to remove this query from the evaluation.

## 3.7 Discussion

A considerable fraction of queries submitted to Web search engines occur infrequently, thus it is virtually impossible to create a representative query sample with relevance judgments to tune parameters.

For short unambiguous queries, constraining the language model to documents containing all query terms adds less noise to the language model. For terms that are ambiguous, forcing their inclusion increases noise, but this is desirable because we are capitalizing on noise in the language model to identify ambiguous queries. In the case that a query is unambiguous, but contains non-content terms, we compensate by selecting terms from the language model that are infrequent in the collection. Thus in Adapted Clarity non-content terms do not harm queries that are otherwise unambiguous.

Tables 3.5 and 3.6 show that similarly to pre-retrieval prediction methods, the performance of post-retrieval approaches also fluctuates widely over different query sets and corpora. For instance, Query Feedback achieves no significant correlation on query set 501-550 both in terms of $r$ and $\tau$, whereas on query set 701-750 it is among the best performing methods with $r = 0.60$ and $\tau = 0.43$ respectively. The range of predictor performance between the query sets of a single corpus is also considerable as evident for instance for Clarity Score and its correlation on query set 351-400 ($r = 0.31$) and on query set 401-450 ($r = 0.57$) respectively. The query sets of the WT10g corpus in general appear to be the most difficult for most of the evaluated post-retrieval methods to predict the query effectiveness for. Clarity Score's correlations are not significantly different from zero for both query sets, while both Query Feedback and *ACSim* perform considerably worse on WT10g's query sets than on the query sets of the other two corpora. The approach least affected by WT10g is *NQC* which exhibits moderate correlations for both query sets of WT10g. On the other hand, *NQC* performs worse than other post-retrieval approaches on GOV2, an observation we cannot explain yet and which requires further investigation.

When we consider the performance of the Adapted Clarity variations in comparison to Clarity Score we observe substantial improvements, which as pointed out in the previous section, are for some query sets large enough to be statistically significant. The largest change in correlation is observed for query set 451-500 of the WT10g corpus, where Clarity Score reaches correlations of $r = 0.26$ and $\tau = 0.13$ respectively whereas Adapted Clarity with automatically set feedback documents and $N = 1\%$ results in $r = 0.59$ and $\tau = 0.37$ respectively. When we consider the average correlation (last column in Tables 3.5 and 3.6) the Adapted Clarity variations with frequency-dependent term selection (the rows are marked as *Fixed*) outperform Clarity Score and in turn the Adapted Clarity variations with frequency-dependent term selection and automatic setting of the number of feedback documents (the rows are marked as *Automatic*) perform better than the variations with a fixed document feedback setting. With the exception of Adapted Clarity with fixed number of feedback documents and $N = 1\%$, each Adapted Clarity variation outperforms Clarity Score for at least six of the eight query sets. These observations hold for both the linear correlation coefficient and Kendall's $\tau$. Adapted Clarity with automatically

set feedback documents and $N = 1\%$ results in the highest average correlation, indicating the benefit of both proposed adaptations.

Notable are the prediction methods that perform closest to Adapted Clarity. In the case of the linear correlation coefficient, the two prediction methods with the highest average correlation after the Adapted Clarity variations are *MaxVAR* and *NQC*, outperforming the more complex prediction methods based on document and query perturbations. With respect to Kendall's $\tau$, *MaxVAR* is the best performing method (except for Adapted Clarity), followed by *ACSim*. This result shows, that post-retrieval approaches do not necessarily perform better than pre-retrieval prediction methods, in fact the pre-retrieval predictor *MaxVAR* outperforms all but the Adapted Clarity variations.

Predicting the quality of queries 451-550 has proven to be the most difficult across a range of predictors. In a Web environment, there are potentially millions of relevant documents for a given query. We hypothesize that the language of news articles and government websites is less varied, and the documents in these collections are more topically cohesive than Web pages. A single Web page contains a large proportion of content not related to the topic of the page itself, and furthermore even among the set of Web pages relevant to a given query, there may be a large number of different genres represented. For example in a Web setting, the set of relevant results may include pages that are largely informational (such as Wikipedia pages), pages that are largely commercial in nature, personal home pages, blogs, etcetera. Whereas the TREC Vol. 4+5 and GOV2 collections can be expected to be free of noisy pages such as spam, WT10g is not.

Furthermore, while the style for news articles is determined by a news organization and enforced to a large extent by the editors at that organization, on the Web the content is written by members of the general public with no style guidelines in place. Thus we hypothesize that one reason for the difficulty of achieving a good performance with Clarity Score on the Web corpus is the large variance in vocabulary, even among topically related documents. Since Clarity Score builds on the hypothesis that relevant documents have a more focused term distribution than non-relevant documents this metric correlates less well with noisy relevant documents (see Section 3.3.1).

## 3.8 Conclusions

The work reported in this chapter has focused on post-retrieval prediction algorithms. We first provided a broad overview of existing prediction methods, then focused on one particular approach: Clarity Score. Based on an analysis of Clarity Score's sensitivity to its parameter settings, we proposed two adaptations, namely setting the number of feedback documents used in the estimation of the query language model individually for each query to the number of documents that contain all query terms, and ignoring high-frequency terms in the KL divergence calculation. We evaluated these changes on three TREC test collections and compared them to a number of strong baseline approaches. We found that on average across

all evaluated query sets, *Adapted Clarity* is the best performing prediction method. Significant differences between Adapted Clarity and the original Clarity Score are only observed consistently for the queries of TREC Vol. 4+5 though. Another notable finding is the observation that the pre-retrieval predictor *MaxVAR* outperforms most evaluated post-retrieval predictors (with the exception of Adapted Clarity).

# Chapter 4

# When is Query Performance Prediction Effective?

## 4.1 Introduction

In the previous two chapters, we have evaluated the quality of query performance prediction methods by reporting how well the predicted performance of a set of queries correlates with the queries' retrieval effectiveness derived for a particular retrieval approach. As correlation measures we relied on the two measures commonly reported, namely the linear correlation coefficient $r$ and the rank correlation coefficient Kendall's $\tau$.

In Chapter 1 we described the perceived benefits of query performance prediction methods and their envisioned applications in adaptive retrieval systems. The current evaluation methodology, however, does not consider, whether - or more accurately *when* - those benefits will indeed be realized. Query effectiveness prediction research focuses on the development of algorithms that increase the correlation between the retrieval effectiveness and the predicted performance. While such an evaluation is straight-forward, it lacks interpretability. For instance, if on a particular data set one prediction method achieves a correlation of $\tau = 0.2$, while another achieves $\tau = 0.4$, does it mean the latter predictor is double as effective in practice? Knowing the correlation of a prediction method does not directly translate to knowing how the method will influence the performance of an adaptive retrieval system. In order to determine the relationship between the *evaluation* and the *application* of query effectiveness prediction methods, it is required to apply them in practice. This step is often not executed as evident in the strong contrast between the number of prediction algorithms that have been proposed over the years, and the number of publications dedicated to applying those prediction methods in an operational setting. It thus remains relatively unknown when a prediction method can be considered to perform well enough to be employable in an adaptive retrieval system.

This is an important knowledge gap, and one that we attempt to bridge in this chapter. Specifically, we investigate the relationship between the rank correlation coefficient $\tau$ a prediction method achieves and the prediction method's effect on

retrieval effectiveness in two operational settings: Meta-Search (MS) [159, 160, 167] and Selective Query Expansion (SQE) [4, 72, 102, 117, 46, 167]. In SQE, pseudo-relevance feedback [82, 126] is not applied uniformly to all queries, instead the decision whether to apply automatic query expansion (AQE) is made for each query individually. In the MS setting that we utilize in our experiments, each query is submitted to a number of systems and the prediction method determines which system's output is best and returned to the user. We chose these two operational settings because they are the two most often named potential practical applications for query performance prediction. Kendall's $\tau$ was chosen as correlation to evaluate as it lends itself naturally to our experiments (Section 4.3.2).

Our goal in this chapter is to determine at what levels of correlation a prediction method can be considered to be of high enough quality to produce tangible positive improvements in retrieval effectiveness in an adaptive retrieval component. If we were able to determine such thresholds, we could infer from a correlation-based evaluation, whether the quality of a prediction method is sufficient for a particular application. Thus, we aim to answer the following question: When is query performance prediction effective? Such a general question naturally leads to a number of more pointed questions that can be investigated:

- Is it worth (does the system improve) running a time consuming prediction method with a recorded performance of $\tau = 0.3$?
- If one prediction method improves the correlation coefficient by $0.05$, how does that affect the effectiveness of an adaptive retrieval system?
- At what levels of correlation can one be reasonably confident that a certain percentage of queries will improve their performance in an adaptive retrieval system over the non-adaptive baseline?

One possible approach to answering these questions is to predict the effectiveness of a set of queries, determine the correlation the predictor achieves and then calculate the retrieval effectiveness of the set of queries on the non-adaptive baseline and the adaptive retrieval system. If the effectiveness of the adaptive system is higher than the effectiveness of the baseline system, we might conclude that the correlation the prediction method achieved, is sufficient for the predictor to be viable in practice. As will become apparent in Section 4.2, such an approach may lead to misleading results. Based on the outcome of a single prediction method and one data set, one cannot draw conclusions about the level of correlation that indicates a high enough predictor accuracy to improve the retrieval effectiveness of an adaptive system in general.

If, on the other hand, we can perform such an experiment multiple times with diverse retrieval systems and predictions methods, we can gain a much better understanding by considering the change in adaptive retrieval performance *on average*. Considering the resources at our disposal for this work, the utilization of diverse retrieval systems and prediction methods covering a wide range of correlations is not possible. In particular, at the time of this thesis' completion, the best prediction methods reach a rank correlation of $\tau \approx 0.6$. For this reason, we base our empirical

studies on several TREC data sets. Instead of the output of existing query effectiveness prediction methods, we rely on generated predictions to analyze the entire spectrum of correlations so as to characterize precisely the relationship between the evaluation of query performance prediction methods and their effectiveness in operational settings.

In this chapter we aim to demonstrate the following:

- the correlation a query performance prediction method needs to achieve on average to be viable in practice, is dependent on the operational setting,
- in SQE, under stringent assumptions, moderate to high $\tau$ coefficients are required to obtain reliable improvements to retrieval performance, and,
- in MS, low to moderate correlations are already sufficient; however, for these improvements to be statistically significant, moderate to high correlations are also required.

The remainder of this chapter is organized as follows: in Section 4.2 we present an overview of works that have attempted to apply query performance prediction in the SQE and MS setting respectively. The section also contains an example of the potential for drawing misleading conclusions when a single predictor is applied. In Section 4.3 we describe our approach to overcoming this problem by generating a substantial number of data sets and predictions of query effectiveness. The SQE and MS experiments based on these generated data sets and predictions are detailed in Sections 4.4 and 4.5. We then discuss the implications of our study in Section 4.6. We close the chapter in Section 4.7 with an overview of the conclusions reached.

## 4.2 Related Work and Motivation

In the following two sections we provide an overview of the literature that describes the application of query performance prediction in the SQE and MS setup. As part of the overview, we also indicate the level of success reported in these works. In Section 4.2.3 we offer a motivation for our study on a concrete example.

### 4.2.1 Applications of Selective Query Expansion

The two SQE scenarios that were evaluated by Yom-Tov et al. [167] are based on the notion that easy queries, that is queries resulting in a high retrieval effectiveness, further improve with the application of pseudo-relevance feedback. Conversely, queries that are deemed difficult and which thus achieve only a low retrieval effectiveness degrade with the application of AQE. The rationale is the following: easy queries will have relevant documents among the top ranked results, and therefore an AQE algorithm [29, 107, 162, 163], which derives additional query terms from the top ranked documents returned for the initial query, is likely to derive terms related to the information need[1]. The ranked list retrieved for the expanded query

[1]Negative feedback from documents assumed not relevant is also possible [154], but will not be discussed further here.

further improves the quality of the results. Difficult queries on the other hand have few or no relevant documents in the top ranks of the result list and thus an AQE algorithm is likely to add irrelevant and misleading terms to the query. This results in query drift when the expanded query is used for retrieval and degrades the result quality. Selective query expansion builds on these assumptions: if we can predict the performance of a query, we selectively expand the queries that perform well according to the predictor method, while not expanding the poorly performing ones. This strategy should lead to an improvement over uniformly applying AQE to all queries as it aims to identify those, which will be hurt by the application of AQE.

In the first scenario reported by Yom-Tov et al. [167], a support vector machine [44, 144] is trained on features derived from the ranked list of results of the original query to classify queries as either to be expanded or not to be expanded. In the second scenario, a query performance prediction method is used to rank the queries according to their predicted effectiveness. The $85\%$ of queries predicted to perform best, are derived from TREC description topics, a procedure that simulates AQE. The queries predicted to be among the bottom $15\%$ performing ones are derived from TREC title topics, simulating the use of non-expanded queries. In both scenarios, selectively expanding queries based on a predictor proves slightly better than uniformly expanding all queries, with a change in MAP of $+0.001$.

A similar scenario with a different predictor is evaluated by Amati et al. [4]. Here, a predicted score threshold is fixed in a heuristic fashion and queries with predicted scores above the threshold are expanded, while queries with predicted scores below the threshold are not. In the experiments, the greatest improvement reported is from a MAP of $0.252$ when all queries are uniformly expanded, to $0.256$ when the queries are selectively expanded. Better results are reported by Cronen-Townsend et al. [46], where the threshold of when (not) to expand a query is learned. Of the data sets evaluated, the largest improvement is an increase in MAP from $0.201$ (AQE on all queries) to $0.212$ (selective AQE). In the worst case though, a considerable degradation in effectiveness is also observed: from a MAP of $0.252$ (AQE on all queries) to $0.221$ (selective AQE).

He and Ounis [72] combine selective query expansion with collection enrichment: depending on how the prediction method predicts a query to perform, it is either left unchanged, expanded based on the ranked list of results of the local corpus or expanded based on the ranked list of results of an external corpus. The evaluation yields mixed results, while for one data set MAP improves from $0.220$ (AQE on all queries) to $0.236$ (selective AQE), no change in effectiveness is observed for a second data set when applying the same approach. A follow-up on this work by Peng et al. [117] applied the same approach to Enterprise document search. Depending on the choice of external corpus and prediction method, the change in effectiveness varies. In the bast case, MAP improves from $0.381$ (AQE on all queries) to $0.396$ (selective AQE), in the worst case MAP degrades from $0.381$ to $0.358$.

Finally, Macdonald and Ounis [102] introduce a "selective candidate topic centric" approach to AQE in the realm of expert search [42]. Here, the decision is not made between which queries to apply AQE to and which queries to leave unaltered but instead a decision is made between which documents to include in the pool of

documents to draw the query expansion terms from and which not. In expert search the task is to retrieve a list of candidate experts in response to a textual query. The expertise of each candidate is represented by a set of documents assigned to her and candidates are ranked according to their document profile. AQE is performed by extracting terms from the top retrieved candidate profiles and their respective documents. Macdonald and Ounis [102] found considerable topic drift in AQE due to candidates with diverse areas of expertise and thus a diverse set of profile documents. They propose the following selective AQE procedure: the cohesiveness of all top ranked profiles is predicted and profiles with a high predicted cohesiveness contribute all their documents to the AQE process, while profiles predicted to be un-cohesive contribute only a small amount of their profile documents. Notable is the fact, that the cohesiveness predictor working best is simply the number of documents assigned to each candidate, outperforming document content based predictors. The experiments on two data sets showed statistically significant improvements, the MAP improves from $0.219$ and $0.561$ (AQE on all queries) to $0.236$ and $0.569$ (selective AQE) respectively.

Based on these mixed results it is difficult to comment conclusively on the suitability of predictors in the operational setting of SQE.

### 4.2.2 Applications of Meta-Search

Yom-Tov et al. [167] also evaluate their predictors in a meta-search setting: a corpus is partitioned into four parts, each query is submitted to each partition, and the result lists of each partition are merged with weights according to their predicted performance. In this experiment, MAP increases from $0.305$ (merging without weights) to $0.315$ when the results are merged according to the predictor based weights.

Wu and Crestani [160] apply a variety of retrieval algorithms to a single corpus. For each query and retrieval algorithm, a ranked list of results is derived and its predicted performance score is determined. Heuristically derived thresholds are used to classify each result list as either performing poorly, mediumly or well. The result lists are then merged with fixed weights according to the predicted classification. The best weighted data fusion method performs $2.12\%$ better than the unweighted baseline.

Lastly, Winaver et al. [159] propose to generate a large number of relevance models [97] for each query and then to pick the relevance model that is predicted to perform best. The results indicate the feasibility of the approach, the predictor-based model selection strategy significantly outperforms the baseline.

### 4.2.3 Motivation

Although the evaluation of query effectiveness predictors by reporting correlation coefficients is the current practice in this research field, this approach to evaluation is not without problems. The following example will reveal three problems:

- very different predictions can lead to similar correlation coefficients,

- a query performance predictor that results in a high correlation with retrieval performance does not necessarily lead to a retrieval performance increase in an operational setting and vice versa a predictor with a low correlation can lead to an optimal increase in retrieval effectiveness, and,
- as a corollary from the previous two points we can observe that a single predictor cannot be a reliable indicator of how large in general the correlation needs to be to lead to a consistent improvement in retrieval effectiveness over a wide range of predictor values.

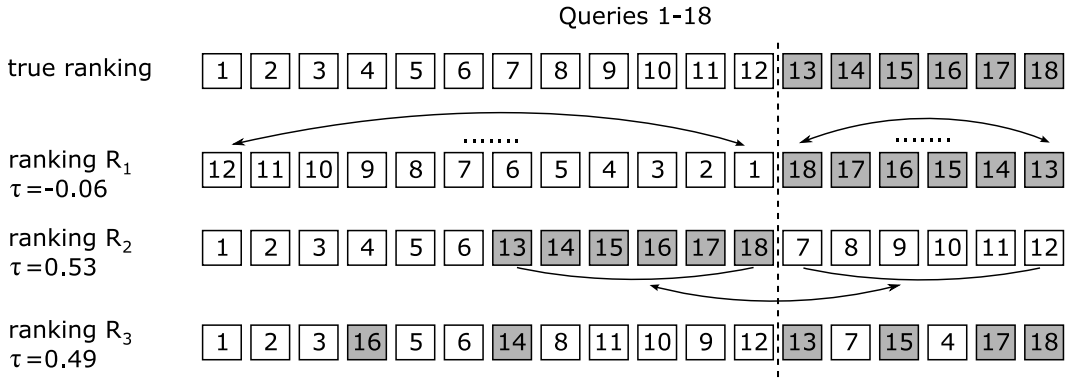



Figure 4.1: The top row contains the true ranking of queries according to their retrieval effectiveness, with the bottom ranked queries (in grey) assumed to benefit from *not* applying AQE. $R_1$, $R_2$ and $R_3$ are predicted rankings of query performance. SQE based on $R_1$ leads to optimal retrieval effectiveness - all AQE decisions are made correctly. $R_2$ leads to a correct AQE decision in one third of the queries. Based on $R_3$, fourteen correct expansion decisions are made.

Figure 4.1 contains an example that highlights the issues just described. In this instance, consider the operational setting of SQE. Let the query set consist of $18$ queries. The *true ranking* is the ranking of the queries based on their retrieval performance where rank $1$ is assigned to the best performing query. Let us further assume that the worst one third performing queries (in grey) benefit from *not* performing AQE. Here, $R_1$, $R_2$ and $R_3$ are examples of predicted rankings by different prediction methods. Ranking $R_1$ does not predict the *rank* of any query correctly which is reflected in its correlation coefficient of $\tau_{R_1} = -0.06$. However, in the SQE setup the correct expansion decision is made for each query which results in the optimal improvement in retrieval effectiveness.

The opposite case is ranking $R_2$ which results in a correlation of $\tau_{R_2} = 0.53$. Note that for twelve queries the wrong decision is made, which leads to a less than optimal retrieval effectiveness. Finally, based on predicted ranking $R_3$, for four queries, the wrong AQE decisions are made, although the correlation $R_3$ achieves is very similar to the correlation of $R_2$: $\tau_{R_3} = 0.49$.

This example shows how similar correlations may have completely different impacts on retrieval performance, while different correlations can lead to counter-intuitive losses or gains in retrieval effectiveness. Although, admittedly, this example is somewhat contrived, it highlights why a single predicted ranking can lead to

misleading conclusions about the rank correlation coefficient required to achieve an adequate retrieval effectiveness in an adaptive retrieval setting.

## 4.3 Materials and Methods

In an ideal setting we would conduct an investigation as follows. Given a large number of query performance prediction methods, a large number of retrieval algorithms and a set of queries:

- let each prediction method predict the queries' quality and determine the prediction method's performance in terms of Kendall's $\tau$,
- use the predictions in an operational setting and perform retrieval experiments to derive a baseline and predictor-based result, and,
- finally determine at what level of correlation the predictor-based retrieval results generally show improvements over the baseline results.

In practice, such a setup is not feasible for two reasons. Most importantly, the predictors are limited in their accuracy, which would not allow us to investigate the change in retrieval performance at higher correlations than $\tau \approx 0.6$. Furthermore, not all settings may strictly adhere to the particular assumptions made, a noise factor which needs to be taken into account. For instance, a common assumption of SQE is that well performing queries improve when AQE is applied. This might not always be the case though. In the experiments described in the next sections, we are able to control for these effects and thus are able to precisely investigate the influence of these factors.

### 4.3.1 Data Sets

To make our results generalizable and less dependent on a particular retrieval approach, we utilize TREC data sets and in particular the runs submitted by the TREC participants to different adhoc retrieval tasks to simulate diverse sets of retrieval approaches. All TREC runs submitted for each query set with a MAP greater than $0.15$ are included in this study. The data sets are listed below. In brackets, the number of runs per data set are shown.

- TREC-6 (49), TREC-7 (77) and TREC-8 (103) based on the corpus TREC Vol. 4+5,
- TREC-9 (53) and TREC-10 (59) based on the corpus WT10g, and,
- Terabyte-04 (35), Terabyte-05 (50) and Terabyte-06 (71) based on the GOV2 corpus.

Please note, that we changed the terminology with respect to the previous chapters (TREC-6 instead of 301-350). We do so deliberately to indicate that we deal with *runs* now instead of a set of queries. The range in retrieval effectiveness is considerable: while the minimum is fixed to $0.15$, the best performing run achieves a MAP of $0.47$.

### 4.3.2 Predictions of Arbitrary Accuracy

Since Kendall's $\tau$ is based on ranks (instead of scores), it is possible to construct predicted rankings for any level of correlation, simply by randomly permutating the true performance ranking of queries. The smaller the number of permutations, the closer $\tau$ is to $1$. Conversely, the larger the number of permutations of the ground truth based ranking, the closer $\tau$ is to $0$. From the full correlation coefficient range of $\tau \in [-1, 1]$, sixteen intervals $CORR = \{c_{0.1}, ..., c_{0.85}\}$ were investigated, each of size $0.05$, starting with $c_{0.1} = [0.1, 0.15)$ and ending with $c_{0.85} = [0.85, 0.9)$. This correlation range is sufficient for our purposes, since negative correlations can be transformed into positive correlations by reversing the ranking and $\tau = 1$ indicates two perfectly aligned rankings.

For each coefficient interval $c_i$, $1000$ rankings were randomly generated with $\tau \in c_i$ with respect to the true ranking. We rely on such a large number of rankings due to the issues outlined in Figure 4.1. By considering the application of $1000$ predicted rankings for each correlation interval $c_i$, we can consider the change in retrieval effectiveness *on average*. Each predicted ranking is utilized in the SQE and MS experiments in place of a query ranking produced by a query performance predictor. This setup allows us to analyze the impact of varying levels of correlation against the change in retrieval effectiveness between the non-adaptive baseline and the prediction-based system. All query rankings were generated once for query set sizes of $m = \{50, 100, 150\}$ and then kept fixed across all experiments.

## 4.4 Selective Query Expansion Experiments

We analyze the relationship between Kendall's $\tau$ as an evaluation measure of query performance prediction and the change in retrieval effectiveness when queries are expanded selectively in a setup analogous to the setup investigated by Yom-Tov et al. [167].

The effect AQE has on retrieval effectiveness varies considerably and is dependent on the particular AQE approach, the retrieval algorithm and the set of queries evaluated. The literature on AQE and pseudo-relevance feedback is vast and contains many different observations, which may substantially contradict each other. What makes pseudo-relevance feedback work is still not perfectly understood, despite significant efforts in the past such as the RIA Workshop [24, 63]. Interest in this research direction has not diminished, as evident by last year's TREC, where the Relevance Feedback track [26] was introduced.

It is beyond the scope of this work, to cover a wide range of literature on AQE; instead we give an overview of findings that are most pertinent to our experiments. Whilst across the whole query set, AQE aids retrieval effectiveness with improvements ranging from $3\%$ to $40\%$ [107, 162, 163], not all queries benefit. The percentage of queries from a query set that perform worse when AQE is applied varies between $20\%$ and $40\%$ [4, 95, 107, 162]. Amati et al. [4], Carpineto et al. [32] and Kwok [95] observe that the worst and the very best performing queries are hurt by

AQE. As reason for the degradation in performance of the best queries is given that a highly effective query is actually diluted if additional unnecessary query terms are added to it and the result quality suffers. Carmel et al. [29] and Xu and Croft [162] on the other hand only report the worst performing queries to be hurt by the application of AQE.

### 4.4.1 Experimental Details

Let us for now assume that all well performing queries improve with the application of AQE, while the worst performing queries degrade with AQE. Let $\theta$ be a rank threshold. Our SQE setup is as follows: given a set of $m$ queries, they are ranked according to their predicted performance. AQE is applied to the best $(\theta \times m - 1)$ performing queries, the remaining queries are not expanded. As this setup only requires predicted rankings, we can use our generated predicted rankings of arbitrary accuracy. To evaluate the retrieval effectiveness of SQE, we require pairs of baseline (no AQE) and AQE runs. Then, we perform SQE based on the predicted rankings and consider SQE to be successful when it improves over the retrieval effectiveness of the AQE run. We derive baseline and AQE run pairs from the runs in our data sets. As we are not interested in the ranked list of results themselves, but in the effectiveness of each run on each query $\mathbf{q}$, for the purpose of this chapter, we consider a *run* to consist of a list of average precision values, thus $\textit{run} = (ap^{\mathbf{q}_1}, ap^{\mathbf{q}_2}, ..., ap^{\mathbf{q}_m})$.

#### Run Pairs

Each run of our data sets is considered as a baseline run $run_{base}$, where no AQE is applied. As the runs consist of $m = 50$ $ap$ values, in order to obtain baseline runs for $m = 100$ and $m = 150$, the original runs were randomly concatenated. For each baseline run a corresponding AQE run $run_{aqe}$ is generated. Recall, that we work with the assumption that AQE improves the effectiveness of the well performing queries, while degrading the effectiveness of poorly performing queries. Thus, for each $ap^{\mathbf{q}_i}_{base}$ in $run_{base}$, a respective $ap^{\mathbf{q}_i}_{aqe}$ value in $run_{aqe}$ is generated such that $ap^{\mathbf{q}_i}_{aqe} > ap^{\mathbf{q}_i}_{base}$ when $ap^{\mathbf{q}_i}_{base}$ is among the top $(\theta \times m - 1)$ performing queries in $run_{base}$, otherwise $ap^{\mathbf{q}_i}_{aqe} < ap^{\mathbf{q}_i}_{base}$. The $ap^{\mathbf{q}_i}_{aqe}$ values are randomly sampled (with the outlined restrictions) from the other runs in the data sets. This strategy is supported by results reported by Billerbeck and Zobel [19] and Carpineto et al. [32], where no correlation between $ap^{\mathbf{q}_i}_{base}$ and the *amount* of improvement, that is $\Delta = (ap^{\mathbf{q}_i}_{aqe} - ap^{\mathbf{q}_i}_{base})$, $\Delta > 0$, was found. The optimal SQE run $run_{opt}$ is the run where the correct AQE decision is made for every query, that is $ap^{\mathbf{q}_i}_{opt} = max(ap^{\mathbf{q}_i}_{base}, ap^{\mathbf{q}_i}_{aqe})$. We only include run pairs where the MAP of $run_{aqe}$ improves by between $15\%$ and $30\%$ over $run_{base}$ and the MAP of $run_{opt}$ improves by at least $3\%$ over $run_{aqe}$. Due to the random component in the run pair generation process $500$ run pairs are created for each setting of $\theta = \{1/2, 2/3, 3/4\}$ and $m = \{50, 100, 150\}$. The choice of $\theta$ is based on results in the literature [107, 162], the settings of $m$ are typical TREC topic set sizes.

Table 4.1 contains basic statistics of all generated run pairs. Here, *av. MAP* is the MAP of the baseline ($base$), the expanded ($aqe$) and the optimal SQE runs ($opt$),

| m | $\theta$ | #pairs | av. $MAP_{base}$ | av. $MAP_{aqe}$ | av. $MAP_{opt}$ |
|---|---|---|---|---|---|
| 50 | 1/2 | 500 | 0.198 | 0.236 | 0.256 |
| | 2/3 | 500 | 0.243 | 0.293 | 0.306 |
| | 3/4 | 500 | 0.270 | 0.325 | 0.337 |
| 100 | 1/2 | 500 | 0.199 | 0.238 | 0.254 |
| | 2/3 | 500 | 0.243 | 0.295 | 0.307 |
| | 3/4 | 500 | 0.281 | 0.335 | 0.347 |
| 150 | 1/2 | 500 | 0.201 | 0.241 | 0.257 |
| | 2/3 | 500 | 0.250 | 0.301 | 0.313 |
| | 3/4 | 500 | 0.278 | 0.331 | 0.342 |

Table 4.1: Statistics of run pairs generated for the SQE setting.

averaged over all 500 run pairs. The difference in retrieval effectiveness between the baseline and the AQE runs are in all instances greater than between the AQE and the optimal SQE runs. This is explained by the fact that the best performing queries all come from the AQE run in the optimal SQE run. Furthermore, as $\theta$ increases, the improvement of the optimal SQE runs over the AQE runs degrades slightly as more queries of the AQE runs occur in the optimal SQE runs.

### SQE Experiment

Given the 1000 generated rankings per correlation coefficient interval $c_i$ and the 500 run pairs ($run_{base}/run_{aqe}$) for each setting of $\theta$ and $m$, SQE is thus performed 500000 times for each $c_i$. A formal description of the experiment is provided in Algorithm 1.

From each run pair and predicted ranking in $c_i$ a selective AQE run $run_{sqe}$ is formed: if according to the predicted ranking $ap^{\mathbf{q_i}}_{base}$ is among the top $(\theta \times m - 1)$ scores in $run_{base}$, then $ap^{\mathbf{q_i}}_{sqe} = ap^{\mathbf{q_i}}_{aqe}$, that is the AQE result is used. The remaining queries are not expanded and it follows that $ap^{\mathbf{q_i}}_{sqe} = ap^{\mathbf{q_i}}_{base}$. Recorded are the MAP of $run_{base}$, $run_{aqe}$, $run_{opt}$ and $run_{sqe}$. We consider SQE to be successful if the MAP of $run_{sqe}$ is higher than the MAP of $run_{aqe}$. Since the run pairs lead to different absolute changes in retrieval effectiveness, we report a normalized value:

$$v_{sqe} = 100 \times \frac{MAP_{sqe} - MAP_{base}}{MAP_{opt} - MAP_{base}}. \tag{4.1}$$

When the correct AQE decision is made for each query, $v_{sqe} = 100$. In contrast, $v_{sqe} < 0$ if the MAP of $run_{sqe}$ is below the baseline's $run_{base}$ MAP. We present the results, derived for each $c_i$, in the form of box plots [142]. Every box marks the lower quartile, the median and the upper quartile of the 500000 $v_{sqe}$ values. The whiskers show the 1.5 inter-quartile range, the remaining separately marked points are outliers. We also include in the plots the median normalized value of the AQE runs as a horizontal line - this is the value $v_{sqe}$ must improve upon in order for SQE to be deemed successful. We chose this type of visualization because it offers a convenient way of depicting the information we are interested in. Each box element marks the interval where the middle 50% of the $v_{sqe}$ scores of all test cases, made

for a single correlation coefficient band, fall. The height of each box indicates the spread of the results. If the entire box is above the horizontal line, at least $75\%$ of $v_{sqe}$ values outperform the normalized AQE value. Conversely, if the box is below the horizontal line, less than $25\%$ of $v_{sqe}$ value outperform the AQE value.

**Algorithm 1:** Selective query expansion

```
1  foreach c ∈ CORR do                                   ▷for all correlation intervals c
2      foreach r ∈ RANK^c do                             ▷for all rankings with τ ∈ c
3          foreach (run_base, run_aqe) ∈ RUNS do         ▷for all run pairs
4              run_sqe = ∅, run_opt = ∅
5              for i ← 1, m do
6                  if r[i] < (θ × m) then                ▷SQE
7                      ap_sqe^qi = ap_aqe^qi
8                  else
9                      ap_sqe^qi = ap_base^qi
10                 end
11                 ap_opt^qi = max(ap_base^qi, ap_aqe^qi)
12             end
13             v_sqe = 100 × (MAP_sqe − MAP_base) / (MAP_opt − MAP_base)
14             v_aqe = 100 × (MAP_aqe − MAP_base) / (MAP_opt − MAP_base)
15         end
16     end
17 end
```

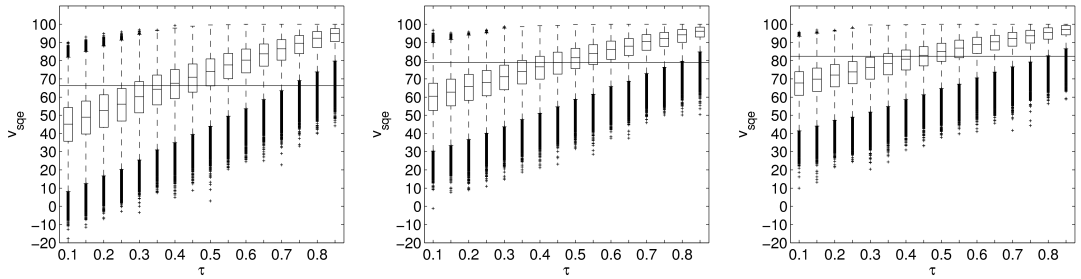
### 4.4.2 Results

We perform two sets of experiments. The first experiment is set up to represent the ideal situation where the assumption we listed about SQE holds. It was designed to test the relationship between $\tau$ and SQE effectiveness in a *noise-free* environment. Thus, the results can be considered as best case. The second experiment tests the robustness of SQE against noise.

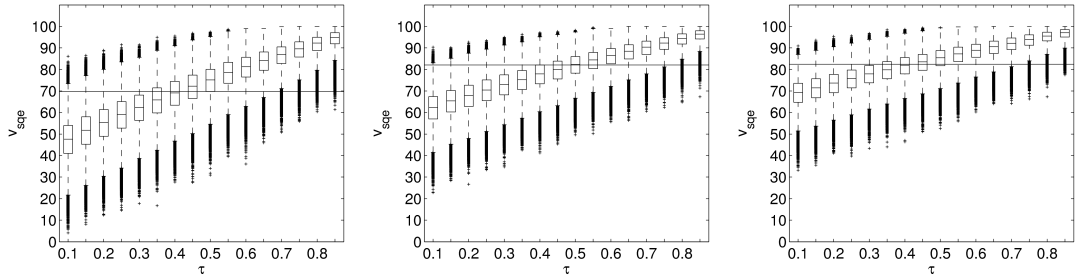
#### Best-Case Scenario

In this experiment, our assumption that AQE only hurts the worst performing queries holds for all run pairs. We also assume $\theta$ to be known. The results of the experiment are shown in Figure 4.2. The first row depicts the boxplots for $m = 50$ queries, the second row contains the results of $m = 100$ queries and the last row shows the development for $m = 150$ queries.

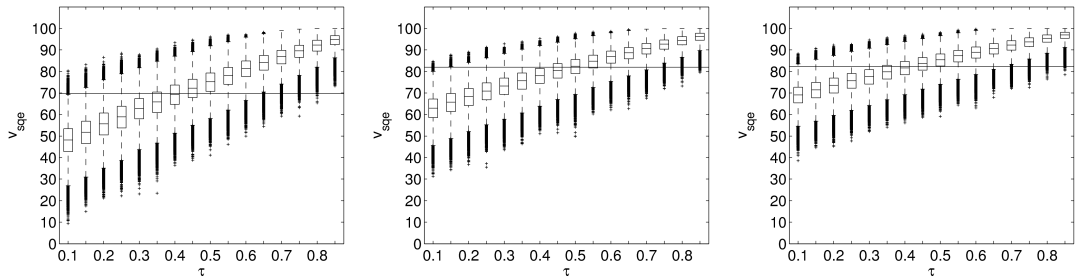
We observe, that independent of the settings of $m$ and $\theta$, for all correlation intervals $c_i$ a number of positive and negative outliers exist, where the MAP of $run_{sqe}$ improves or degrades over $run_{aqe}$'s MAP. Thus, even if $\tau = 0.1$, a predictor can be successful by chance. In contrast, a predicted ranking with $\tau = 0.9$ can still result




(a) $m = 50$, $\theta = 1/2$ (b) $m = 50$, $\theta = 2/3$ (c) $m = 50$, $\theta = 3/4$




(d) $m = 100$, $\theta = 1/2$ (e) $m = 100$, $\theta = 2/3$ (f) $m = 100$, $\theta = 3/4$




(g) $m = 150$, $\theta = 1/2$ (h) $m = 150$, $\theta = 2/3$ (i) $m = 150$, $\theta = 3/4$

Figure 4.2: SQE best-case scenario. Listed on the x-axis is the starting value of each correlation interval $c_i$, that is the results for $c_{0.2} = [0.2, 0.25)$ are shown at position $0.2$. The horizontal lines mark the normalized median value of the performance of the AQE runs, which $v_{sqe}$ must improve upon for SQE to be considered successful.

in a negative change of retrieval effectiveness. This supports the view that a single experiment and predictor are inadequate indicators to show a predictor method's utility in practice.

When the correlation of the predicted ranking with respect to the ground truth is low, $run_{sqe}$ may perform worse than $run_{base}$. This is for instance visible in Figure 4.2a for $\theta = 1/2$ and $m = 50$, where the $v_{sqe}$ values are negative. Recall that the generation process of run pairs (Section 4.4.1) ensures that $run_{base}$ performs between $15\%$ and $30\%$ worse than $run_{aqe}$, thus a poor predictor can significantly degrade the effectiveness of a system. An increase in $\tau$ generally leads to a smaller spread in performance (the height of the boxes in the plot) of $v_{sqe}$, indicating that outliers are rarer and the performance drop is less pronounced.

Increasing the setting of $\theta$ yields rises in $v_{aqe}$, as can be expected: when $\theta = \frac{3}{4}$,

| **m** | $\theta$ | **25%** | **50%** | **75%** | **min. 95% opt.** |
|---|---|---|---|---|---|
| 50 | 1/2 | 0.30 | 0.40 | 0.50 | 0.20 |
| | 2/3 | 0.35 | 0.45 | 0.60 | 0.10 |
| | 3/4 | 0.35 | 0.45 | 0.60 | 0.10 |
| 100 | 1/2 | 0.35 | 0.45 | 0.55 | 0.35 |
| | 2/3 | 0.40 | 0.50 | 0.60 | 0.30 |
| | 3/4 | 0.35 | 0.45 | 0.55 | 0.20 |
| 150 | 1/2 | 0.35 | 0.45 | 0.50 | 0.50 |
| | 2/3 | 0.45 | 0.50 | 0.55 | 0.45 |
| | 3/4 | 0.35 | 0.45 | 0.50 | 0.35 |

Table 4.2: SQE best-case scenario: summary of $\tau$ necessary to improve $25\%$, $50\%$ and $75\%$ of $run_{sqe}$ over the median of $run_{aqe}$. The final column contains the minimum correlation coefficient interval where at least in one instance $run_{sqe}$ reaches $95\%$ retrieval effectiveness of $run_{opt}$.

three quarters of the queries in $run_{aqe}$ outperform the queries of $run_{base}$, only very poorly performing queries have a slightly worse performance when expanded. If on the other hand $\theta = \frac{1}{2}$, the medium performing queries can degrade and the range of possible degradation is larger.

Finally, the number $m$ of queries also influences the outcome: with increased query set size the spread of results decreases and the results become more stable. This is visible in the plots by the decreased height of the boxes as well as the fewer extreme cases. Thus, the more queries are used in the evaluation, the better the correspondence between the evaluation measure and the performance in an operational setting.

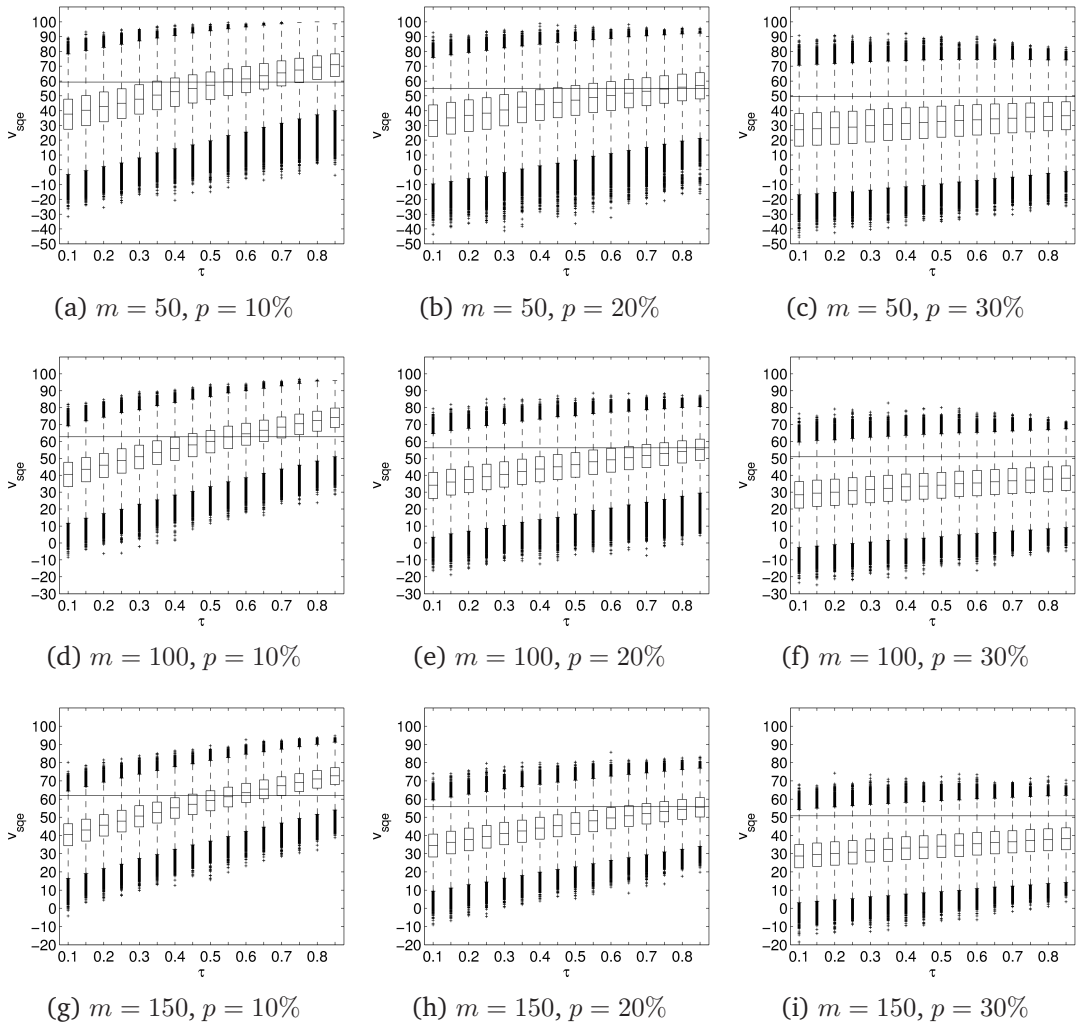
To provide an overview of the results, we summarize the most important correlation thresholds in Table 4.2. Reported are the thresholds of $\tau$ where $25\%$, $50\%$ and $75\%$ of the $500000$ test cases overcome the horizontal line in the plots. The final column of Table 4.2 shows the minimum correlation coefficient for which in at least one test case $run_{sqe}$ reaches $95\%$ retrieval effectiveness of $run_{opt}$. This value is particularly low for $m = 50$, that is, even if the correlation is not statistically significantly different from $\tau = 0$, the ranking can lead to a close to optimal result in the SQE setting. In short, in the best-case scenario of SQE, where we have perfect knowledge about which queries improve effectiveness when being expanded and which degrade, medium correlations are sufficient to improve the effectiveness.

### Random Perturbations

The best-case scenario presented in the previous section is unrealistic, as we do not have perfect knowledge about what queries will improve and degrade with the application of AQE. Thus, to complement the first experiment, we now investigate how imperfect knowledge of the influence of AQE changes the results. This experiment is motivated by the divergent observations by Amati et al. [4], Kwok [95], Mitra et al. [107] and Xu and Croft [162] about the change in effectiveness of the best

performing queries when AQE is applied.

To simulate such violation we turn to perturbing $run_{aqe}$. Given a pair of runs $(run_{base}, run_{aqe})$, we randomly select a query $\mathbf{q_i}$ from the top $(\theta \times m - 1)$ performing queries of $run_{aqe}$ and perturb its score $ap^{\mathbf{q_i}}_{aqe}$ to $\hat{ap}^{\mathbf{q_i}}_{aqe}$, which is a random value below $ap^{\mathbf{q_i}}_{base}$. To keep the MAP of $run_{aqe}$ constant, the difference $(ap^{\mathbf{q_i}}_{aqe} - \hat{ap}^{\mathbf{q_i}}_{aqe})$ is randomly distributed among the remaining $ap$ values of $run_{aqe}$. This ensures that the overall effectiveness of $run_{aqe}$ remains between $15\%$ and $30\%$ better than $run_{base}$. This procedure is performed for $p = \{10\%, 20\%, 30\%\}$ of the top $(\theta \times m - 1)$ performing queries. Specifically, the number of queries perturbed for each value of $p$ were $\{3, 5, 8\}$ for $m = 50$, $\{5, 10, 15\}$ for $m = 100$ and $\{8, 15, 23\}$ for $m = 150$.




(a) $m = 50$, $p = 10\%$ (b) $m = 50$, $p = 20\%$ (c) $m = 50$, $p = 30\%$

(d) $m = 100$, $p = 10\%$ (e) $m = 100$, $p = 20\%$ (f) $m = 100$, $p = 30\%$

(g) $m = 150$, $p = 10\%$ (h) $m = 150$, $p = 20\%$ (i) $m = 150$, $p = 30\%$

Figure 4.3: SQE random perturbation scenario. $\theta$ is fixed to $1/2$.

The results of this experiment are shown in Figure 4.3. Note, that we fixed $\theta$ to $1/2$ in order to avoid having another changing parameter in the experiments. It is evident, that even a small number of perturbed queries already has great influence on the usability of a query performance predictor in the SQE setting. When $p = 10\%$ of queries are perturbed (Figures 4.3a, 4.3d and 4.3g), the perturbation can still be

compensated, and a correlation $\tau \geq 0.55$ is necessary to ensure that for more than $50\%$ of the test cases the selectively expanded runs improve over the fully expanded runs. The trend of improvement however is already less visible for $p = 20\%$. A further increase in the number of perturbed queries leads to the situation as visible for $p = 30\%$ (Figures 4.3c, 4.3f and 4.3i), where independent of the actual accuracy of the predicted ranking, in less than $25\%$ of all test cases an improvement of effectiveness over the fully expanded run is possible, although positive outliers still exist across all levels of $c_i$. Also, notably different from the optimal scenario in the best-case scenario is the performance of the negative outliers: when $p = 30\%$, for all $m$ and correlation intervals there exist test cases that perform considerably worse than $run_{base}$ such that $v_{sqe}$ is negative.

It should also be pointed out that the median of the normalized AQE value (the horizontal line), decreases slightly with increasing $p$. This effect is due to the way $run_{aqe}$ is perturbed. Since the MAP is kept constant for $run_{aqe}$ and $run_{base}$ when perturbing the queries and $v_{aqe}$ is defined as (Algorithm 1):

$$v_{aqe} = 100 \times \frac{MAP_{aqe} - MAP_{base}}{MAP_{opt} - MAP_{base}}, \tag{4.2}$$

it follows, that the decrease of $v_{aqe}$ is the result of a small increase of $MAP_{opt}$, created by the random distribution of $(ap^{\mathbf{q_i}}_{aqe} - \hat{ap}^{\mathbf{q_i}}_{aqe})$ remainders.

| m | p | 25% | 50% | 75% | min. 95% opt. |
|---|---|---|---|---|---|
| 50 | 10% | 0.35 | 0.55 | 0.75 | 0.35 |
| | 20% | 0.45 | 0.80 | - | 0.40 |
| | 30% | - | - | - | - |
| 100 | 10% | 0.45 | 0.55 | 0.75 | 0.65 |
| | 20% | 0.65 | - | - | - |
| | 30% | - | - | - | - |
| 150 | 10% | 0.45 | 0.60 | 0.70 | - |
| | 20% | 0.65 | 0.85 | - | - |
| | 30% | - | - | - | - |

Table 4.3: SQE random perturbations scenario: summary of $\tau$ necessary to improve $25\%$, $50\%$ and $75\%$ of $run_{sqe}$ over the median of $run_{aqe}$. The final column contains the minimum correlation coefficient interval where at least in one instance $run_{sqe}$ reaches $95\%$ retrieval effectiveness of $run_{opt}$. In all experiments, $\theta = 1/2$ is fixed.

Table 4.3 summarizes the levels of correlation required in the random perturbation scenario where $25\%$, $50\%$ and $75\%$ of the test cases improve over the AQE run. Feasible thresholds can only be achieved for $p = 10\%$ perturbations.

### 4.4.3 Out-of-the-Box Automatic Query Expansion

The pseudo-relevance feedback mechanism employed to determine which terms to enhance a query with can be very complex with many parameters requiring substantial tuning. In practice however, out-of-the-box mechanisms are often used as

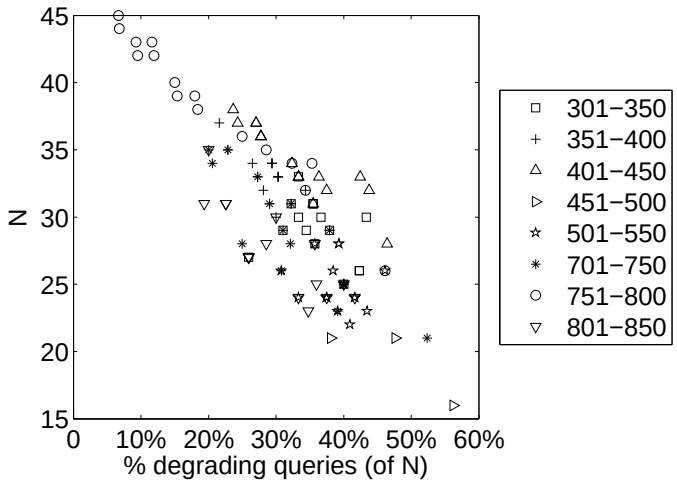



Figure 4.4: Scatter plot of behavior of out of the box AQE runs. Each marker stands for one AQE run. The y-axis shows the number of queries $N$ for which AQE improves the queries' effectiveness. The x-axis shows the percentage of queries among the top $N$ performing unexpanded baseline queries that do not improve with AQE.

supplied by Information Retrieval toolkits such as Lucene, Terrier or Lemur/Indri. In this section, we determine how many of the top performing queries indeed improve with the application of AQE. For this analysis, we relied on the query sets and corpora already utilized in Chapters 2 and 3. The retrieval approach is fixed to Language Modeling with Dirichlet smoothing ($\mu = 500$). The title part of the TREC topics is utilized to form the queries. The AQE experiment is performed with Indri's default AQE setup. We varied the number of feedback terms and documents to use for AQE between $5$, $10$, $20$ and $50$ for both parameters. Thus, for each query set, sixteen (4x4 parameter pairings) different AQE runs are created. We then compare these runs to the baseline runs where no AQE is employed.

The results of this analysis are shown in Figure 4.4. Each point represents one AQE run. The vertical axis depicts the number of queries $N$ (out of $50$ for each query set) for which the AQE run exhibits a higher average precision than the non-expanded baseline run. The horizontal axis depicts the percentage of queries among the top $N$ performing baseline queries that do *not* improve with AQE. The closer that value is to $0\%$ the better the assumption holds that the top performing queries improve their effectiveness when AQE is applied.

The scatter plot shows that how well the AQE assumption holds is not only dependent on the parameter settings of the AQE mechanism, but also on the query set. When we consider the results of query set 751-800 the number of queries (out of 50) that improve with AQE vary between $26$ and $45$, depending on the setting of the number of feedback terms and documents. A comparison between the runs of query set 751-800 and query set 801-850 shows that although both are based on the same corpus (GOV2), the results differ considerably. For query set 801-850, in the most favorable parameter setting $N = 31$ queries improve when AQE is applied, a strong contrast to most runs of query set 751-800.

The scatter plot also shows that only a small minority of runs overall have less than $20\%$ of non-improving queries in the top ranks; for most runs this number lies between $25\%$ and $45\%$. In general, the fewer queries $N$ improve overall, the larger the percentage of queries performing contrary to the common SQE assumption.

## 4.5 Meta-Search Experiments

The second operational setting for query performance prediction under investigation is meta-search. From a high level perspective, MS may be described as any setting where two or more ranked lists of results are returned for a single input query. The result lists can either contain documents from the same collection or documents retrieved from different corpora. In the former case, the input query can be varied, for instance, by adding or deleting query terms such that different results are returned for each query variation. Alternatively, a range of retrieval approaches can be employed to retrieve a wider variety of documents. This is the setting we employ in our experiments, which is analogous to Winaver et al. [159]'s experiments: given a query and a set of $t$ runs, we pick the run that is predicted to perform best for the query. We chose this setup over the data fusion setup evaluated by Yom-Tov et al. [167] and by Wu and Crestani [160], where the merging algorithm introduces an additional dimension, as it allows us to better control the experimental setting.

### 4.5.1 Experimental Details

Selecting the run that is *predicted to perform best* requires some consideration since we evaluate the rank correlation coefficient $\tau$ and thus rely on predicted ranks. The obvious mechanism is to consider the rank the query is predicted to have in each system and then use the system where the query is ranked highest (the $rank$ approach). This can be misleading though, in particular when the difference in systems' performances is large. To illustrate this point, we formulate the following example. Let for three queries the average precision scores of system $S_1$ be $(0.2, 0.4, 0.1)$ and for system $S_2$ let them be $(0.8, 0.6, 0.4)$. An accurate prediction method will predict the middle query to have rank $1$ in $S_1$ and rank $2$ in $S_2$, therefore, based on ranks, the output of $S_1$ would be picked. This is clearly incorrect.

This problem can be alleviated by transforming the ranks into average precision scores, that is the $i^{th}$ predicted rank is transformed into $i^{th}$ highest average precision score of the set of queries. Then for each query, the system with the highest *predicted average precision score* is chosen (the $score$ approach). Since this experiment assumes knowledge of the true average precision scores, the results can be considered as a lower bound for $\tau$ necessary to reliably improve retrieval effectiveness. The major results that follow are based on the $score$ approach. For comparison purposes, we report a number of results of the $rank$ approach as well.

In preliminary experiments we found two parameters influencing the retrieval effectiveness of predictor based meta-search. These are:

- the number $t$ of runs participating in the MS setup, and,

- the percentage $\gamma$ of improvement in MAP between the worst ($run_{low}$) and the best ($run_{high}$) performing run in the set of $t$ runs.

The experimental setup reflects these findings. We derived $500$ sets of $t$ runs from the TREC data sets for various settings of $\gamma$: $0\% - 5\%$, $5\% - 10\%$, $10\% - 15\%$, $15\% - 20\%$, $30\% - 35\%$ and $50\% - 55\%$. A range of $0\% - 5\%$ means, that all $t$ runs perform very similar with respect to MAP, while in the extreme setting of $\gamma$, the MAP of the best run in the set is between $50\%$ and $55\%$ higher than of the worst run. No limitations exist for the runs that are neither the best nor the worst in a set of runs.

It should be pointed out, that the setting of $\gamma$ influences two other factors. Let us briefly assume $t = 2$ and $MAP_{r_1} \leq MAP_{r_2}$. The parameter $\gamma$ influences the number of times a query of $r_1$ has a higher average precision than the corresponding query of $r_2$. It also influences the percentage of increase from $MAP_{r_2}$ to $MAP_{opt}$, the latter being the $MAP$ of the optimal meta-search run, formed by always picking the better performing query from $r_1$ and $r_2$. By not further restricting the parameters, we implicitly make the assumption that the positions at which one run outperforms another are random and that the amount of improvement or degradation is random as well.

In order to generate each set of runs, $t$ runs are randomly selected from the TREC runs of our data sets (for $m > 50$, runs are randomly concatenated). A set is valid if the maximum percentage of retrieval improvement lies in the specified interval of $\gamma$. In order to avoid sets of runs where a single spike in the difference in $ap$ overshadows the other query items, no query pair may have a difference in $ap$ larger than $0.4$[2]. Recall, that $1000$ predicted rankings exist per correlation coefficient interval $c_i$. As we require $t$ predicted rankings per set of runs, $t$ rankings are randomly chosen from all rankings of a given $c_i$. This implies that query performance predictors have similar performances in terms of $\tau$ over different systems.

Algorithm 2 offers a formal description of the experiment. The meta-search run $run_{meta}$ is created by selecting for each query the result of the run with the highest predicted $ap$ score. The optimal run $run_{opt}$ is derived by $ap_{opt}^{\mathbf{q_i}} = max(ap_1^{\mathbf{q_i}}, ..., ap_t^{\mathbf{q_i}})$. As in to the SQE experiments, we report the normalized performance of $run_{meta}$:

$$v_{meta} = 100 \times \frac{MAP_{meta} - MAP_{low}}{MAP_{opt} - MAP_{low}}, \tag{4.3}$$

where $MAP_{low}$ is the MAP value of the worst of the $t$ runs. When $v_{meta} < 0$, $run_{meta}$ performs worse than the worst run of the set, a value of $v_{meta} = 100$ on the other hand implies that $run_{meta}$'s performance is optimal, that is for every query the correct run is chosen. For MS to be considered a success, $run_{meta}$ needs to outperform the best individual run in the set of $t$ runs. As in the SQE experiments, this threshold is indicated by the horizontal lines in the box plots, which is the normalized median of the best runs' MAP across all sets of runs.

[2]This value was empirically chosen after evaluating the data sets.

**Algorithm 2:** Meta-search

**foreach** $c \in CORR$ **do** ▷*for all correlation intervals* $c$
  **for** $i \leftarrow 1, 1000$ **do**
    $r_1 = random(r \in RANK^c)$
    ...
    $r_t = random(r \in RANK^c)$
    **foreach** $(run_1, .., run_t) \in RUNS$ **do** ▷*for all run sets*
      $run_{meta} = \emptyset, run_{opt} = \emptyset$
      **for** $j \leftarrow 1, m$ **do**
        $s = max(score(run_1, r_1[j]), .., score(run_t, r_t[j]))$
        $x = run\_index\_of(s)$
        $ap_{meta}^{\mathbf{q_j}} = ap_x^{\mathbf{q_j}}$ ▷*MS*
        $ap_{opt}^{\mathbf{q_j}} = max(ap_1^{\mathbf{q_j}}, ..., ap_t^{\mathbf{q_j}})$
      **end**
      $MAP_{low} = min(MAP_1, .., MAP_t)$
      $MAP_{high} = max(MAP_1, .., MAP_t)$
      $v_{meta} = 100 \times \frac{MAP_{meta} - MAP_{low}}{MAP_{opt} - MAP_{low}}$
      $v_{high} = 100 \times \frac{MAP_{high} - MAP_{low}}{MAP_{opt} - MAP_{low}}$
    **end**
  **end**
**end**

Another aspect we consider is whether in cases where the meta-search run improves over the best run in the set of runs, the improvement is statistically significant. We performed a paired t-test [127] with significance level $0.05$ for each of those run pairs. We report this statistic by the percentage of run results for each correlation coefficient interval (that is, all 500 sets of systems times 1000 permutations) where the meta-search run significantly outperforms the best individual run.

### 4.5.2 Results

To gain an overview, we experimented with run sets of sizes $t = \{2, 3, 4, 5\}$. In the following two sections the results for $t = 2$ and $t = 4$ are detailed. Before describing the results, we define one more quantity, namely

$$ratio_{low} = \frac{w}{m}, \tag{4.4}$$

which is the fraction of queries $w$ in $run_{opt}$ that were chosen from the worst performing run; $\sigma(ratio_{low})$ is the corresponding standard deviation over all sets of runs. With increasing $\gamma$, less queries from the worst performing run can be expected to be utilized in the optimal run. For comparison, if the systems were to be chosen at random for each query, one would expect approximately $1/t$ queries to come from each system.

### Meta-Search with $t = 2$ Systems

First, we restrict ourselves to sets consisting of two systems each, that is $t = 2$, and evaluate the influence the parameter $\gamma$ has in the settings of $0\% - 5\%$, $5\% - 10\%$, $10\% - 15\%$ and $15\% - 20\%$. To get an impression of how $\gamma$ influences the quantity $ratio_{low}$, consider the results shown Table 4.4. The table contains the development for all stages of $\gamma$. If the queries had been drawn at random from both runs with equal probability, the mean would have been approximately $0.5$. Evidently, with increasing difference in the runs' effectiveness, less queries are selected on average from the worse performing run. To provide more detailed information, the last three columns of Table 4.4 show the average MAP of the worse and better performing run per set of systems as well as the MAP of the optimal run, which is derived by choosing the better performing result for each query from both runs.

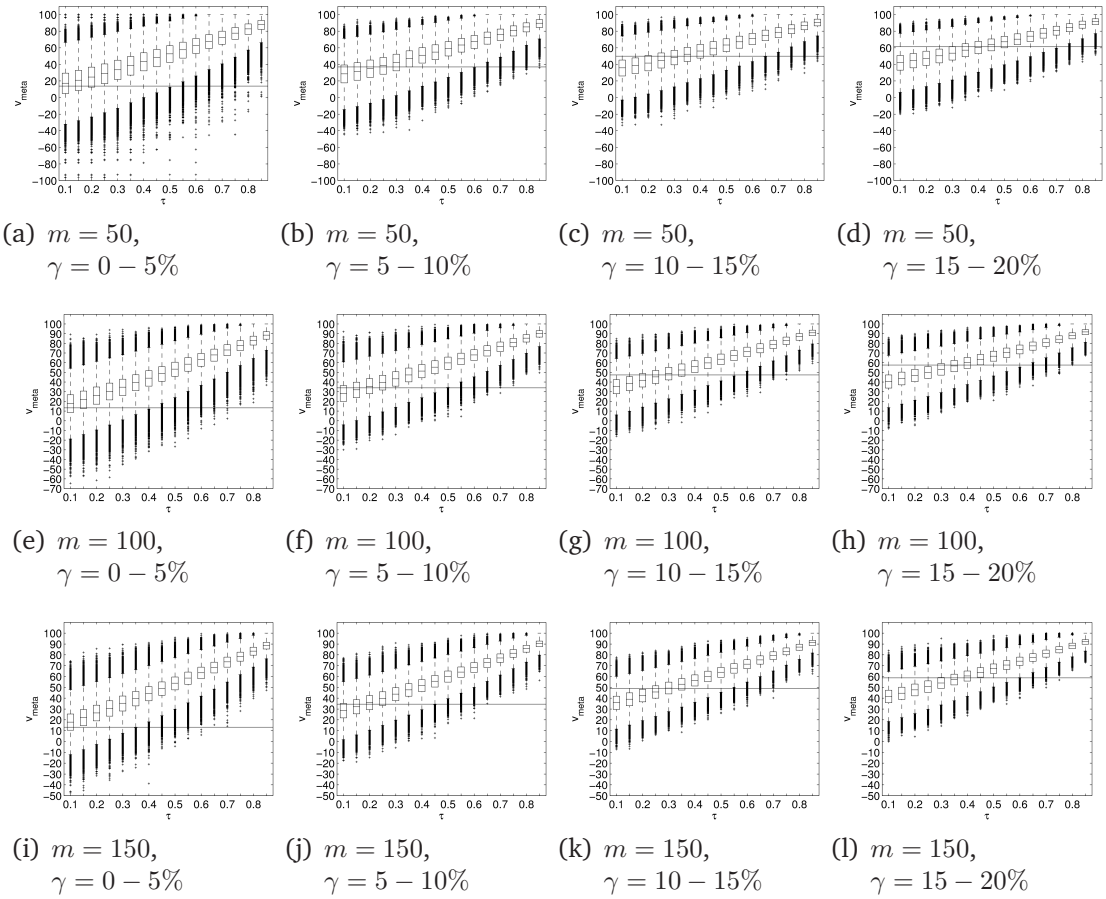
We also investigated if queries at certain positions within the query set are more likely to come from the worse performing run, but the results showed that the positions across the entire query set were covered uniformly.

| **m** | $\gamma$ | **#pairs** | $ratio_{low}$ | $\sigma(ratio_{low})$ | *av. MAP*$_{\mathbf{low}}$ | *av. MAP*$_{\mathbf{high}}$ | *av. MAP*$_{\mathbf{opt}}$ |
|---|---|---|---|---|---|---|---|
| 50 | $0-5\%$ | 500 | 0.478 | 0.057 | 0.254 | 0.260 | 0.298 |
| | $5-10\%$ | 500 | 0.432 | 0.053 | 0.249 | 0.268 | 0.300 |
| | $10-15\%$ | 500 | 0.394 | 0.056 | 0.235 | 0.264 | 0.293 |
| | $15-20\%$ | 500 | 0.355 | 0.055 | 0.227 | 0.267 | 0.292 |
| 100 | $0-5\%$ | 500 | 0.502 | 0.080 | 0.227 | 0.232 | 0.268 |
| | $5-10\%$ | 500 | 0.457 | 0.083 | 0.229 | 0.246 | 0.278 |
| | $10-15\%$ | 500 | 0.414 | 0.067 | 0.215 | 0.241 | 0.271 |
| | $15-20\%$ | 500 | 0.386 | 0.076 | 0.202 | 0.238 | 0.264 |
| 150 | $0-5\%$ | 500 | 0.514 | 0.080 | 0.236 | 0.242 | 0.277 |
| | $5-10\%$ | 500 | 0.468 | 0.078 | 0.227 | 0.244 | 0.276 |
| | $10-15\%$ | 500 | 0.427 | 0.075 | 0.217 | 0.244 | 0.272 |
| | $15-20\%$ | 500 | 0.395 | 0.076 | 0.208 | 0.244 | 0.269 |

Table 4.4: Statistics of the sets of runs, generated for the $t = 2$ experiments.

The results of the MS experiment with the $score$ approach applied are shown in Figure 4.5 in the form of box plots. Most notably, in contrast to the SQE experiment, a very low correlation $\tau$ can be sufficient to improve the effectiveness of $run_{meta}$ over the effectiveness of the better performing individual run. Specifically, the smaller $\gamma$, the lower the correlation $\tau$ may be to still lead to an improvement of $run_{meta}$. At the same time, the results show that for $\gamma = 0-5\%$ in particular, the spread of outliers is very wide, in the positive as well as the negative direction. Thus, while the majority of test cases improve at low $\tau$, in some instances an extremely poor performance is recorded, as evident by negative values of $v_{meta}$ across all parameter settings. Again the observation can be made that a larger value of $m$ gives more reliable results as the spread of results degrades.

To gain an understanding of the influence of the $score$ versus the $rank$ approach in Algorithm 2 (line 9), consider the results in Table 4.5. Analogous to the SQE experiments, we report the thresholds of $\tau$ to improve at least a quarter, half and three quarters of test cases where $run_{meta}$ improves over $run_{high}$ respectively. This



(a) $m = 50$, $\gamma = 0 - 5\%$ (b) $m = 50$, $\gamma = 5 - 10\%$ (c) $m = 50$, $\gamma = 10 - 15\%$ (d) $m = 50$, $\gamma = 15 - 20\%$

(e) $m = 100$, $\gamma = 0 - 5\%$ (f) $m = 100$, $\gamma = 5 - 10\%$ (g) $m = 100$, $\gamma = 10 - 15\%$ (h) $m = 100$, $\gamma = 15 - 20\%$

(i) $m = 150$, $\gamma = 0 - 5\%$ (j) $m = 150$, $\gamma = 5 - 10\%$ (k) $m = 150$, $\gamma = 10 - 15\%$ (l) $m = 150$, $\gamma = 15 - 20\%$

Figure 4.5: Meta-Search results with $score$ approach and $t = 2$ systems in a set.

gives an indication of what kind of $\tau$ is necessary to achieve adequate performance when applying query performance prediction in the MS setting. When $\gamma$ is low and thus the difference in effectiveness between the runs is low, there is no difference between $rank$ and $score$, the thresholds are the same. However, when $\gamma$ increases and thus the difference in performance between the participating systems increases, the $\tau$ required by the $rank$ approach to achieve the same MS performance as the $score$ approach is between $0.05$ and $0.1$ higher.

The results of the significance tests are shown in Figure 4.6, where we investigate to what extent $run_{meta}$ performs better than $run_{high}$ in a statistically significant manner. The plot shows the percentage of test cases in which $run_{meta}$ significantly improves over $run_{high}$ for each correlation interval $c_i$. Although we showed that low $\tau$ is sufficient to improve the retrieval effectiveness, significant improvements are considerably more difficult to achieve. The trend is clear: the lower $m$ and the higher $\gamma$, the more difficult it is to obtain significant improvements. If we aim for at least $50\%$ significant improvements in the test cases, $\tau \geq 0.4$ for the setting of $m = 150$ and $\gamma = 0 - 5\%$. If $m = 50$, a correlation of $\tau \geq 0.6$ is required.

| **m** | $\gamma$ | **25%** | | **50%** | | **75%** | | **min. 95% opt.** | |
|---|---|---|---|---|---|---|---|---|---|
| | | *score* | *rank* | *score* | *rank* | *score* | *rank* | *score* | *rank* |
| 50 | $0-5\%$ | 0.10 | 0.10 | 0.10 | 0.10 | 0.25 | 0.25 | 0.10 | 0.10 |
| | $5-10\%$ | 0.10 | 0.10 | 0.25 | 0.25 | 0.35 | 0.40 | 0.35 | 0.35 |
| | $10-15\%$ | 0.20 | 0.30 | 0.35 | 0.40 | 0.45 | 0.55 | 0.15 | 0.45 |
| | $15-20\%$ | 0.30 | 0.40 | 0.45 | 0.55 | 0.55 | 0.65 | 0.40 | 0.50 |
| 100 | $0-5\%$ | 0.10 | 0.10 | 0.10 | 0.10 | 0.20 | 0.20 | 0.45 | 0.45 |
| | $5-10\%$ | 0.10 | 0.10 | 0.20 | 0.20 | 0.30 | 0.35 | 0.35 | 0.45 |
| | $10-15\%$ | 0.20 | 0.25 | 0.30 | 0.40 | 0.40 | 0.45 | 0.40 | 0.50 |
| | $15-20\%$ | 0.30 | 0.35 | 0.35 | 0.45 | 0.45 | 0.55 | 0.45 | 0.60 |
| 150 | $0-5\%$ | 0.10 | 0.10 | 0.10 | 0.10 | 0.15 | 0.15 | 0.50 | 0.60 |
| | $5-10\%$ | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.30 | 0.55 | 0.65 |
| | $10-15\%$ | 0.25 | 0.25 | 0.30 | 0.35 | 0.40 | 0.45 | 0.60 | 0.65 |
| | $15-20\%$ | 0.35 | 0.40 | 0.40 | 0.50 | 0.50 | 0.55 | 0.60 | 0.75 |

Table 4.5: Meta-search scenario with $t = 2$ systems in a set: summary of $\tau$ necessary to improve $25\%$, $50\%$ and $75\%$ of the sets of systems over the median of the best individual system for the $rank$ and $score$ approach. The final two columns contain the minimum $\tau$ where at least one MS run reaches $95\%$ performance of the optimal MS run.

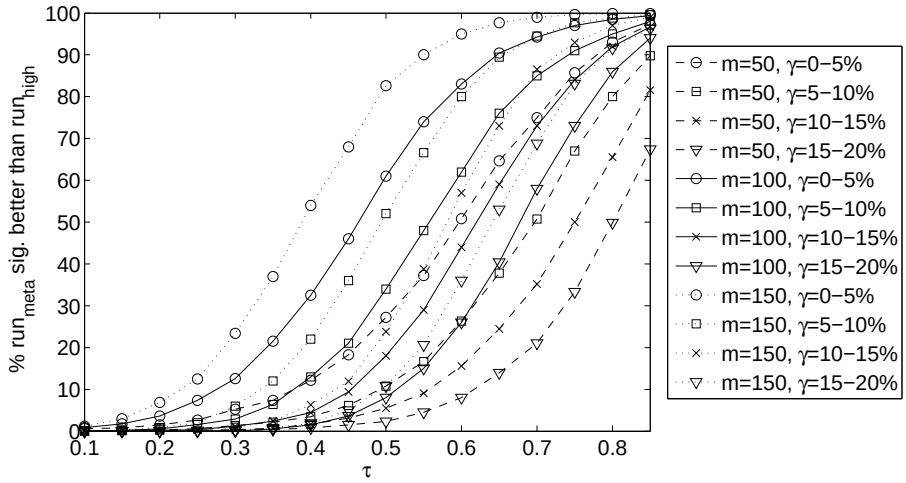



Figure 4.6: Significance testing for the $t = 2$ systems setup: percentage of test cases where $run_{meta}$ significantly improves over $run_{high}$.

### Meta-Search with $t = 4$ Systems

A MS setting with only two participating systems might not be very realistic. For this reason we further we experiment with increasing the number of systems. In this section specifically, the results of the experiments with $t = 4$ systems are described. Apart from changing $t$, larger intervals of $\gamma$ are evaluated: $0-5\%$, $15-20\%$, $30-35\%$ and $50-55\%$. As more systems participate in the meta-search setup, the likelihood increases that at least one system performs particularly poorly or well.

The statistics of the generated sets of systems are detailed in Table 4.6. At $t = 4$, if the run selection were to be random, we expect $25\%$ of the results to come from each run. As expected, with increasing $\gamma$, less results from the worst performing

| **m** | $\gamma$ | **#pairs** | $ratio_{low}$ | $\sigma(ratio_{low})$ | av. $MAP_{\mathbf{low}}$ | av. $MAP_{\mathbf{high}}$ | $MAP_{\mathbf{opt}}$ |
|---|---|---|---|---|---|---|---|
| 50 | $0-5\%$ | 500 | 0.234 | 0.077 | 0.236 | 0.245 | 0.328 |
| | $15-20\%$ | 500 | 0.184 | 0.067 | 0.220 | 0.259 | 0.334 |
| | $30-35\%$ | 500 | 0.145 | 0.059 | 0.209 | 0.277 | 0.344 |
| | $50-55\%$ | 500 | 0.115 | 0.051 | 0.196 | 0.298 | 0.356 |
| 100 | $0-5\%$ | 500 | 0.229 | 0.073 | 0.230 | 0.239 | 0.340 |
| | $15-20\%$ | 500 | 0.192 | 0.068 | 0.212 | 0.250 | 0.341 |
| | $30-35\%$ | 500 | 0.171 | 0.061 | 0.198 | 0.262 | 0.341 |
| | $50-55\%$ | 500 | 0.138 | 0.060 | 0.180 | 0.275 | 0.351 |
| 150 | $0-5\%$ | 500 | 0.235 | 0.065 | 0.222 | 0.231 | 0.345 |
| | $15-20\%$ | 500 | 0.200 | 0.062 | 0.211 | 0.249 | 0.347 |
| | $30-35\%$ | 500 | 0.175 | 0.057 | 0.195 | 0.259 | 0.348 |
| | $50-55\%$ | 500 | 0.145 | 0.056 | 0.182 | 0.277 | 0.354 |

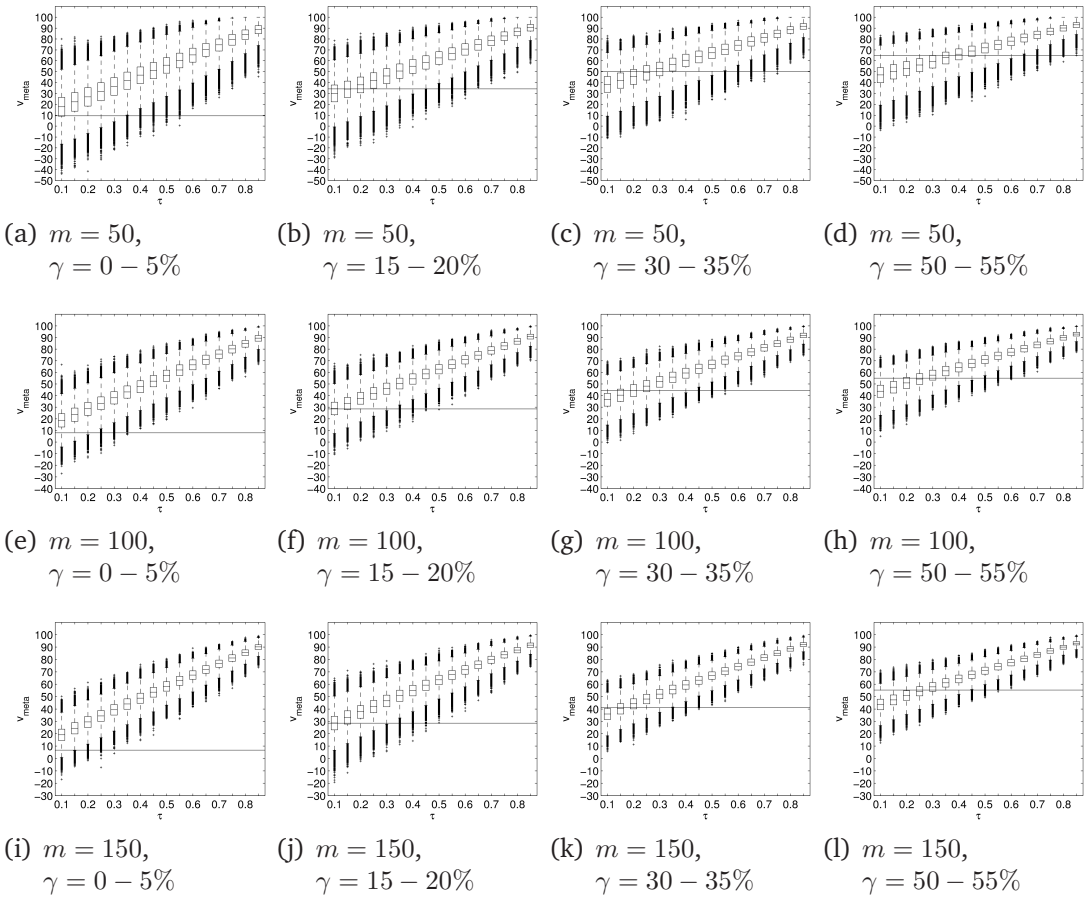
Table 4.6: Statistics of the sets of runs, generated for the $t = 4$ experiments.

run are utilized in the optimal run. Though even for large differences in retrieval effectiveness ($\gamma = 50\% - 55\%$), the worst performing run still contributes towards the optimal run. Moreover, with increasing $\gamma$, the MAP of the worst performing run consequently degrades whereas the MAP of the best performing run increases. The performance of the optimal run on the other hand is not as strongly dependent on $\gamma$, a slight increase in MAP is observed when $\gamma$ increases.

| **m** | $\gamma$ | **25%** | | **50%** | | **75%** | | **min. 95% opt.** | |
|---|---|---|---|---|---|---|---|---|---|
| | | *score* | *rank* | *score* | *rank* | *score* | *rank* | *score* | *rank* |
| 50 | $0-5\%$ | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.15 | 0.55 | 0.65 |
| | $15-20\%$ | 0.10 | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.60 | 0.65 |
| | $30-35\%$ | 0.20 | 0.30 | 0.30 | 0.35 | 0.35 | 0.45 | 0.60 | 0.65 |
| | $50-55\%$ | 0.30 | 0.50 | 0.40 | 0.55 | 0.50 | 0.65 | 0.55 | 0.60 |
| 100 | $0-5\%$ | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.70 | 0.75 |
| | $15-20\%$ | 0.10 | 0.10 | 0.10 | 0.15 | 0.20 | 0.20 | 0.70 | 0.75 |
| | $30-35\%$ | 0.15 | 0.20 | 0.20 | 0.30 | 0.30 | 0.35 | 0.70 | 0.75 |
| | $50-55\%$ | 0.20 | 0.30 | 0.30 | 0.35 | 0.35 | 0.45 | 0.65 | 0.80 |
| 150 | $0-5\%$ | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.15 | 0.80 | 0.80 |
| | $15-20\%$ | 0.10 | 0.10 | 0.10 | 0.15 | 0.20 | 0.20 | 0.70 | 0.80 |
| | $30-35\%$ | 0.15 | 0.15 | 0.20 | 0.25 | 0.25 | 0.30 | 0.75 | 0.80 |
| | $50-55\%$ | 0.25 | 0.30 | 0.30 | 0.35 | 0.35 | 0.45 | 0.75 | 0.85 |

Table 4.7: Meta-search scenario with $t = 4$ systems per set: summary of $\tau$ necessary to improve $25\%$, $50\%$ and $75\%$ of the sets of systems over the median of the best individual system for the $rank$ and $score$ approach. The final two columns contain the minimum $\tau$ where at least one MS run reaches $95\%$ performance of the optimal MS run.

The results of the experiments with $t = 4$ systems are depicted in Figure 4.7. The trends are similar to the experiment with $t = 2$ systems (Figure 4.5). When $\gamma$ is low and thus the difference in retrieval effectiveness between the retrieval systems is minor, predictions that achieve low correlations are sufficient to improve $run_{meta}$ over the effectiveness of the best run in the set. In contrast to the results for $t = 2$, however, the spread of results is considerably smaller; in particular the degradation over the worst performing individual run is less developed. Although this is partially



(a) $m = 50$, $\gamma = 0 - 5\%$ (b) $m = 50$, $\gamma = 15 - 20\%$ (c) $m = 50$, $\gamma = 30 - 35\%$ (d) $m = 50$, $\gamma = 50 - 55\%$

(e) $m = 100$, $\gamma = 0 - 5\%$ (f) $m = 100$, $\gamma = 15 - 20\%$ (g) $m = 100$, $\gamma = 30 - 35\%$ (h) $m = 100$, $\gamma = 50 - 55\%$

(i) $m = 150$, $\gamma = 0 - 5\%$ (j) $m = 150$, $\gamma = 15 - 20\%$ (k) $m = 150$, $\gamma = 30 - 35\%$ (l) $m = 150$, $\gamma = 50 - 55\%$

Figure 4.7: Meta-search results with $score$ approach and $t = 4$ systems in a set.

due to the greater values of $\gamma$ used as compared to the results in Figure 4.5, this behavior can also be observed for $\gamma = 0 - 5\%$. So, we can deduce that the chance of *not* randomly picking the worst performing system is higher for $t = 4$ than for $t = 2$ systems. Finally, again, we can observe that greater $m$ leads to considerably more reliable results, indicated by the decreased height of the boxes.

Consider Table 4.7 for an overview of the $\tau$ thresholds necessary to improve different quartiles of the meta-search test cases over the best individual systems. As before, both the $rank$ and the $score$ approaches are reported. The results are similar to Table 4.5, as with increasing $\gamma$ the thresholds increase; this is less pronounced for $m = 150$ as compared to $m = 100$ and $m = 50$. On the other hand, the results of the last column, which contains the correlation coefficient where first close to optimal performance is achieved, are considerably different from the results in Table 4.5. We observe that while for $t = 2$ and $m = 50$ at $\tau = 0.1$ at least one test case already achieved nearly optimal performance, for $t = 4$ systems the threshold lies at $\tau = 0.65$ and it further increases with increasing $m$.

The results of the significance tests over the pairs of best individual and meta-search runs are presented in Figure 4.8. The plot shows the percentage of samples where $run_{meta}$ significantly outperforms the best individual run in the run set. At

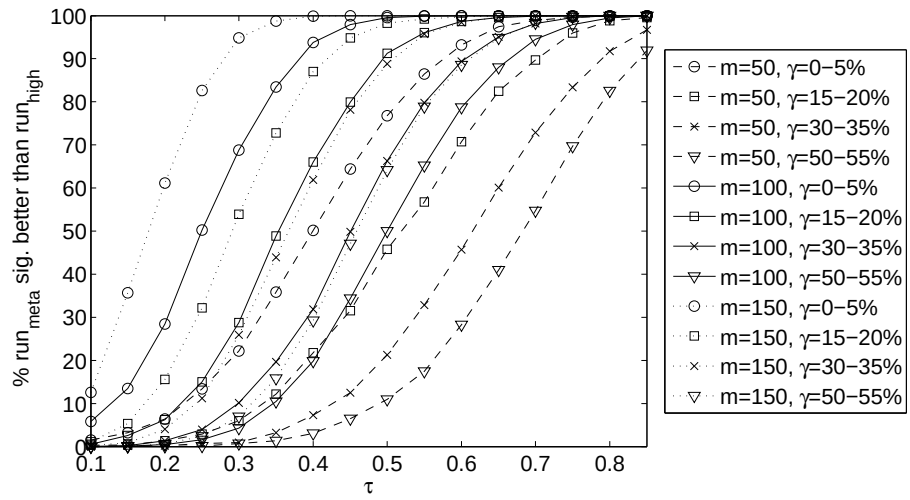



Figure 4.8: Significance testing for the $t = 4$ systems setup: percentage of test cases where $run_{meta}$ significantly improves over $run_{high}$.

$m = 50$, the correlation threshold of $\tau$ to improve $50\%$ of the samples significantly, ranges from $\tau = 0.4$ for $\gamma = 0\% - 5\%$ to $\tau = 0.7$ at $\gamma = 50\% - 55\%$. The thresholds are lower for $m = 150$: $\tau = 0.2$ ($\gamma = 0\% - 5\%$) and $\tau = 0.5$ ($\gamma = 50\% - 55\%$) respectively.

## 4.6 Discussion

The previous two sections established thresholds for $\tau$ that a query effectiveness predictor should reach to be reasonably confident that an adaptive retrieval system employing that prediction method will improve over the non-adaptive baseline. Here, for the sake of comparison, we report once more the rank correlation coefficients of a small number of methods that were introduced in Chapter 2 and 3: *MaxIDF*, *MaxSCQ*, *MaxVAR*, Clarity Score and the best performing setup of Adapted Clarity Score.

The results in Table 4.8 are shown for different levels of smoothing. Additionally, those correlations are marked, that in our SQE and MS experiments have lead to an – in the case of MS *significant* – improvement of at least $50\%$ of the test cases over the non-adaptive approach. The main point to be made is that in most instances, the reported correlations are sufficient for one specific setting only: meta-search with $t = 4$ and $\gamma = 0 - 5\%$. Of the remaining MS experiments, only Adapted Clarity and *MaxVAR* attain a high enough correlation, though only for one particular test collection (TREC Vol. 4+5). Thus, in the MS setting, the usability of query performance predictors depends to a large extent on how stark the difference in performance between the participating systems is.

In the case of SQE, the outcome is less encouraging. Apart from Adapted Clarity, which is adequate in SQE's best-case scenario for TREC Vol. 4+5, none of the reported correlations are high enough. When small perturbations of $p = 10\%$ are

| | TREC Vol.4+5 | | | WT10g | | | GOV2 | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\mu = 100$ | $\mu = 1500$ | $\mu = 5000$ | $\mu = 100$ | $\mu = 1500$ | $\mu = 5000$ | $\mu = 100$ | $\mu = 1500$ | $\mu = 5000$ |
| **MaxIDF** | $0.34^{5}$ | $0.35^{5}$ | $0.40^{3,5}$ | $0.25^{5}$ | $0.28^{5}$ | $0.30^{5}$ | $0.26^{5}$ | $0.25^{5}$ | $0.32^{5}$ |
| **MaxSCQ** | $0.33^{5}$ | $0.34^{5}$ | $0.38^{5}$ | $0.28^{5}$ | $0.34^{5}$ | $0.37^{5}$ | $0.27^{5}$ | $0.28^{5}$ | $0.33^{5}$ |
| **MaxVAR** | $0.40^{3,5}$ | $0.41^{3,5}$ | $0.44^{3,5}$ | $0.29^{5}$ | $0.33^{5}$ | $0.36^{5}$ | $0.27^{5}$ | $0.29^{5}$ | $0.30^{5}$ |
| **Clarity Score** | $0.39^{5}$ | $0.30^{5}$ | $0.21^{5}$ | 0.20 | 0.19 | 0.16 | $0.28^{5}$ | $0.31^{5}$ | $0.32^{5}$ |
| **Adap. Clarity** | $0.47^{1,3}_{5}$ | $0.48^{1,3}_{5}$ | $0.50^{1,3}_{5,6}$ | $0.25^{5}$ | $0.26^{5}$ | $0.26^{5}$ | $0.28^{5}$ | $0.34^{5}$ | $0.36^{5}$ |

Table 4.8: Overview of Kendall's $\tau$ correlation coefficients over different levels of smoothing for the Language Modeling retrieval approach. The markers indicate when a $\tau$ lead to a - for MS *significant* - improvement in at least 50% of the test cases in our experiments. The markers stand for the different types of experiments: SQE best-case ($^{1}$), SQE $p = 10\%$ ($^{2}$), MS $t = 2 \wedge \gamma = 0 - 5\%$ ($^{3}$), MS $t = 2 \wedge \gamma = 15 - 20\%$ ($^{4}$), MS $t = 4 \wedge \gamma = 0 - 5\%$ ($^{5}$) and MS $t = 4 \wedge \gamma = 50 - 55\%$ ($^{6}$). The $\tau$ thresholds of the MS experiments used are based on the $score$ approach.

introduced to the SQE setup, Adapted Clarity also fails across all corpora. The perturbation result indicates, that in the SQE setting it is not sufficient to apply a good enough query performance predictor method. It is also crucial that the initial SQE assumption (well performing queries improve, poorly performing queries degrade with AQE) is investigated for the particular AQE method used in the application. Furthermore, in our experiments we made the simplifying assumption that the value of $\theta$, which is the threshold to which AQE improves retrieval effectiveness, is known to us. In a practical setting this value can only be approximated, further increasing the difficulty to a successful application of query performance predictors in the SQE setting.

It should also be emphasized once more, that across all parameter settings for both the SQE and MS experiments, improvements over the baseline are reached at all levels of $\tau$, that is, even if $\tau = 0.1$ and the majority of test cases degrade, there are always outliers, that by chance lead to an improved effectiveness of the adaptive retrieval system.

## 4.7 Conclusions

In this chapter we have investigated the relationship of one standard evaluation measure of query performance prediction, namely Kendall's $\tau$, and the change in retrieval effectiveness when predictors are employed in two operational settings: selective query expansion and meta-search. We have primarily focused on investigating when prediction methods can be considered good enough to be viable in practice. To this end, we performed a comprehensive evaluation based on TREC data sets.

We found that the required level of correlation varies and depends on the particular operational setting a prediction method is employed in. In the case of SQE, in the best-case scenario, $\tau \geq 0.4$ is found to be the minimum level of correlation for the SQE runs to outperform the AQE runs in $50\%$ of the samples. In a second experiment we were able to show the danger of assuming AQE to behave in a certain

way; slightly violating a commonly made AQE assumption already requires predictors with a correlation of $\tau \geq 0.75$ for them to be viable in practice, as the more accurate predictor is able to compensate the less accurate assumption made about when AQE is beneficial.

The outcome of our investigation was different in the case of meta-search. Here, the level of correlation was shown to be dependent on the performance differences of the participating systems but also on the number of systems employed. If the participating runs are similar, prediction methods with low correlations, that are not significantly different from zero are sufficient to improve $50\%$ of the runs. If the differences in retrieval effectiveness between the systems are large, a correlation of $\tau = 0.3$ is required. To achieve statistically significant improvements for $50\%$ of the runs under large system differences, correlations of $\tau = 0.7$ ($m = 50$) and $\tau = 0.5$ ($m = 150$) can be considered as lower boundaries.

The above results may be summed as showing that query performance prediction methods need further improvement to become viable in practice, in particular for the SQE setting. Also, as query set sizes $m$ increase, the evaluation in terms of Kendall's $\tau$ relates better to the change in effectiveness in an operational setting.

This analysis has serious implications for the area of query performance prediction. It indicates that predictors should not only be tested in isolation, they should also be studied in the context of an application in order to contextualize the effectiveness, particularly if $\tau$ is not very high. While this research provides a guide that shows the strength of correlation needed in order to achieve improvements, there are always cases where outliers are likely to have a significantly adverse affect on performance. From the analysis, it becomes evident that current methods for query effectiveness prediction need to be further improved in order to realize the potential gains.

# Chapter 5

# A Case for Automatic System Evaluation

## 5.1 Introduction

Ranking retrieval systems according to their retrieval effectiveness *without* relying on costly relevance judgments is a challenging task which was first explored by Soboroff et al. [133]. The motivation for this research stems from the high costs involved in the creation of test collections, coupled with more and larger collections becoming available. As an illustration, while the GOV2 corpus [38], which was introduced to TREC in 2004, contains roughly 25 million documents, the ClueWeb09 corpus, introduced to TREC in 2009, contains already more than one billion documents, a forty-fold increase in size.

Moreover, in a dynamic environment such as the World Wide Web, where the collection [58, 113] and user search behavior change over time [155, 156], regular evaluation of search engines with human relevance assessments is not feasible [79, 133]. If it becomes possible to determine the relative effectiveness of a set of retrieval systems, reliably and accurately, without the need for relevance judgments, then the cost of evaluation could be greatly reduced.

Additionally, such estimated ranking of retrieval systems could not only serve as a way to compare systems, it can also provide useful information for other applications, such as retrieval model selection [159], where we are interested in finding the best retrieval model in a query dependent fashion, or data fusion, where the estimated ranking of systems can be relied upon to derive merging weights for each system [160, 167].

In recent years, a number of *system ranking estimation* approaches have been proposed [9, 114, 133, 135, 161], which attempt to rank a set of retrieval systems (for a given topic set and test corpus) without human relevance judgments. All these approaches estimate a performance based ranking of retrieval systems by considering the relationship of the top retrieved documents across all or a number of selected systems. While the initial results highlighted the promise of this new direction, the utility of system ranking estimation methods remains unclear, since they have been shown to consistently underestimate the performance of the best systems,

an observation which is attributed to the "tyranny of the masses" effect [9]. This is a very important limitation, as, in practice, it is often the best systems that we are most interested in identifying accurately, rather than the average systems.

In the analysis presented in this chapter, we will show that the problem of mis-ranking the best systems is not inherent to system ranking estimation methods. In previous work [9, 114, 133, 135, 161], the evaluations were mostly performed on the TREC-{3,5,6,7,8} data sets. Note that when we refer to a TREC data set, such as TREC-3, we mean all retrieval runs submitted to TREC for the topics of that task. Since a retrieval run is the output of a retrieval system, by ranking retrieval runs, we rank retrieval systems. We will use the terms *run* and *system* mostly interchangeably. In our work, we evaluate system ranking estimation methods on a much wider variety of data sets than previously. We consider a total of sixteen different TREC data sets. They include a range of non-adhoc task data sets, such as expert search [42] and named page finding [43], as well as adhoc tasks on non-traditional corpora, for instance the Blog [115] and Genomics [75] corpora. We observe that the extent of mis-ranking the best systems varies considerably between data sets and is indeed strongly related to the degree of human intervention in the best runs of a data set. This finding suggests that under certain conditions, automatic system evaluation is a viable alternative to human relevance judgments based evaluations.

In a second set of experiments, we also investigate the number of topics required to perform system ranking estimation. In all existing approaches, the retrieval results of the full TREC topic set are relied upon to form an estimate of system performance. However, in [61, 109] it is demonstrated that some topics are better suited than others to differentiate the performance of retrieval systems. Whilst these works were not performed in the context of system ranking estimation, we consider this observation as a starting point for our work. We hypothesize, that with the right subset of topics, the current methods for estimating system rankings without relevance judgment can be significantly improved. To verify this claim, we implement five different approaches [50, 114, 133, 135] to system ranking estimation and compare their performances to each other. We experimentally evaluate the extent of the topic dependent performance and perform a range of experiments to determine the degree to which topic subsets can improve the performance of system ranking estimation approaches. Finally, we attempt to automatically find a good subset of topics to use for system ranking estimation.

Specifically, in this chapter we will show that:

- the ability to accurately identify the best system of a set of retrieval systems is strongly related to the amount of human intervention applied to the system: the larger the amount of human intervention, the less able we are to identify it correctly,
- across the evaluated system ranking estimation approaches, the original work by Soboroff et al. [133] is the most consistent and gives the best performance overall,
- the ability of system ranking estimation methods to estimate a ranking of systems *per topic* varies highly,

- topic subset selection improves on average the performance of the approach proposed by Soboroff et al. [133] by $26\%$ and up to a maximum of $56\%$ (similar improvements are observed for the other system ranking estimation methods), and,
- on average, a subset size of $10$ topics yields the highest system ranking estimation performance, a result that is consistent across all data sets and corpora, independent of the number of topics contained in the full TREC topic set.

This chapter is organized as follows: in Section 5.2, we provide an overview of related work in the area of system ranking estimation. Then, in Section 5.3, we introduce the motivation for our experiments in topic subset selection. In Section 5.4, the experimental setup is described and the data sets under investigation are outlined. The empirical analysis, which forms the main part of this chapter, is described in Section 5.5. It contains a comparison of different ranking estimation approaches on the full set of topics (Section 5.5.1) and an analysis of the methods' abilities to determine the correct ranking of systems for each individual topic (Section 5.5.2). The amount of possible performance gain when relying on a subset of topics is discussed in Section 5.5.3. A first attempt to automatically find a good subset of topics is then made in Section 5.5.4. The chapter concludes with a summary in Section 5.6.

## 5.2 Related Work

Research aiming to reduce the cost of evaluation has been conducted along two lines. Specifically, a number of approaches focus on *reducing* the amount of manual assessments required, while others rely on *fully automatic* evaluation, foregoing the need for manual assessments altogether. Approaches in the first category include the determination of good documents to judge [33, 153], the proposal of alternative pooling methods [8] in contrast to TREC's depth pooling, the proposal of alternative evaluation measures for incomplete judgments [8, 27], the usage of term relevance judgments instead of document relevance judgments [6] and the reliance on manually created queries to derive pseudo-relevant documents [55].

Whilst the aforementioned methods have been developed in the context of TREC, the works by Abdur Chowdhury and his colleagues [16, 17, 37, 79] investigate the evaluation of Web search engines through automatically generated known item queries from query logs and manually built Web directories such as the ODP. Despite the fact, that the queries are derived automatically, we still consider these approaches to be part of the efforts aimed at reducing the amount of manual assessments, as the ODP is constantly maintained by human editors.

In this chapter, we consider approaches of the second category, that is, we focus on algorithms that are completely automatic and require no manual assessments at all. The first work in the ranking of retrieval systems which did not include manually derived relevance judgments is attributed to Soboroff et al. [133] and was motivated by the fact that the relative ranking of retrieval systems remains largely

unaffected by assessor disagreement in the creation of relevance judgments [149]. This observation led to the proposed use of automatically created *pseudo* relevance judgments. In this case, the pseudo relevant documents are derived in the following manner: first, the top retrieved documents across the TREC runs for a particular topic are pooled together such that a document that is retrieved by $x$ systems, appears $x$ times in the pool[1]. Then, a number of documents are drawn at random from the pool; those are now considered to be the relevant documents. This process is performed for each topic and the subsequent evaluation of each system is performed with pseudo relevance judgments instead of relevance judgments. In the end, a system's effectiveness is estimated by its pseudo mean average precision. To determine the accuracy of this estimate, the pseudo relevance judgment based system ranking is compared against the ground truth ranking, that is the ranking of systems according to a retrieval effectiveness measure such as MAP. All experiments reported in [133] were performed on the data sets TREC-{3,5,6,7,8}. Although the reported correlations are significant, one major drawback was discovered: whereas the ranking of the poorly and moderately performing systems is estimated quite accurately with this approach, the ranks of the best performing systems are always underestimated. This is not a small issue, the extent of mis-ranking the best system(s) is severe. For instance, in the TREC-8 data set, where $129$ systems are to be ranked, the best system according to the ground truth in MAP is estimated to be ranked at position $113$ (Table 5.2), which means it is estimated to be one of the worst performing systems. It was later suggested by Aslam and Savell [9] that this observation can be explained by the "tyranny of the masses" effect, where the best systems are estimated to perform poorly due to them being different from the average. The evaluation can therefore be considered to be based more on popularity than on performance.

The exploitation of pseudo relevant documents has been further investigated by Nuray and Can [114], on a very similar data set, specifically TREC-{3,5,6,7}. In contrast to Soboroff et al. [133], not all available retrieval systems participate in the derivation of pseudo relevance judgments. The authors experiment with different approaches to find a good subset of $P\%$ of systems. Overall, the best approach is to select those systems that are most different from the average system. Once a subset of systems is determined, the top $b$ retrieved documents of each selected system are merged and the top $s\%$ of the merged result list constitute the pseudo relevance judgments. Different techniques are evaluated for merging the result lists; the best performing approach is to rely on Condorcet voting where each document in the list is assigned a value according to its rank. In this way, it is not only the frequency of occurrence of a document in various result lists that is a factor as in [133], but also the rank the document is retrieved at. By placing emphasis on those systems, that are most dissimilar from the average, this work directly addresses the "tyranny of the masses" criticism [9] levelled at the approach in [133]. The results reported in [114] outperform the results reported in [133] for the data sets evaluated. However, in this chapter, we perform an extensive evaluation across a

[1]Removal of duplicates from the pool was also investigated in [133], but proved to be less successful.

larger number of more recent and varied data sets, and will show that this approach does not always deliver a better performance.

Another direction of research is to directly estimate a ranking of systems based on the document overlap between different result lists, instead of deriving pseudo relevance judgments. Wu and Crestani [161] propose to rank the retrieval systems according to their *reference count*. The reference count of a system and its ranked list for a particular topic is the number of occurrences of documents in the ranked lists of the other retrieval systems; a number of normalized and weighted counting methods are also proposed. Experiments on data sets TREC-{3,5,6,7,10} generally yield lower correlations than in [114, 133].

A variation, which relies on the *structure of overlap* of the top retrieved documents between different retrieval systems, is proposed by Spoerri [135]. Spoerri suggests that instead of ranking all systems at once, as done in the previous approaches, it is beneficial to repeatedly rank a set of five randomly chosen systems based on their document overlap structure and average the results across all trials to gain a ranking over all systems. It is hypothesized, that adding all available systems at once creates a considerable amount of "noise" as near duplicate systems are entered, thus boosting some documents in a biased way. While the reported experiments on TREC-{3,6,7,8} exhibit considerably higher correlations than previous work, and the best systems are consistently ranked in at least the top half of the ranking, it needs to be emphasized that the results are not directly comparable to earlier work. In [135], the evaluation is based only on automatic short runs with the further restriction that only the best performing run per participating group is included. This means for instance, instead of basing the evaluation of TREC-8 on 129 systems as in [9, 114, 133, 161], only 35 systems are evaluated. We will show, that when including all available systems of a data set in the experiments, this method does not perform better than previously introduced approaches.

Finally, in [34] it is proposed to circumvent the problem of mis-ranking the best systems, by assuming those system to be known and reweighting their contribution towards the pseudo-relevance judgments accordingly. The evaluation, performed on TREC-{3,4,5,6,7,8}, shows the validity of the approach. Such an approach however leads to a circular argument - we rely on automatic evaluation methods to rank the systems according to their performance, but to do so, we require the best systems to be known in advance.

The aforementioned methods have all assumed that all topics of a TREC topic set are equally useful in estimating the ranking of systems. However, recent research on evaluation which relies on manual judgments to rank systems has found that only a subset of topics is needed [61, 109]. We discuss this in the next section and consider how the same idea can be applied when relevance judgments are not available.

## 5.3 Topic Subset Selection

In order to explore the relationship between a set of TREC topics and a set of retrieval systems, Mizzaro and Robertson [109] took a network analysis based view.

They proposed the construction of a complete bipartite *Systems-Topic graph* where systems and topics are nodes and a weighted edge between a system and a topic represents the retrieval effectiveness of the pair.

Network analysis can then be performed on the graph, in particular, Mizzaro & Robertson employed HITS [87], a method that returns a hub and authority value for each node. The correspondence between those measures and systems and topics was found to be as follows

- the authority of a system node indicates the system's effectiveness,
- the hubness of a system node indicates the system's ability to estimate topic difficulty,
- the authority of a topic node is an indicator for topic difficulty, and, finally,
- the hubness of a topic node indicates the topic's ability to estimate system performance.

While the study in [109] was more theoretic in nature, a recent follow up on this work by Guiver et al. [61] has shown experimentally that when selecting the right subset of topics, the resulting relative system performance is very similar to the system performance on the full topic set, thus allowing the reduction of the number of topics required. The same work though concedes concrete ideas of how to select those topics to future work.Mizzaro [108] also proposed a novel evaluation metric, the Normalized MAP value, which takes the difficulty of a topic into account when evaluating systems.

The finding that individual topics vary in their ability to indicate system performance provides the basis for our work as it implies that there might be subsets of topics that are as suited to estimate the system performance as the full set of topics provided for a TREC task. While the motivation in [61, 109] is to reduce the cost of evaluation by reducing the topic set size, in this work, we are motivated by the fact that system ranking estimation does not perform equally well across all topics.

We examine the following research questions:

- By reducing the topic set, can the performance of current system ranking estimation methods be improved and if so to what extent?
- Can the reduced topic sets that improve the estimation be selected automatically?
- To what extent does the performance of system ranking estimation approaches depend on the set of systems to rank and the set of topics available?

## 5.4 Materials and Methods

In order to improve the validation power of our results we conduct our analysis on sixteen data sets and five system ranking estimation methods. We shall refer to the estimation methods that we employ in the following fashion:

- the *Data Fusion* ($DF$) approach by Nuray and Can [114],
- the *Random Sampling* ($RS$) approach by Soboroff et al. [133],
- the *Structure of Overlap* ($SO$) approach by Spoerri [135],
- the *Autocorrelation based on Document Scores* (*ACScore*) approach by Diaz [50], and,
- the *Autocorrelation based on Document Similarity and Scores* (*ACSimScore*) approach by Diaz [50].

Whereas the first three approaches have already been introduced in Section 5.2, the latter two (*ACSimScore* and *ACScore*) have not been applied to system ranking estimation yet. They are proposed in [50] to evaluate aspect EA3 of Figure 1.1. The main motivation for evaluating specifically those approaches is their mix of information sources. In particular, $RS$ relies on document overlap as shown in [9], $SO$ considers small subsets of systems for ranking, while *ACScore* and $DF$ take the particular retrieval score and the rank respectively a system assigns to a document into account. Finally, the *ACSimScore* approach goes a step further and considers the content similarity between ranked documents and the retrieval scores to determine the relative effectiveness of a system.

Each approach derives a performance score for each pair $(t_i, s_j)$ of topic $t_i$ and system $s_j$. In order to derive a system's performance score over a particular topic set, the scores the system achieves across all topics in the set are averaged. Based on the scores, assigned to each system by a system ranking estimation approach, the ranking of retrieval systems is estimated. This ranking is then correlated against the ground truth ranking of systems. In our experiments, we will rely on two ground truth rankings of systems:

- Foremost, we rely on the ground truth ranking based on the retrieval effectiveness measure over the entire topic set, which corresponds to aspect EA4 of Figure 1.1. In most instances, this is the ranking of systems according to MAP. Estimating this ranking correctly is the ultimate goal of system ranking estimation approaches. It is utilized in most experiments, the only exception being the experiments in Section 5.5.2.
- In Section 5.5.2, we are interested in how well the ranking of systems can be estimated for each individual topic. Thus, the ground truth ranking is based on the retrieval effectiveness measure of a single topic, which corresponds to aspect EA3 of Figure 1.1. In the experiments of that section, for all but two data sets, the ground truth is the ranking of systems according to average precision. This ranking may or may not coincide with the system ranking based on the retrieval effectiveness measure over the full topic set.

Since we are interested in the ranking of retrieval systems, the evaluation is performed by reporting the rank correlation coefficient Kendall's $\tau$.

### 5.4.1 Data Sets

As previous work, we also rely on TREC adhoc tasks over different years in our experiments. However, whereas earlier studies focused mainly on TREC-{3,5,6,7,8}, we investigate a wider variety of data sets, that include more recent adhoc task data sets, a range of non-adhoc task data sets, as well as adhoc tasks on non-traditional corpora. In the previous chapters, we restricted our experiments to three corpora, namely TREC Vol. 4+5, WT10g and GOV2 and their corresponding adhoc retrieval tasks. The main reasons are the availability of the corpora and the importance of the adhoc retrieval task. In the context of system ranking estimation, however, corpus information or training data is not always required. This is the case for the $RS$ approach for instance, which relies exclusively on the document identifiers of the top retrieved documents to determine a ranking of retrieval systems. Such a document-content independent approach makes it possible to include a larger number of data sets. In the experiments in this chapter, we take advantage of this fact and evaluate a cross-section of different tasks that have been introduced to TREC over the years. All data sets, that is all the runs submitted by the groups participating in TREC, can be downloaded from the TREC website.

In particular, all experiments are performed on the following TREC data sets:

- **TREC-{6,7,8}**: adhoc retrieval tasks on TREC Vol. 4+5 [148],
- **TREC-{9,10}**: adhoc retrieval tasks on WT10g [132],
- **TB-{04,05,06}**: adhoc retrieval tasks on the TeraByte corpus GOV2 [38],
- **CLIR-01**: the Cross Language track of 2001 [60] which aims at retrieving Arabic documents from a mix of English, French and Arabic topics,
- **NP-02**: the Named Page finding task of 2002 [43] where the task is to find a particular page in a corpus of Web documents,
- **EDISC-05**: the Enterprise Discussion search task of 2005 [42] which relies on a test corpus of e-mails and aims to retrieve e-mails that discuss positive and negative aspects of a topic,
- **EEXP-05**: the Enterprise Expert search task of 2005 [42] which focuses on finding people who are experts on a topic area,
- **BLTR-06**: the Blog track [115], introduced in 2006 to TREC with its topic relevance task as an adhoc-style task on a corpus of blog entries,
- **GEN-07**: the Genomics track of 2007 [75] which focuses on entity based question answering tasks on a collection of biomedical journal articles[2],
- **LEGAL-07**: the Legal track of 2007 [141], a recall-oriented track which centers around searching documents in regulatory and litigation settings, and,
- **RELFB-08**: the Relevance Feedback track [26] which intends to study the effects of relevance feedback when different amounts of true relevance feedback is available.

[2]Although the Genomics task itself calls for passage retrieval, the submitted runs were also evaluated on the document level. The latter interpretation is the one we rely on in our evaluation.

As in Chapter 4 we use a different notation and terminology from Chapters 2 and 3 (data set instead of topic set and TREC-6 instead of 301-350 for instance) to distinguish the current experiments on system ranking estimation from the earlier experiments on query effectiveness prediction.

In all but two data sets, the retrieval effectiveness of a system is measured in MAP. In the NP-02 data set, where the ranking of one particular document is of importance, mean reciprocal rank is the evaluation measure of choice, while in the RELFB-08 data set, the effectiveness measure is statistical MAP [8].

The number of retrieval systems to rank varies between a minimum of $37$ (EEXP-05) and a maximum of $129$ (TREC-8). The number of topics in the topic set ranges from $25$ topics for CLIR-01 to $208$ topics for RELFB-08. A comprehensive overview of the number of topics and systems for each data set can be found in Table 5.1. We include all available runs in our experiments, automatic as well as manual and short as well as long runs. This also includes runs, that are not part of the official TREC runs, but are nevertheless available from the TREC website. In the setting of TREC, a run is labelled automatic, if no human intervention was involved in its creation, otherwise it is considered to be manual. The amount of human intervention in manual runs varies, it can range from providing explicit relevance judgments and manually re-ranking documents to small changes in the topic statement to make it accessible for a specific retrieval system. Runs are also categorized as short or long according to the TREC topic part they employ, either the title, description or narrative.

Only one of the approaches we evaluate, namely, *ACSimScore*, is based on document content. As in earlier chapters, we preprocessed the corpora by applying Krovetz stemming [90] and stopword removal.

### 5.4.2 Algorithms

The following sections introduce six system ranking estimation approaches, four of which - $DF$, $RS$, $SO$ and *ACScore* - will be investigated throughout this chapter. The document-content based approach *ACSimScore* is only used for comparison in the first set of experiments, as we only have the corpora available for TREC-{6-10} and TB-{04-06}. The system similarity approach by Aslam and Savell [9] is also briefly described, but excluded from further analysis as it is very similar in spirit to $RS$. It also relies on document overlap without consideration the rank or retrieval score of a document, while at the same time being less effective than $RS$.

#### Data Fusion ($DF$)

We implemented the variation of the data fusion approach, that performed best in [114], namely Condorcet voting and biased system selection. In this approach, a number of parameters need to be set, specifically, (i) the percentage $P\%$ of systems to select in a biased way that favors non-average systems for data fusion, (ii) the number $b$ of top retrieved documents to select from each selected system and (iii) the percentage $s\%$ of documents to use as pseudo relevance judgments from the

merged list of results. We evaluated a range of values for each parameter:

$$\begin{aligned} s &= \{1\%, 5\%, 10\%, 20\%, 30\%, 40\%, 50\%\} \\ b &= \{10, 20, 30, ..., 100, 125, 150, 175, 200, 250\} \\ P &= \{10\%, 20\%, 30\%, ..., 100\%\}. \end{aligned}$$

Each of the $1050$ possible parameter combinations was tested. To determine the best parameter settings for each data set, we trained on the remaining data sets available for that corpus, that means for instance, that the parameters of the TREC-6 data set were those that led to the best ranking estimation performance for data sets TREC-7 and TREC-8. Depending on the training data sets, widely different parameter combinations were learned, for instance TREC-9 is evaluated on $s = 1\%$, $b = 250$ and $P = 50\%$ while TB-05 is run with $s = 50\%$, $b = 60$ and $P = 80\%$.

Data sets for training are only available for TREC-{6-10} and TB-{04-06} though. For the remaining data sets, we evaluated all combinations of parameter settings that were learned for TREC-{6-10} and TB-{04-06}. We evaluated each setting on the eight data sets without training data (CLIR-01 to RELFB-08) and chose the setting that across these data sets gave the best performance: $s = 10\%$, $b = 50$ and $P = 100\%$. Thus, the best results are achieved when *not* biasing the selection of systems ($P = 100\%$) towards non-average systems. Since in effect, we optimized the parameters on the test sets, we expect $DF$ to perform very well on those data sets.

### Random Sampling ($RS$)

We follow the methodology from [133] and rely on the $100$ top retrieved documents per retrieval system. We pool the results of *all* systems that are to be ranked, not just the official TREC runs. The percentage of documents to sample from the pool is sampled from a normal distribution with a mean according to the mean percentage of relevant documents in the relevance judgments and a standard deviation corresponding to the deviation between the different topics. Note, that this requires some knowledge about the distribution of relevance judgments; this proved not to be problematic however, as fixing the percentage to a small value ($5\%$ of the number of unique documents in the pool) actually yielded little variation in the results. As in [133], due to the inherent randomness of the process, we perform $50$ trials. In the end, we average the pseudo average precision values for each pair $(t_i, s_j)$ of topic and system and rank the systems according to pseudo mean average precision.

### System Similarity

A simplification of the $RS$ process was proposed by Aslam and Savell [9], who observed that retaining duplicate documents in the pool leads to a system ranking estimate that is geared towards document popularity. That is, systems that retrieve many popular documents, are assigned top ranks. Put differently, the more similar a system is to all other systems, the more popular documents it retrieves and thus the better its rank is estimated to be.

The similarity between two systems $s_i$ and $s_j$ is determined by the document overlap of their respective ranked lists of documents $R_i$ and $R_j$, expressed by the Jaccard similarity coefficient:

$$\textit{SysSimilarity}(s_i, s_j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|}.$$

The estimated effectiveness score of a system $s_o$ is then the average over all pairwise similarities:

$$\textit{Avg}(s_o) = \frac{1}{n-1} \sum_{s_i \neq s_o} \textit{SysSimilarity}(s_o, s_i).$$

A system's estimated score decreases with decreasing similarity towards the average system.

### Structure of Overlap ($SO$)

Recall, that the structure of overlap approach [135], in contrast to the previously introduced approaches, does not rank all systems at once, but instead repeatedly ranks random sets of five systems. Let there be $n$ systems to be ranked. A total of $n$ random groupings of five systems each are then created such that each system appears in exactly five groupings. Subsequently, for each grouping and for each of the topics, the percentage $\%$*Single* of documents in the ranked lists of the top retrieved 50 documents found by only one and the percentage $\%$*AllFive* of documents found by all five systems is determined. The three scores of $\%$*Single*, $\%$*AllFive* and the difference ($\%$*Single* $-$ $\%$*AllFive*) were proposed as estimated system score. These scores are further averaged across all topics. Since each system participates in five groupings, the scores across those groupings are again averaged, which leads to the final system score.

### Autocorrelation based on Document Scores (*ACScore*)

This approach, proposed by Diaz [50] and denoted by $\rho(\mathbf{y}, \mathbf{y}_\mu)$ in his work, is based on document overlap and the particular retrieval scores a system assigns to each document. Essentially, a retrieval system is estimated to perform well if its scores are close to the average scores across all systems. First of all, the document scores are normalized in order to make them comparable across systems [110]. Then, the average system vector of scores $\mathbf{y}_\mu$ is determined as follows: a set $\mathcal{U}$ of the top $75$ retrieved documents of all systems is formed and the average score for each element in the set is calculated. Thus the length of vector $\mathbf{y}_\mu$ is $m = |\mathcal{U}|$. The linear correlation coefficient $r$ between $\mathbf{y}_\mu$ and the vector $\mathbf{y}$ of scores of the top $m$ retrieved documents per system is then the indicator of a system's estimated quality where $r$ is high when both vectors are very similar to each other.

**Autocorrelation based on Document Similarity and Scores (*ACSimScore*)**

We also evaluate a second approach by Diaz [50] which combines the *ACSim* approach (introduced in Chapter 3, Section 3.2.2) with *ACScore*. This method, originally referred to as $\rho(\tilde{\mathbf{y}}, \mathbf{y}_\mu)$, is based on the notion that well performing systems are likely to fulfill the cluster hypothesis, while poorly performing systems are not. Based on a document's score vector $\mathbf{y}$, a perturbed score vector $\tilde{\mathbf{y}}$ is derived, which is based on the similarity between the ranked documents of a system. Each element $y_i$ is replaced by the weighted average of scores of the $5$ most similar documents (based on TF.IDF) in the ranked list. If the cluster hypothesis is fullfilled, we expect that the most similar documents will also receive a similar score from the retrieval system ($y_i$ and $\tilde{y}_i$ will be similar), while in the opposite case, high document similarity is not expressed in similar scores and $\tilde{y}_i$ will be different from $y_i$. To score each system, the linear correlation coefficient between $\mathbf{y}_\mu$ and the average system vector of scores, $\tilde{\mathbf{y}}$, is determined.

## 5.5 Experiments

In Section 5.5.1, we compare the introduced ranking estimation approaches across the different data sets. Then, in Section 5.5.2, we will show that the system ranking cannot be estimated equally well for each topic. In Section 5.5.3, we perform a number of motivational experiments to determine whether it is possible to exploit this observation. Finally, in Section 5.5.4, we make a first attempt at automatically selecting a good subset of topics from the full topic set.

### 5.5.1 System Ranking Estimation on the Full Set of Topics

In this section, we replicate the experiments reported in [114, 133]. In contrast to [135], we apply $SO$ to rank all available systems per data set. Additionally, we investigate *ACSimScore* and *ACScore* for their ability to rank retrieval systems, instead of ranking systems for single topics as reported in [50]. The results of our experiments are shown in Table 5.1. The highest correlation achieved for each data set is given in bold; the correlations that are not significantly different from the best one are underlined. All correlations reported are significantly different from zero with a p-value $< 0.005$.

When comparing the correlations in Table 5.1 with those reported in earlier chapters for the task of predicting the query effectiveness for a particular retrieval system (Figure 1.1, EA2) , it is evident that the task of estimating a ranking of retrieval systems is less difficult to achieve. The weakest approach, *ACSimScore* estimates the ranking of systems with a correlation between $\tau = 0.42$ and $\tau = 0.65$. Noteworthy are the high correlations the five estimators achieve on the TREC-{9,10} data sets in comparison to the TREC-{6,7,8} data sets. For the task of query effectiveness estimation the opposite observation was made in previous chapters: the performance of queries of TREC Vol. 4+5 can be predicted very well, while

| | #sys | #top | Kendall's Tau | | | | |
|---|---|---|---|---|---|---|---|
| | | | DF | ACSimScore | ACScore | SO | RS |
| **TREC-6** | 73 | 50 | **0.600** | 0.425 | 0.429 | 0.470 | 0.443 |
| **TREC-7** | 103 | 50 | **0.486** | 0.417 | 0.421 | 0.463 | 0.466 |
| **TREC-8** | 129 | 50 | 0.395 | 0.467 | 0.438 | 0.532 | **0.538** |
| **TREC-9** | 105 | 50 | 0.527 | 0.639 | 0.655 | 0.634 | **0.677** |
| **TREC-10** | 97 | 50 | 0.621 | 0.649 | **0.663** | 0.598 | 0.643 |
| **TB-04** | 70 | 50 | 0.584 | 0.647 | 0.687 | 0.614 | **0.708** |
| **TB-05** | 58 | 50 | 0.606 | 0.574 | 0.547 | 0.604 | **0.659** |
| **TB-06** | 80 | 50 | 0.513 | 0.458 | **0.528** | 0.447 | 0.518 |
| **CLIR-01** | 47 | 25 | 0.697 | - | 0.700 | 0.650 | **0.702** |
| **NP-02** | 70 | 150 | 0.667 | - | **0.696** | 0.668 | 0.693 |
| **EDISC-05** | 57 | 59 | **0.668** | - | 0.560 | 0.614 | 0.666 |
| **EEXP-05** | 37 | 50 | 0.589 | - | **0.682** | 0.502 | 0.483 |
| **BLTR-06** | 56 | 50 | 0.482 | - | 0.485 | 0.357 | **0.523** |
| **GEN-07** | 66 | 36 | **0.578** | - | 0.500 | 0.362 | 0.563 |
| **LEGAL-07** | 68 | 43 | **0.754** | - | 0.680 | 0.749 | 0.741 |
| **RELFB-08** | 117 | 208 | 0.537 | - | **0.599** | 0.544 | 0.559 |

Table 5.1: System ranking estimation on the full set of topics. Reported is Kendall's $\tau$. All correlations reported are significant ($p < 0.005$). The highest correlation per topic set is bold. The correlations that are not statistically different from the best one are underlined. Column **#sys** shows the number of systems to rank, **#top** shows the number of topics in a data set.

for the queries of WT10g predicting the effectiveness is difficult. As will be shown later in this chapter, system ranking estimation is more difficult on TREC-{6,7,8} due to the greater amount of human intervention in the best runs. Manual runs can be very different from automatic runs, containing many unique documents that are not retrieved by other systems. This is a problem as system ranking estimation is to some extent always based on document popularity. A simple solution would be to prefer runs, that retrieve many unique documents, however this is not possible since the worst performing runs also retrieve a lot of documents that are not retrieved by any other run.

$DF$ outperforms $RS$ on TREC-{6,7} as already reported in [114]. The poor result on TREC-8 is due to an extreme parameter setting found to perform best on TREC-{6,7}, which was subsequently used to evaluate $DF$ on TREC-8. On the remaining data sets were $DF$'s parameters were trained (TREC-{9,10} and TB-{04,05,06}), $RS$ outperforms $DF$, in two instances significantly. The highly collection dependent behavior of $DF$ is due to the method's inherent bias in the way in which the subset of systems to select the pseudo relevant documents from are determined. A system that is dissimilar to the average system, can either perform very well or very poorly. On the data sets without training data (CLIR-01 to RELFB-08), $DF$ performs similarly to $RS$, which is not surprising as the best performing parameter setting of $DF$ means that the most popular documents are included in

the pseudo relevant documents, just as for $RS$. The only exception is data set EEXP-05, where $DF$ achieves a correlation of $\tau = 0.59$, while $RS$ achieves a correlation of $\tau = 0.48$. This variation can be explained by the small number of systems to rank (37) - here a small difference in system rankings has a considerable effect on Kendall's $\tau$.

Relying on TF.IDF based content similarity does not help, shown by *ACSimScore*'s performance. In four out of eight evaluated data sets, its performance is significantly worse than the best performing approach. Although the approach ranks systems higher that are closer to the average (just like $RS$), the TF.IDF similarity might not be reflected in the score similarity as is expected in this approach. In particular more advanced retrieval systems are likely to include more than basic term statistics such as evidence from external corpora, anchor text, the hyperlink structure, etcetera, all of which influences the retrieval scores a system assigns to a document.

The $SO$ approach performs similarly to $RS$ on TREC-{6,7,8,9}, while for the remaining data sets the differences in performance generally become larger. We suspect that the ranking of five systems is less stable, than the ranking of all systems at once.

*ACScore*, which takes the score a retrieval system assigns to a document into account, is also well performing, for five data sets it achieves the highest correlation. A potential disadvantage of *ACScore* though is that it requires knowledge of the retrieval scores a retrieval system assigns to each document, while $DF$, $SO$ and $RS$ require no such knowledge.

In Table 5.1 we have refrained from reporting a mean correlation across all data sets for each estimator on purpose, due to the different data set sizes, that is the number of retrieval systems to rank. Instead, we point out, that $RS$ exhibits the highest correlation for six data sets, while $DF$ and *ACScore* record the highest correlation on five data sets each. Additionally, $RS$'s performance is significantly worse than the best performing approach in only three instances, $DF$ is significantly worse in four and *ACScore* is significantly worse in six data sets. Taking into account, that $DF$'s parameters were optimized on eight of the sixteen data sets, we conclude that in contrast to earlier work which was performed on a small number of TREC data sets [9, 114, 135, 161], when evaluating a broader set of data sets, the random sampling approach $RS$ is the most consistent and overall the best performing method.

### Rank Estimate of the Best Performing System

As discussed in Section 5.2, the commonly cited problem of automatic system evaluation is the mis-ranking of the best systems. As in previous work evaluations have mostly been carried out on TREC-{3,5,6,7,8}, where the problem of underestimating the best systems occurs consistently, it has been assumed to be a general issue. When considering more recent and diverse data sets, we find this problem to be dependent on the set of systems to rank.To give an impression of the accuracy of the rankings, in Figure 5.1 scatter plots of the estimated system ranks versus the ground truth system ranks are shown for a number of data sets and system ranking

estimation approaches along with the achieved correlation and the estimated rank (ER) of the best system. Each data point stands for one of the $n$ systems and the best system is assigned rank $1$ in the ground truth ranking. In the ideal case, when the ranks of all systems are estimated correctly and therefore $\tau = 1.0$, the points would lie on a straight line from $(1, 1)$ to $(n, n)$.

| | Best Run | M/A | #sys | Estimated Rank DF | ACScore | SO | RS |
|---|---|---|---|---|---|---|---|
| **TREC-6** | *uwmt6a0* [41] | M | 73 | **52** | 55 | 56 | 57 |
| **TREC-7** | *CLARIT98COMB* [56] | M | 103 | **48** | 70 | 78 | 74 |
| **TREC-8** | *READWARE2* [1] | M | 129 | 112 | 117 | **104** | 113 |
| **TREC-9** | *iit00m* [36] | M | 105 | 83 | 79 | **76** | **76** |
| **TREC-10** | *iit01m* [2] | M | 97 | **80** | 84 | 87 | 83 |
| **TB-04** | *uogTBQEL* [120] | A | 70 | **23** | 26 | 30 | 30 |
| **TB-05** | *indri05AdmfL* [106] | A | 58 | 35 | 45 | **30** | 32 |
| **TB-06** | *indri06AtdnD* [105] | A | 80 | 12 | **5** | 28 | 20 |
| **CLIR-01** | *BBN10XLB* [164] | A | 47 | 3 | **2** | 6 | **2** |
| **NP-02** | *thunp3* [173] | A | 70 | 18 | **16** | 20 | 17 |
| **EDISC-05** | *TITLETRANS* [101] | A | 57 | **1** | 2 | 2 | **1** |
| **EEXP-05** | *THUENT0505* [59] | A | 37 | 8 | **3** | 12 | 10 |
| **BLTR-06** | *wxoqf2* [165] | A | 56 | 5 | **4** | 13 | 5 |
| **GEN-07** | *NLMinter* [48] | M | 66 | **1** | **1** | 2 | **1** |
| **LEGAL-07** | *otL07frw* [140] | M | 68 | **4** | 15 | 8 | **4** |
| **RELFB-08** | *Brown.E1* [98] | M | 117 | 64 | **61** | **61** | 65 |

Table 5.2: Estimated rank of the system that performs best according to the ground truth. The evaluation metric of the ground truth is mean reciprocal rank (NP-02), statistical MAP [8] (RELFB-08) and MAP (all other data sets) respectively. **M/A** indicates if the best system is manual (M) or automatic (A) in nature. **#sys** shows the number of systems (or runs) to rank. The last four columns depict the estimated rank of the best system. Rank 1 is the top rank.

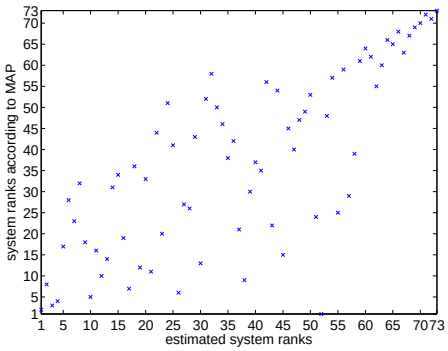
The scatter plots in Figure 5.1a, 5.1b and 5.1c reveal the extent of mis-ranking the best systems of data sets TREC-{6,8,10} respectively. In case of TREC-6, only the best system is severely mis-ranked, while in data set TREC-8 the best ten systems are estimated to perform poorly with estimated ranks of 70 or worse, in fact, the best system is estimated to be ranked at position 113 out of 129. In the TREC-10 data set, the two best performing systems are ranked together with the worst performing systems, while the other well performing systems are ranked towards the top of the ranking. When we consider the TB-04 data set (Figure 5.1d), a decrease in the amount of mis-ranking of the best systems is evident. The best correspondence between estimated and ground truth based ranking can be found in Figure 5.1g where the results of data set LEGAL-07 are shown. The correlation of $\tau = 0.75$ indicates the quality of the estimated ranking, and the best system has an estimated rank of four. Better in terms of the best systems performs only $DF$ on EDISC-05 (Figure 5.1e), where the two best performing systems are estimated correctly at

ranks one and two.

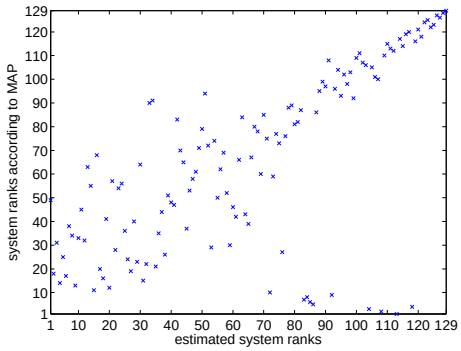
An overview of the estimated rank of the best system across all data sets is given in Table 5.2, where the best rank estimate of a data set is indicated in bold. For comparison purposes, we also list the number of systems to rank once more. Note, that we exclude the approach *ACSimScore* from further experiments, as it is not available for all data sets and moreover has shown to be the weakest performing method on the evaluated data sets. We observe that independent of the system ranking estimation approach, the problem of underestimating the ranking of the best system decreases considerably for the data sets TB-{04-06} in comparison to TREC-{6-10}. With the exception of RELFB-08, the ranks of the best systems are estimated to a much greater accuracy, in fact five for data sets (CLIR-01, EDISC-05, BLTR-06, GEN-07, LEGAL-07) $DF$ and $RS$ estimate the best system within the top five ranks. When we investigated this discrepancy in estimating the rank of the best system between the different data sets, it became apparent, that the reason for this behavior lies in the makeup of the best run. Table 5.2 also lists the best system of each data set according to the ground truth and an indicator if the best system is manual or automatic in nature. For data sets TREC-{6-10} in all cases, the top performing run according to the MAP based ground truth ranking is manual. The amount of manual intervention in each run is significant:

- In TREC-6, the run *uwmt6a0* [41] was created by letting four human assessors spent a total of 105 hours (2.1 hours on average per topic) on the creation of queries and the judging of documents, which lead to $13064$ judgments being made.
- In TREC-7, the run *CLARIT98COMB* [56] was created by having each topic judged by four different assessors. The judged documents were then included as relevance feedback in the final result run, with additional resorting to move the documents manually labeled as relevant to the top of the ranking.
- In TREC-8, the run *READWARE2* [1] was created by letting a retrieval system expert create numerous queries for each TREC topic by considering the top retrieved documents and reformulating the queries accordingly. On average, 12 queries were created per TREC topic.
- In TREC-9, the run *iit00m* [36] was created by letting an expert derive a query with constraints for each TREC topic.
- In TREC-10, the run *iit01m* [2] was created with the aid of manual relevance feedback: the top ranked documents were assessed and reranking was performed accordingly.

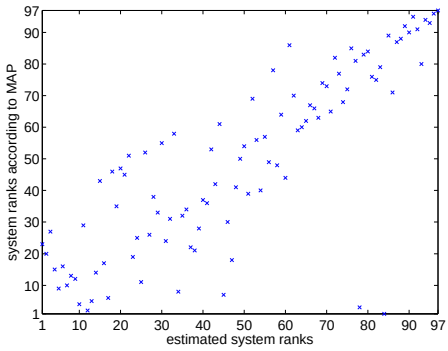
The runs created this way are very different from automatic runs (which one way or another are dependent on the collection frequencies of the query terms) and are bound to have a small amount of overlap in the top retrieved documents in comparison to the automatic runs. This explains why system ranking estimation approaches uniformly estimate them to be among the worst performing runs. In contrast, the estimated ranks of the best systems of GEN-07 and LEGAL-07, which are also classified as manual, are highly accurate. This is explained by the fact,



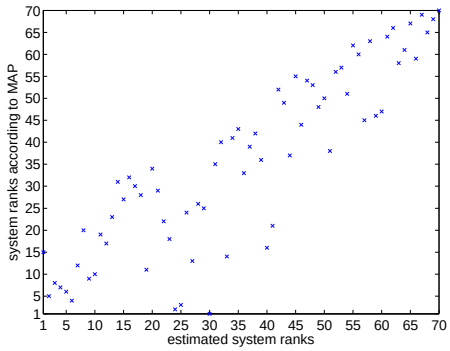

(a) TREC-6, $DF$, $\tau = 0.60$, $ER = 52$



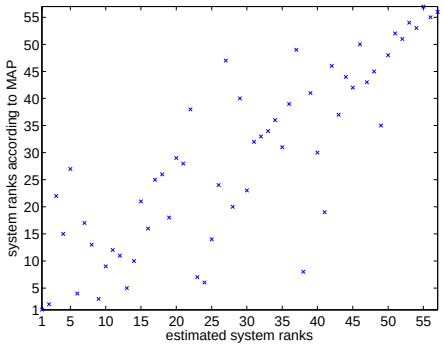

(b) TREC-8, $RS$, $\tau = 0.54$, $ER = 113$



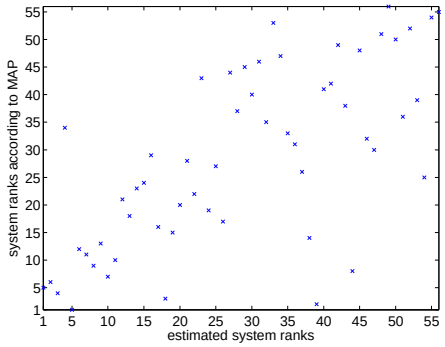

(c) TREC-10, *ACScore*, $\tau = 0.66$, $ER = 84$



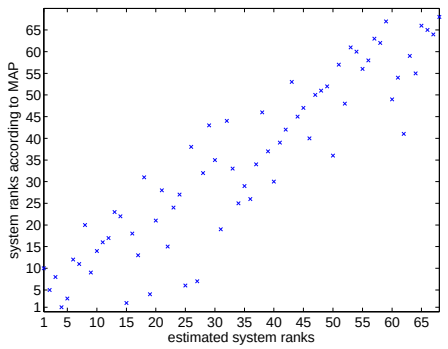

(d) TB-04, $RS$, $\tau = 0.71$, $ER = 30$



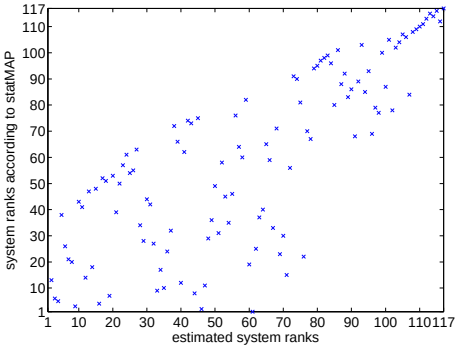

(e) EDISC-05, $DF$, $\tau = 0.67$, $ER = 1$




(f) BLTR-06, $RS$, $\tau = 0.52$, $ER = 5$




(g) LEGAL-07, $DF$, $\tau = 0.75$, $ER = 4$




(h) RELFB-08, $ACScore$, $\tau = 0.60$, $ER = 61$

Figure 5.1: Scatter plots of system ranks according to system ranking estimation approaches (x-axis) versus system ranks according to the ground truth (y-axis). Each marker stands for one retrieval system. Rank 1 is assigned to the best system.

that in both instances, the runs were created with little human intervention. The best run of GEN-07, *NLMinter* [48], is tagged as manual, because the Genomics topics were manually transcribed into queries for a domain specific external search engine and the documents retrieved from this engine were used for collection enrichment. The best run of LEGAL-07, *otL07frw* [140], is even less manual. Here, the provided TREC topics were manually transformed into queries suitable for the search engine without adding any additional knowledge by the human query transcriber. Finally, we note that the poor estimation performance on RELFB-08 and its best run, *Brown.E1* [98], is a result of the task [26], where the performance of pseudo-relevance feedback algorithms was investigated, by providing four different sets of relevance judgments, the smallest set with one relevant document per topic, the largest set with between 40-800 judged documents per topic. We hypothesize that based on the very different type of relevance information between the runs, document overlap might not be a good indicator.

A comparison of the performances of $DF$ on data sets TREC-6 and GEN-07 reveals, that reporting both the correlation coefficient $\tau$ and the estimated rank of the best performing system offers better insights into the abilities of a system ranking estimation method. For both data sets, $DF$ performs similarly with respect to the rank correlation, $\tau = 0.60$ (TREC-6) and $\tau = 0.58$ (GEN-07) respectively. However, the performances with respect to the estimated rank of the best system are very different, while the estimate of TREC-6 is highly inaccurate ($ER = 52$ out of $73$ systems), the best system of GEN-07 is identified correctly ($ER = 1$ out of $66$ systems).

Considering the success of estimating the rank of the best system across the four system ranking estimation approaches, we note that $DF$ and *ACScore* outperform $SO$ and $RS$ by providing the best estimates for seven data sets, while $RS$ and $SO$ provide the best estimates on five and four data sets respectively. In most instances, the estimated ranks are similar across all approaches, exceptions are data sets TREC-7 and TB-06 where the maximum difference in $ER$ is $30$ and $23$ respectively. Although on TREC-7 all four ranking estimation approaches result in similar correlations (between $\tau = 0.42$ and $\tau = 0.47$), $DF$'s estimate of the best system is considerably better than of the remaining approaches. This reiterates the previous point, that both $\tau$ and $ER$ should be reported to provide a more comprehensive view of an algorithm's performance.

### 5.5.2 Topic Dependent Ranking Performance

In this section, we show that the ability of system ranking estimation approaches to rank retrieval systems correctly differs significantly between the topics of a topic set. While for a number of topics the estimated rankings are highly accurate and close to the actually observed rankings, for other topics system ranking estimation fails entirely.

We set up the following experiment: for each topic, we evaluated the estimated ranking of systems (Figure 1.1, EA3) by correlating it against the ground truth ranking that is based on average precision, reciprocal rank or statistical average preci-

sion. This is different from the ground truth ranking based on MAP, mean reciprocal rank or statMAP. Here, we are *not* interested in how well a single topic can be used to approximate the ranking of systems over the entire topic set. Instead, we are interested in how well a system ranking estimation method performs for each individual topic. To evaluate the range of performances, we record the topic for which the least correlation is achieved and the topic for which the highest correlation is achieved. The results are shown in Table 5.3.

| | **DF** | | **ACScore** | | **SO** | | **RS** | |
|---|---|---|---|---|---|---|---|---|
| | **min.** $\tau$ | **max.** $\tau$ | **min.** $\tau$ | **max.** $\tau$ | **min.** $\tau$ | **max.** $\tau$ | **min.** $\tau$ | **max.** $\tau$ |
| **TREC-6** | 0.008 | 0.849† | −0.161 | 0.812† | −0.147 | 0.752† | −0.105 | 0.814† |
| **TREC-7** | −0.061 | 0.765† | 0.053 | 0.695† | −0.008 | 0.764† | −0.004 | 0.693† |
| **TREC-8** | 0.053 | 0.792† | 0.087 | 0.740† | 0.080 | 0.723† | 0.143 | 0.731† |
| **TREC-9** | −0.234† | 0.835† | 0.018 | 0.760† | 0.096 | 0.760† | 0.179 | 0.730† |
| **TREC-10** | −0.094 | 0.688† | 0.031 | 0.722† | 0.054 | 0.707† | 0.130 | 0.821† |
| **TB-04** | 0.002 | 0.906† | −0.161 | 0.777† | −0.057 | 0.784† | −0.025 | 0.882† |
| **TB-05** | 0.040 | 0.769† | −0.161 | 0.716† | −0.052 | 0.709† | −0.083 | 0.827† |
| **TB-06** | −0.070 | 0.728† | 0.055 | 0.644† | −0.159 | 0.710† | −0.152 | 0.760† |
| **CLIR-01** | 0.268 | 0.862† | 0.378† | 0.837† | 0.220 | 0.876† | 0.248 | 0.862† |
| **NP-02** | −0.264 | 0.607† | −0.129 | 0.760† | −0.239 | 0.621† | −0.257 | 0.649† |
| **EDISC-05** | −0.019 | 0.573† | −0.038 | 0.589† | −0.021 | 0.526† | 0.024 | 0.640† |
| **EEXP-05** | −0.250 | 0.845† | −0.224 | 0.808† | −0.294 | 0.764† | −0.208 | 0.770† |
| **BLTR-06** | 0.044 | 0.534† | 0.018 | 0.507† | −0.192 | 0.436† | 0.206 | 0.562† |
| **GEN-07** | 0.151 | 0.795† | 0.040 | 0.700† | 0.078 | 0.627† | 0.180 | 0.774† |
| **LEGAL-07** | 0.027 | 0.690† | −0.004 | 0.583† | −0.058 | 0.691† | −0.008 | 0.690† |
| **RELFB-08** | −0.183† | 0.797† | −0.115 | 0.749† | −0.172 | 0.774† | −0.137 | 0.775† |

Table 5.3: Topic dependent ranking performance: minimum and maximum estimation ranking accuracy in terms of Kendall's $\tau$. Significant correlations ($p < 0.005$) are marked with †.

The results are very regular across all data sets and system ranking estimation methods: the spread in correlation between the best and worst case are extremely wide; in the worst case, there is no correlation ($\tau \approx 0$) between the ground truth and the estimated ranking or in rare cases a significant negative correlation is observed (such as for data sets TREC-9 and RELFB-08). In the best case on the other hand, the estimated rankings are highly accurate, and with few exceptions $\tau > 0.7$. Overall, $DF$ exhibits the highest correlation on TB-04, where the maximum achievable correlation is $\tau = 0.91$. Though not explicitly shown, we note that the topics for which the minimum and maximum $\tau$ are recorded vary between the different system ranking estimation approaches.

These findings form the main motivation for our work: if we were able to determine a subset of topics for which the system ranking estimation algorithms perform well, we hypothesize that this would enable us to achieve a higher estimation accuracy of the true ranking across the full set of topics.

### 5.5.3 How Good are Subsets of Topics for Ranking Systems?

Having shown in the previous section that the quality of ranking estimation varies across individual topics, we now turn to investigating whether selecting a subset of topics from the full topic set is useful in the context of system ranking estimation algorithms. That is, we attempt to determine whether we can improve the accuracy of the approaches over the results reported in Section 5.5.1 on the full set of topics. We can choose a subset of topics, for instance, by removing those topics from the full set of topics the system ranking approach performs most poorly on. To investigate this point, we experiment with selecting subsets of topics according to different strategies as well as evaluating a large number of randomly drawn subsets of topics.

Each of the evaluated topic sets consists of $m$ topics, $m$ varies from $25$ to $208$ (Table 5.1). We therefore test subsets of cardinality $c = \{1, 2, .., m\}$. In the ideal case, for each cardinality we would test all possible subsets. This is not feasible though, as for each cardinality $c$, a total of $\binom{m}{c}$ different subsets exist; for a topic set with $m = 50$ topics and subsets of cardinality $c = 6$ for instance this already amounts to nearly sixteen million subset combinations, that is $\binom{50}{6} = 15890700$. For this reason, for each $c$, we randomly sample $10000$ subsets of topics. Apart from this random strategy, we also include a number of iterative topic selection strategies, that will be described shortly.

For the topic subsets of each cardinality, we determine the correlation between the estimated ranking of systems (based on this subset) and the ground truth ranking of systems based on the retrieval effectiveness across the full set of topics. In contrast to Section 5.5.2, we are now indeed interested in how well a subset of one or more topics can be used to approximate the ranking of systems over the entire topic set.

In total, we report results for five subset selection strategies, two based on samples of subsets and three iterative ones:

- **worst sampled subset**: given the 10000 sampled subsets of a particular cardinality $c$, reported is the $\tau$ of the subset resulting in the lowest correlation,
- **average sampled subset**: given the 10000 sampled subsets of a particular cardinality $c$, reporeted is the average $\tau$ across all samples,
- **greedy approach**: an iterative strategy; at cardinality $c$, that topic, from the pool of unused topics, is added to the existing subset of $c-1$ topics, for which the new subset reaches the highest correlation with respect to the ground truth ranking based on MAP; this approach performs usually as well as or better than the best sampled subset, which is therefore not listed separately,
- **median AP**: an iterative strategy; at cardinality $c$ that topic is added to the existing subset of $c-1$ topics, that exhibits the highest median average precision across all systems; this means that first the easy topics (on which many systems achieve a high average precision) are added and then the difficult ones,
- **estimation accuracy**: an iterative strategy; at cardinality $c$ that topic is added to the existing subset of $c-1$ topics, that best estimates the ranking of systems

according to average precision for that topic; thus, first those topics are added to the subset that the system ranking estimation method achieves the highest estimation accuracy for (this strategy draws from results of Section 5.5.2).

We should stress here, that the latter three strategies (greedy, median AP and estimation accuracy) all require knowledge of the true relevance judgments. This experiment was set up to determine whether it is at all beneficial to rely on subsets instead of the full topic set. These strategies were not designed to find a subset of topics automatically. Therefore this section should be viewed as an indicator that subset selection is indeed a useful research subject that should be pursued further.

The results of this analysis are shown in Figure 5.2 for selected data sets. After a visual inspection it becomes immediately evident that the general trend of the results is similar across all examples. The greedy approach, especially at small subset sizes between $c = 5$ and $c = 15$, yields significantly higher correlations than the baseline, which is the correlation the method achieves at the full topic set size of $m$ topics. After a peak, the more topics are added to the topic set, the lower the correlation. The amount of change of $\tau$ is data set dependent, the largest change in Figure 5.2 is observed for TREC-9 and the $DF$ approach, where $\tau$ increases from the baseline correlation of $\tau = 0.53$ to $\tau = 0.80$ at the peak of the greedy approach. The worst subset strategy on the other hand shows the potential danger of choosing the wrong subset of topics: $\tau$ is significantly lower than the baseline for small cardinalities.

When averaging $\tau$ across all sampled subsets (the average subset strategy) of a cardinality, at subset sizes of about $m/3$ topics, the correlation is only slightly worse than the baseline correlation.

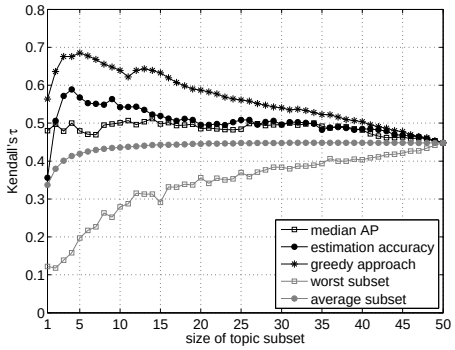
When considering the median AP strategy, which first adds easy topics (topics with a high median AP) to the subset of topics, the gains in correlation over the baseline are visible in Figures 5.2a and 5.2f but they are topic dependent and far less pronounced than the best possible improvement, as exemplified by the greedy approach.

Better than the median AP strategy is the performance of the estimation accuracy strategy, where first those topics are added to the topic subset, for whom the ranking of systems is estimated most accurately as measured by Kendall's $\tau$. This strategy is based on the results of Section 5.5.2. Particularly high correlations are achieved in Figures 5.2b, 5.2c, 5.2e and 5.2f. Here, the development of the correlation coefficient achieved by the estimation accuracy strategy across different cardinalities mirrors the development of the greedy subset approach. However, the improvements are also not consistent across all data sets. In the worst case as seen in Figure 5.2g, the correlations are not better than for the average subset approach. Overall though, the estimation accuracy strategy is also a subset selection method worth pursuing. It remains to be seen though, whether an automatic procedure can be devised that allows us to estimate the accuracy of the ranking to a high degree.

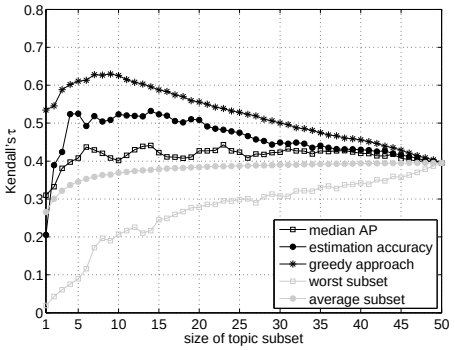
A summary of the most important results of $DF$, *ACScore*, $SO$ and $RS$ on all data sets are shown in Table 5.4. Listed are the the correlation coefficients on the full set of topics as well as the correlation of the best performing subset in the

| | DF | | | ACScore | | | SO | | | RS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **full set** | **greedy** | ±% | **full set** | **greedy** | ±% | **full set** | **greedy** | ±% | **full set** | **greedy** | ±% |
| | $\tau$ | $\tau$ | | $\tau$ | $\tau$ | | $\tau$ | $\tau$ | | $\tau$ | $\tau$ | |
| **TREC-6** | 0.600 | **0.804** | +34.0% | 0.429 | 0.723 | +68.5% | 0.470 | 0.731 | +55.5% | 0.443 | 0.654 | +47.6% |
| **TREC-7** | 0.486 | **0.762** | +56.8% | 0.421 | 0.591 | +40.4% | 0.463 | 0.633 | +36.7% | 0.466 | 0.584 | +25.3% |
| **TREC-8** | 0.395 | 0.630 | +59.5% | 0.438 | 0.606 | +38.4% | 0.532 | **0.661** | +24.2% | 0.538 | 0.648 | +20.4% |
| **TREC-9** | 0.527 | **0.800** | +51.8% | 0.655 | 0.780 | +19.1% | 0.634 | 0.775 | +22.2% | 0.677 | 0.779 | +15.1% |
| **TREC-10** | 0.621 | **0.761** | +22.5% | 0.663 | 0.755 | +13.9% | 0.598 | 0.711 | +18.9% | 0.643 | 0.734 | +14.2% |
| **TB-04** | 0.584 | **0.898** | +53.8% | 0.687 | 0.829 | +20.7% | 0.614 | 0.804 | +30.9% | 0.708 | 0.846 | +19.5% |
| **TB-05** | 0.606 | 0.800 | +32.0% | 0.547 | 0.743 | +35.8% | 0.604 | 0.786 | +30.1% | 0.659 | **0.812** | +23.2% |
| **TB-06** | 0.513 | 0.682 | +32.9% | 0.528 | **0.707** | +33.9% | 0.447 | 0.632 | +41.4% | 0.518 | 0.704 | +35.9% |
| **CLIR-01** | 0.697 | 0.785 | +12.6% | 0.700 | **0.815** | +16.4% | 0.650 | 0.771 | +18.5% | 0.702 | 0.808 | +15.0% |
| **NP-02** | 0.667 | 0.839 | +25.8% | 0.696 | **0.875** | +25.7% | 0.668 | 0.838 | +25.4% | 0.693 | 0.853 | +23.0% |
| **EDISC-05** | 0.668 | 0.776 | +16.2% | 0.560 | 0.703 | +25.5% | 0.614 | 0.773 | +25.9% | 0.666 | **0.801** | +20.3% |
| **EEXP-05** | 0.589 | **0.900** | +52.9% | 0.682 | 0.874 | +28.2% | 0.502 | 0.745 | +48.5% | 0.483 | 0.718 | +48.4% |
| **BLTR-06** | 0.482 | **0.617** | +28.0% | 0.485 | 0.603 | +24.4% | 0.357 | 0.538 | +51.0% | 0.523 | 0.601 | +14.9% |
| **GEN-07** | 0.578 | **0.685** | +18.5% | 0.500 | 0.672 | +34.5% | 0.362 | 0.569 | +57.2% | 0.563 | 0.680 | +20.9% |
| **LEGAL-07** | 0.754 | 0.864 | +14.6% | 0.680 | 0.808 | +18.9% | 0.749 | 0.874 | +16.6% | 0.741 | **0.865** | +16.7% |
| **RELFB-08** | 0.537 | 0.878 | +63.5% | 0.599 | **0.895** | +49.4% | 0.544 | 0.859 | +57.9% | 0.559 | 0.872 | +56.1% |

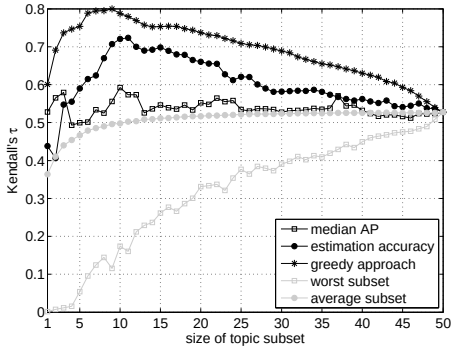
Table 5.4: Summary of topic subset selection experiments. In bold, the highest correlation coefficient of the greedy strategy per topic set. The columns marked with ± show the percentage of change between $\tau$ achieved on the full topic set and the greedy approach. All correlations reported are significant ($p < 0.005$). All differences between the best greedy $\tau$ and the $\tau$ of the full topic set are statistically significant.



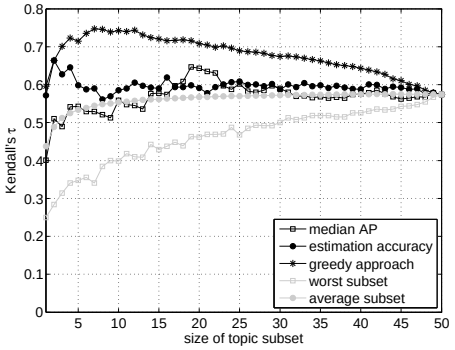

(a) TREC-6, $RS$



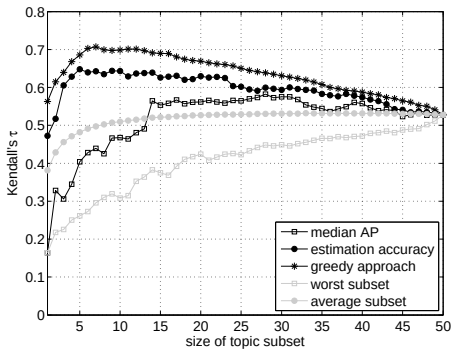

(b) TREC-8, $DF$



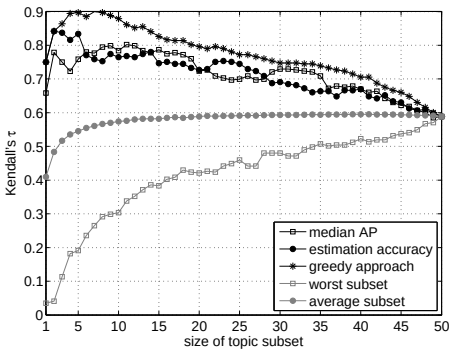

(c) TREC-9, $DF$



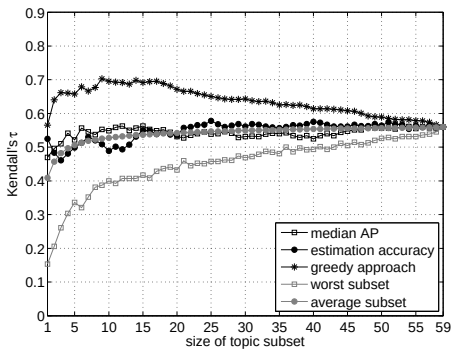

(d) TB-05, *ACSimScore*



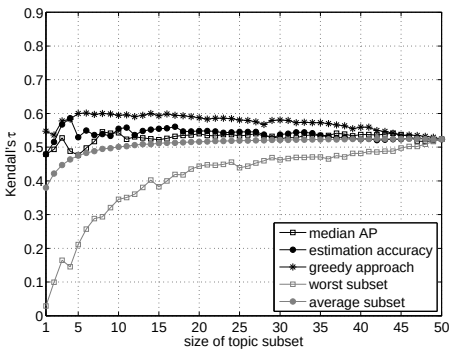

(e) TB-06, *ACScore*




(f) EEXP-05, $DF$




(g) EDISC-05, *ACScore*




(h) BLTR-06, $RS$

Figure 5.2: Topic subset selection experiments.

greedy approach, which is the maximum amount of improvement we assume possible. Though this is not entirely correct, the fact that the best sample among the random subsets does not perform better than the greedy approach suggests, that this is an adequate approximation of the true optimal performance. Across all pairings of system ranking estimation approach and topic set, subsets of topics indeed exist that would greatly improve the performance of system ranking estimation algorithms. Consider for instance, the results of $RS$ on RELFB-08: with the "right" topic subset, a rank correlation of $\tau = 0.87$ can be reached, a $56\%$ increase over the performance on the full topic set ($\tau = 0.56$). Given in bold, is the best possible correlation for each data set. Here, the $DF$ approach shows the most potential, in particular for eight out of sixteen data sets it records the highest possible improvement.

### 5.5.4 Automatic Topic Subset Selection

The observations made in the previous two sections can only be useful in practice if it becomes possible to automatically identify those subsets of topics that lead to improved system ranking estimation performance. In this section, we make a first step in that direction.

As $RS$ proved overall to be the best performing algorithm in Section 5.5.1, we focus on it now. Recall, that $RS$ is based on document popularity, that is, the most often retrieved documents have the highest chance of being sampled from the pool and thus being declared pseudo-relevant. This approach therefore assumes that *popularity* $\approx$ *relevance*. It is clear, that this assumption is not realistic, but we can imagine cases of topics where it holds: in the case of *easy* topics. Easy topics are those where all or most systems do reasonably well, that is, they retrieve the truly relevant document towards the top of the ranking and then relevance can be approximated by popularity.

The above observation leads to the basic strategy we employ: adding topics to the set of topics according to their estimated difficulty. Again, as we do not have access to relevance judgments, we have to rely on an estimate of *collection topic hardness* [7] (see Figure 1.1, EA1), as provided by the Jensen-Shannon Divergence ($JSD$) approach by Aslam and Pavlu [7]. The $JSD$ approach estimates a topic's difficulty with respect to the collection and in the process also relies on different retrieval systems: the more diverse the result lists of different retrieval systems as measured by the Jensen-Shannon Divergence, the more difficult the topic is with respect to the collection.

Therefore, we perform a prediction task on two levels. First, a ranking of topics according to their inherent difficulty is estimated by the $JSD$ approach and then we rely on the topics that have been predicted to be the easiest, to perform system ranking estimation. The only parameter of the $JSD$ approach is the document cutoff. We relied on the parameter settings recommended in [7], that is a cutoff of $100$ documents for TB-$\{$04-06$\}$ and a cutoff of $20$ documents for the remaining data sets.

The results of this two-level prediction approach are presented in Table 5.5.

Shown are Kendall's $\tau$ on the full set of topics and the correlation achieved by $JSD$ based selected subsets of $c = 10$ and $c = 20$ of topics. The particular size of the topic subset is of no great importance as seen in the small variation in $\tau$. For nine out of sixteen data sets, we can observe improvements in correlation, though none of the improvements are statistically significant. The largest improvement in correlation is observed for EEXP-05, where the correlation on the full topic set $\tau = 0.48$ increases to $\tau = 0.62$ when the easiest $c = 10$ topics are evaluated. The correlation change of the data sets that degrade with $JSD$ based topic subset selection is usually slight, the most poorly performing data set is NP-02, where $\tau = 0.69$ on the full set of topics degrades to $\tau = 0.60$. Considering the potential amount of improvements with the "right" subsets of topics as evident in Section 5.5.3 and Table 5.4, this result is somewhat disappointing. We suspect two reasons for the low levels of change the $JSD$ approach achieves: apart from the fact, that the median AP strategy does in a few instances only perform little better than the baseline correlation (Section 5.5.3), the $JSD$ approach itself does not estimate the topic's difficulty to a very high degree. When evaluating the accuracy of the collection topic hardness results (Figure 1.1, EA1) , $JSD$ reaches correlations between $\tau = 0.41$ and $0.63$, depending on the data set.

We also evaluated three further automatic topic subset selection mechanisms. First, we attempted to exploit general knowledge we have about the performance of a number of retrieval approaches such as the fact that TF.IDF usually performs worse than BM25 or Language Modeling. In order to rank $n$ systems, we assumed to have an additional $k << n$ systems available for which we know the performance ranking based on past experience. The ranking of the $n+k$ systems is then estimated as usual,and, topic subset selection is performed by first selecting those topics, for which the $k$ systems are ranked according to our assumption. The hypothesis being, that topics for which the system ranking estimator is able to derive the ranking of the known $k$ system correctly, are more likely to also produce good estimates for the unknown $n$ systems.

A second strategy is to cluster the estimated rankings derived for each topic and then to choose all topics of the largest cluster as topic subset. The motivation behind this approach can be explained by the results of Figure 5.3. We took the $10000$ random subsets created for topic subsets of cardinality $c = 10$ in the TREC-7 data set and the $RS$ approach. We sorted the subset samples according to the correlation they achieve with respect to the ground truth ranking, that is the MAP on the full set of $m = 50$ topics. Then, we created two sets: the set of *good* subsets, that are the $250$ subsets with the highest correlation and the set of *bad* subsets, that are the $250$ subsets with the lowest correlation with the ground truth. Now, for each of the $50$ topics in the full topic set, we determined how often it appears in the *good* and *bad* sets. The ratio $|G|/(|G|+|B|)$ is $1$ if a topic only appears in good sets, while it is $0$ if a topic only appears in bad sets, a value of $0.5$ means that the topic appears to the same extent in both types. In the plot in Figure 5.3 each point represents one topic. Two variations are given: the topics and their correlation to the MAP based ground truth and the topics and their correlation to the AP based ground truth. It is evident, that the best subsets are made out of topics which achieve a

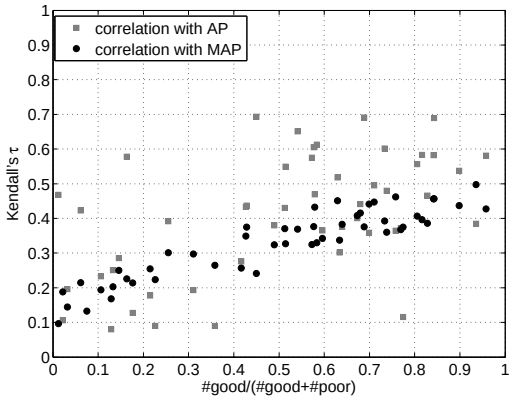



Figure 5.3: Distribution of topics in the best and worst subsets ($RS$, TREC-7): Of the $10000$ random samples of topic subsets with cardinality $c = 10$, the best and worst performing $250$ subsets are kept. For each of the $50$ topics in TREC-7, it is recorded how often it appears in the good and bad subsets, the x-axis contains the ratio. Shown are the topics and their correlation with AP and their correlation with MAP.

high correlation with MAP. The topics that are predicted to the highest degree (their correlation with AP) are not necessarily those that are always found in the best sets – there are inter-relations with the other topics in the subset. As the best subsets are made out of topics that estimate rankings with a good correlation to the MAP based ground truth, we clustered the estimated rankings and chose as subset the cluster with the largest number of topics, estimating that this might be a cluster of good topics.

Finally, we attempted to approximate the estimation accuracy of a topic (Section 5.5.2) by introducing noise to the estimated performance scores and testing the robustness of the ranking against randomly introduced perturbations. If one estimator estimates the performance scores of three documents to be $(0.2, 0.201, 0.22)$ while a second estimator derives the scores $(0.2, 0.6, 1.0)$ we would have more confidence in the ranking of documents by the second estimator, as the score differences are larger, whereas the confidence in the document ranking is smaller for the first estimator, as the estimated performance scores are very similar. We tested this intuition by adding Gaussian noise to the estimated performances scores and determining by how much the ranking of documents after the introduction of noise differs from the ranking of documents based on the unperturbed scores. This method was motivated by the query performance prediction methods that work on query and document perturbations (Chapter 3).

The evaluation of those three more advanced strategies, however, failed to achieve better results than the $JSD$ strategy. The reasons for the failure to identify valuable topic subsets with either of these mechanisms are not well understood yet and require further investigation.

| | RS | JSD | |
|---|---|---|---|
| | full set | c=10 | c=20 |
| **TREC-6** | 0.443 | **0.455** | **0.485** |
| **TREC-7** | 0.466 | **0.489** | **0.505** |
| **TREC-8** | 0.538 | **0.585** | **0.588** |
| **TREC-9** | 0.677 | 0.649 | 0.644 |
| **TREC-10** | 0.643 | 0.634 | 0.635 |
| **TB-04** | 0.708 | **0.760** | **0.733** |
| **TB-05** | 0.659 | **0.670** | 0.612 |
| **TB-06** | 0.518 | 0.495 | 0.508 |
| **CLIR-01** | 0.702 | **0.706** | 0.698 |
| **NP-02** | 0.693 | 0.623 | 0.597 |
| **EDISC-05** | 0.666 | **0.709** | **0.729** |
| **EEXP-05** | 0.483 | **0.616** | **0.616** |
| **BLTR-06** | 0.523 | 0.501 | **0.528** |
| **GEN-07** | 0.563 | 0.530 | 0.556 |
| **LEGAL-07** | 0.741 | 0.695 | 0.728 |
| **RELFB-08** | 0.559 | **0.589** | **0.638** |

Table 5.5: Overview of Kendall's $\tau$ achieved by $RS$ on the full set of topics and on topic subsets of cardinality $c = 10$ and $c = 20$ of the $JSD$ topic subset selection strategy. In bold, improvements over the full topic set. All correlations reported are significant ($p < 0.005$), though none are statistically significantly different from the highest correlation per data set.

## 5.6 Conclusions

In this chapter, we have investigated the task of system ranking estimation, which attempts to rank a set of retrieval systems, for a given topic set and test corpus, according to their relative performance *without* relying on relevance judgments. This type of automatic evaluation could in the ideal case be used in the context of formal evaluations though currently the results suggest that this is not a realistic goal yet. We have described the most common approaches and performed an evaluation of them on a wider variety of data sets than done previously. In contrast to earlier findings [9, 114, 133, 135, 161] on a small number of older TREC data sets, we found the initially proposed approach by Soboroff et al. [133] to be the most stable and the best performing one. Moreover, we found the commonly reported problem of system ranking estimation methods, namely the severe underestimation of the performance of the best systems, not to be an inherent problem of system ranking estimation approaches. Instead we argue that this is a data set dependent issue, in particular it depends on the amount of human intervention in the best systems of a data set. If the best system is automatic in nature, or is derived with a small amount of human intervention, it can often be identified with a high degree of accuracy, or for some data sets, even correctly. This result suggests, that especially in practical applications, where we have a choice of different retrieval approaches it can be possible to automatically determine the best (or close to the best) performing one.

In terms of evaluation, it also proved beneficial to report not only the rank correlation coefficient Kendall's $\tau$ as evaluation measure of system ranking estimation approaches, but also to report the estimated rank of the best system as this measure provides an alternative view of an approach's performance.

In a second set of experiments, we turned to investigating the ability of system ranking estimation approaches to estimate the ranking of systems for each individual topic. We showed that the quality of the estimated rankings vary widely within a topic set. Based on this result, we designed a number of motivational experiments with different subset selection strategies. We were able to confirm the hypothesis that there exist subsets of topics that are better suited for the system ranking estimation task than others. Having found this regularity is only the first step however, for this knowledge to be useful in a practical setting, automatic methods are required that can identify those good subsets of topics to rely on.

We also proposed a strategy to automatically identify good subsets of topics by relying on topics that have been estimated to be easy. This strategy yielded some improvements, though they were not consistent across all data sets. Considering the amount of potential improvement, this can only be considered as a first attempt at subset selection.

# Chapter 6

# Conclusions

In this thesis we have investigated the prediction of query and retrieval system effectiveness. As we introduced the topic we clearly identified its pertinent evaluation aspects (Figure 1.1) and set the focus on two aspects in particular, namely predicting the effectiveness of queries for a particular system (EA2) and predicting the relative effectiveness of systems (EA4).

The motivation for our research efforts stems primarily from the enormous benefits originating from successfully predicting the quality of a query or a system. Accurate predictions enable the employment of adaptive retrieval components which would have a considerable positive effect on the user experience. Furthermore, if we would achieve sufficiently accurate predictions of the quality of retrieval systems, the cost of evaluation would be significantly reduced.

We have conducted our research along four lines: the pre-retrieval and post-retrieval prediction of query effectiveness, the contrast between the evaluation of predictors and their effect in practice, and, lastly, the prediction of system effectiveness.

## 6.1 Research Themes

### 6.1.1 Pre-Retrieval Prediction

Pre-retrieval prediction methods are used by retrieval systems to predict the quality of a ranked list of results retrieved in response to a query *without* actually retrieving the result list. Instead of considering the content of the result list, the methods rely on collection statistics and external resources such as semantic dictionaries to derive a prediction. The first research theme **RT1** revolves around these methods and considers the following research questions: On what heuristics are the prediction algorithms based? Can the algorithms be categorized in a meaningful way? How similar are different approaches with respect to their behavior to each other? How sensitive are the algorithms to a change in the retrieval approach? What gain can be achieved by combining different approaches?

We conclude in Chapter 2 that prediction methods are distinguished, in the literature, in four different classes according to the heuristics they exploit to predict the

effectiveness of a query. As such, *specificity* based prediction methods relate more specific query terms to a better performance, while *ambiguity* based predictors rely on the query terms' level of ambiguity to determine the performance. Ambiguous query terms are predicted to lead to a poor retrieval effectiveness while unambiguous query terms are viewed as evidence for a high retrieval effectiveness. A number of prediction methods also rely on the degree of *relatedness* between query terms to infer the query's performance: related query terms are predicted to lead to a better search result than unrelated query terms. Finally, the *ranking sensitivity* based prediction methods attempt to infer how difficult it will be for a retrieval approach to rank the documents that contain the query terms.

We performed an analytical and empirical evaluation of the prediction methods within each class and showed substantial similarities between them. When evaluating the prediction methods according to their ability to predict the effectiveness of queries on three different corpora we found their accuracy to be dependent on the retrieval approach, the query set and the corpus under investigation. We also showed that the dependency on the retrieval approach is very pronounced, not only when considering diverse retrieval approaches, but also when considering the different parameter settings of a single retrieval approach.

Overall, our results have indicated that when comparing predictor performances, a single retrieval setting can be misleading, and when possible, a variety of retrieval methods should be evaluated before conclusive observations are drawn about the merits of individual predictors. The general lack of predictor robustness as evinced from our work also brings into question the merits of pre-retrieval predictors; if they are unstable and often result in poor prediction accuracy, then the advantage of being low cost in terms of processing time is lost.

Finally the potential gain in accuracy when combining prediction methods has been explored. Specifically, we investigated the utility of penalized regression as a principled approach to combine predictors. The evaluation showed potential, for two of our three corpora the penalized regression methods led to improvements over the best single individual predictor.

### 6.1.2 Post-Retrieval Prediction

Approaches predicting the effectiveness of a query's result list by indeed considering the result list are employed after the initial retrieval stage and are thus called post-retrieval. The questions posed as part of the second research theme **RT2** were set around the post-retrieval predictor Clarity Score [45] and were as follows: How sensitive is this post-retrieval predictor to the retrieval algorithm? How does the algorithm's performance change over different test collections? Is it possible to improve upon the prediction accuracy of existing approaches?

In Chapter 3 we were able to show on two concrete predictor examples, one of which was Clarity Score, that post-retrieval prediction methods are as sensitive to the parameter settings of the retrieval approach as pre-retrieval predictors. The same observation holds for the performance of Clarity Score on different test corpora; the prediction accuracy varies widely depending on the corpus and the partic-

ular query set under investigation. We proposed two adaptations to Clarity Score: (i) setting the number of feedback documents used in the estimation of the query language model individually for each query to the number of documents that contain all query terms, and, (ii) ignoring high-frequency terms in the KL divergence calculation. These adaptations were thoroughly tested on three TREC test collections. With the exception of one set of queries, one or more of the proposed variations always outperformed the Clarity Score baseline, often by a large margin.

The main conclusion we draw from the investigations of Chapter 3 is that *Adapted Clarity* is a highly competitive post-retrieval approach which, on average across all evaluated corpora, outperforms all other tested pre- as well as post-retrieval predictors.

### 6.1.3 Contrasting Evaluation and Application

The third research theme dealt with the relationship of the current evaluation methodology for query performance prediction and the change in retrieval effectiveness of adaptive systems that employ a predictor for selective query expansion or meta-search. In selective query expansion a predictor is expected to predict when the application of automatic query expansion will lead to a higher quality result list. When applied in meta-search, for each query there exists a choice of result lists and a predictor is expected to identify the one of highest quality.

In particular the posed questions of **RT3** were: What is the relationship between the correlation coefficient as an evaluation measure for query effectiveness estimation and the effect of such a method on retrieval effectiveness? At what levels of correlation can we be reasonably sure that a query performance prediction method will be useful in an operational setting?

In Chapter 4 we provide first answers to these questions. We chose these two operational settings as they are most often mentioned as potential applications of query effectiveness prediction. Our experiments have shown that the level of Kendall's $\tau$ required to be confident that a prediction method is viable in practice is dependent on the particular operational setting it is employed in. In the case of selective query expansion, a value of $\tau \geq 0.4$ has been found to be the minimum level of correlation that should be attained provided perfect knowledge of the behavior of the employed automatic query expansion mechanism is available. A second experimental inquiry evaluated the effect of overly optimistic assumptions such as that query expansion aids all queries with initially high effectiveness. Under these circumstances predictors need to achieve a correlation of $\tau \geq 0.75$ for them to be viable.

In the meta-search setting, the level of correlation required to reliably improve the retrieval effectiveness of a meta-search system is shown to be dependent on the performance differences of the participating systems as well as on the number of systems employed. Notably, when the effectiveness of all systems is similar, prediction methods acheiving low levels of correlation are already sufficient. However, when the differences in system performance are large and we are interested in statistically significant improvements, the level of correlation necessary varies between

$\tau = 0.5$ ($m = 150$) and $\tau = 0.7$ ($m = 50$) depending on the number $m$ of queries participating in the experiment.

Based on the knowledge we gained in Chapter 2 and Chapter 3 we can convincingly state our main conclusion as follows: current query effectiveness prediction methods are not sufficiently accurate to lead to consistent and significant improvements when applied to meta-search and selective query expansion.

### 6.1.4 System Effectiveness Prediction

In Chapter 5 we turned to estimating the ranking of retrieval systems as set by the fourth research theme **RT4**. The questions posed were: Is the performance of system ranking estimation approaches as reported in previous studies comparable with their performance for more recent and diverse data sets? What factors influence the accuracy of system ranking estimation? Can the accuracy be improved when selecting a subset of topics to rank retrieval systems?

In order to answer these questions, we have investigated a wide range of data sets covering a variety of retrieval tasks and a variety of test collections. We found that in contrast to earlier studies which were mostly conducted on the same small number of data sets, there are indeed differences in the ability to rank retrieval systems depending on the data set. The issue that has long prevented this line of evaluation to be used in practice has been shown to be the mis-ranking of the best systems. In the extreme case, the most effective systems are estimated to be among the worst performing ones. In our experiments however, we have discovered this not to be an inherent problem of system ranking estimation approaches. The extent of the mis-ranking problem was shown to be data set dependent and, more specifically, dependent on the amount of human intervention in the best system of a data set. We conclude that in cases where the best system is (largely) automatic, the best system can often be identified with a high degree of accuracy.

The evaluation of retrieval systems has always been performed based on some set of topics. To answer the final question on accuracy improvement we first investigated the variability between topics, that is we evaluated how well the systems can be ranked for each individual topic. The result of this investigation motivated the follow up question on whether we can improve the ability of estimating a ranking of systems when relying on a subset of topics. In a motivational study we have shown that selecting topic subsets from the full set of topics can lead to a significantly higher accuracy.

The most important conclusion to have emerged from the work in Chapter 5 is that automatic system ranking estimation methods are *not* a lost cause. They are in fact capable of high quality estimations in contrast to previous findings.

## 6.2 Future Work

A number of future research avenues have become evident in the course of this work. One particular direction is the exploration of alternatives to the currently

employed correlation coefficients, namely Kendall's $\tau$ and the linear correlation coefficient $r$. Blest [21] proposes a rank correlation coefficient that weights errors at the top end of the ranking more than errors at the bottom of the ranking, which stands in contrast to Kendall's $\tau$ where all errors are weighted equally. In particular in the context of system ranking estimation, where we are often most interested in identifying the best performing systems correctly such an evaluation measure can be useful. In the context of Information Retrieval, Yilmaz et al. [166] propose a rank correlation coefficient based on average precision that penalizes errors at the top of the ranking to a higher degree. Both of the above may be considered as alternative evaluation measures in future work.

Most experiments in the realm of query performance prediction, including the experiments reported in this thesis, have been performed on informational queries. There are also other types of queries, such as navigational queries or transactional queries [23]. How the current approaches for informational queries can be translated to those query types largely remains an open question.

With the introduction of the Million Query track [3] to TREC, a much larger number of topics ($10000$ topics to be exact) has recently become available than the standard topic set size of between $50$ and $250$ topics for earlier test collections. Relevance judgments for such a large number of topics cannot be derived in the same manner as for a small set of topics though and, therefore, instead of average precision, new effectiveness measures had to be introduced such as statistical AP [8]. This development naturally leads to two further research questions; first to evaluate the performance of query performance prediction methods for such topic set sizes, and, second, to investigate if the novel evaluation measures can also be predicted reasonably well.

One future work prospect of Chapter 2 is the evaluation of the robustness of the prediction methods with respect to TREC runs. Since the predictor performances vary widely, it would be beneficial to analyze for which kind of retrieval approaches the different prediction methods perform well and for which they fail. Such an analysis would require an extensive review of all TREC runs and the methods they employ.

Future work related to Chapter 3 could focus on setting the feedback document parameter more effectively, specifically by taking into account the dependency between the query terms. Furthermore, the question of how best to set the $N$ parameter automatically arises. An alternative line of investigation in particular for the Web corpus WT10g would be to preprocess the documents by filtering out the non-topical content such as navigational information, page decoration, etcetera. Such an approach has been shown to improve the effectiveness of pseudo-relevance feedback of the WT10g data set [168], and might also be beneficial for query effectiveness prediction.

A central assumption of Clarity Score is that for an unambiguous query, the top retrieved documents are more focused than the corpus. Although this is a valid assumption if each document contains exactly one topic, often documents covering multiple topics occur frequently in a collection and unnecessary noise is added to the query language model. Therefore, a future investigation could be the segmen-

tation of each document according to subtopics in order to alleviate these effects. The TextTiling [74] or C99 [35] segmentation algorithms could be employed for instance and those segmented passages may then be used where query terms occur in the creation of the query language model rather than relying on the entire document.

The study of Chapter 4, which investigated the contrast between the evaluation of query performance prediction and the application of prediction methods in practice, also offers diverse lines of follow-up research. In this work, we restricted ourselves to an analysis of Kendall's $\tau$, however, a future effort might perform a similar analysis of the linear correlation coefficient $r$. In contrast to the rank-based $\tau$, $r$ is based on raw scores, which adds another dimension to the study, namely the distribution of raw scores. A second measure which might also be investigated further is the *area between the MAP curves* [151], already briefly discussed in Section 2.3.2. Although it has not been widely used in the query performance prediction literature, we hypothesize that it is particularly useful for the operational setting of selective query expansion, as it emphasizes the worst performing topics.

Finally, the experiments on automatic system evaluation described in Chapter 5 could also be further explored. On the one hand, for topic subset selection to be beneficial, it is still necessary to develop an automatic method that identifies the most suitable subsets; here one could concentrate on identifying features that enable us to distinguish the topics that appear mostly in subsets which improve system ranking estimation, from those topics that appear mostly in poorly performing subsets. Another direction to consider is the adaptation of the $RS$ approach by selectively boosting some documents in the pool of documents to sample from. This idea is motivated by the fact that in the case of easy topics, the very best systems will retrieve ranked lists of documents similar to average systems, while for more difficult topics the result lists will diverge. If we can identify the systems, that appear average on easy topics and unlike average systems on harder topics, we can boost the number of documents entered into the pool by them. This would require a comparison of document overlap across different topics, which is a deviation from current work where each topic is viewed in isolation.

Effectiveness predictors have great potential as adaptive systems that take the correct query-dependent actions are bound to outperform systems applying a one-size-fits-all approach. Although this potential is not yet fulfilled as shown in this thesis, current state-of-the-art methods are slowly beginning to reach the levels of accuracy required in practical settings, motivating future research in this direction.

# Appendix A

# ClueWeb09 User Study

In order to investigate how well proficient Web searchers are able to predict the quality of search results retrieved in response to a query, we took the fifty queries released for the adhoc task of TREC 2009 on the ClueWeb09 corpus, and asked the users to judge for each query whether the search results will be of low, medium or high quality. Since the users were asked to judge the queries without looking at the result list, they in fact acted as human pre-retrieval predictors. We created an online questionnaire with the following task description:

> For each of the fifty queries, please judge what you would expect the results to be, if you would submit the query to a Web search engine. Do you expect the top results to be of high quality (i.e. many results are relevant to the query), of medium quality or do you believe the top ranked results will be of low quality (i.e. few or none results will be relevant)? If you have no intuition about the query, please use "unknown". Please do NOT actually submit the queries to a search engine, rely on your intuition only.

A total of thirty-three users (twenty-six male, seven female) participated in our study. They were recruited from the Database group and the Human Media Interaction group, Department of Electrical Engineering, Mathematics and Computer Science at the University of Twente. All users stated to use the Internet daily, thirty of them use Web search engines four or more times a day, while three users use them at least once a day. The age of the participants ranged between $20-29$ years (nineteen users), $30-39$ years (nine users), $40-49$ years (two users), $50-59$ years (two users) and $60$ years or older (one user).

The queries and the user responses are listed in Table A.1. The last column contains the average query difficulty score we derived for each query based on the user responses. The scores of $1$, $2$ and $3$ were assigned to the judgments of low, medium and high quality respectively and then averaged. The user judgment "unknown" was ignored. Then, the average score column was correlated against the ground truth. As ground truth we relied on the best and the median estimated AP score of each query over all runs submitted to TREC 2009 for the adhoc task. The linear correlation between human prediction and ground truth evaluated to $r = 0.35$ (best estimated AP score) and $r = 0.46$ (median estimated AP score) respectively.

| Query | Users Judgments | | | | Average Score |
|---|---|---|---|---|---|
| | unknown | low | medium | high | |
| obama family tree | 1 | 1 | 10 | **21** | 2.625 |
| french lick resort and casino | 7 | 6 | 7 | **13** | 2.269 |
| getting organized | 0 | **21** | 9 | 3 | 1.455 |
| toilet | 0 | **24** | 7 | 2 | 1.333 |
| mitchell college | 1 | 3 | 12 | **17** | 2.438 |
| kcs | **21** | 6 | 6 | 0 | 1.500 |
| air travel information | 0 | **14** | 12 | 7 | 1.788 |
| appraisals | 6 | **23** | 2 | 2 | 1.222 |
| used car parts | 2 | 8 | **19** | 4 | 1.871 |
| cheap internet | 0 | **18** | 11 | 4 | 1.576 |
| gmat prep classes | 6 | 6 | **11** | 10 | 2.148 |
| djs | 8 | **23** | 2 | 0 | 1.080 |
| map | 0 | **26** | 3 | 4 | 1.333 |
| dinosaurs | 1 | 4 | **18** | 10 | 2.188 |
| espn sports | 6 | 2 | 5 | **20** | 2.667 |
| arizona game and fish | 3 | 8 | **16** | 6 | 1.933 |
| poker tournaments | 2 | 5 | **18** | 8 | 2.097 |
| wedding budget calculator | 2 | 1 | 13 | **17** | 2.516 |
| the current | 2 | **29** | 1 | 1 | 1.097 |
| defender | 1 | **27** | 4 | 1 | 1.188 |
| volvo | 0 | 2 | 11 | **20** | 2.545 |
| rick warren | 5 | 3 | 10 | **15** | 2.429 |
| yahoo | 0 | 2 | 8 | **23** | 2.636 |
| diversity | 1 | **23** | 5 | 4 | 1.406 |
| euclid | 3 | 4 | **16** | 10 | 2.200 |
| lower heart rate | 0 | 8 | **16** | 9 | 2.030 |
| starbucks | 0 | 4 | 5 | **24** | 2.606 |
| inuyasha | **17** | 2 | 2 | 12 | 2.625 |
| ps 2 games | 3 | **10** | **10** | **10** | 2.000 |
| diabetes education | 1 | 3 | **17** | 12 | 2.281 |
| atari | 2 | 3 | **15** | 13 | 2.323 |
| website design hosting | 0 | 10 | **16** | 7 | 1.909 |
| elliptical trainer | 5 | 4 | **15** | 9 | 2.179 |
| cell phones | 1 | **15** | 11 | 6 | 1.72 |
| hoboken | **15** | 6 | 6 | 6 | 2.000 |
| gps | 0 | 12 | **16** | 5 | 1.788 |
| pampered chef | **12** | 10 | 8 | 3 | 1.667 |
| dogs for adoption | 0 | 4 | **19** | 10 | 2.182 |
| disneyland hotel | 0 | 3 | 7 | **23** | 2.606 |
| michworks | **15** | 2 | 7 | 9 | 2.389 |
| orange county convention center | 1 | 1 | 4 | **27** | 2.813 |
| the music man | 6 | **17** | 9 | 1 | 1.407 |
| the secret garden | 3 | 11 | **13** | 6 | 1.833 |
| map of the united states | 0 | 0 | 5 | **28** | 2.848 |
| solar panels | 0 | 2 | **19** | 12 | 2.303 |
| alexian brothers hospital | 5 | 0 | 4 | **24** | 2.857 |
| indexed annuity | 10 | 4 | **13** | 6 | 2.087 |
| wilson antenna | 7 | 3 | **13** | 10 | 2.269 |
| flame designs | 2 | 10 | **15** | 6 | 1.871 |
| dog heat | 4 | **16** | 11 | 2 | 1.517 |

Table A.1: User judgments of the queries of the TREC 2009 Web adhoc task.

# Appendix B

# Materials and Methods

## B.1 Test Corpora

To perform the experiments, the adhoc retrieval task was evaluated on three different corpora, namely, TREC Volumes 4+5 minus the Congressional Records (TREC Vol. 4+5) [148], WT10g [132] and GOV2 [38]. The three corpora, although in the general domain, are quite different from each other. To illustrate this, consider their basic statistics, shown in Table B.1. The table lists the number of documents in each corpus ($\#docs$), the indexed average document length and the linear correlation coefficient between the document frequency and term frequency of the vocabulary ($r(cf, df)$). The average document length is dependent on the particular preprocessing steps such as HTML parsing, tokenization and stemming. All reported results are based on the Lemur Toolkit for Language Modeling and Information Retrieval[1], version 4.3.2. The corpora were stemmed with the Krovetz stemmer [90] and stopwords were removed[2].

TREC Vol. 4+5 consists of news reports and is the smallest of the three corpora. WT10g is a corpus which was extracted from a crawl of the Web; it is rather noisy and contains numerous pages without text (pages containing images only for example), pages containing up to two terms only ("test page", "under construction"), copyright notices, etc. The largest corpus is GOV2. It was derived from a crawl of the .gov domain and resembles to some extend an intranet structure and thus can be expected to be less noisy than WT10g. To assess this assumption informally, the ten most frequently occurring terms of documents with a maximum indexed length of ten and fifty terms respectively are shown in Table B.2. The high frequency terms *construct* and *copyright* in WT10g indicate, that short documents in WT10g might often not be useful for informational queries.

A different view of the amount of informational pages of a corpus can be gained by considering the percentage of stopwords per document. TREC Vol. 4+5 consists of news reports of which most can be considered information bearing. The situation is different for Web pages. Imagine a web site consisting of 2 frames, one

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

[2] stopword list: http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words

| | TREC Vol. 4+5 | WT10g | GOV2 |
|---|---|---|---|
| $\#docs$ | 528155 | 1692095 | 25205179 |
| $av.\ doc.\ length$ | 266.4 | 377.6 | 665.3 |
| $r(cf, df)$ | 0.95 | 0.88 | 0.67 |

Table B.1: Basic corpora statistics.

| TREC Vol. 4+5 | | WT10g | | GOV2 | |
|---|---|---|---|---|---|
| $\leq 10$ | $\leq 50$ | $\leq 10$ | $\leq 50$ | $\leq 10$ | $\leq 50$ |
| edition | ft | page | page | slide | library |
| 1990 | edition | home | home | state | session |
| 1989 | news | return | 1996 | class | public |
| home | home | 1996 | copyright | library | time |
| final | company | click | information | locate | click |
| pm | 94 | ye | web | washington | kalamazoo |
| thursday | said | construct | return | law | 2002 |
| county | people | homepage | use | patent | day |
| friday | world | web | mail | subclass | webcat |
| sunday | 93 | 1 | right | body | 2000 |

Table B.2: The most frequently occurring terms in documents with less or equals to ten and fifty indexed terms respectively.

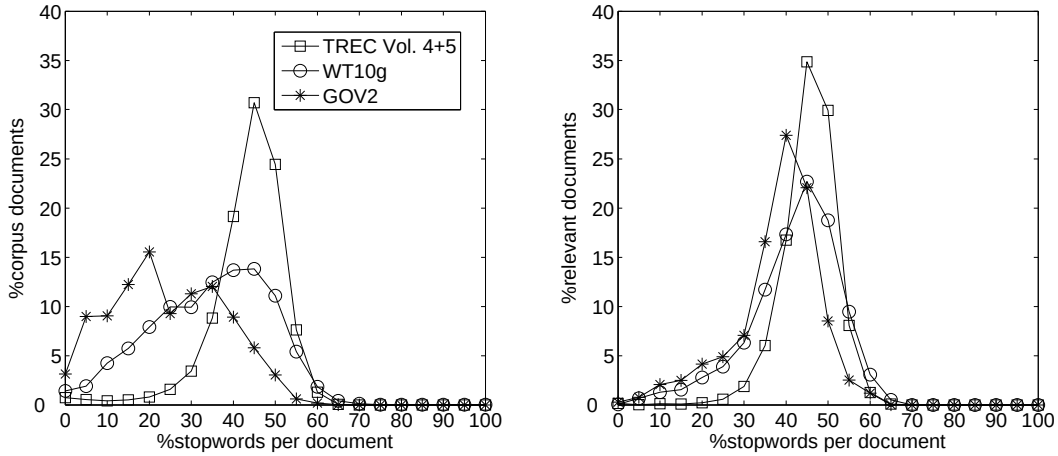



Figure B.1: Distribution of the amount of stopwords over all (left) and only the relevant documents (right).

navigational, and one with informative content. The former will appear as a long list of keywords hardly containing any stopwords, whereas the latter will appear as an ordinary content-bearing document. When, as in this case, the stopwords are removed in the preprocessing step, the two document types (keyword list and content page) appear similar in the index. If the percentage of stopwords is plotted against the percentage of documents, as done in the left part of Figure B.1, the difference becomes apparent. The vast majority of news reports contain between $30\%$ and

55% stopwords per document. The graphs for WT10g and GOV2 show a different picture: the percentage of stopwords varies considerably, there is no pronounced peak as for TREC Vol. 4+5. For an effective comparison, the right part of Figure B.1 contains the stopword distribution of known content-bearing pages, which in this example are the relevant documents available for the query sets in our experiments. The stopword distribution of the relevant documents appears to be similar across all three corpora: it has a peak at 40% (GOV2) and 45% (TREC Vol. 4+5, WT10g) respectively and is less spread then the distribution over all collection documents.

## B.2 Query Sets

The algorithms are evaluated on informational queries, the most common query type in TREC evaluations. The system is posed an informational query and returned are the documents deemed most probable by the system to be relevant. In the adhoc task, the TREC topics usually consist of a title, description and narrative part. The title part of a topic contains mostly between 1 and 3 terms. For instance, the title part of TREC topic 485 is "gps clock", the description part is shown below:

> Clock reliance is a very important consideration in the operation of a global positioning system (GPS) What entity is responsible for clock accuracy and what is the accuracy?

In Table B.3 we present an overview of the query sets used in our experiments and the corpus they belong to. For reasons of comparison, the average length of the title queries and description queries, which are derived from the TREC title topics and TREC description topics respectively, are also shown. Whereas the average length of title queries is relatively stable across all query sets, the average length of description queries is significantly longer for the query sets of TREC Vol. 4+5 than of WT10g and GOV2. All reported retrieval experiments are based on queries derived from TREC title topics.

| Corpus | Query Set | Av. Title Length | Av. Descr. Length |
|---|---|---|---|
| TREC Vol. 4+5 | 301-350 | 2.54 | 13.24 |
| | 351-400 | 2.50 | 11.04 |
| | 401-450 | 2.40 | 9.62 |
| WT10g | 451-500 | 2.43 | 7.26 |
| | 501-550 | 2.84 | 4.78 |
| GOV2 | 701-750 | 3.10 | 5.84 |
| | 751-800 | 2.94 | 5.74 |
| | 801-850 | 2.86 | 6.60 |

Table B.3: Overview of query sets.

In query set 451-500, we manually identified and corrected three spelling errors. Our focus is on investigating performance prediction algorithms and we assume the ideal case of error-free queries. In practical applications, spelling error correction would be a preprocessing step.

## B.3 The Retrieval Approaches

The goal of prediction algorithms is to predict the actual retrieval performance as best as possible. The average precision of

- Language Modeling with Dirichlet Smoothing [170],
- Okapi [125], and,
- TF.IDF [13]

is used as ground truth. Additionally, the retrieval runs submitted by the participants of the TREC tasks are also utilized as they offer diverse retrieval approaches, which do not solely rely on the document content.

### B.3.1 Language Modeling, Okapi and TF.IDF

Table B.4 shows the performance of the three retrieval approaches in mean average precision (MAP) over all title topic based query sets; the smoothing level of the Language Modeling approach is varied between $\mu = \{100, 500, 1000, 1500, 2000, 2500\}$. Larger values of $\mu$ show no further improvements in retrieval effectiveness.

| | | TF.IDF | Okapi | Language Modeling with Dirichlet Smoothing | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | $\mu = 100$ | $\mu = 500$ | $\mu = 1000$ | $\mu = 1500$ | $\mu = 2000$ | $\mu = 2500$ |
| TREC | 301-350 | 0.109 | 0.218 | 0.216 | **0.227** | 0.226 | 0.224 | 0.220 | 0.218 |
| Vol. 4+5 | 351-400 | 0.073 | 0.176 | 0.169 | 0.182 | 0.187 | 0.189 | **0.190** | 0.189 |
| | 401-450 | 0.088 | 0.223 | 0.229 | 0.242 | **0.245** | 0.244 | 0.241 | 0.239 |
| WT10g | 451-500 | 0.055 | 0.183 | 0.154 | 0.195 | **0.207** | 0.206 | 0.201 | 0.203 |
| | 501-550 | 0.061 | 0.163 | 0.137 | 0.168 | 0.180 | 0.185 | **0.189** | 0.189 |
| GOV2 | 701-750 | 0.029 | 0.230 | 0.212 | 0.262 | **0.269** | 0.266 | 0.261 | 0.256 |
| | 751-800 | 0.036 | 0.296 | 0.279 | 0.317 | **0.324** | 0.324 | 0.321 | 0.318 |
| | 801-850 | 0.023 | 0.250 | 0.247 | 0.293 | **0.297** | 0.292 | 0.284 | 0.275 |

Table B.4: Overview of mean average precision over different retrieval approaches. In bold the top performing retrieval run for each query set.

Due to the nature of some prediction methods, it is expected that the amount of smoothing in the Language Modeling approach will have a considerable influence on their quality. To evaluate the influence of high levels of smoothing, in addition to the moderate settings of $\mu$ listed in Table B.4, $\mu$ is evaluated for the settings of $5 \times 10^3$, $1 \times 10^4$, $1.5 \times 10^4$, $2 \times 10^4$, $2.5 \times 10^4$, $5 \times 10^4$, $1 \times 10^5$, $1.5 \times 10^5$, $2 \times 10^5$, $2.5 \times 10^5$, $3 \times 10^5$ and $3.5 \times 10^5$. The development of the retrieval performance in MAP over all levels of smoothing is illustrated in Figure B.2. Across all query sets, a performance drop is visible for $\mu > 2000$. The query sets of the GOV2 collection are the most sensitive to the setting of $\mu$, the drop in performance is considerably steeper than for the query sets of WT10g and TREC Vol. 4+5.

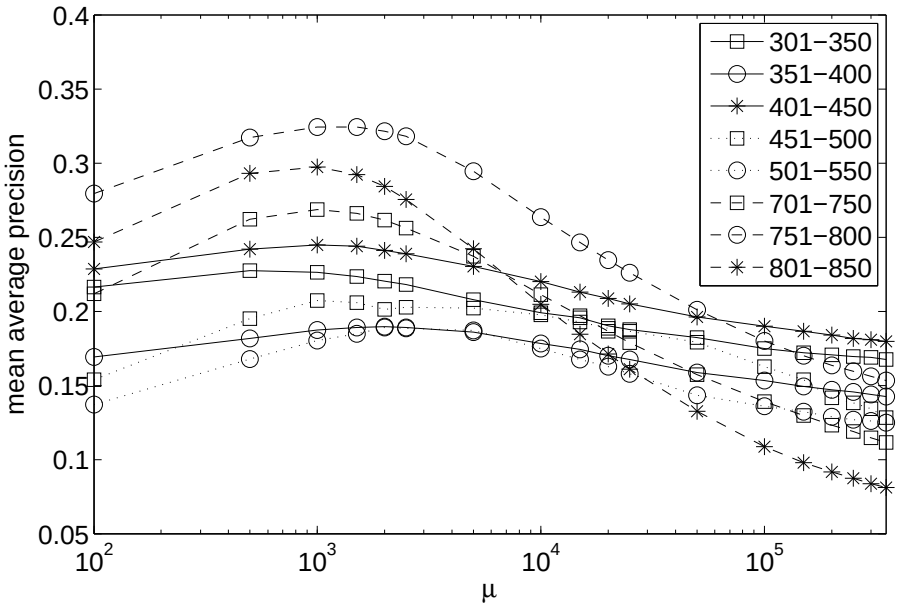



Figure B.2: Development of retrieval effectiveness over a range of smoothing values ($\mu$) in the Language Modeling Approach with Dirichlet Smoothing.

### B.3.2 TREC Runs

In contrast to the standard retrieval methods introduced in the last section, retrieval runs submitted by the participants of TREC tasks cover a variety of retrieval approaches and consider evidence from diverse sources. Such sources include not just the content of the documents, but for instance also the link structure, anchor texts (in the case of WT10g and GOV2) and document titles. Some approaches also employ collection enrichment strategies.

| Corpus | Topic set | Number of Runs | Mean MAP | Median MAP | Min. MAP | Max. MAP |
|---|---|---|---|---|---|---|
| **TREC Vol. 4+5** | 351-400 | 23 | 0.189 | 0.187 | 0.115 | 0.261 |
| | 401-450 | 18 | 0.247 | 0.256 | 0.139 | 0.306 |
| **WT10g** | 451-500 | 28 | 0.157 | 0.163 | 0.106 | 0.201 |
| | 501-550 | 54 | 0.169 | 0.178 | 0.107 | 0.223 |
| **GOV2** | 701-750 | 33 | 0.206 | 0.210 | 0.107 | 0.284 |
| | 751-800 | 40 | 0.282 | 0.297 | 0.112 | 0.389 |
| | 801-850 | 47 | 0.293 | 0.299 | 0.120 | 0.374 |

Table B.5: Overview of automatic TREC title topic runs with a MAP above 0.1.

For each topic set, all runs submitted to TREC in the year of the topic set's introduction have been taken into consideration. All automatic runs exploiting only the title part of the TREC topics with a MAP above 0.1 are utilized. An overview of the number of runs and the minimum, maximum, median and mean MAP across all runs is given in Table B.5. Since for topics 301-350 no summary information is

available about which part of the TREC topic was used by the participants, the topic set is excluded from the experiments.

# Bibliography


[1] Tom Adi, O.K. Ewell, and Patricia Adi. High Selectivity and Accuracy with READWARE's Automated System of Knowledge Organization. In *Proceedings of the Eighth Text REtrieval Conference (TREC 8)*, 1999.

[2] M. Aljlayl, S. Beitzel, E. Jensen, A. Chowdhury, D. Holmes, M. Lee, D. Grossman, and O. Frieder. IIT at TREC-10. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, 2001.

[3] J. Allan, B. Carterette, and B. Dachev. Million Query Track 2007 Overview. In *Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.

[4] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. Query Difficulty, Robustness and Selective Application of Query Expansion. In *ECIR '04: Proceedings of the 26th European Conference on IR Research on Advances in Information Retrieval*, pages 127–137, 2004.

[5] Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, 20(4):357–389, 2002.

[6] Einat Amitay, David Carmel, Ronny Lempel, and Aya Soffer. Scaling IR-system evaluation using term relevance sets. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pages 10–17, 2004.

[7] Javed A. Aslam and Virgil Pavlu. Query Hardness Estimation Using Jensen-Shannon Divergence Among Multiple Scoring Functions. In *ECIR '07: Proceedings of the 29th European Conference on IR Research on Advances in Information Retrieval*, pages 198–209, 2007.

[8] Javed A. Aslam, Virgil Pavlu, and Emine Yilmaz. A statistical method for system evaluation using incomplete judgments. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, pages 541–548, 2006.

[9] Javed A. Aslam and Robert Savell. On the effectiveness of evaluating retrieval systems in the absence of relevance judgments. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval*, pages 361–362, 2003.

[10] Leif Azzopardi and Vishwa Vinay. Accessibility in Information Retrieval. In *ECIR '08: Proceedings of the 30th European Conference on IR Research on Advances in Information Retrieval*, pages 482–489, 2008.

[11] Francis R. Bach. Bolasso: model consistent Lasso estimation through the bootstrap. In *ICML '08: Proceedings of the 25th international conference on machine learning*, pages 33–40, 2008.

[12] Ricardo Baeza-Yates, Vanessa Murdock, and Claudia Hauff. Efficiency trade-offs in two-tier web search systems. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval*, pages 163–170, 2009.

[13] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley; 1st edition, 1999.

[14] Satanjeev Banerjee and Ted Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, pages 136–145, 2002.

[15] Satanjeev Banerjee and Ted Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *IJCAI '03: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 805–810, 2003.

[16] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, and David Grossman. Using titles and category names from editor-driven taxonomies for automatic evaluation. In *CIKM '03: Proceedings of the twelfth international conference on information and knowledge management*, pages 17–23, 2003.

[17] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, David Grossman, and Ophir Frieder. Using manually-built web directories for automatic evaluation of known-item retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval*, pages 373–374, 2003.

[18] Suma Bhat and Kenneth Church. Variable selection for ad prediction. In *ADKDD '08: Proceedings of the 2nd International Workshop on Data Mining and Audience Intelligence for Advertising*, pages 45–49, 2008.

[19] Bodo Billerbeck and Justin Zobel. When query expansion fails. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval*, pages 387–388, 2003.

[20] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer; 1 edition, 2007.

[21] D. C. Blest. Rank correlation: an alternative measure. *Australian and New Zealand Journal of Statistics*, 42:101–111, 2000.

[22] Mohan John Blooma, Alton Y. K. Chua, and Dion Hoe-Lian Goh. A predictive framework for retrieving the best answer. In *SAC '08: Proceedings of the 2008 ACM symposium on applied computing*, pages 1107–1111, 2008.

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

[24] Chris Buckley. Reliable information access final workshop report. Technical report, Northeast Regional Research Center (NRRC), 2004.

[25] Chris Buckley. Topic prediction based on comparative retrieval rankings. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pages 506–507, 2004.

[26] Chris Buckley and Stephen Robertson. Relevance Feedback Track Overview: TREC 2008. In *Proceedings of the Seventeenth Text REtrieval Conference (TREC 2008)*, 2008.

[27] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pages 25–32, 2004.

[28] L.J. Cao and F.E.H. Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, 14(6):1506–1518, 2003.

[29] David Carmel, Eitan Farchi, Yael Petruschka, and Aya Soffer. Automatic query refinement using lexical affinities with maximal information gain. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, pages 283–290, 2002.

[30] David Carmel, Elad Yom-Tov, Adam Darlow, and Dan Pelleg. What makes a query difficult? In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, pages 390–397, 2006.

[31] David Carmel, Elad Yom-Tov, and Haggai Roitman. Enhancing digital libraries using missing content analysis. In *JCDL '08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 1–10, 2008.

[32] Claudio Carpineto, Renato de Mori, Giovanni Romano, and Brigitte Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, 19(1):1–27, 2001.

[33] Ben Carterette and James Allan. Incremental test collections. In *CIKM '05: Proceedings of the 14th ACM international conference on information and knowledge management*, pages 680–687, 2005.

[34] Ben Carterette and James Allan. Research Methodology in Studies of Assessor Effort for Information Retrieval Evaluation. In *Proceedings of RIAO 2007*, 2007.

[35] Freddy Y. Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 26–33, 2000.

[36] A. Chowdhury, S. Beitzel, E. Jensen, M. Sai-lee, D. Grossman, and O. Frieder. IIT TREC-9 - Entity Based Feedback with Fusion. In *Proceedings of the Ninth Text REtrieval Conference (TREC 9)*, 2000.

[37] Abdur Chowdhury and Ian Soboroff. Automatic evaluation of world wide web search services. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, pages 421–422, 2002.

[38] Charles Clarke, Nick Craswell, and Ian Soboroff. Overview of the TREC 2004 Terabyte Track. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, 2004.

[39] Norman Cliff and Ventura Charlin. Variances and Covariances of Kendall's Tau and Their Estimation. *Multivariate Behavioral Research*, 26(4):693–707, 1991.

[40] Kevyn Collins-Thompson and Paul N. Bennett. Estimating query performance using class predictions. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval*, pages 672–673, 2009.

[41] Gordon V. Cormack, Charles L. A. Clarke, Christopher R. Palmer, and Samuel S. L. To. Passage-Based Refinement (MultiText Experiments for TREC-6). In *Proceedings of the Sixth Text REtrieval Conference (TREC 6)*, 1997.

[42] Nick Craswell, Arjen P. de Vries, and Ian Soboroff. Overview of the TREC-2005 Enterprise Track. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, 2005.

[43] Nick Craswell and David Hawking. Overview of the TREC-2002 Web Track. In *Proceedings of the Eleventh Text Retrieval Conference (TREC 2002)*, 2002.

[44] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[45] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, pages 299–306, 2002.

[46] Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. A framework for selective query expansion. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 236–237, 2004.

[47] Arjen P. de Vries, Anne-Marie Vercoustre, James A. Thom, Nick Craswell, and Mounia Lalmas. Overview of the INEX 2007 Entity Ranking Track. In *Focused Access to XML Documents*, pages 245–251, 2008.

[48] Dina Demner-Fushman, Susanne M. Humphrey, Nicholas C. Ide, Russell F. Loane, James G. Mork, Patrick Ruch, Miguel E. Ruiz, Lawrence H. Smith, W. John Wilbur, and Alan R. Aronson. Combining resources to find answers to biomedical questions. In *Proceedings of the The Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.

[49] Marcel Dettling and Peter Bühlmann. Finding predictive gene groups from microarray data. *Journal of Multivariate Analysis*, 90(1):106–131, 2004.

[50] Fernando Diaz. Performance prediction using spatial autocorrelation. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, pages 583–590, 2007.

[51] Fernando Diaz and Jaime Arguello. Adaptation of offline vertical selection predictions in the presence of user feedback. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval*, pages 323–330, 2009.

[52] Fernando Diaz and Rosie Jones. Using temporal profiles of queries for precision prediction. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pages 18–24, 2004.

[53] Fernando Diaz and Donald Metzler. Improving the estimation of relevance models using large external corpora. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, pages 154–161, 2006.

[54] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[55] Miles Efron. Using multiple query aspects to build test collections without human relevance judgments. In *ECIR '09: Proceedings of the 31st European Conference on IR Research on Advances in Information Retrieval*, pages 276–287, 2009.

[56] David A. Evans, Alison Huettner, Xiang Tong, Peter Jansen, and Jeffrey Bennett. Effectiveness of Clustering in Ad-Hoc Retrieval. In *Proceedings of the Seventh Text REtrieval Conference (TREC 7)*, 1998.

[57] Christiane Fellbaum, editor. *WordNet - An Electronic Lexical Database*. The MIT Press, 1998.

[58] Dennis Fetterly, Mark Manasse, Marc Najork, and Janet Wiener. A large-scale study of the evolution of web pages. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 669–678, 2003.

[59] Yupeng Fu, Wei Yu, Yize Li, Yiqun Liu, Min Zhang, and Shaoping Ma. THUIR at TREC 2005: Enterprise Track. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, 2005.

[60] Fredric C. Gey and Douglas W. Oard. The TREC-2001 Cross-Language Information Retrieval Track: Searching Arabic Using English, French or Arabic Queries. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, 2001.

[61] John Guiver, Stefano Mizzaro, and Stephen Robertson. A few good topics: Experiments in topic set reduction for retrieval evaluation. *ACM Transactions on Information Systems (in press)*, 2009.

[62] Donna Harman. Overview of the First Text REtrieval Conference (TREC-1). In *Proceedings of the First Text REtrieval Conference (TREC-1)*, 1992.

[63] Donna Harman and Chris Buckley. The NRRC reliable information access (RIA) workshop. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pages 528–529, 2004.

[64] Claudia Hauff and Leif Azzopardi. When is query performance prediction effective? In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval*, pages 829–830, 2009.

[65] Claudia Hauff, Leif Azzopardi, and Djoerd Hiemstra. The Combination and Evaluation of Query Performance Prediction Methods. In *ECIR '09: Proceedings of the 31st European Conference on IR Research on Advances in Information Retrieval*, pages 301–312, 2009.

[66] Claudia Hauff, Leif Azzopardi, Djoerd Hiemstra, and Franciska de Jong. Relying on Topic Subsets for System Ranking Estimation. In *CIKM '09: Proceedings of the 18th ACM conference on information and knowledge management*, pages 1859–1862, 2009.

[67] Claudia Hauff, Leif Azzopardi, Djoerd Hiemstra, and Franciska de Jong. Query Performance Prediction: Evaluation Contrasted with Effectiveness. In *ECIR '10: Proceedings of the 32nd European Conference on IR Research on Advances in Information Retrieval*, 2010, *to appear*.

[68] Claudia Hauff, Djoerd Hiemstra, Leif Azzopardi, and Franciska de Jong. A Case for Automatic System Evaluation. In *ECIR '10: Proceedings of the 32nd European Conference on IR Research on Advances in Information Retrieval*, 2010, *to appear*.

[69] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. A survey of pre-retrieval query performance predictors. In *CIKM '08: Proceeding of the 17th ACM conference on information and knowledge management*, pages 1419–1420, 2008.

[70] Claudia Hauff, Vanessa Murdock, and Ricardo Baeza-Yates. Improved query difficulty prediction for the web. In *CIKM '08: Proceeding of the 17th ACM conference on information and knowledge management*, pages 439–448, 2008.

[71] Ben He and Iadh Ounis. Inferring Query Performance Using Pre-retrieval Predictors. In *The Eleventh Symposium on String Processing and Information Retrieval (SPIRE)*, pages 43–54, 2004.

[72] Ben He and Iadh Ounis. Combining fields for query expansion and adaptive query expansion. *Information Processing and Management*, 43(5):1294–1307, 2007.

[73] Jiyin He, Martha Larson, and Maarten de Rijke. Using Coherence-Based Measures to Predict Query Difficulty. In *ECIR '08: Proceedings of the 30th European Conference on IR Research on Advances in Information Retrieval*, pages 689–694, 2008.

[74] Marti A. Hearst. TextTiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[75] William Hersh, Aaron Cohen, Lynn Ruslen, and Phoebe Roberts. TREC 2007 Genomics Track Overview. In *Proceedings of the The Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.

[76] Djoerd Hiemstra, Stephen Robertson, and Hugo Zaragoza. Parsimonious language models for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pages 178–185, 2004.

[77] H. Hotelling. The selection of variates for use in prediction with some comments on the general problem of nuisance parameters. *Annals of Mathematical Statistics*, 11: 271–283, 1940.

[78] Bernard J. Jansen, Danielle L. Booth, and Amanda Spink. Determining the user intent of web search engine queries. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1149–1150, 2007.

[79] Eric C. Jensen, Steven M. Beitzel, Abdur Chowdhury, and Ophir Frieder. Repeatable evaluation of search services in dynamic environments. *ACM Transactions on Information Systems*, 26(1):1–38, 2007.

[80] Eric C. Jensen, Steven M. Beitzel, David Grossman, Ophir Frieder, and Abdur Chowdhury. Predicting query difficulty on the web by learning visual clues. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval*, pages 615–616, 2005.

[81] Jiwoon Jeon, Bruce W. Croft, Joon Ho Lee, and Soyeon Park. A framework to predict the quality of answers with non-textual features. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, pages 228–235, 2006.

[82] J.J. Rocchio Jr. *The Smart system - experiments in automatic document processing*, chapter Relevance feedback in information retrieval, pages 313–323. Englewood Cliffs, NJ: Prentice Hall Inc., 1971.

[83] Tapas Kanungo and David Orr. Predicting the readability of short web summaries. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 202–211, 2009.

[84] M.G. Kendall. *Rank Correlation Methods*. New York: Hafner Publishing Co., 1955.

[85] Lyndon S. Kennedy, Shih-Fu Chang, and Igor V. Kozintsev. To search or to label?: predicting the performance of search-based automatic image classifiers. In *MIR '06: Proceedings of the 8th ACM international workshop on multimedia information retrieval*, pages 249–258, 2006.

[86] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka. Stock market prediction system with modular neural networks. In *IJCNN: Proceedings of the International Joint Conference on Neural Networks*, pages 1–6, 1990.

[87] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[88] Arnd Christian König, Michael Gamon, and Qiang Wu. Click-through prediction for news queries. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval*, pages 347–354, 2009.

[89] Wessel Kraaij, Thijs Westerveld, and Djoerd Hiemstra. The Importance of Prior Probabilities for Entry Page Search. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, pages 27–34, 2002.

[90] Robert Krovetz. Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on research and development in information retrieval*, pages 191–202, 1993.

[91] S. Kullback and R. A. Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[92] Giridhar Kumaran and James Allan. Selective user interaction. In *CIKM '07: Proceedings of the sixteenth ACM conference on conference on information and knowledge management*, pages 923–926, 2007.

[93] Giridhar Kumaran and James Allan. Effective and efficient user interaction for long queries. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval*, pages 11–18, 2008.

[94] Giridhar Kumaran and Vitor R. Carvalho. Reducing long queries using query quality predictors. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval*, pages 564–571, 2009.

[95] K.L. Kwok. An Attempt to Identify Weakest and Strongest Queries. In *ACM SIGIR '05 Workshop: Predicting Query Difficulty - Methods and Applications*, 2005.

[96] Hao Lang, Bin Wang, Gareth Jones, Jin-Tao Li, Fan Ding, and Yi-Xuan Liu. Query Performance Prediction for Information Retrieval Based on Covering Topic Score. *Journal of Computer Science and Technology*, 23(4):590–601, 2008.

[97] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, pages 120–127, 2001.

[98] Matthew Lease. Incorporating Relevance and Pseudo-relevance Feedback in the Markov Random Field Model. In *Proceedings of the Seventeenth Text REtrieval Conference (TREC 2008)*, 2008.

[99] Jure Leskovec, Susan Dumais, and Eric Horvitz. Web projections: learning from contextual subgraphs of the web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 471–480, 2007.

[100] Jianhua Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[101] Jimmy Lin, Eileen Abels, Dina Demner-Fushman, Douglas W. Oard, Philip Wu, and Yejun Wu. A Menagerie of Tracks at Maryland: HARD, Enterprise, QA, and Genomics, Oh My! In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, 2005.

[102] Craig Macdonald and Iadh Ounis. Expertise drift and query expansion in expert search. In *CIKM '07: Proceedings of the sixteenth ACM conference on conference on information and knowledge management*, pages 341–350, 2007.

[103] X.L. Meng, R. Rosenthal, and D.B. Rubin. Comparing correlated correlation coefficients. *Psychological Bulletin*, 111:172–175, 1992.

[104] Donald Metzler and Bruce W. Croft. A Markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval*, pages 472–479, 2005.

[105] Donald Metzler, Trevor Strohman, and W.B. Croft. Indri at TREC 2006: Lessons Learned From Three Terabyte Tracks. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.

[106] Donald Metzler, Trevor Strohman, Yun Zhou, and W.B. Croft. Indri at TREC 2005: Terabyte Track. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, 2005.

[107] Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval*, pages 206–214, 1998.

[108] Stefano Mizzaro. The Good, the Bad, the Difficult, and the Easy: Something Wrong with Information Retrieval Evaluation? In *ECIR '08: Proceedings of the 30th European Conference on IR Research on Advances in Information Retrieval*, pages 642–646, 2008.

[109] Stefano Mizzaro and Stephen Robertson. Hits hits TREC: exploring IR evaluation results with network analysis. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, pages 479–486, 2007.

[110] Mark Montague and Javed A. Aslam. Relevance score normalization for metasearch. In *CIKM '01: Proceedings of the tenth international conference on information and knowledge management*, pages 427–433, 2001.

[111] Josiane Mothe and Ludovic Tanguy. Linguistic features to predict query difficulty - a case study on previous TREC campaigns. In *ACM SIGIR '05 Workshop: Predicting Query Difficulty - Methods and Applications*, 2005.

[112] Jerome L. Myers and Arnold D. Well. *Research Design and Statistical Analysis (2nd edition)*. Lawrence Erlbaum, 2002.

[113] Alexandros Ntoulas, Junghoo Cho, and Christopher Olston. What's new on the web?: the evolution of the web from a search engine perspective. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 1–12, 2004.

[114] Rabia Nuray and Fazli Can. Automatic ranking of information retrieval systems using data fusion. *Information Processing and Management*, 42(3):595–614, 2006.

[115] Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. Overview of the TREC-2006 Blog Track. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.

[116] Siddharth Patwardhan and Ted Pedersen. Using WordNet Based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together*, pages 1–8, 2006.

[117] Jie Peng, Ben He, and Iadh Ounis. Predicting the Usefulness of Collection Enrichment for Enterprise Search. In *ICTIR '09: Proceedings of the 2nd International Conference on Theory of Information Retrieval*, pages 366–370, 2009.

[118] Joaquin Perez-Iglesias and Lourdes Araujo. Ranking List Dispersion as a Query Performance Predictor. In *ICTIR '09: Proceedings of the 2nd International Conference on Theory of Information Retrieval*, pages 371–374, 2009.

[119] Nina Phan, Peter Bailey, and Ross Wilkinson. Understanding the relationship of information need specificity to search query length. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, pages 709–710, 2007.

[120] Vassilis Plachouras, Ben He, and Iadh Ounis. University of Glasgow at TREC2004: Experiments in Web, Robust and Terabyte tracks with Terrier. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004)*, 2004.

[121] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval*, pages 275–281, 1998.

[122] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[123] Guang Qiu, Kangmiao Liu, Jiajun Bu, Chun Chen, and Zhiming Kang. Quantify query ambiguity using ODP metadata. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, pages 697–698, 2007.

[124] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30, 1989.

[125] Stephen E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, 1999.

[126] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[127] Mark Sanderson and Justin Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval*, pages 162–169, 2005.

[128] F. Scholer, H.E. Williams, and A. Turpin. Query association surrogates for web search. *Journal of the American Society for Information Science and Technology*, 55(7):637–650, 2004.

[129] Mark R. Segal, Kam D. Dahlquist, and Bruce R. Conklin. Regression approaches for microarray data analysis. *Journal of Computational Biology*, 10(6):961–980, 2003.

[130] Anna Shtok, Oren Kurland, and David Carmel. Predicting Query Performance by Query-Drift Estimation. In *ICTIR '09: Proceedings of the 2nd International Conference on Theory of Information Retrieval*, pages 305–312, 2009.

[131] Amit Singhal, Gerard Salton, and Chris Buckley. Length Normalization in Degraded Text Collections. In *Proceedings of Fifth Annual Symposium on Document Analysis and Information Retrieval*, pages 15–17, 1995.

[132] Ian Soboroff. Does WT10g look like the web? In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, pages 423–424, 2002.

[133] Ian Soboroff, Charles Nicholas, and Patrick Cahan. Ranking retrieval systems without relevance judgments. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, pages 66–73, 2001.

[134] Anselm Spoerri. How the overlap between the search results of different retrieval systems correlates with document relevance. *Proceedings of the American Society for Information Science and Technology*, 42(1), 2005.

[135] Anselm Spoerri. Using the structure of overlap between search results to rank retrieval systems without relevance judgments. *Information Processing and Management*, 43(4):1059–1070, 2007.

[136] Masao Takaku, Keizo Oyama, and Akiko Aizawa. An Analysis on Topic Features and Difficulties Based on Web Navigational Retrieval Experiments. In *AIRS 2006*, pages 625–632, 2006.

[137] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval*, pages 163–170, 2008.

[138] Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[139] S. Tomlinson. Robust, Web and Terabyte Retrieval with Hummingbird SearchServer at TREC 2004. In *Proceedings of the 13th Text REtrieval Conference (TREC 2004)*, 2004.

[140] Stephen Tomlinson. Experiments with the Negotiated Boolean Queries of the TREC 2007 Legal Discovery Track. In *Proceedings of the The Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.

[141] Stephen Tomlinson, Douglas W. Oard, Jason R. Baron, and Paul Thompson. Overview of the TREC 2007 Legal Track. In *Proceedings of the The Sixteenth Text REtrieval Conference (TREC 2007)*, 2007.

[142] John Wilder Tukey. *Exploratory Data Analysis*, chapter Box-and-Whisker Plots. Reading, MA: Addison-Wesley, 1977.

[143] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, Second Edition, 1979.

[144] Vladimir Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[145] Vishwa Vinay, Ingemar J. Cox, Natasa Milic-Frayling, and Ken Wood. On ranking the effectiveness of searches. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval*, pages 398–404, 2006.

[146] Vishwa Vinay, Natasa Milic-Frayling, and Ingmar Cox. Estimating retrieval effectiveness using rank distributions. In *CIKM '08: Proceeding of the 17th ACM conference on information and knowledge management*, pages 1425–1426, 2008.

[147] Ellen Voorhees. Overview of the TREC 2001 Question Answering Track. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, 2001.

[148] Ellen Voorhees and Donna Harman. Overview of the Sixth Text REtrieval Conference (TREC-6). In *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*, 1997.

[149] Ellen M. Voorhees. Variations in Relevance Judgments and the Measurement of Retrieval Effectiveness. *Information Processing and Management*, 36:697 – 716, 2000.

[150] Ellen M. Voorhees. Overview of the TREC 2002. In *Proceedings of the Eleventh Text Retrieval Conference (TREC 2002)*, 2002.

[151] E.M. Voorhees. Overview of the TREC 2003 Robust Retrieval Track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, 2003.

[152] E.M. Voorhees. Overview of the TREC 2004 Robust Track. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, 2004.

[153] Huyen-Trang Vu and Patrick Gallinari. A machine learning based approach to evaluating retrieval systems. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 399–406, 2006.

[154] Xuanhui Wang, Hui Fang, and ChengXiang Zhai. A study of methods for negative relevance feedback. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval*, pages 219–226, 2008.

[155] Harald Weinreich, Hartmut Obendorf, Eelco Herder, and Matthias Mayer. Not quite the average: An empirical study of Web use. *ACM Transactions on the Web*, 2(1): 1–31, 2008.

[156] Ryen W. White and Steven M. Drucker. Investigating behavioral variability in web search. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 21–30, 2007.

[157] Ryen W. White, Matthew Richardson, Mikhail Bilenko, and Allison P. Heath. Enhancing web search by promoting multiple search engine use. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval*, pages 43–50, 2008.

[158] E.J. Williams. Significance of difference between two non-independent correlation coefficients. *Biometrics*, 15:135–136, 1959.

[159] Mattan Winaver, Oren Kurland, and Carmel Domshlak. Towards robust query expansion: model selection in the language modeling framework. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, pages 729–730, 2007.

[160] Shengli Wu and Fabio Crestani. Data fusion with estimated weights. In *CIKM '02: Proceedings of the eleventh international conference on information and knowledge management*, pages 648–651, 2002.

[161] Shengli Wu and Fabio Crestani. Methods for ranking information retrieval systems without relevance judgments. In *SAC '03: Proceedings of the 2003 ACM symposium on applied computing*, pages 811–816, 2003.

[162] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval*, pages 4–11, 1996.

[163] Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112, 2000.

[164] Jinxi Xu, Alexander Fraser, and Ralph Weischedel. TREC 2001 Cross-lingual Retrieval at BBN. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, 2001.

[165] Kiduk Yang, Ning Yu, Alejandro Valerio, and Hui Zhang. WIDIT in TREC-2006 Blog track. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.

[166] Emine Yilmaz, Javed A. Aslam, and Stephen Robertson. A new rank correlation coefficient for information retrieval. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval*, pages 587–594, 2008.

[167] Elad Yom-Tov, Shai Fine, David Carmel, and Adam Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval*, pages 512–519, 2005.

[168] Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 11–18, 2003.

[169] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on information and knowledge management*, pages 403–410, 2001.

[170] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to Ad Hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, pages 334–342, 2001.

[171] ChengXiang Zhai and John Lafferty. Two-stage language models for information retrieval. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, pages 49–56, 2002.

[172] Guoqiang Zhang, B. Eddy Patuwo, and Michael Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1): 35–62, 1998.

[173] Min Zhang, Ruihua Song, Chuan Lin, Shaoping Ma, Zhe Jiang, Yijiang Jin, Yiqun Liu, and Le Zhao. THU TREC2002 Web Track Experiments. In *Proceedings of the Eleventh Text Retrieval Conference (TREC 2002)*, 2002.

[174] Ying Zhao, Falk Scholer, and Yohannes Tsegay. Effective Pre-retrieval Query Performance Prediction Using Similarity and Variability Evidence. In *ECIR '08: Proceedings of the 30th European Conference on IR Research on Advances in Information Retrieval*, pages 52–64, 2008.

[175] Yun Zhou. Measuring Ranked List Robustness for Query Performance Prediction. *Knowledge and Information Systems*, 16(2):155–171, 2007.

[176] Yun Zhou and W. Bruce Croft. Ranking robustness: a novel framework to predict query performance. In *CIKM '06: Proceedings of the 15th ACM international conference on information and knowledge management*, pages 567–574, 2006.

[177] Yun Zhou and W. Bruce Croft. Query performance prediction in web search environments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval*, pages 543–550, 2007.

[178] Ji Zhu and Trevor Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427–443, 2004.

[179] Hui Zou and Trevor Hastie. Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005.

# Abstract

In this thesis we consider users' attempts to express their information needs through queries, or search requests and try to predict whether those requests will be of high or low quality. Intuitively, a query's quality is determined by the outcome of the query, that is, whether the retrieved search results meet the user's expectations. The second type of prediction methods under investigation are those which attempt to predict the quality of search systems themselves. Given a number of search systems to consider, these methods estimate how well or how poorly the systems will perform in comparison to each other.

The motivation for this research effort stems primarily from the enormous benefits originating from successfully predicting the quality of a query or a system. Accurate predictions enable the employment of adaptive retrieval components which would have a considerable positive effect on the user experience. Furthermore, if we would achieve sufficiently accurate predictions of the quality of retrieval systems, the cost of evaluation would be significantly reduced.

In a first step, pre-retrieval predictors are investigated, which predict a query's effectiveness before the retrieval step and are thus independent of the ranked list of results. Such predictors base their predictions solely on query terms, collection statistics and possibly external sources such as WordNet or Wikipedia. A total of twenty-two prediction algorithms are categorized and their quality is assessed on three different TREC test collections, including two large Web collections. A number of newly applied methods for combining various predictors are examined to obtain a better prediction of a query's effectiveness. In order to adequately and appropriately compare such techniques the current evaluation methodology is critically examined. It is shown that the standard evaluation measure, namely the linear correlation coefficient, can provide a misleading indication of performance. To address this issue, the current evaluation methodology is extended to include cross validation and statistical testing to determine significant differences.

Building on the analysis of pre-retrieval predictors, post-retrieval approaches are then investigated, which estimate a query's effectiveness on the basis of the retrieved results. The thesis focuses in particular on the Clarity Score approach and provides an analysis of its sensitivity towards different variables such as the collection, the query set and the retrieval approach. Adaptations to Clarity Score are introduced which improve the estimation accuracy of the original algorithm on most evaluated test collections.

The utility of query effectiveness prediction methods is commonly evaluated by reporting correlation coefficients, such as Kendall's Tau and the linear correlation coefficient, which denote how well the methods perform at predicting the retrieval effectiveness of a set of

queries. Despite the significant amount of research dedicated to this important stage in the retrieval process, the following question has remained unexplored: what is the relationship of the current evaluation methodology for query effectiveness prediction and the change in effectiveness of retrieval systems that employ a predictor? We investigate this question with a large scale study for which predictors of arbitrary accuracy are generated in order to examine how the strength of their observed Kendall's Tau coefficient affects the retrieval effectiveness in two adaptive system settings: selective query expansion and meta-search. It is shown that the accuracy of currently existing query effectiveness prediction methods is not yet high enough to lead to consistent positive changes in retrieval performance in these particular settings.

The last part of the thesis is concerned with the task of estimating the ranking of retrieval systems according to their retrieval effectiveness without relying on costly relevance judgments. Five different system ranking estimation approaches are evaluated on a wide range of data sets which cover a variety of retrieval tasks and a variety of test collections. The issue that has long prevented this line of automatic evaluation to be used in practice is the severe mis-ranking of the best systems. In the experiments reported in this work, however, we show this not to be an inherent problem of system ranking estimation approaches, it is rather data set dependent. Under certain conditions it is indeed possible to automatically identify the best systems correctly. Furthermore, our analysis reveals that the estimated ranking of systems is not equally accurate for all topics of a topic set, which motivates the investigation of relying on topic subsets to improve the accuracy of the estimate. A study to this effect indicates the validity of the approach.

# SIKS Dissertation Series

Since 1998, all dissertations written by Ph.D. students who have conducted their research under auspices of a senior research fellow of the SIKS research school are published in the SIKS Dissertation Series.

**2010-05** Claudia Hauff (UT), *Predicting the Effectiveness of Queries and Retrieval Systems*
**2010-04** Olga Kulyk (UT), *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*
**2010-03** Joost Geurts (CWI), *A Document Engineering Model and Processing Framework for Multimedia documents*
**2010-02** Ingo Wassink (UT), *Work flows in Life Science*
**2010-01** Matthijs van Leeuwen (UU), *Patterns that Matter*
**2009-46** Loredana Afanasiev (UvA), *Querying XML: Benchmarks and Recursion*
**2009-45** Jilles Vreeken (UU), *Making Pattern Mining Useful*
**2009-44** Roberto Santana Tapia (UT), *Assessing Business-IT Alignment in Networked Organizations*
**2009-43** Virginia Nunes Leal Franqueira (UT), *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*
**2009-42** Toine Bogers (UvT), *Recommender Systems for Social Bookmarking*
**2009-41** Igor Berezhnyy (UvT), *Digital Analysis of Paintings*
**2009-40** Stephan Raaijmakers (UvT), *Multinomial Language Learning: Investigations into the Geometry of Language*
**2009-39** Christian Stahl (TUE, Humboldt-Universitaet zu Berlin), *Service Substitution – A Behavioral Approach Based on Petri Nets*
**2009-38** Riina Vuorikari (OU), *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*
**2009-37** Hendrik Drachsler (OUN), *Navigation Support for Learners in Informal Learning Networks*
**2009-36** Marco Kalz (OUN), *Placement Support for Learners in Learning Networks*
**2009-35** Wouter Koelewijn (UL), *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*
**2009-34** Inge van de Weerd (UU), *Advancing in Software Product Management: An Incremental Method Engineering Approach*
**2009-33** Khiet Truong (UT), *How Does Real Affect Affect Affect Recognition In Speech?*
**2009-32** Rik Farenhorst (VU) and Remco de Boer (VU), *Architectural Knowledge Management: Supporting Architects and Auditors*
**2009-31** Sofiya Katrenko (UVA), *A Closer Look at Learning Relations from Text*
**2009-30** Marcin Zukowski (CWI), *Balancing vectorized query execution with bandwidth-optimized storage*
**2009-29** Stanislav Pokraev (UT), *Model-Driven Semantic Integration of Service-Oriented Applications*
**2009-27** Christian Glahn (OU), *Contextual Support of social Engagement and Reflection on the Web*
**2009-26** Fernando Koch (UU), *An Agent-Based Model for the Development of Intelligent Mobile Services*
**2009-25** Alex van Ballegooij (CWI), *RAM: Array Database Management through Relational Mapping*
**2009-24** Annerieke Heuvelink (VUA), *Cognitive Models for Training Simulations*
**2009-23** Peter Hofgesang (VU), *Modelling Web Usage in a Changing Environment*
**2009-22** Pavel Serdyukov (UT), *Search For Expertise: Going beyond direct evidence*
**2009-21** Stijn Vanderlooy (UM), *Ranking and Reliable Classification*
**2009-20** Bob van der Vecht (UU), *Adjustable Autonomy: Controling Influences on Decision Making*
**2009-19** Valentin Robu (CWI), *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*
**2009-18** Fabian Groffen (CWI), *Armada, An Evolving Database System*
**2009-17** Laurens van der Maaten (UvT), *Feature Extraction from Visual Data*
**2009-16** Fritz Reul (UvT), *New Architectures in Computer Chess*
**2009-15** Rinke Hoekstra (UVA), *Ontology Representation - Design Patterns and Ontologies that Make Sense*
**2009-14** Maksym Korotkiy (VU), *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*
**2009-13** Steven de Jong (UM), *Fairness in Multi-Agent Systems*
**2009-12** Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin), *Operating Guidelines for Services*
**2009-11** Alexander Boer (UVA), *Legal Theory, Sources of Law & the Semantic Web*
**2009-10** Jan Wielemaker (UVA), *Logic programming for knowledge-intensive interactive applications*
**2009-09** Benjamin Kanagwa (RUN), *Design, Discovery and Construction of Service-oriented Systems*
**2009-08** Volker Nannen (VU), *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*
**2009-07** Ronald Poppe (UT), *Discriminative Vision-Based Recovery and Recognition of Human Motion*
**2009-06** Muhammad Subianto (UU), *Understanding Classification*
**2009-05** Sietse Overbeek (RUN), *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*

**2009-04** Josephine Nabukenya (RUN), *Improving the Quality of Organisational Policy Making using Collaboration Engineering*
**2009-03** Hans Stol (UvT), *A Framework for Evidence-based Policy Making Using IT*
**2009-02** Willem Robert van Hage (VU), *Evaluating Ontology-Alignment Techniques*
**2009-01** Rasa Jurgelenaite (RUN), *Symmetric Causal Independence Models*
**2008-35** Ben Torben Nielsen (UvT), *Dendritic morphologies: function shapes structure*
**2008-34** Jeroen de Knijf (UU), *Studies in Frequent Tree Mining*
**2008-33** Frank Terpstra (UVA), *Scientific Workflow Design; theoretical and practical issues*
**2008-32** Trung H. Bui (UT), *Toward Affective Dialogue Management using Partially Observable Markov Decision Processes*
**2008-31** Loes Braun (UM), *Pro-Active Medical Information Retrieval*
**2008-30** Wouter van Atteveldt (VU), *Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content*
**2008-29** Dennis Reidsma (UT), *Annotations and Subjective Machines – Of Annotators, Embodied Agents, Users, and Other Humans*
**2008-28** Ildiko Flesch (RUN), *On the Use of Independence Relations in Bayesian Networks*
**2008-27** Hubert Vogten (OU), *Design and Implementation Strategies for IMS Learning Design*
**2008-26** Marijn Huijbregts (UT), *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*
**2008-25** Geert Jonker (UU), *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*
**2008-24** Zharko Aleksovski (VU), *Using background knowledge in ontology matching*
**2008-23** Stefan Visscher (UU), *Bayesian network models for the management of ventilator-associated pneumonia*
**2008-22** Henk Koning (UU), *Communication of IT-Architecture*
**2008-21** Krisztian Balog (UVA), *People Search in the Enterprise*
**2008-20** Rex Arendsen (UVA), *Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven.*
**2008-19** Henning Rode (UT), *From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search*
**2008-18** Guido de Croon (UM), *Adaptive Active Vision*
**2008-17** Martin Op 't Land (TUD), *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*
**2008-16** Henriëtte van Vugt (VU), *Embodied agents from a user's perspective*
**2008-15** Martijn van Otterlo (UT), *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.*
**2008-14** Arthur van Bunningen (UT), *Context-Aware Querying; Better Answers with Less Effort*
**2008-13** Caterina Carraciolo (UVA), *Topic Driven Access to Scientific Handbooks*
**2008-12** József Farkas (RUN), *A Semiotically Oriented Cognitive Model of Knowledge Representation*
**2008-11** Vera Kartseva (VU), *Designing Controls for Network Organizations: A Value-Based Approach*
**2008-10** Wauter Bosma (UT), *Discourse oriented summarization*
**2008-09** Christof van Nimwegen (UU), *The paradox of the guided user: assistance can be counter-effective*
**2008-08** Janneke Bolt (UU), *Bayesian Networks: Aspects of Approximate Inference*
**2008-07** Peter van Rosmalen (OU), *Supporting the tutor in the design and support of adaptive e-learning*
**2008-06** Arjen Hommersom (RUN), *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*
**2008-05** Bela Mutschler (UT), *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*
**2008-04** Ander de Keijzer (UT), *Management of Uncertain Data – towards unattended integration*
**2008-03** Vera Hollink (UVA), *Optimizing hierarchical menus: a usage-based approach*
**2008-02** Alexei Sharpanskykh (VU), *On Computer-Aided Methods for Modeling and Analysis of Organizations*
**2008-01** Katalin Boer-Sorbán (EUR), *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*
**2007-25** Joost Schalken (VU), *Empirical Investigations in Software Process Improvement*
**2007-24** Georgina Ramírez Camps (CWI), *Structural Features in XML Retrieval*
**2007-23** Peter Barna (TUE), *Specification of Application Logic in Web Information Systems*
**2007-22** Zlatko Zlatev (UT), *Goal-oriented design of value and process models from patterns*
**2007-21** Karianne Vermaas (UU), *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*
**2007-20** Slinger Jansen (UU), *Customer Configuration Updating in a Software Supply Network*
**2007-19** David Levy (UM), *Intimate relationships with artificial partners*
**2007-18** Bart Orriëns (UvT), *On the development an management of adaptive business collaborations*
**2007-17** Theodore Charitos (UU), *Reasoning with Dynamic Networks in Practice*
**2007-16** Davide Grossi (UU), *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*
**2007-15** Joyca Lacroix (UM), *NIM: a Situated Computational Memory Model*
**2007-14** Niek Bergboer (UM), *Context-Based Image Analysis*
**2007-13** Rutger Rienks (UT), *Meetings in Smart Environments; Implications of Progressing Technology*
**2007-12** Marcel van Gerven (RUN), *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*
**2007-11** Natalia Stash (TUE), *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*
**2007-10** Huib Aldewereld (UU), *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*
**2007-09** David Mobach (VU), *Agent-Based Mediated Service Negotiation*
**2007-08** Mark Hoogendoorn (VU), *Modeling of Change in Multi-Agent Organizations*
**2007-07** Nataša Jovanović (UT), *To Whom It May Concern – Addressee Identification in Face-to-Face Meetings*
**2007-06** Gilad Mishne (UVA), *Applied Text Analytics for Blogs*
**2007-05** Bart Schermer (UL), *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*

**2007-04** Jurriaan van Diggelen (UU), *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach*
**2007-03** Peter Mika (VU), *Social Networks and the Semantic Web*
**2007-02** Wouter Teepe (RUG), *Reconciling Information Exchange and Confidentiality: A Formal Approach*
**2007-01** Kees Leune (UvT), *Access Control and Service-Oriented Architectures*
**2006-28** Börkur Sigurbjörnsson (UVA), *Focused Information Access using XML Element Retrieval*
**2006-27** Stefano Bocconi (CWI), *Vox Populi: generating video documentaries from semantically annotated media repositories*
**2006-26** Vojkan Mihajlović (UT), *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
**2006-25** Madalina Drugan (UU), *Conditional log-likelihood MDL and Evolutionary MCMC*
**2006-24** Laura Hollink (VU), *Semantic Annotation for Retrieval of Visual Resources*
**2006-23** Ion Juvina (UU), *Development of Cognitive Model for Navigating on the Web*
**2006-22** Paul de Vrieze (RUN), *Fundaments of Adaptive Personalisation*
**2006-21** Bas van Gils (RUN), *Aptness on the Web*
**2006-20** Marina Velikova (UvT), *Monotone models for prediction in data mining*
**2006-19** Birna van Riemsdijk (UU), *Cognitive Agent Programming: A Semantic Approach*
**2006-18** Valentin Zhizhkun (UVA), *Graph transformation for Natural Language Processing*
**2006-17** Stacey Nagata (UU), *User Assistance for Multitasking with Interruptions on a Mobile Device*
**2006-16** Carsten Riggelsen (UU), *Approximation Methods for Efficient Learning of Bayesian Networks*
**2006-15** Rainer Malik (UU), *CONAN: Text Mining in the Biomedical Domain*
**2006-14** Johan Hoorn (VU), *Software Requirements: Update, Upgrade, Redesign – towards a Theory of Requirements Change*
**2006-13** Henk-Jan Lebbink (UU), *Dialogue and Decision Games for Information Exchanging Agents*
**2006-12** Bert Bongers (VU), *Interactivation – Towards an ecology of people, our technological environment, and the arts*
**2006-11** Joeri van Ruth (UT), *Flattening Queries over Nested Data Types*
**2006-10** Ronny Siebes (VU), *Semantic Routing in Peer-to-Peer Systems*
**2006-09** Mohamed Wahdan (UM), *Automatic Formulation of the Auditor's Opinion*
**2006-08** Eelco Herder (UT), *Forward, Back and Home Again – Analyzing User Behavior on the Web*
**2006-07** Marko Smiljanic (UT), *XML schema matching – balancing efficiency and effectiveness by means of clustering*
**2006-06** Ziv Baida (VU), *Software-aided Service Bundling – Intelligent Methods & Tools for Graphical Service Modeling*
**2006-05** Cees Pierik (UU), *Validation Techniques for Object-Oriented Proof Outlines*
**2006-04** Marta Sabou (VU), *Building Web Service Ontologies*
**2006-03** Noor Christoph (UVA), *The role of metacognitive skills in learning to solve problems*
**2006-02** Cristina Chisalita (VU), *Contextual issues in the design and use of information technology in organizations*
**2006-01** Samuil Angelov (TUE), *Foundations of B2B Electronic Contracting*
**2005-21** Wijnand Derks (UT), *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*
**2005-20** Cristina Coteanu (UL), *Cyber Consumer Law, State of the Art and Perspectives*
**2005-19** Michel van Dartel (UM), *Situated Representation*
**2005-18** Danielle Sent (UU), *Test-selection strategies for probabilistic networks*
**2005-17** Boris Shishkov (TUD), *Software Specification Based on Re-usable Business Components*
**2005-16** Joris Graaumans (UU), *Usability of XML Query Languages*
**2005-15** Tibor Bosse (VU), *Analysis of the Dynamics of Cognitive Processes*
**2005-14** Borys Omelayenko (VU), *Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics*
**2005-13** Fred Hamburg (UL), *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*
**2005-12** Csaba Boer (EUR), *Distributed Simulation in Industry*
**2005-11** Elth Ogston (VU), *Agent Based Matchmaking and Clustering – A Decentralized Approach to Search*
**2005-10** Anders Bouwer (UVA), *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*
**2005-09** Jeen Broekstra (VU), *Storage, Querying and Inferencing for Semantic Web Languages*
**2005-08** Richard Vdovjak (TUE), *A Model-driven Approach for Building Distributed Ontology-based Web Applications*
**2005-07** Flavius Frasincar (TUE), *Hypermedia Presentation Generation for Semantic Web Information Systems*
**2005-06** Pieter Spronck (UM), *Adaptive Game AI*
**2005-05** Gabriel Infante-Lopez (UVA), *Two-Level Probabilistic Grammars for Natural Language Parsing*
**2005-04** Nirvana Meratnia (UT), *Towards Database Support for Moving Object data*
**2005-03** Franc Grootjen (RUN), *A Pragmatic Approach to the Conceptualisation of Language*
**2005-02** Erik van der Werf (UM)), *AI techniques for the game of Go*
**2005-01** Floor Verdenius (UVA), *Methodological Aspects of Designing Induction-Based Applications*
**2004-20** Madelon Evers (Nyenrode), *Learning from Design: facilitating multidisciplinary design teams*
**2004-19** Thijs Westerveld (UT), *Using generative probabilistic models for multimedia retrieval*
**2004-18** Vania Bessa Machado (UvA), *Supporting the Construction of Qualitative Knowledge Models*
**2004-17** Mark Winands (UM), *Informed Search in Complex Games*
**2004-16** Federico Divina (VU), *Hybrid Genetic Relational Search for Inductive Learning*
**2004-15** Arno Knobbe (UU), *Multi-Relational Data Mining*
**2004-14** Paul Harrenstein (UU), *Logic in Conflict. Logical Explorations in Strategic Equilibrium*
**2004-13** Wojciech Jamroga (UT), *Using Multiple Models of Reality: On Agents who Know how to Play*
**2004-12** The Duy Bui (UT), *Creating emotions and facial expressions for embodied agents*
**2004-11** Michel Klein (VU), *Change Management for Distributed Ontologies*
**2004-10** Suzanne Kabel (UVA), *Knowledge-rich indexing of learning-objects*
**2004-09** Martin Caminada (VU), *For the Sake of the Argument; explorations into argument-based reasoning*
**2004-08** Joop Verbeek (UM), *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise*

**2004-07** Elise Boltjes (UM), *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*
**2004-06** Bart-Jan Hommes (TUD), *The Evaluation of Business Process Modeling Techniques*
**2004-05** Viara Popova (EUR), *Knowledge discovery and monotonicity*
**2004-04** Chris van Aart (UVA), *Organizational Principles for Multi-Agent Architectures*
**2004-03** Perry Groot (VU), *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*
**2004-02** Lai Xu (UvT), *Monitoring Multi-party Contracts for E-business*
**2004-01** Virginia Dignum (UU), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*
**2003-18** Levente Kocsis (UM), *Learning Search Decisions*
**2003-17** David Jansen (UT), *Extensions of Statecharts with Probability, Time, and Stochastic Timing*
**2003-16** Menzo Windhouwer (CWI), *Feature Grammar Systems – Incremental Maintenance of Indexes to Digital Media Warehouses*
**2003-15** Mathijs de Weerdt (TUD), *Plan Merging in Multi-Agent Systems*
**2003-14** Stijn Hoppenbrouwers (KUN), *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*
**2003-13** Jeroen Donkers (UM), *Nosce Hostem – Searching with Opponent Models*
**2003-12** Roeland Ordelman (UT), *Dutch speech recognition in multimedia information retrieval*
**2003-11** Simon Keizer (UT), *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*
**2003-10** Andreas Lincke (UvT), *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*
**2003-09** Rens Kortmann (UM), *The resolution of visually guided behaviour*
**2003-08** Yongping Ran (UM), *Repair Based Scheduling*
**2003-07** Machiel Jansen (UvA), *Formal Explorations of Knowledge Intensive Tasks*
**2003-06** Boris van Schooten (UT), *Development and specification of virtual environments*
**2003-05** Jos Lehmann (UVA), *Causation in Artificial Intelligence and Law – A modelling approach*
**2003-04** Milan Petković (UT), *Content-Based Video Retrieval Supported by Database Technology*
**2003-03** Martijn Schuemie (TUD), *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*
**2003-02** Jan Broersen (VU), *Modal Action Logics for Reasoning About Reactive Systems*
**2003-01** Heiner Stuckenschmidt (VU), *Ontology-Based Information Sharing in Weakly Structured Environments*
**2002-17** Stefan Manegold (UVA), *Understanding, Modeling, and Improving Main-Memory Database Performance*
**2002-16** Pieter van Langen (VU), *The Anatomy of Design: Foundations, Models and Applications*
**2002-15** Rik Eshuis (UT), *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*
**2002-14** Wieke de Vries (UU), *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*
**2002-13** Hongjing Wu (TUE), *A Reference Architecture for Adaptive Hypermedia Applications*
**2002-12** Albrecht Schmidt (Uva), *Processing XML in Database Systems*
**2002-11** Wouter C.A. Wijngaards (VU), *Agent Based Modelling of Dynamics: Biological and Organisational Applications*
**2002-10** Brian Sheppard (UM), *Towards Perfect Play of Scrabble*
**2002-09** Willem-Jan van den Heuvel (KUB), *Integrating Modern Business Applications with Objectified Legacy Systems*
**2002-08** Jaap Gordijn (VU), *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*
**2002-07** Peter Boncz (CWI), *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*
**2002-06** Laurens Mommers (UL), *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*
**2002-05** Radu Serban (VU), *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*
**2002-04** Juan Roberto Castelo Valdueza (UU), *The Discrete Acyclic Digraph Markov Model in Data Mining*
**2002-03** Henk Ernst Blok (UT), *Database Optimization Aspects for Information Retrieval*
**2002-02** Roelof van Zwol (UT), *Modelling and searching web-based document collections*
**2002-01** Nico Lassing (VU), *Architecture-Level Modifiability Analysis*
**2001-11** Tom M. van Engers (VUA), *Knowledge Management: The Role of Mental Models in Business Systems Design*
**2001-10** Maarten Sierhuis (UvA), *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*
**2001-09** Pieter Jan 't Hoen (RUL), *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*
**2001-08** Pascal van Eck (VU), *A Compositional Semantic Structure for Multi-Agent Systems Dynamics.*
**2001-07** Bastiaan Schonhage (VU), *Diva: Architectural Perspectives on Information Visualization*
**2001-06** Martijn van Welie (VU), *Task-based User Interface Design*
**2001-05** Jacco van Ossenbruggen (VU), *Processing Structured Hypermedia: A Matter of Style*
**2001-04** Evgueni Smirnov (UM), *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*
**2001-03** Maarten van Someren (UvA), *Learning as problem solving*
**2001-02** Koen Hindriks (UU), *Agent Programming Languages: Programming with Mental Models*
**2001-01** Silja Renooij (UU), *Qualitative Approaches to Quantifying Probabilistic Networks*
**2000-11** Jonas Karlsson (CWI), *Scalable Distributed Data Structures for Database Management*
**2000-10** Niels Nes (CWI), *Image Database Management System Design Considerations, Algorithms and Architecture*
**2000-09** Florian Waas (CWI), *Principles of Probabilistic Query Optimization*
**2000-08** Veerle Coupé (EUR), *Sensitivity Analyis of Decision-Theoretic Networks*
**2000-07** Niels Peek (UU), *Decision-theoretic Planning of Clinical Patient Management*
**2000-06** Rogier van Eijk (UU), *Programming Languages for Agent Communication*
**2000-05** Ruud van der Pol (UM), *Knowledge-based Query Formulation in Information Retrieval.*
**2000-04** Geert de Haan (VU), *ETAG, A Formal Model of Competence Knowledge for User Interface Design*
**2000-03** Carolien M.T. Metselaar (UVA), *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.*
**2000-02** Koen Holtman (TUE), *Prototyping of CMS Storage Management*
**2000-01** Frank Niessink (VU), *Perspectives on Improving Software Maintenance*

**1999-08** Jacques H.J. Lenting (UM), *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.*
**1999-07** David Spelt (UT), *Verification support for object database design*
**1999-06** Niek J.E. Wijngaards (VU), *Re-design of compositional systems*
**1999-05** Aldo de Moor (KUB), *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*
**1999-04** Jacques Penders (UM), *The practical Art of Moving Physical Objects*
**1999-03** Don Beal (UM), *The Nature of Minimax Search*
**1999-02** Rob Potharst (EUR), *Classification using decision trees and neural nets*
**1999-01** Mark Sloof (VU), *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*
**1998-05** E.W. Oskamp (RUL), *Computerondersteuning bij Straftoemeting*
**1998-04** Dennis Breuker (UM), *Memory versus Search in Games*
**1998-03** Ans Steuten (TUD), *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*
**1998-02** Floris Wiesman (UM), *Information Retrieval by Graphically Browsing Meta-Information*
**1998-01** Johan van den Akker (CWI), *DEGAS – An Active, Temporal Database of Autonomous Objects*