THE UNIVERSITY OF MELBOURNE

DOCTORAL THESIS

Improving the Reliability and Robustness of Information Retrieval Evaluation

Author:
Ziying Yang
ORCID: 0000–0001–7705–3280

Supervisors:
Prof. Alistair Moffat
Prof. Andrew Turpin

Doctor of Philosophy

May 2019

School of Computing and Information Systems

A thesis submitted in total fulfilment of the requirements for the degree of Doctor of Philosophy

Abstract

Batch evaluation techniques are often used to measure and compare the performance of Information Retrieval (IR) systems. In these approaches, IR evaluation metrics score the systems' runs against a set of ground-truth knowledge represented as relevance judgments for each one of a set of topics. Those system-topic scores are then compared, so that the superior system – if one exists – can be identified. Chapter 2 describes these processes in detail, including defining several commonly-used IR evaluation metrics, and introducing a range of associated techniques for collecting relevance judgments.

Chapter 3 considers what happens when the document-scoring model creates ties; that is, when the similarity scores assigned to documents by the IR system are the same. In particular, the role of tied similarity scores in past TREC experimentation is measured, and possible strategies for handling ties in IR evaluations are introduced. Tied similarity scores may be caused by score rounding within similarity metrics, usually undertaken for efficiency. In further experiments we suggest deliberately grouping documents as ties, to discover the extent to which similarity score rounding can be tolerated in a quest to allow faster query processing without greatly losing effectiveness.

Chapter 4 explores the potential risk to IR evaluation reproducibility that might result from the use of incomplete relevance judgments, focusing on estimating the reliability of IR system effectiveness scores when evaluated by recall-based metrics. Such effectiveness scores for metrics such as average precision (AP) and normalized discounted cumulative gain (NDCG) can be associated with corresponding parameter values for utility-based metrics such as rank-biased precision (RBP), for which residual scores can be computed and used to bound the uncertainty associated with unjudged documents. We found that while the uncertainty of recall-based metrics could be very high when the number of unjudged documents is large, in practical measurements the effect was less. Even so, we suggest that researchers should report uncertainties of system effectiveness scores via the residual of a weighted-precision metrics such as RBP, in addition to carrying out statistical tests for establishing metric score consistency.

The relevance judgments that form a foundation of all batch evaluations have conventionally been assessed by small numbers of trained experts using ordinal relevance scales with two or more relevance categories. Judgments collected by such scales often contain large numbers of ties: documents in the same category that cannot be separated by relevance. Collecting judgments on a scale with higher fidelity can help understand users' perceptions of relevance, and allow the functions used for mapping relevance categories to numeric gain scores to be refined. Chapter 5 considers these issues in detail, and proposes a judgment solicitation approach that requires pairwise forced-choice decisions to collect relevance judgments using three different scales: preference, absolute relevance, and relevance ratio, using a crowd-sourcing platform which provides a large number of non-specialist assessors. We investigate the variation of the normalized relevance judgments generated by answers associated with these three methods, and compare them

with three forms of previous judgments: NIST binary, Sormunen and Magnitude Estimation. We measure the number of assessed documents, average assessing speed, average document length, assessing inconsistency, accuracy and method preferences of workers, and consider which of those factors might affect the quality of relevance assessments.

Chapter 6 brings together these three related investigations, summarizes the findings of the thesis, and describes a range of avenues for possible future work.

Declaration

I, Ziying Yang, declare that this thesis titled, "Improving the Reliability and Robustness of Information Retrieval Evaluation" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.
- The thesis is fewer than the maximum word limit in length, exclusive of tables, maps, bibliographies and appendices as approved by the Research Higher Degrees Committee.

Credits

Portions of the material in this thesis have previously appeared in the following refereed publications:

- **Chapter 3**
  Ziying Yang, Alistair Moffat, Andrew Turpin. "How Precise Does Document Scoring Need to Be?" In: *Proc. Asia Information Retrieval Societies Conf. (AIRS)* 2016: 279-291. doi: 10.1007/978-3-319-48051-0_21

- **Chapter 4**
  Alistair Moffat, Falk Scholer, Ziying Yang. "Estimating Measurement Uncertainty for Information Retrieval Effectiveness Metrics". In: *J. Data and Information Quality* 10(3), 10:1-10:22 (2018). doi: 10.1145/3239572

- **Chapter 5**
  Ziying Yang. "Relevance Judgments: Preferences, Scores and Ties". In: *Proc. ACM Conf. on Research and Development in Information Retrieval (SIGIR)*. Doctoral Consortium Abstract. 2017: 1373. doi: 10.1145/3077136.3084154

  Ziying Yang, Alistair Moffat, Andrew Turpin. "Pairwise Crowd Judgments: Preference, Absolute, and Ratio". In: *Proc. Australasian Document Computing Symp. (ADCS)*. 2018: 3:1-3:8. doi: 10.1145/3291992.3291995

The relevance judgments obtained in Chapter 5 were collected with approval from the Human Ethics Sub-Committee of The University of Melbourne (ID number 1749112).

Acknowledgments

Firstly, I would particularly acknowledge my parents. Thank you for supporting me for such a long time, allowing me to embrace so many opportunities and challenges overseas. I have been trying live up to your expectation, and be the girl that you desired me to be, just like my name in Chinese, 杨孜颖, hardworking and talented. In Chinese: 感谢多年来父母亲对我的支持和培养，让我能站到高处拥有更广阔的眼界，寻找和实现梦想。很幸运生而为你们的女儿，希望能成为你们的骄傲，没有辜负我的名字，孜孜不倦，脱颖而出。

I would like to thank my supervisors, Professor Alistair Moffat and Professor Andrew Turpin, for providing invaluable guidance and support with my study plan, endless corrections for my thesis writing, and encouragement when I was crestfallen, or made slow progress. I have always been motivated when being recognized. Alistair, thank you for recommending me be a tutor for your subject during my PhD candidature when I was not confident about my English speaking. As you said, we need to face and focus on what we are not good at. This experience greatly improved my confidence and presentation skills in seminars and conferences. Thank you for pushing me outside my comfort zone, and rise to the challenging. A big thank you to my co-supervisor, Andrew Turpin, whose whereabouts were often a mystery, but even so, he was always there when he was needed. Thanks to both of you for providing novel and useful suggestions when I was in a dilemma struggling with research problems. Most of all, thank you for training me how to learn and to write.

I gratefully acknowledge my friends, Qingyu Chen, Unni Krishnan, Alfan Wicaksono, Xiaolu Lu and other colleagues in University of Melbourne and RMIT. I enjoyed the time when we discussed our research, supported each other, participated in seminars and conferences, and simply when walking together in the streets of Tokyo. I will always remember these great and cheerful memories.

Ziying Yang

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Information Retrieval

Searching for information on the Web has become the first choice for most people when looking for resources that can satisfy an information need, for example, answering a question, solving a problem, researching a holiday destination, and so on. Thirty years ago, people could only go to the library, check the catalogs, seek the possible useful documents independently, and search for the desired information using print resources such as books, newspapers and encyclopedias. Encyclopedia Britannica has in total of 32 volumes, $32,640$ pages, and costs about US$\$400$. Even if there is an article matching the information need, each of the subsequent year books needed to be checked to ensure that the found knowledge is still up to date. Similarly, twenty years ago, people looking for real estates could only read the information in the newspaper of the day in the morning, and then attend inspections. The validity of such information could not be guaranteed, and there might be no update at all. In the past, information searching consumed time and money, and sometimes the resources were updated slowly, or not at all.

However, information searching is now convenient, fast and effective, because of the huge advances in computing and the Internet. Today, people only need to type keywords (as queries) about their information need, press the "search" button, and the selected search engine will return a ranked list containing a set of possible related documents sorted in order of estimated relevance. All the computations and rankings are completed by the retrieval model of search engines within a few tenth of a second. When searching knowledge in encyclopedia, the website Wikipedia[1] has become the major source, replacing most encyclopedia books, for its free access, abundant and extensive content, fast updating, and coverage of multiple languages and cultures. Wikipedia has been compared with Encyclopedia Britannica, and shown to provide information with similar quality [34]. Nowadays, searching is seemingly almost free, and can happen anywhere and anytime with electronic equipment. The information that is effective in short period can also be updated in a few seconds.

[1] https://www.wikipedia.org

Why are the search engines so powerful? The credit should be given to a range of research in the area of Information Retrieval (IR) over the past thirty years. With achievements in IR studies, search engines have been developed for retrieving relevant documents for different kinds of tasks, and optimized for data at a large scale that would have been impossible to contemplate thirty years ago. Nor are user queries the only approach that is supported by the retrieval methods and algorithms. Other tasks such as filtering, recommending, classification are also included in the development of search engines. People need IR systems in various practical situations, such as desktop search, enterprise search and so on. This absolute reliance on IR then raises a key question: how to implement a good IR system, or improve its search performance? How can we measure the quality of IR systems? We need evaluation techniques.

1.2 IR Evaluation

IR evaluation is for comparing IR systems in terms of efficiency and effectiveness. In this thesis, we mainly study the effectiveness of IR systems; that is, their ability to retrieve the "right" information [25]. Starting with a user query, the compared systems estimate the relevance of documents to the query by computing a similarity score for each document, and then return a ranked list containing documents in decreasing similarity score order.

As the retrieval methods and algorithms used by systems are different, systems usually estimate the "relevance" in distinct ways. With a same user query, the ranked lists returned by systems are different most of time. To evaluate system effectiveness, we need to look into the retrieved documents (Web pages, in Web search) in the ranking, and judge their relevance to the given query, called *relevance judgments* in IR evaluation. And then, some established *effectiveness metrics* can be used to compute a score for each retrieved list, which represents the effectiveness of the system considered by the used metric. The system with better performance for the given query can therefore be discovered by comparing the metric scores.

For example, we search the query "phd graduate jobs australia" using Google[2] and Bing[3] respectively. The retrieved top results and the interface are shown in Figure 1.1. The results retrieved by Google are shown on the left-hand side, containing links and snippets of the most related five web pages. It states that there are about in total of $53{,}200{,}000$ results found, and returned using only 0.67 seconds. The ranked list of an advertisement and top five relevant web pages considered by Bing is shown on the right-hand side.

The quality of the top five ranks of these two search engines can be reviewed by examining the relevance of the retrieved Web pages using a Binary relevance scale (with 1 to represent relevant, and 0 to represent non-relevant). After clicking though the provided

[2]https://www.google.com.au
[3]https://www.bing.com

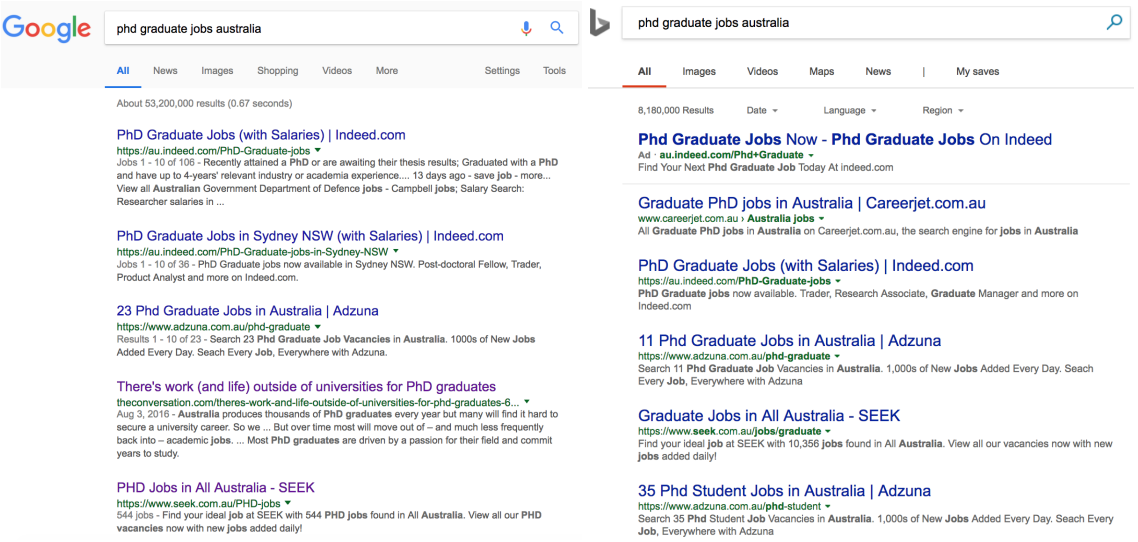


Figure 1.1: The interface and search results of search engines, Google and Bing, for query: *phd graduate jobs australia* (searched on 2018.4.20).

links and reading the contents, the relevance scores of the five Web pages returned by Google are (given here according to the opinion of the query creator, the thesis author):

$$1, 1, 1, 1, 0.$$

That is the first four Web pages are judged as relevant, and the fifth result is irrelevant, according to the judgments by the author. Similarly, the relevance of the top-5 Web page retrieved by Bing is:

$$1, 1, 1, 0, 1.$$

Note that ideally the relevance of documents to the information need is usually assessed by the query creator, but this might not always be possible.

With such relevance judgments, how to measure the effectiveness of the compared systems? As can be seen, both systems find four relevant Web pages at the first five ranks. If we only consider the proportion of relevant documents within the evaluation depth (that is, five, in this example), the two systems are equally good. However, if the rank of the relevant Web pages also matters, as Google found the fourth relevant Web page earlier than Bing, Google might be deemed as being more effective than Bing for this query.

In IR evaluation, the effectiveness of IR systems is quantified by effectiveness metrics. Metrics place different emphasis in different aspects, such as total number of relevant documents found, how soon the first relevant document is found, and the total relevance gained by the user if consider documents at top ranks more important than those in the bottom. Studying the user models, properties and assumptions of metrics is essential for selecting metrics for IR evaluations having users of distinct types, and tasks of different

complexity levels.

In the example above, only the results of the first five ranks are examined. But in practice, the returned ranked list could is arbitrarily long, potentially $53{,}000{,}000$ in the case of Google shown in Figure 1.1. It is impossible to assess the relevance of all the documents for the query. Current research has focused on analyzing user's action log data which is highly correlated with the relevance of the retrieved result, such as clickthrough, to evaluate systems. But most companies and research groups still prefer to evaluate systems using smaller *test collections* in laboratory experiments [25].

A test collection, such as Text REtrieval Conference (TREC)[4], typically consists of a collection of documents, a set of topics with descriptions of the information needs and sample queries, a set of relevance judgments for the topics. The relevance judgments are usually assessed by the topic creators, and they are also experts in the research area of IR. Even though the size of the document collection is much smaller than Web, it is still prohibitive to collect relevance judgments for all of the topic-document combinations. Thus, for each topic, a *pool* of top-k (k is usually 100 in the past TREC rounds) documents returned by the participated systems is built for collecting relevance judgments from human experts [111]. With the relevance judgments, a range of metrics can be used to compute effectiveness scores for systems, and therefore compare the quality of the participated systems. This evaluation process is known as *batch evaluation*.

In an experiment based on test collections, some participating systems may assign the same similarity scores for more than one document (called ties), that is, several documents are considered as equally relevant to the given query. As documents need to be sequentially presented in the ranked list, some strategies for ordering tied documents are employed by systems. However, ordering tied documents using different methods may affect system effectiveness to varying degrees. In **Chapter 3**, we explore how these ties were handle in IR evaluation, how reliable of evaluation results were using different tie handling strategies, and how ordering of the tied documents may affect the robustness of IR system comparison results.

As the pool does not contain all of the documents, documents outside the pool would not be judged by human experts. In the evaluation, they are usually treated as irrelevant documents by the metrics. Zobel [120] argued that there were only 50%–70% relevant documents had been found. Some other relevant documents were outside the pool, and had not been judged. If we increase the size of relevance judgments by pooling more documents, and find more relevant documents, the metric score of systems may change. We call the upper limit of the score change caused by unjudged documents the *uncertainty* of the metric scores.

Some metrics such as Rank Biased Precision (RBP) [73] can compute the uncertainty as a residual score that considers the ranks of all the unjudged documents in the evaluated ranked list. For some other metrics, such as Average Precision (AP), there is no

[4] http://trec.nist.gov

method to compute their uncertainties, or measure the reliability of them. Thus in **Chapter 4**, we propose a strategy to estimate these metrics using RBP, and measure their score uncertainties via RBP residual scores.

The described relevance judgments for the search results in Figure 1.1 were based on assessments by the thesis author. Other people may have different opinions about the relevance of the retrieved Web pages. Suppose that another user of the compared search engines considers the relevance score of the top-5 Web pages retrieved by Google as:

$$0, 1, 1, 0, 0.$$

And the relevance judgments for the results of Bing are:

$$1, 0, 0, 1, 1.$$

Now which system is better than the other?

The users of IR systems may have distinct backgrounds, ages, search purposes, and so on. Their opinions may also change with time. For one topic-document combination, some people may think the document is relevant to the topic, but others may not. Instead of Binary scale, if multiple-level relevance scales (such as 5-point, 7-point scales) are used, the agreement between users may further decrease. If we aim to build a set of relevance judgments agreed by general users, the opinions about the relevance of each topic-document pair need to be collected from a large number assessors, which is expensive and impractical.

Thus, in addition to the large scale, another challenge we face when collecting relevance judgments is: relevance is subjective, and people's perceptions of relevance can vary widely.

Relevance judgments were conventionally assessed by small numbers of trained experts using ordinal relevance scales with two or more relevance categories. Judgments collected by such scales often contain a lot of ties: documents in the same category cannot be separated by relevance. In the example described above, if we only focus on the top-3 ranks, using the relevance judgments of the author, as both of Google and Bing retrieve three documents with relevance score of 1, the effectiveness of these two systems will be deemed as equal by metrics.

To enhance the robustness of IR evaluation, a technique of collecting reliable and high fidelity relevance judgments without high costs is indispensable. In **Chapter 5**, we explore the use of pairwise scales to collect relevance judgments. In Figure 1.2, an example of comparing system X and system Y using conventional low fidelity judgments (on the left), and pairwise judgments (on the right) respectively are demonstrated. System X returns a ranked list of documents A, C, D, and B. System Y retrieved A, D, C, and B. The only difference between these two rankings is the ordering of documents C and D. But unfortunately, in the judgments with a small number of relevance levels shown on the

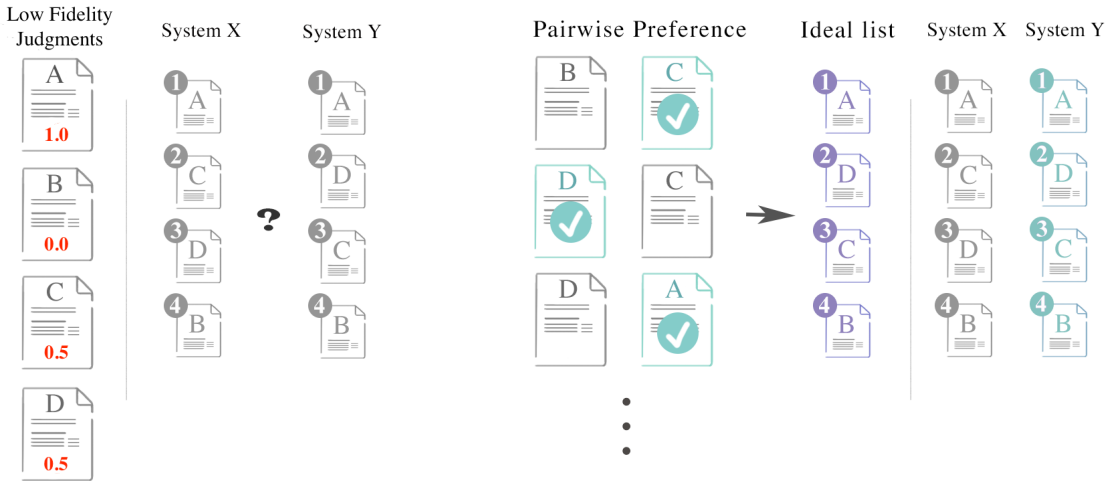


Figure 1.2: An example of using low fidelity relevance judgment and pairwise judgments to compare the effectiveness of the ranked lists returned by system X and system Y.

left, documents C and D receive the same relevance scores, 0.5. Thus we cannot separate systems X and Y using the judgments on the left. However, using the pairwise preference scale (one of the methods that we combine and propose to use in our work), as shown on the right-hand side of Figure 1.2, documents are paired, and assessors only need to select the document that is more relevant than the other in each pair. Based on the preference judgments, we can generate an ideal list (or an ordering) of documents sorted in decreasing preference order. As the relevance order between any two documents are known in preference judgments, system Y is therefore evaluated as more effective than system X.

The system discrimination of metrics, and the extent of tolerating similarity score ties are both related to the fidelity of the employed relevance judgments. Collecting judgments on a scale with higher fidelity can help for better understanding of user's perception of relevance, and relevance gain profiles might be refined.

Moreover, pairwise scales can be used to collect general opinions from different assessors. We design pairwise questions where a human judge must choose which is more relevant for each document pair. We collect relevance judgments on the crowd-sourcing platform which provide a large number of human workers. With quality control processes, we can obtain high level of judgments from crowd workers with cheap costs. We also investigate which factors of assessing speed, workload, topic difficulty, and assessing method, might affect the quality of relevance assessments, and conclude that which relevance scale has more advantages than other in which kind of judgment collection tasks.

1.3 Thesis Structure

Chapter 2 reviews commonly-used evaluation techniques for IR. The test collections and batch evaluation techniques are briefly described. The potential problems of currently widely-used relevance judgments are reviewed. The pooling strategies, well established IR evaluation metrics and their user models, and methods for collecting relevance judgments are introduced and compared. Some correlation measurements and data analysis models for IR evaluations are also described.

Chapter 3 explores the role of tied similarity scores in the past TREC runs, and possible strategies to handle them in IR evaluations. Tied similarity scores can be caused by score rounding within similarity metrics, usually undertaken for efficiency. We explain our proposed strategy that deliberately groups documents as ties, and discover to which extent the similarity score rounding can be tolerated without affecting system discriminations of IR metrics.

Chapter 4 describes the potential risk of IR evaluation on reproducibility, associated with the unjudged documents in relevance judgments. We propose a strategy to estimate the reliability of system effectiveness scores evaluated by recall-based metrics using RBP residuals. Then, we investigate the uncertainty of recall-based metrics when the pool depth is reduced, that is, the number of unjudged documents is large. We suggest researchers report uncertainties of system effectiveness scores, which could be computed by weighted-precision metrics (such as RBP), in addition to statistical test results for examining metric score consistency.

Chapter 5 describes a combined method to collect relevance judgments using three different pairwise scales: preference, absolute relevance and relevance ratio. We explain our experiment design on a crowd-sourcing platform that provides a large number of non-specialist assessors. We measure the variation of the normalized relevance judgments generated by answers associated with these three methods, and compare them with previous judgments: NIST Binary, Sormunen and Magnitude. Furthermore, we investigate the number of assessed documents, average assessing speed, average document length, assessing inconsistency, accuracy and method preferences of workers, and analyze which factors might affect the quality of relevance assessments.

Chapter 6 concludes the findings and contributions of the previous chapters, and recommends directions for future works.

Chapter 2

Background

Information Retrieval (IR) systems are usually designed to consider a *query* formed by a human user, interpret the query as being a user *information need*, search for topically related *documents* (in a text collection), and return a ranked list (or *run*) containing documents in decreasing estimated relevance order back to the user, as the process shown in Figure 2.1. The core component of IR systems are the employed *retrieval models* which are used to match queries and documents and compute a topical *similarity score* relative to the given query for each document. Documents in the returned run are usually sorted in decreasing similarity score order and displayed as the search result. The quality of the runs retrieved by different systems are often compared in batch evaluation processes (see Section 2.1.2). The effectiveness of runs, called *effectiveness scores*, are scored as numeric values by *IR effectiveness metrics* using *relevance judgments*. Relevance judgments record how the documents are related to the topics, and are typically determined by human experts using an ordinal relevance scale. However, in recent years, researchers proposed some new methods (described in Section 2.2) to collect relevance judgments on *Crowd-sourcing* platforms (see Section 2.2.5).

IR effectiveness metrics (a range of choices are available, detailed in Section 2.3) simulate different user behaviors and expectations for distinct kinds of tasks. In other words, metrics measure different aspects of system performance (described in Section 2.3.2). For comparing systems using their effectiveness scores, the run scores of each system could be aggregated into a single system score, or statistical tests for determining whether the compared two systems are significantly different or not, could be carried out. The commonly-used technologies for data analysis, as well as measurements employed by projects described in later chapters, are described in Section 2.4.

2.1 IR Evaluation Methodologies

Evaluation is a critical process for building better IR systems. The performance of a new system design can be examined by retrieval experiments to determine if it has measurable advantages compared to existing baseline systems. The general usability of a system could be evaluated in terms of effectiveness, efficiency, and satisfaction. In this thesis, we focus primarily on the effectiveness of IR systems. However, if effectiveness is defined

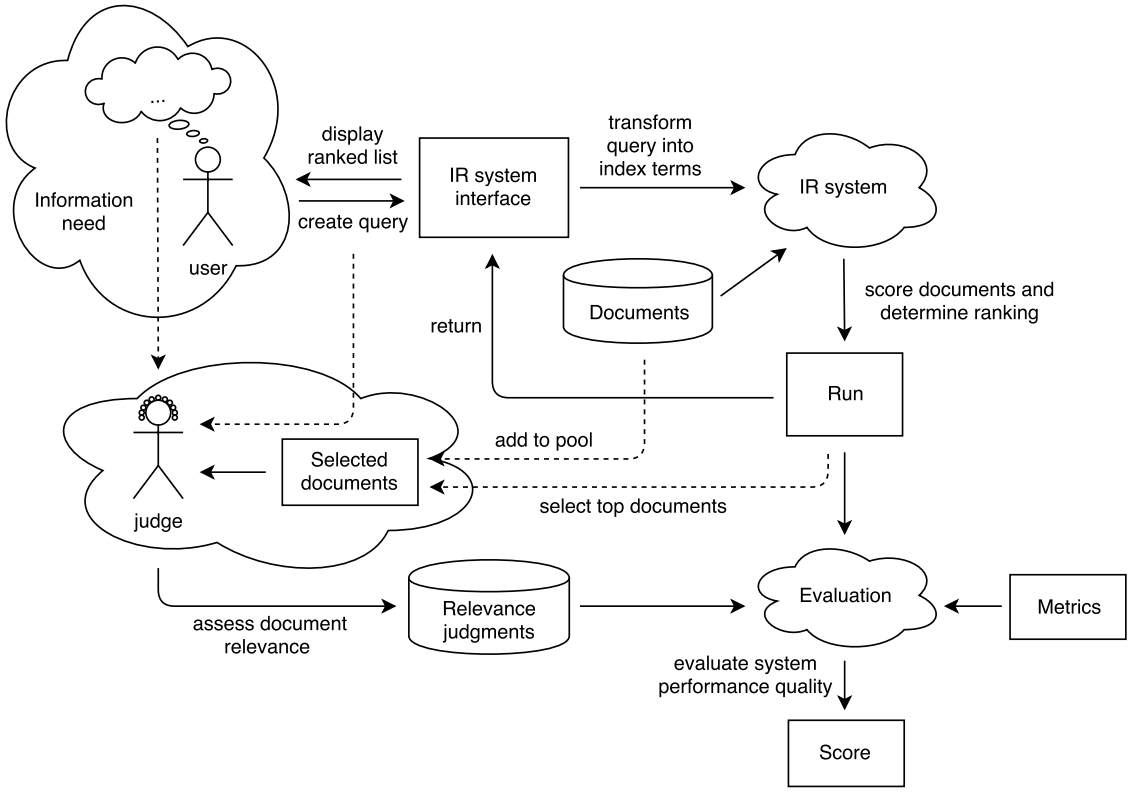


Figure 2.1: The work flow of Information Retrieval. The top part of the diagram shows the main working processes of IR systems, the bottom part describes the IR evaluation process.

as determining information utility that is gained by the IR system user, evaluating the effectiveness of IR systems becomes extremely difficult [101].

2.1.1 User-Based Evaluation

As the goal of IR evaluation is to measure how well IR systems meet the information needs of users, the most direct method of evaluation seems to be user-based, which samples human assessors from actual users of IR systems, and determines how satisfied they are with the results retrieved by the systems being measured. Compared to system-centered evaluation (such as Batch Evaluation Technique, described in the next section), user-based evaluation directly studies user interactions with systems, and also reflects user experience [49].

In user-based evaluation, the sampled users are provided a scenario that they are searching for information about the given topic (query) using the system that is to be evaluated. Documents retrieved by the system are sorted in a ranked list in decreasing estimated relevance order, and shown in the system interface. Users are required to examine the ranked list for each topic, including clicking through and reading, and provide

feedback on the extent to which they are satisfied with the results retrieved by the system. Note that, the user interface itself can be evaluated independently of the quality of documents returned, but that is outside the scope of this thesis.

The number of representative assessors involved in an user-based evaluation needs to be sufficiently large (but there is no exact answer to the question "how many participants are enough" [28]), and such experiments are usually expensive. There are also some requirements for this human-centered evaluation, such as each assessor needing to be equally trained on all systems; learning effects need to be carefully considered; the interfaces of all systems need to be well-designed and appropriate for assessors to use [101, 111]; the environment (for example, the lab in which the testing occurs) and the equipment used during the testing should be as similar as possible to the actual situation when users perform the search task [28]; and so on. Jiang and Allan [45] find that user's persistence when sequentially examining documents usually changes with the quality of the viewed results. If the retrieved results are off-topic, users might have low persistences to continue the examining to avoid wasting time. As topics are various in terms of difficulty, novelty, and other aspects, users persistences might not be the same when they examine results of distinct topics. Therefore, *Latin square design* that performs all the combinations of different elements (for example, the tested systems, and topics) in balance to ensure fairness is usually employed by user-based evaluation experiments, so that the variation caused by factors such as display ordering, topic difficulty, user's assessment experience built and so on in the experiment could be controlled.

In user-based evaluation, the effectiveness of systems could be measured via a variety of distinct ways, such as eye tracking, analyzing user's action logs, rating scores provided by users, time to completion, and some methods modified based on traditional measures.

Mao et al. [66] explore the relationship between relevance, usefulness, and user satisfaction in a user-based experiment. They discover that highly relevant documents in relevance judgments used in system-centered evaluations may not be highly useful for practical users, and that document usefulness is greatly correlated with user satisfaction feedback. Thus, the authors propose two methods for collecting usefulness labels in Web searches, and evaluate their reliability and validity. Chen et al. [21] then investigate how user satisfaction can be modeled by off-line and on-line metrics in different search environments, and propose to include hover information into "clicks" log analysis for a better estimation of user satisfaction when using on-line metrics to evaluate systems.

Fox et al. [33] exploit "clicks" as indicators of relevance in the search logs of users to obtain relevance judgments. However this kind of technology requires detecting and removing noise and bias from the data, which is related to studies of user interactions with IR systems, and not easy to develop [90]. There are two categories of bias defined by Joachims et al. [46]: *trust bias* – users are willing to trust the employed search engine, and tend to click documents at the top of the retrieved list even when some non-relevant documents are deliberately placed in top ranks; and *quality bias* – users tend to click less

relevant documents when the overall quality of the retrieved results is poor. Instead of extracting absolute relevance of documents from log data, Joachims et al. [46] propose to extract pairwise preference between documents (for example, when $\boldsymbol{d}_1$ is ranked before $\boldsymbol{d}_2$, if the user skips $\boldsymbol{d}_1$ and clicks $\boldsymbol{d}_2$, $\boldsymbol{d}_2$ will be deemed as more relevant than $\boldsymbol{d}_1$) to generate relevance judgments (relevance scales are discussed in details in Section 2.2.4).

Query logs can also be used to analyze and understand user behaviors during search tasks. Agichtein et al. [1] subtract trust bias and split the query log into a training set and a testing set. The system is trained using the training set, learning to predict which documents the user would click on for their submitted queries in the testing set. This technology can be used to determine the accuracy of user models, and to customize search results for individual users. Chapelle et al. [19] combine the click data from query logs and relevance judgments assessed by human experts to compare a range of proposed evaluation metrics (which are used to examine the performance of IR systems, described in Section 2.3).

As the scale of IR systems increases, online experimentation has been introduced and developed to deal with the challenges of scalability and complexity in IR system evaluations [39]. Online evaluation experimentations for IR, usually developed in research projects in interactive IR [48], focus on studying user iterations with IR systems, and can also be used to evaluate and tune IR systems directly based on the online performance of systems. A commonly-used technology in online experimentations is A/B testing, which is a controlled experiment comparing two versions of systems – the original version (also called control, or champion, or "A"), and the redesigned version (also called treatment, or challenger, or "B"). The two versions are shown to many different users at random. The target quantity (such as number of clicks, time, inferred gain, and so on) are measured, and then analyzed by statistical significance tests (described in Section 2.4.2) to determine which version performs better, and whether that observed differences are significant. Using A/B testing, even small differences between the two tested versions could be uncovered out of millions of exposures of the redesigned version.

However, A/B testing has limitations in terms of the scalability of comparisons [39]. In the lab study described above, many sources of variance, such as the information need and the perception of relevance, need to be controlled to avoid high variance in the experiment results. Although these kinds of variance could be reduced by large sample sizes, A/B testing requires a long time to collect enough data to gain significant insights. Recently, some technologies for estimating online performance based on offline data analysis [55, 56], or online metrics [4, 51], have been proposed to address the problem of scalability faced by conventional A/B testing. Overall, online experimentation expands the traditional short-term user-based evaluation from small lab scale to a web-wide practical experimentations, which allows IR researchers to obtain more informative insights of user behaviors, and evaluate IR systems based on opinions from a larger number of real users.

In general, user-based evaluations allow system developers to receive feedbacks directly from their users. However, none of the measures described above have become a new standard for human-centered evaluation [49]. User-based experiments are usually too expensive and hard to control [111]. Meanwhile, system-centered evaluations have been deemed reliable and repeatable, and have become a commonly-used evaluation technique in IR research.

2.1.2 Test Collections and Batch Evaluation Technique

In order to reduce the cost and boost the repeatability of IR experiments, instead of performing user-based evaluations, IR researchers carry out comparative experiments on *test collections* to compare the effectiveness of IR systems. These "laboratory tests" of system-centered evaluations consider more about behaviors of the system, rather than those of the user. Document rankings are evaluated by effectiveness metrics using relevance judgments (which record the relevance overlap between the document and topic, described in Section 2.2), and the computed scores are deemed as the effectiveness of the evaluated systems.

The modern IR evaluation conferences such as Text REtrieval Conference (TREC) [1], NII-NACSIS Test Collection for IR Systems (NTCIR) [2] and Cross-Language Evaluation Forum (CLEF) [3], are all based on the Cranfield paradigm [23] which was an initial investigation for finding the best alternative among several index languages [111]. There are three assumptions made in Cranfield paradigm: (1) that relevance could be measured via topical similarity; (2) that relevance judgments can be generated by assessors who represent the user population; (3) that all the relevant documents can be identified (in the relevance judgments). Moreover, the Cranfield approach also assumes that the information need of users would not change after having viewed documents; that is, relevant documents are independently and equally desired by users.

Voorhees [111] notes that these assumptions of the Cranfield paradigm might not be strictly true in IR test collections. IR has to face the challenges that the same concept could be expressed in various ways, and the size of collections (such as the number of documents) is large, making it prohibitively expensive to judge every document against every topic. Thus, modern collections usually employ *pooling* techniques in which a small subset of the collection is judged for each topic, and documents outside the pool are assumed to be irrelevant to that topic. As the assumption is generally not true, the generated noise in system evaluations need to be reduced by careful experiment designs so that it has only limited effect on the evaluation result. As a consequence, a single system score, measured by a effectiveness metric, has little meaning except when compared with the

[1] https://trec.nist.gov
[2] http://research.nii.ac.jp/ntcir/index-en.html
[3] http://clef.isti.cnr.it

scores of other systems which are evaluated in the same experiment. Thus, the effectiveness of IR systems is commonly assessed or compared using *batch evaluation techniques*, in which systems are tested with a same set of test collections and effectiveness metrics (shown in the lower half part of Figure 2.1).

Test collections have kept changing over the years to reflect the diversity of communications between users and data. To build a TREC collection, for example, NIST [4] provide a set of *topics* (queries, or detailed descriptions of information needs) and a collection of text documents. Each topic includes the topic number, query, description, and narrative (note that, topics in some TREC rounds might contain additional information such as concept(s), factor(s), and so on), which are formed into Standard Generalized Markup Language (SGML). An example of topic 405 in TREC-8 is shown as below:

```
<top>
<num> Number:  405
<title> cosmic events

<desc> Description:
What unexpected or unexplained cosmic events or celestial
phenomena, such as radiation and supernova outbursts or new
comets, have been detected?

<narr> Narrative:
New theories or new interpretations concerning known celestial
objects made as a result of new technology are not relevant.
<\top>
```

The document files usually record more attributes (or tags) than topics, such as resources, date, author, and so on, which might also be different for documents in distinct tracks. A typical example is shown below:

```
<DOC>
<DOCNO> LA120390-0126 </DOCNO>
<DOCID> 317273 </DOCID>
<DATE>December 3, 1990, Monday, P.M. Final</DATE>
<SECTION>Part P; Page 2; Column 4; Late Final Desk</SECTION>
<LENGTH>323 words</LENGTH>
<HEADLINE>NASA WORKS TO FIX TELESCOPES ABOARD SHUTTLE</HEADLINE>
<BYLINE>From Associated Press</BYLINE>
<DATELINE>SPACE CENTER, Houston</DATELINE>
```

[4] https://www.nist.gov

```
<TEXT>
Trouble with the $150-million Astro observatory today delayed
Columbia astronauts' scientific research, and as the hours
ticked by scientists on the ground said some planned
observations would not be made.

"There's a definite loss as we go," said mission scientist
Ted Gull.  "Some objects are just going to slip off the list."

...
</TEXT>

</DOC>
```

The initial TREC test collection, consisting mostly of newspaper, newswire articles and government documents, is about two gigabytes of text [109]. The size of other TREC tracks could be smaller or larger, and different from year to year. Before TREC-3 (1994), only Routing and Ad Hoc tracks are available. From TREC-3 onwards, various tracks for different purposes were created for matching the research interests of more groups. In TREC-2017 (2017), there were totally eight distinct tracks involved.

With a subset of the test collection available to be used for *training*, participants (such as research groups in universities and companies) run their IR systems to retrieve documents from the provided text collection for each given topic. The retrieved runs are submitted to TREC, and top-d documents of a subset of the runs are selected and merged into the the pool (usually $d = 100$ [110], but note that some TREC rounds re-use the relevance judgments of past rounds, thus the number of pooled runs and pooling depths of those round might vary). Relevance judgments are then formed only for the documents in the pool.

Relevance judgments of each topic are usually (but not always) assessed by the human experts who proposed the information need, created the query and descriptions for the topic, and are familiar with the collection and its documents [25]. The relevance of documents in TREC collection is mostly assessed by a binary relevance scale (`0` for irrelevant, `1` for relevant), but multiple-level scales are also used in some TREC collections, such as GOV2. With the relevance judgments, the collections could then be used for comparing the performance of IR systems.

In batch evaluation techniques, the collection for *testing* contains a set of topics, a collection of text documents and relevance judgments for each topic [90] (note that, topic and document sets in testing collections might not be the same sets as those in training collections). For each of the given topics $\boldsymbol{t}$, each system $\boldsymbol{s}$ that is to be evaluated searches

Topic ID	Doc. ID	Binary	Sormunen
401	A	1	H
401	B	0	N
401	C	1	M
401	D	1	H
...	...	...	...

Table 2.1: An example of relevance judgments in Binary (0 or 1) and Sormunen categories [99] (H–highly relevant; M–marginally relevant; R–relevant; N–not relevant).

the text collection and using a retrieval model assigns a similarity score for each document $\boldsymbol{d}$, denoted $\mathsf{Sim}(\boldsymbol{d}, \boldsymbol{s})$, which estimates how closely the document is related to the given topic. The returned runs contain documents sorted in decreasing similarity score order. Denote $\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(i)$ as the document that is ranked at ith ($i \geq 1$) position in the ranked list (or run) $\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}$ and its similarity score given by system $\boldsymbol{s}$ as $\mathsf{Sim}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(i), \boldsymbol{s})$.

The *relevance score* of document $\boldsymbol{d}$, denoted as $\mathsf{Rel}(\boldsymbol{d}, \boldsymbol{t})$, indicates the relevance status between the document and the topic $\boldsymbol{t}$ (see Section 2.2), and is stored in the *qrel* files of relevance judgments. A variety of established IR evaluation metrics (discussed in Section 2.3) such as *Average Precision* (AP) can then be used to examine the performance of the system by computing an *effectiveness score* for each run. For each system, these individual run scores across topics are then aggregated into a single numeric score, usually by calculating their arithmetic mean, as the measurement of the retrieval quality. If the evaluated system achieves a statistically significant higher effectiveness score, it can be regarded as being better than the baseline system.

2.2 Relevance Judgments

To record the pertinence (according to some definition) of documents to topics, relevance judgments are typically assessed using ordinal scales at binary or multiple levels [22]. They are registered in the form of *qrels*, a list of tuples each containing a topic identifier, a document identifier and the relevance score about how much pertinence (overlap) there is between the document and the topic. Table 2.1 shows examples of relevance judgments in two scales, Binary and Sormunen [99]. For instance, the first row indicates that document A is considered as relevant (relevance score is 1) in a Binary sense, and highly relevant (relevance category is H) in the Sormunen graded categories to topic number 401.

2.2.1 Pooling

It is all but impossible to collect judgments for all documents against all topics when the collection size is large. For example, if we want to collect full relevance judgments for TREC-8 Ad Hoc test collection (about $5{,}000{,}000$ topic-document combinations), optimistically assume that a human assessor spends two minutes [52] on judging the relevance of one topic-document combination, the judgment collection will cost about 80 years of one assessor. An alternative process that select documents which have high probabilities as being relevant to the given topic, and high importance in terms of separating the systems being measured, called *pooling* (and now also known as Depth@k) proposed by Spärck Jones and Van Rijsbergen [100], and only collect relevance judgments for these pooled documents. The pooling methodologies described in the follow content are for generating a pool for a single topic. The strategies could be repeatedly applied to each of the given topics in the test collection.

The Depth@k approach has been employed by TREC since the starting rounds in the early 1990s. It forms the union of the top-k documents retrieved by the submitted runs (which are retrieved by participating systems which were developed by participants such as researchers and organizations). As Figure 2.1 describes, the human *expert* assesses the relevance of these selected pooled documents for each topic and builds the *qrel* file for IR evaluation based on the pooled subset of the possible documents. The pooling depth k needs to be carefully chosen for distinct text collections (in practice, it may be dictated primarily by budget available). For example, TREC-8 [112] involved 129 systems and 50 topics, and pooled the top-100 documents returned for each topic by a subset of 71 systems. On average, each topic received 1736.6 judgments, with 94.6 (about 5.4%) of them judged as relevant by the NIST assessors.

The bias of the evaluation using these relevance judgments is caused by documents outside the pool which are never considered as relevant [58]. This occurs because only pooled documents are presented to the assessors, and judged as relevant or irrelevant to the given topic. The relevance of documents which are not in the pool is unknown, and all such documents are usually deemed as irrelevant in recall-based evaluations. Utility-based metrics, such as *Rank-Biased Precision* (explained in Section 2.3), measure the extent of potential risks caused by unjudged documents via computed residual scores, concerned with the pooling depth k and the chosen evaluation metric (the uncertainty of IR evaluations caused by documents outside the pool will be described in details in Chapter 4). However, the effectiveness score of systems which did not contribute to the pool still suffer from this bias. Zobel [120] argued that the relevance judgments of TREC, pooled using Depth@100, only found at best 50%–70% relevant documents. As all unjudged documents are deemed as irrelevant in the evaluation, the performance of novel IR systems and "new" systems that did not contributed to the pool might be distorted and underestimated. Sanderson and Zobel [91] discovered that, in contrast to the methodology used by TREC, building test collections with more topics but shallower

pooling depth might result in more reliable system comparisons.

In addition to the standard method Depth@k, there have been other pooling strategies proposed to improve the efficiency of creating a reliable (the opposite of *biased*) pool. The strategy of Take@N [60] assigns each document $\boldsymbol{d}$ the highest rank, $\boldsymbol{h_d}$, that document $\boldsymbol{d}$ is placed in all contributed (selected) systems. For example, document $\boldsymbol{d}_0$ is ranked at 6, 2 and 4 in runs retrieved by system $\boldsymbol{s}_1, \boldsymbol{s}_2$ and $\boldsymbol{s}_3$ respectively. The $\boldsymbol{h}_{\boldsymbol{d}_0}$ is the highest rank that $\boldsymbol{d}_0$ is placed in a run, that is 2. Then Take@N pool $\boldsymbol{N}$ documents with the highest $\boldsymbol{h_d}$. Take@N could guarantee the size of the pool with the fixed number of pooled documents ($\boldsymbol{N}$), and might pool more documents than others [61]. But compared with Depth@k, instead of pooling exact k documents to the pool from each contributed system (that is, the contributed systems equally contribute to the pool), Take@N pools first N documents with the highest $\boldsymbol{h_d}$, without considering which system contribute the document. For one topic, Depth@k and Take@N are very similar and pool almost same set of documents [61]. But when pooling documents for multiple topics, Take@N can guarantee that the number of the pooled documents for every topic is the same, while Depth@k cannot.

BordaTake@N [3], developed based on the methodology of *Borda count*, selects the most preferred $\boldsymbol{N}$ documents by the contributed systems. Each document $\boldsymbol{d}$ is assigned the sum of ranks that it is placed by the contributed systems. In general, the lower the rank sum the document receives, the more it is preferred by the contributed system. Documents with the $\boldsymbol{N}$ smallest rank sums are then pooled. Compared to approaches described above, BordaTake@N takes the preference of documents considered by all contributed systems into account, but Depth@k and Take@N only consider the earliest rank of documents [61].

Similarly, CondocertTake@N guarantees the *Condocert Criterion*, and select top-$\boldsymbol{N}$ Condorcet winners (documents which are ordered higher by more contributed systems than others) into the pool [61]. Firstly, a document set $\boldsymbol{D}$ that contains all the documents retrieved by the contributed systems (that is, the union set of documents in all contributed runs) is generated. Then, for each document pair $\langle \boldsymbol{d}_i, \boldsymbol{d}_j \rangle$ in $\boldsymbol{D}$ (where $\boldsymbol{d}_i, \boldsymbol{d}_j \in \boldsymbol{D}$, and $i \neq j$), compare the ordering of $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$ in each of runs of the contributed systems. If $\boldsymbol{d}_i$ is ranked before $\boldsymbol{d}_j$, increase the counter of this document pair by one, otherwise decrease by one. After iterating though all the runs returned by the contributed systems, if the counter for $\langle \boldsymbol{d}_i, \boldsymbol{d}_j \rangle$ is positive, $\boldsymbol{d}_i$ will have higher priority to be pooled than $\boldsymbol{d}_j$. In general, CondocertTake@N selects documents which win a majority of votes by the contributed systems, which could be satisfied by previously described pooling strategies. However, Condorcet winners do not always exist according to the given votes, known as Condorcet's voting paradox [24]. Thus the pool containing $\boldsymbol{N}$ Condorcet winner documents is not guaranteed to exist if CondocertTake@N is used.

The pooling strategies described above consider the rank of documents, rather than the similarity score assigned to each document. A series of CombTake@N strategies construct a new ranking list (for each topic) containing all documents sorted in decreasing

mapped (by some fusion method) score order, and then pool the top-N documents in the new list [61]. The mapping function is different in each sub-strategy of CombTake@N. The similarity scores of documents are firstly normalized into the score range $[0, 1]$ across each run of contributed systems and each topic. CombMAXTake@N assigns each document d the greatest normalized similarity score that d received in all the pooled runs. This strategy is similar to Take@N except that it considers similarity scores rather than ranks.

CombMINTake@N assigns the minimum similarity score that contributed systems compute for the document d. This strategy minimizes the probability that non-relevant documents are selected into the pool, while CombMAXTake@N maximizes the probability that relevant documents are pooled.

Except using the highest and lowest similarity score of each document d as the mapped score of d in the new list, by CombMAXTake@N and CombMINTake@N respectively, other fusion methods that compute the median (CombMEDTake@N), the sum (CombSUMTake@N), and the average of non-zero normalized scores (CombANZTake@N) of each document over all contributed systems are also proposed [61]. In addition, CombMNZTake@N takes the product of the number of systems that retrieve the document d and the sum of the normalized similarity scores of document d as the mapped score of d in the new document list. It considers similarity scores given by all contributed systems, meanwhile assign higher priors to documents retrieved by more systems.

Lipani et al. [61] compared the nine pooling strategies described above by generating fixed sizes (the costs of collecting relevance judgments are therefore fixed) of TREC-8 pools using nine pooling strategies respectively, pool size N from $5{,}000$ to $80{,}000$ increase by $5{,}000$. The bias of pools generated by each strategy is measured by computing the variations in the TREC-8 run scores, evaluated by AP, NDCG, and Prec(100) respectively (evaluation metrics are described in Section 2.3), using a set of ***leave-one-out*** experiments [120] with systems, that is, starting with the original set of documents pooled by this strategy, then removing documents solely contributed by runs submitted by one participant each time, and forming qrels file for the remaining documents. The variations were calculated by the difference of run scores (Mean Absolute Error), and the distinction of the rank of runs (System Rank Error [59]). The two-tailed paired t-test was be employed to test if the variations were statistically significant (System Rank Error with Statistical Significance [59]).

By comparing the bias of relevant judgments with the same size but pooled by different strategies, Lipani et al. [61] found that the nine tested methods all reduced to Depth@100 when the pool was large enough (the methods have to converge in these experiments because they would end with pooling all the available judged documents). The authors stated that, the biases of pools generally decrease when the sizes of pools increase, and when the pool size reaches around $40{,}000$, the biases of all strategies are close to zero (however, in their fundamental experiments, the reference was the TREC

judgments; no document outside the reference was added to the pool when the pool size increased, thus document sets pooled by all different strategies have to converge to the reference). The relevance judgments pooled by CombMINTake@N and CondocertTake@N were shown as less reliable if compared with judgments of Take@N strategy. The proportion of non-relevant documents pooled by CombMINTake@N was too high, and therefore made the relevance judgments unstable. The CondocertTake@N, as mentioned before, cannot guarantee that "the most preferred N documents" exist. Meanwhile, the strategy of CombMAXTake@N, CombSUMTake@N and CombMNZTake@N always perform better than the baseline of Take@N in terms of building pool with lower bias.

Lipani et al. [61] also explored which pooling strategies were more stable when evaluating systems using the metric of AP, NDCG, and Prec(100) respectively. For example, AP and NDCG treat relevant documents at top ranks as being more important than those at deeper ranks (known as *top-weightedness* described in Section 2.3), however relevant documents ranked before depth 100 are equally weighted by Prec(100). Thus, for pooling strategies, such as CombMAXTake@N and CombMNZTake@N, which pooled more top-ranked documents in the runs could generate more stable relevant judgments for AP and NDCG. The CombMEDTake@N, similarly, is the most reliable pooling strategy if AP and NDCG are employed for the system evaluation, but it is the most biased strategy if Prec(100) is used. Finally, Lipani et al. [61] discovered that AP is least affected by pooling strategies, compared to NDCG and Prec(100).

Moffat, Webber, and Zobel [72] proposed a methodology to adaptively pool documents for identifying the best systems, so that the pool size could be greatly reduced. As the goal of this strategy is to find the best systems, instead of ordering all the systems, documents only retrieved by poor systems would not be considered by the pooling. As the pooling proceeds, more effort is spent on pooling and judging documents that are critical to distinguish top systems. The authors found that, an average of 200 judgments per topic (conventionally there was an average of 1,737 judgments for each topic) was sufficient to identify the effectiveness bounds of top systems for TREC-8. And they also demonstrated that RBP scores of systems were more reliable than AP scores for significance tests in system comparisons.

A number of methods of deciding when to stop pooling new documents based on either existing judgments, or estimates of relevance in new judgments were proposed by Losada, Parapar, and Barreiro [62]. They discovered that some of their proposed methods only required 5% to 7% documents to be pooled to archive accurate results.

2.2.2 Judgment Ordering Effect

Relevance is a subjective, dynamic and multidimensional concept [93], the quantification of which depends on the document content as well as the assessor's perception of relevance at a specific moment [38]. Several researchers have studied whether the order of

presenting documents (sequentially) when collecting relevance judgment would affect assessor's decisions.

Eisenberg and Barry [29] sorted a list of documents in decreasing and increasing relevance order respectively and tested if assessors would underestimate, or overestimate, the relevance of documents using a 7-level scale. The observed trends were weak, and became insignificant when using other scales. Huang and Wang [42] explored the order effect in regard of document positions (ranks) in the list. They showed that the ordering of presenting documents affected relevance judgments of documents from rank 15 to 30, but did not for others.

Xu and Wang [118] found that the pattern of order effect for relevance assessments was curvilinear (U-shape), and which was different for documents in distinct relevance levels. The authors believed that this relationship was consistent with the combined effects of learning, sub-need scheduling, and cursoriness. However none of these effects could be isolated for evaluation. Their work confirmed the conclusion in the previous study [42] that ordering only affects documents in the some positions. It had minor effects on the assessment of documents at top and bottom ranks. In general, Xu and Wang [118] concluded that order effects existed, but its impact to the relevance assessment was relatively small. However in reality, some documents in the retrieved results might be completely ignored because of order effects, which might influence mainstream opinions (such as in news searching).

2.2.3 Relevance Judgment Variation

In early work, Katter [47] observed that the inter-judge and intra-judge reliability of relevance judgments was low. Conventionally, relevance was subjectively judged by individuals (experts) independently. Even when relevance is assessed by groups, as was argued by Fairthorne [30], there are still great disparities between assessors' opinions on judgments for topic-document pairs [5]. In the relevance assessment of TREC-4, up to 200 relevant documents plus 200 randomly selected irrelevant documents judged by the primary assessor (who created the topic) became a pool which was then passed over to another two secondary assessors [30]. Although these assessors had similar background and were trained for the TREC task, the study showed that the *overlaps* of their assessment agreements were under 50% [30]. Turpin and Scholer [104] carried out similar tests (using another data set, TREC GOV2) and concluded that assessors did not agree well (only around 70%) with each other in both task and topic levels.

There are several factors that may affect relevance assessments, including those noted by Burgin [14], Turpin and Scholer [104], and Kelly [48]:

- *human*: gender, age, background, preference, experience;
- *document*: representation, style, presented order, other documents in the set;
- *task*: time limitation, expectation, relevance scale used; and

- *topic*: rarity, saliency, ambiguity.

In particular, assessors (or users) observe and measure relevance according to their own definitions [48], which may lead to conflicting results for the evaluation of IR systems.

In contrast, other researchers such as Salton and Lesk [89], have demonstrated that if the assessor consistency is low, their disagreements tend to be mostly found at deep ranks, and that the top relevant documents of a run tended to be evaluated as relevant by most assessors, meaning that in most cases the variation of judges would have a reduced offset on system ranking.

Scholer, Turpin, and Sanderson [95] analyze human assessment error via exploring the inconsistency of relevance judgments which are in distinct *qrels* of TREC test collections, for same topic-document combinations. Their results show that about 18% of documents receive inconsistent judgments from distinct *qrels*, indicating notable intra-assessor errors. Scholer, Turpin, and Sanderson [95] further explore the factors that might affect the quality of relevance assessment. They find that time is a key effect, that is, assessment consistencies of assessors usually shift over time. Moreover, except the test collection of GOV2 and WT10G, assessors tend to assign the same relevance values to adjacent documents (inertia). Scholer, Turpin, and Sanderson [95] note that it may be because assessors anticipate that relevant documents appear together when they viewed documents in the *qrels* order, that is document ID sorted linear order. The inconsistency of duplicated documents in other relevance judgments which are collected using an alternative approach, rather than document ID sorting methods, is then investigated. The observed low inconsistency demonstrates that the judgments gathered using the alternative method is potentially more accurate than TREC judgments.

In addition, Scholer, Turpin, and Sanderson [95] split relevance judgments according to their order in *qrels* into different judgment sets, and measure the correlation of system orderings evaluated using judgments made in early and late assessment process respectively. The system orderings are shown to be significantly different, which indicate that the ordering of judgments when forming *qrels* is a important factor that affects the evaluation of IR systems.

2.2.4 Relevance Scales

Absolute Relevance In early work, a binary relevance scale (`0` for irrelevant, `1` for relevant) was commonly used for relevance assessing. From the 1990s, relevance scales, which divide relevance into multiple levels, allowed modeling of the relationship of document and topic in multiple degrees. The pertinence of documents and topics can be interpreted into several levels or categories using *ordinal relevance scales*. For example in TREC-2005 [22], the relevance level for irrelevant documents is `0`, for relevant documents is `1`, and for highly relevant documents is `2`. The number of some categorical scales such as Sormunen [99] (which has four relevance categories: `H`–highly relevant; `M`–marginally relevant; `R`–relevant; `N`–not relevant) is even more.

However the criteria that assessors (whose decisions should be representative of practical user's opinions) have to determine the distinctions of relevance levels is diverse. Some experiment results have shown that *generous* users (defined as assessing more than 50% of level 0 documents as relevant) are more likely to judge documents as relevant, but *parsimonious* users (defined as assigning less than 50% of level 1 documents as relevant) usually only judge the level 2 documents as relevant [96, 94]. In IR evaluation, system effectiveness scores can be greatly affected by the criteria of assessors who generate judgments. But using generous relevance judgments may not always result in increment of system scores, similarly parsimonious judgments do not always hurt system scores. The evaluation results depend on the employed metrics as well. However, the probability that systems cannot be significantly distinguished by IR evaluation will be high if the criteria of the used judgments is extreme (too generous, or too parsimonious).

Han et al. [36] proposed a method of transforming existing absolute relevance judgments to different absolute scales with smaller number of relevance levels. For example, one of their experiments was converting Sormunen judgments to Binary. They discovered that it was better to transform judgments topic by topic. And there was no unique method that could be applied to every type of judgments conversion.

The relevance scale used for judgment assessing is usually preset by experiment creators instead of assessors. But the distinction between relevance levels expected by assessors may be disparate. Assessor perceptions of relevance are hard to incorporate into a single relevance scale [105]. Thus, using single absolute relevance scales for relevance assessing can still cause judgment variation.

Pairwise Preference Carterette and Petkova [16] suggested a method to merge distinct lists of retrieved results: break items in lists into pairs and discover the binary preference of each pair so that an ideal ranking list integrated from distinct ranking lists can be generated based on *pairwise preferences* (PP). This strategy was then considered to collect relevance judgments for IR evaluation. Carterette et al. [17] proposed to collect *preference judgments* that only record which document is preferred over the other in a document pair. Judges only need to make a preference choice for each pair instead of assigning absolute relevance score for each document in the conventional relevance assessing.

Compared to ordinal relevance scales, pairwise preference helps to reduce the complexity and increase the consistency of relevance assessing [20], meaning that the cost of collecting valid relevance judgments with high fidelity may therefore decrease. For each topic, the preference between documents in pairs can therefore be known and used to learn the relevance ranking of all judged documents. With this rank list with much fewer relevance *ties* (documents with the same relevance score), for most pairs of documents, their relevance order can therefore be discovered. For example, when a Binary relevance scale (a 2-level scale) is used, the scope for documents having the same tied relevance score is high, whereas if relevance is measured on Sormunen category (a 4-level

scale) [99], there will be most likely be (but, of course, not guaranteed to be) fewer ties. Documents C and D in Table 2.1 can be deemed as a example that their relevance scores are tied in Binary judgments but in Sormunen, document D is judged as more relevant than C for topic 401.

Pairwise preferences can be used to collect judgments with fewer ties with sufficient pairs generated and assessments made. However, the costs (the number of pairs, the number of assessment per pair and payment per pair to an expert assessor) and the fidelity (the precision of relevance scores) of judgments using this method all need to be considered and traded off. We explore this further in Chapter 5.

Magnitude Estimation User perception of relevance varies and it is hard to choose a single ordinal relevance scale to capture all individual views. Preference judgments may require a large number of assessments. Turpin et al. [105] used a psychophysical technique, *magnitude estimation* (ME), working with a crowd-sourcing platform (users of which are paid to complete human intelligence tasks, or HIT; crowd-sourcing is described in Section 2.2.5), to collect relevance judgments from users.

Magnitude Estimation is a scaling technique used to estimate the magnitude of answers based on a ratio by comparing the current subject with the previous one, which is claimed to allow replication across subject groups and keeps the measurement consistent [7].

Each task unit designed by Turpin et al. [105] included eight documents (a gold standard is constituted by two of them, denoted as H_u and N_u, whose relevance levels are known as *highly relevant* and *not relevant* respectively) and was randomly assigned to participants. Documents were presented in a random order and the worker could assign any positive score to the first document. The next document was assessed with regard to the previous one. For example, if the current document was twice as relevant as the previous one, its score should be double the previously assigned score. After all the assessments of topic $\boldsymbol{t}$ were complete, the relevance score of document $\boldsymbol{d}$ was normalized as a single numeric value as below:

$$\mathsf{Rel}(\boldsymbol{d}, \boldsymbol{t}) = \exp(\log score_{\boldsymbol{d}} - \mu_u + \mu). \tag{2.1}$$

where $score_{\boldsymbol{d}}$ is the score of document $\boldsymbol{d}$ assigned by a participant, μ_u is the arithmetic mean of log scores of the set of 8 documents in the unit u, and μ is the average of all the documents judged for this topic.

The quality of each assessment using ME is controlled by (1) two tests requiring correct answers to continue the task; (2) requiring at least 20 seconds for each document assessment; (3) gold standard check, whereby the assigned ME score of H_u should be larger than that assigned to N_u.

Using ME, participants do not need to interpret the distinctions between relevance levels but can use any numbers that seem suitable to them. They only need to keep the

assessment consistent within the unit (eight documents) instead of the whole text collection for the given query. Magnitude estimation is considered as a suitable relevance scale for IR evaluation by Turpin et al. [105], since its judgments are consistent with ordinal judgments. They agree more with each other than binary judgments in document orderings.

With different relevance scales and evaluation metrics used, the system ordering is substantially disparate. But the distribution of magnitudes demonstrates that user perceptions of relevance are neither readily fitted by a single profile, nor easily captured.

S100 More recently, Roitero et al. [80] propose an approach they call *S100* which complements ME, to collect relevance judgments on crowd-sourcing platforms. The authors state that ME allows assessors to use any positive numbers they like to judge the document as more or less relevant than previous documents they have judged so far, but it is not a natural approach for human assessors who usually use ordinal relevance scales (such as the five-star rating scale) to judge value in a wide range of contents other than the Web.

With S100 employed to judge the relevance of documents, assessors are required to use a bounded slider to choose a number from 0 to 100, instead of entering numbers into text fields, as the relevance score of document. S100, similar to ME, allows assessors to start with any number they prefer to judge documents. But distinguished from ME, if S100 is employed, the assigned scores should be limited within the range $[0, 100]$, rather than unbounded.

Similar to the experiment design of ME, S100 also includes eight documents in each crowd-sourcing unit (or HIT), and quality checks are performed to reduce the assessment noise. If assessors fail the quality check of a unit, they have up to three chances to restart judging the HIT and change answers.

Judgments of each document collected by S100 can be directly aggregated by computing the arithmetic mean of individual scores. According to the results of judgment comparisons over all topics, Roitero et al. [80] state that the normalization process in ME makes judgments less comparable across assessors and topics, and that is why S100 judgments have higher agreement with Binary and 4-level relevance judgments than ME judgments do. In addition, S100, compared with ME, is easier to learn for crowd-workers, and so costs less time (on average) of assessors to complete the judging task of one document. Using TREC binary judgments as the reference, the pairwise agreements of the reference and judgments collected by S100 and ME respectively are compared. The results show that, when using shorter HITs (the number of documents in each HIT is smaller than eight), the pairwise agreement of S100 judgments is higher than it of ME, for any HIT length from two to seven. Moreover, when the HIT length is eight, to obtain judgments with similar qualities, S100 requires about (on average) three fewer judgments

(from different assessors) per document than ME. Therefore Roitero et al. [80] conclude that S100 is more robust than ME.

However, the bounds 0 and 100 are also somewhat arbitrary, albeit a common marking scale in many assessments. Thus, S100 might be easily and quickly adapted by assessors who often use this scale in their daily life, but not for people who get used to 3-level, 5-level, or 7-level. Even using a same scale, the distance between relevance levels expected by people from different areas could also be distinct, and which is hard to capture and measure.

Greisdorf and Spink [35] used 4-level, 77-level, and 100-level ordinal relevance scale to collect relevance judgments from a group of human assessors, and they conclude that assessors have ability to measure the relevance of documents using any type of ordinal scale.

Likert-type Scale The most widely-used technique of scaling attitudes in survey-based research is the *Likert-type Scale* [57]. Likert [57] describes the problem when measuring attitudes of people – as it is possible to classify and subclassify stimuli indefinitely, any given person could possesses almost infinite number of attitudes, which is statistically absurd in a research study. In his paper, Likert described a detailed technique on how to construct a good scale to measure human attitudes, including how to choose the number of response categories (or levels), how to select statements for each category, how to measure the capacity of the scale, and so on. A good Likert-type scale needs both *symmetry* – the number of positive and negative response categories should be equal, and the categories should be bilaterally symmetric to the "zero", or neutral, position (for example, the 5-level scale with categories "strongly disagree", "disagree", "neutral", "agree" and "strongly agree"); and *balance* – the distance between any two categories is the same. A quantitative value (maybe not shown to the assessors) can also be assigned to each category in the scale, so that the quantitative analysis can be carried out using categorical responses.

The most suitable scale could be found by asking the same group of people the same questions using different scales, and comparing the results in regard of reliability, internal consistency, and so on to find the most reliable and suitable scale for the given case. Matell and Jacoby [68] discover that internal consistency, test-retest stability, concurrent validity, predictive validity, and proportion of the scale used [67] are not related to the number of response categories provided. The optimum number of response categories could be different from case to case. Dawes [27] stated that data collected using a 5-level scale could be transferred to 7-level format using a simple rescaling method without any effect to the results. In addition, they found that the mean scores produced by scales with more response categories (such as 10-level scale used in their experiments) are *slightly* lower, relative to the upper bound of the scale. And for other data characteristics, the

scale formats have little effect, in terms of variation about the mean, skewness or kurtosis, to the data.

In IR evaluation, NIST Binary used a 2-level relevance scale. Some categorical relevance judgments, such as Sormunen [99] which employ a 4-level scale, describe document relevance to a higher fidelity using multi-level scales. But how precise the relevance score needs to be has no unique answer, which usually depends on the budget, the size of data, and the metric discrimination for the compared systems. In Chapter 5, we collect relevance judgments with high precision, and explore mapping them into categories.

2.2.5 Crowd-Sourcing

In recent years, crowd-sourcing, an online model for problem-solving, has become very popular in various areas such as creative and design industries (`Threadless`[5]), scientific research and development (`InnoCentive`[6]), business marketing and so on. In researches of IR evaluation, crowd-sourcing has also become a useful technology to collect relevance judgments from a great number of human workers with low costs.

In 2006, Howe [41] described the web-base business model crowd-sourcing as "crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call".

The online T-shirt company `Threadless` allows users to submit their T-shirt designs, which will then be available to the voting community for scoring. The design with the highest score in competitions will be printed and sold on the website. The winner designer will be paid US$2,000, which is a nontrivial payment in terms of crowd-sourcing, but a quite low payment for design services. `Threadless` was "selling 60,000 T-shirts a month", "had a profit margin of 35%" and "was on track to gross $18 million in 2006" [40].

`InnoCentive`, a crowd-sourcing platform that enable the crowd-sourcing scientists (that is, `InnoCentive` users) to "receive professional recognition and financial award for solving research and development (R&D) challenges" [43]. The users of `InnoCentive` submit their solutions to R&D problems posted by seeker companies. The companies review the solutions and awards a pre-set cash prize to the "best" solution which meets all of the requirements. Lakhani et al. [54] state that opening up the unsolved scientific problem to a large community of anonymous "outsider" on the crowd-sourcing platform can not only obtain innovative solutions, but also boost research productivity. Employing crowd-sourcing for solving R&D problems is potentially an effective strategy.

`EMOS`[7], an AI emotion analytics company, sell products that interpret emotion and sentiment from audio and textual conversations with models trained via deep learning technologies. The models were trained using limited academic datasets with noise, so

[5]https://www.threadless.com
[6]https://www.innocentive.com
[7]https://www.emos.ai/

they cannot handle different cases well in the real world. Thus EMOS aimed to collect more training data with high quality from human contributors on the crowd-sourcing platform, Figure8[8]. Although crowd-workers do not need to be trained to understand different emotions of humans, emotion is very subjective, and whose intent is also hard to understand. EMOS collaborated with Figure8 to design a workflow that assign audios with the same speakers to each annotator to reduce the confusion, and prioritize tasks with strong emotions. Using the obtained training data from Figure8, EMOS achieved "80% accuracy in their emotional detection algorithms, an improvement of an additional 30% to their model's performance" [31].

Similar to the problems just described, in conventional relevance assessments in IR, the number of assessors is relatively small because they are discipline and topic experts. They make an assessment, and their decision is final and taken to be "gospel". If researchers want to study other aspects (not only textual or topical relevance) of retrieved documents such as novelty and freshness, it is very hard and expensive to collect new judgments from experts. Moreover, the relevance of documents can be subjectively defined by practical users in different ways. Therefore, IR research groups have started performing experiments on crowd-sourcing platforms, as described above, which provide a great number of human participants who are drawn from a large community and paid a small amount of money to complete human intelligence tasks (HIT) published by task creators [2].

The crowd-sourcing platforms which are commonly-used by IR researchers, such as Mechanical Turk[9] and Figure8, combine the advantages of human intelligence and automation to enhance the data gathered using human knowledge or opinions. As suggested by Figure8 in their online documentation[10], the ideal tasks for Figure8 are jobs that could not be automatically completed by machines but need to obtain human intelligence. The task needs to be designed using objective rules, and instructions should be clear so that crowd-workers can follow them and provide valid answers. Figure8 also lists some common use cases (it is also easier and more productive to create these kinds of tasks in this platform): sentiment analysis, categorization, data collection and enhancement, data verification, training data creation for an algorithm, and so on.

The typical question forms on Figure8 are single choice, multiple choice, textboxes and so on. The platforms support Javascript and CSS so that task creators could create or customize functions such as answer validations, and time recording. The information (such as worker ID, crowd-group belonged, country, crowd-task experience level, and so on) of crowd-workers who have contributed to the task could be seen by task creators. The platforms also allow task creators to set qualification requirements of workers, for

[8] https://www.figure-eight.com

[9] https://www.mturk.com

[10] https://success.figure-eight.com/hc/en-us/articles/202703295-Getting-Started-Ideal-Jobs-for-Crowdsourcing

example, participants should be from English-speaking countries, before launching tasks. The pay rate (workers get paid per HIT) of task is also set by task creators.

As already noted in connection with magnitude estimation and S100, crowd-sourcing platforms can be employed to collect relevance judgments from a certain number of workers (participants) at low cost. Although participants on crowd-sourcing are "bronze standard" judges according to the categorization of Bailey et al. [5], the assessing noise can be reduced by quality checks such as involving test questions and *gold standards* which are designed by task creators, and by collecting a large volume of opinions and then synthesizing them into single outcome for each evaluation unit. On `Figure8`, task creators could enable the *quiz mode* (explained in more detail in Chapter 5) to remove workers with low accuracy before they enter the *work mode* to do the real task. Each participant may complete a large number of distinct small tasks and each task can be performed by several individuals. `Figure8` also provides an option of choosing participants in different quality levels. The number of participants in the level of *high quality* (most experienced, highest accuracy assessors) is the smallest. Thus it generally requires a longer time to collect judgments from workers in the higher levels, and they may also expect higher payment rates.

A number of IR researchers have employed crowd-sourcing platforms to collect judgments or study different aspects of relevance. Chen et al. [20] dynamically generated document pairs based on the collected answers given by crowd-workers to minimize the costs of collecting preference judgments on `Mechanical Turk` platform. Chandar and Carterette [18] employed pairwise preferences to study different aspects of relevance such as novelty on `Mechanical Turk`. As described in the previous section, relevance judgments of ME [65, 105] and S100 [80] were all collected from `Figure8` platform, and were shown to have high agreement with conventional judgments created by experts.

In our project of collecting relevance judgments for the same set of topic-document combinations on a crowd-sourcing platform using different pairwise scales (see Chapter 5), we create HITs and gather judgments for document pairs on `Figure8`. The prepared data sheets (each row of which contains a topic and two paired documents, the detailed methodology for generating pairs is described in Chapter 5) needs to be upload to `Figure8`, which could be protected as being private by upgrading to an enterprise account. Crowd-workers get paid only if they successfully complete a page of assessments (answer all questions and pass quality control validations). Each page could contain more than one row, and rows could be linked (for example, hide or populate the row based on worker's answers of previous rows) by logic statements using CML (HTML like) programming language on `Figure8`. In the stage of interface design on `Figure8`, task questions could only be implemented by CML code for one row, which is quite inconvenient in some cases, for example, when a page contains multiple rows but their task questions are not the same. In our experiment design, survey question about which relevance scales they prefer to use, is only shown after the normal questions of the last

row of the page, and needs to hide for previous rows. We could only use Javascript to solve this problem.

As described before, test questions are used to control the quality of worker's assessments, which are usually questions with definite answers. Test question rows are usually chosen before launching the tasks, and task creators need to set "correct" answers (could be multiple) for these test questions. The workers will be marked as wrong if their answers do not match either correct answer of the test question. Survey or open-end question tasks usually do not need test questions. If the test question accuracy of the worker is above the accuracy threshold set by the task creator, the worker is deemed a *trusted* contributor whose answers in the main tasks will be retained in the results. As mentioned before, the quiz mode (if enabled by the task creator) is for filtering "bad" workers out by asking only test questions. If the worker pases the quiz mode, they still need to answer one test question, which has not been seen by this worker before, in the work mode per HIT (page). Note that, trusted workers could become untrusted if the accuracies of their answers to test questions in the work mode drop below the minimum accuracy requirement, the answers of untrusted workers would be removed from the results even they have already got paid.

In the `Figure8` settings page, task creators could set the lower limit of the time that each worker need to spend on a HIT (to avoid workers answering questions without careful consideration); the upper bound of the number of HITs that a worker could complete (to avoid individual opinion bias); the number of different workers who contribute for each row, denoted as $\boldsymbol{Y}$; the minimum test question accuracy of the worker whose answers could be trusted; if some untrusted answers of *finalized* rows (have contributed by exact $\boldsymbol{Y}$ distinct workers) are removed from the results because some trusted workers become untrusted, whether the system should re-open the finalized rows to collect trusted answers, or not; whether rows need to be completed in order, or randomly; and the pay rate per HIT.

After launching a job, task creators are able to keep watch on job details (for example, the number of rows remaining) in the dashboard page of `Figure8`. The result sheets are downloaded from `Figure8` when each job completes.

Collecting research data from crowd-sourcing platforms has been popular in recent years. Researchers can design and implement their experiments on the platforms easily and pay small amounts of money to the human participants. The crowd-sourced tasks are usually completed by people from different countries and processed efficiently. The effective data could be collected by experiment designs with high levels of quality control processes. The platforms such as `Figure8` would also collaborate to select qualified workers for the tasks. As the size of data increases, crowd-sourcing has become a useful and reliable methodology used to collect or analyze data requiring human intelligence. We make use of it in Chapter 5.

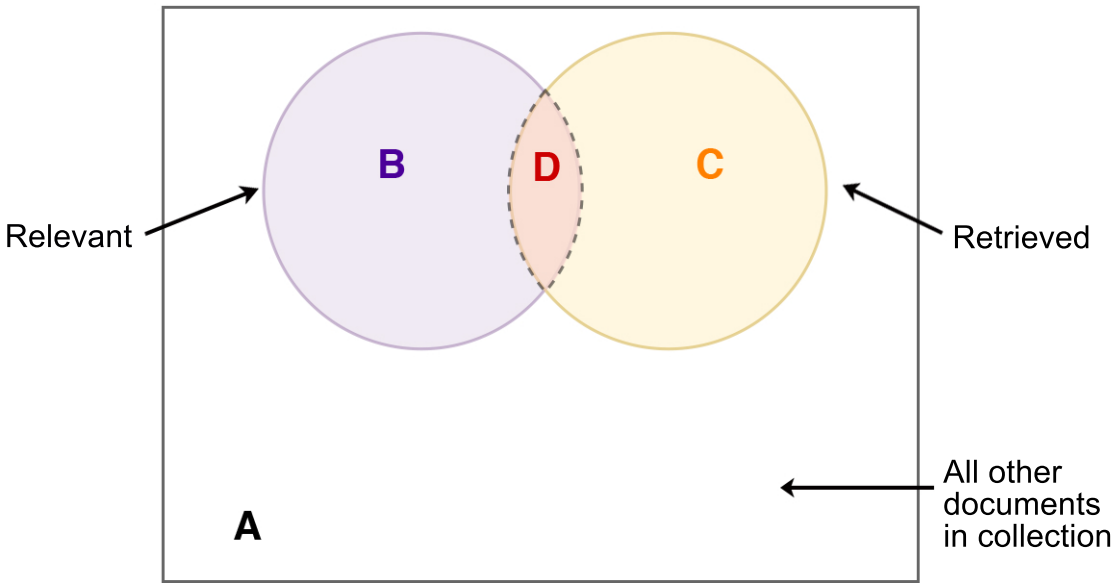


Figure 2.2: Document classifications concerned with Recall and Prec, ignoring relevance judgments. The set of all relevant documents for the given topic t is represented by the left circle ($\mathbf{B} \cup \mathbf{D}$). The set of all retrieved documents in the run $r_{s,t}$ is represented by the right circle ($\mathbf{C} \cup \mathbf{D}$). The area $\mathbf{A} \cup \mathbf{B} \cup \mathbf{C} \cup \mathbf{D}$ is the universe.

2.3 IR Evaluation Measurements

Once relevance judgments have been collected and formed into a qrels file, the retrieval quality of IR systems can be evaluated by a wide range of effectiveness metrics, with different metrics emphasizing different aspects of the system runs. Suppose that the system s retrieves a run $r_{s,t}$ for a given topic t. In this section, all the examples of metric score computations are assumed for a single run, $r_{s,t}$, with some prefix of it taken to be an answer. Techniques for aggregating run scores over a set of topics into a system score will be described in Section 2.4.1.

2.3.1 Metrics

Recall and Precision *Recall* and *Precision* are common measures to evaluate the effectiveness of search results. Without considering how the relevance judgments are arrived at, assume all the relevant documents for the given topic t are known and represented by the area of **B** together with **D** (the left circle) in Figure 2.2. The right circle, constructed by the areas **C** and **D**, represents the set of retrieved documents (regardless the ordering of documents in the retrieved run). The area **A** is the set of all other documents in collection.

The red oval (area **D**) is the intersection of these two circles, and is the set containing all the relevant documents in the retrieved set, known as *true positives*. The error of not retrieving relevant documents, shown by area **B** in Figure 2.2, is called *false negatives*, and the area **C** is the set of non-relevant documents in the retrieved set, called *false positives*.

The metric Recall and Prec are defined as:

$$\text{Recall} = \frac{\textit{true positive}}{\textit{true positive} + \textit{false negative}} = \frac{|\mathbf{D}|}{|\mathbf{B} \cup \mathbf{D}|}$$

$$\text{Prec} = \frac{\textit{true positive}}{\textit{true positive} + \textit{false positive}} = \frac{|\mathbf{D}|}{|\mathbf{C} \cup \mathbf{D}|}$$

In practice, not every relevant document is identified by relevance judgment collection. In the most common form of the process of pooling (see Section 2.2.1 for more on pooling), for example, only the top-d documents in the submitted runs are judged by humans. In Figure 2.3, the set of pooled documents is shown as the bottom circle. The actual Recall and true Prec are then computed by:

$$\text{Recall} = \frac{|\mathbf{E} \cup \mathbf{H}|}{|\mathbf{E} \cup \mathbf{H} \cup \mathbf{F} \cup \mathbf{K}|}$$

$$\text{Prec} = \frac{|\mathbf{E} \cup \mathbf{H}|}{|\mathbf{E} \cup \mathbf{H} \cup \mathbf{G} \cup \mathbf{L}|}.$$

However, relevance labels for documents in areas **E**, **L** and **K**, excluded from the relevance judgments, are not known, the size of which is assumed as zero by Recall in practice.

Documents in the areas **K** and **E** are usually assumed to be empty (or irrelevant) by most evaluation technologies. If the size of area **E** is large, the uncertainty (possible score change if additional documents are judged) of computed effectiveness score is likely to be great. The large size of **E** also indicates an inferior pooling process. To build more reliable relevance judgments, more documents retrieved by good IR systems need to be pooled and judged. A straight-forward way is to increase the pooling depth (that is, enlarge the bottom circle in Figure 2.3), which needs to be traded off with the judging costs (see more discussions in Chapter 4). But there is no guarantee that this will enlarge $\mathbf{F} \cup \mathbf{H}$.

In real search task, documents in the retrieved set are sequentially ordered in the ranked list (or run), and usually examined by the user from the top to the bottom. Therefore, the size of upper right circle in Figure 2.3 is equal to the rank where the user stops checking the results in the ranked list, also called *evaluation depth* (note that, not all metrics have evaluation depth, some of them only have expected evaluation depth), but not the length of the whole ranked list retrieved by the system. That is, the right circle in Figure 2.2 may be smaller than (but still the subset of) the set of all retrieved documents.

In the metric Prec(k), an example of Prec with evaluation depth $k = 10$, only the top-10 documents in $\boldsymbol{r}_{s,t}$ are deemed as retrieved documents ($|\mathbf{E} \cup \mathbf{H} \cup \mathbf{G} \cup \mathbf{L}| = 10$), and documents after rank 10 are all ignored by Prec(k).

An example of computing Recall and Prec at each depth k of ranked lists returned by Google and Bing for query *"phd graduate jobs australia"* described in Chapter 1 is shown in Table 2.2. From shallow ranks to deep ranks, the Recall scores are non-decreasing and

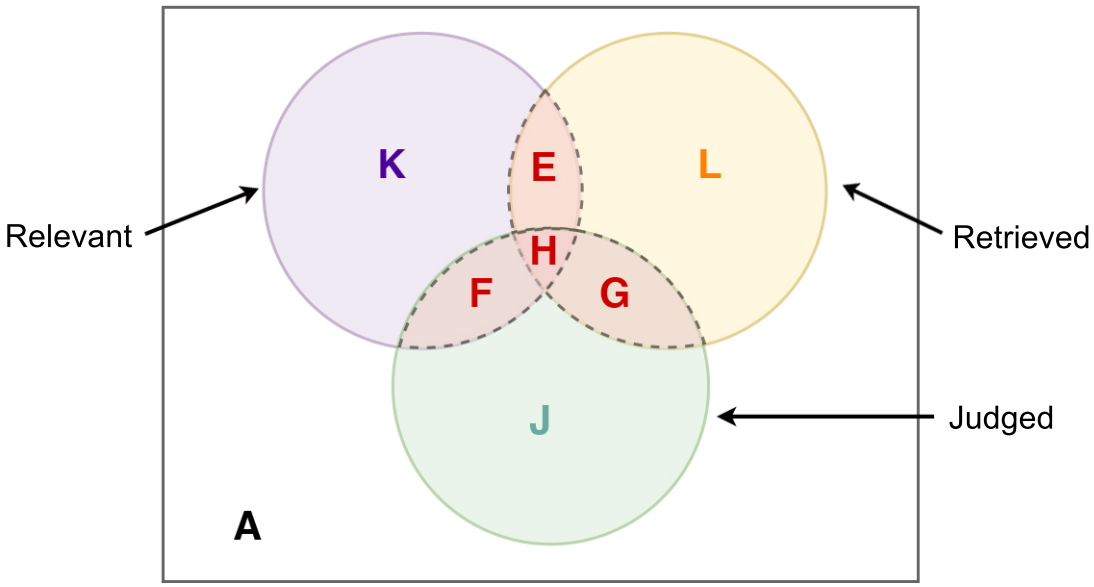


Figure 2.3: Document classifications concerned with Recall and Prec. Documents in relevance judgments for the given topic t are represented by the bottom circle ($\mathbf{J} \cup \mathbf{F} \cup \mathbf{H} \cup \mathbf{G}$). The set of all relevant documents for topic t is represented by the left circle ($\mathbf{E} \cup \mathbf{H} \cup \mathbf{F} \cup \mathbf{K}$). The set of all retrieved documents in the run $r_{s,t}$ is represented by the right circle ($\mathbf{E} \cup \mathbf{H} \cup \mathbf{G} \cup \mathbf{L}$).

increase at depths where relevant documents are found. However as the total number of relevant documents for the query is unknown, it can only be represented by R_t (defined as the total number of relevant documents, that is the upper left circle in Figure 2.3) in the result of Recall. In practice, R_t is hard to know and may be distinct for different users. For search engine users, Prec and Recall always draw the same conclusion when comparing systems. If the $\text{Prec}(r_{s1,t}, k)$ for the run $r_{s1,t}$ at depth k is greater than $\text{Prec}(r_{s2,t}, k)$ (where $r_{s1,t}$ and $r_{s2,t}$ are two different runs), we will see $\text{Recall}(r_{s1,t}, k)$ is higher than $\text{Recall}(r_{s2,t}, k)$ as well. However, Recall requires the value of R_t, Prec could be simply computed by dividing the given evaluation depth k. Therefore only the Prec at depth k (pre-defined) is calculated when comparing two rankings for a given query in most published research.

As Table 2.2 shows, the Prec(10) score for $r_{G,t}$ is $7/10$ and for $r_{B,t}$ is $5/10$. As we stated above, the total number of relevant documents is very difficult to know, so R-Prec and Recall cannot be computed in this example.

Generally, Recall measures the system performance of finding relevant documents in total and Prec measures how well the system filters irrelevant documents out from the retrieved ranking list. Another well-known measure computing the *harmonic mean* of Recall and Prec, called *F measure*, is introduced by Rijsbergen [77] as:

$$\text{F} = \frac{2}{1/\text{Recall} + 1/\text{Prec}} = \frac{2 \cdot \text{Recall} \cdot \text{Prec}}{\text{Recall} + \text{Prec}}. \tag{2.2}$$

k	1	2	3	4	5	6	7	8	9	10	...
$\mathsf{Rel}(\boldsymbol{r_{G,t}}(k), \boldsymbol{t})$	1	1	1	1	0	0	1	1	1	0	...
$\mathsf{Recall}(\boldsymbol{r_{G,t}}, k)$	$1/\boldsymbol{R_t}$	$2/\boldsymbol{R_t}$	$3/\boldsymbol{R_t}$	$4/\boldsymbol{R_t}$	$4/\boldsymbol{R_t}$	$4/\boldsymbol{R_t}$	$5/\boldsymbol{R_t}$	$6/\boldsymbol{R_t}$	$7/\boldsymbol{R_t}$	$7/\boldsymbol{R_t}$	...
$\mathsf{Prec}(\boldsymbol{r_{G,t}}, k)$	1/1	2/2	3/3	4/4	4/5	4/6	5/7	6/8	7/9	7/10	...
$\mathsf{Rel}(\boldsymbol{r_{B,t}}(k), \boldsymbol{t})$	1	1	1	0	1	0	0	0	1	0	...
$\mathsf{Recall}(\boldsymbol{r_{B,t}}, k)$	$1/\boldsymbol{R_t}$	$2/\boldsymbol{R_t}$	$3/\boldsymbol{R_t}$	$3/\boldsymbol{R_t}$	$4/\boldsymbol{R_t}$	$4/\boldsymbol{R_t}$	$4/\boldsymbol{R_t}$	$4/\boldsymbol{R_t}$	$5/\boldsymbol{R_t}$	$5/\boldsymbol{R_t}$	...
$\mathsf{Prec}(\boldsymbol{r_{B,t}}, k)$	1/1	2/2	3/3	3/4	4/5	4/6	4/7	4/8	5/9	5/10	...

Table 2.2: The binary relevance score of document, Recall and Prec at depth k for ranked lists retrieved by Google ($\boldsymbol{r_{G,t}}$) and Bing ($\boldsymbol{r_{B,t}}$) respectively, for the query $\boldsymbol{t}$ = *"phd graduate jobs australia"* mentioned in Chapter 1, with denoting $\boldsymbol{R_t}$ the total number of relevant documents for the topic $\boldsymbol{t}$.

The effectiveness in different aspects measured by Recall and Prec can be summarized by the F score into a single number.

R-Precision Instead of consider documents in sets regardless of ordering, now we turn back run $\boldsymbol{r_{s,t}}$ (the ranked list retrieved by the system $\boldsymbol{s}$ for the topic $\boldsymbol{t}$).

The Prec ($\boldsymbol{R}$) (also named as R-Prec) is the Prec at rank $\boldsymbol{R}$ for the topic which has $\boldsymbol{R}$ relevant documents in the pool (that is, $\boldsymbol{R} = |\mathbf{F} \cup \mathbf{H}|$). Users are assumed to know the value of $\boldsymbol{R}$ before scanning the retrieved result, and examine the ranked list $\boldsymbol{r_{s,t}}$ from top to bottom till rank $\boldsymbol{R}$. As can be seen, R-Prec is sensitive to $\boldsymbol{R}$, but not top-weighted (treat documents at top ranks more important than documents at bottom ranks). Therefore, the metric of average precision is introduced.

Average Precision *Average Precision* (AP) at depth k, a widely used metric for IR, average values of all $\mathsf{Prec}(\boldsymbol{r_{s,t}}, i)$ where a relevant document is retrieved at depth i, that is (assuming binary judgments are used):

$$\mathsf{AP}(\boldsymbol{r_{s,t}}, k) = \frac{1}{\boldsymbol{R_t}} \sum_{i=1}^{k} \mathsf{Prec}(\boldsymbol{r_{s,t}}, i) \cdot \mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) \tag{2.3}$$

where $\boldsymbol{R_t}$ is the total number of relevant documents in the relevance judgments for topic $\boldsymbol{t}$, and k is the evaluation depth. In many cases AP (without specifying k) is used, assumed to be the sum over all known relevant documents.

For the example in Table 2.2, the AP, with evaluation depth $k = 10$, for $\boldsymbol{r_{G,t}}$ and $\boldsymbol{r_{B,t}}$ can be calculated as:

$$\begin{aligned}\mathsf{AP}(\boldsymbol{r_{G,t}}, 10) &= (1/1 + 2/2 + 3/3 + 4/4 + 5/7 + 6/8 + 7/9)/\boldsymbol{R_t} \\ &= 6.242/\boldsymbol{R_t} \\ \mathsf{AP}(\boldsymbol{r_{B,t}}, 10) &= (1/1 + 2/2 + 3/3 + 4/5 + 5/9)/\boldsymbol{R_t} \\ &= 4.356/\boldsymbol{R_t}\end{aligned}$$

As can be seen in the calculation, computing an AP score requires the total number of relevant documents, that is $\boldsymbol{R_t}$. If $\boldsymbol{R_t}$ is unknown before the evaluation, and the evaluation depths are shallow (such as $k = 10$ in the example), AP scores can be compared, even if they cannot be computed.

AP takes into account the ranking of all of the relevant documents and also heavily depends on the positions of the top-ranked relevant documents. That is, it measures how well the system finds all of the relevant documents, and how well it then clusters them at the top of the run.

Reciprocal Rank *Reciprocal Rank* (RR) is used to measure how high in the ranking the first relevant document is retrieved, and is computed as:

$$\mathsf{RR}(\boldsymbol{r_{s,t}}, k) = \begin{cases} \frac{1}{\min\{\, i \mid \mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) = 1 \,\}} & i \leq k \\ 0 & i > k. \end{cases}$$

This metric is employed when the users of the system are presumed to need only one relevant document, and to stop checking the ranked list once they find one, such as users carrying out question answering task.

Discounted Cumulative Gain Another well established IR evaluation metric is *Discounted Cumulative Gain* (DCG) proposed by Järvelin and Kekäläinen [44], which assumes both that users gain more from documents with higher relevance score, and that documents at deeper ranks in the run are less helpful than the documents at top. The DCG score is the accumulated discounted relevance gained by users starting from the top of the ranking and proceeds though to some depth k to the deep ranks.

The relevance of documents can be in multiple levels rather than binary. Categorical document relevance categories (such as `H`, `R`, `M` and `N` in Sormunen [99]) cannot be directly used in mathematical computations, thus a *gain* function which maps categories to numeric values needs to be employed. One widely-used gain function of rank i, denoted as $\mathsf{Gain}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t})$, is defined to map the relevance score (or category) $\mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t})$ to the relevance gain value obtained by users, which can be defined (in one version) as (assuming

ordinal $\mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) \in [0, \boldsymbol{max}]$):

$$\mathsf{Gain}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) = \frac{2^{\mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t})} - 1}{2^{\boldsymbol{max}} - 1}. \tag{2.4}$$

Another gain function, for assigning numeric values to categories of Sormunen judgments, can be (for example):

$$\mathsf{Gain}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) = \begin{cases} 1 & \text{if } \mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) = \texttt{H} \\ 0.5 & \text{if } \mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) = \texttt{R} \\ 0.25 & \text{if } \mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) = \texttt{M} \\ 0 & \text{if } \mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) = \texttt{N}. \end{cases}$$

As discussed in Section 2.2, for different people, perceptions of relevance could be different. Some research for refining gain functions was carried out by Turpin et al. [105]. They found that a linear profile (view of relevance), rather than exponential profile, was closer to users in their experiment in general. But they noted that a single gain profile might not be sufficient for IR system evaluations. In Chapter 5, we also explore the distribution of relevance in pairwise judgments by looking at the normalized relevance scores of documents and the relevance ratios of document pairs assigned by crowd-workers, in order to obtain a gain function based on the opinions of human assessors.

Similarly, one version of the *discount function*, which could also be varied for different users, denoted as $\mathsf{Disc}(i)$, provides the reduction (weight) to the relevance gain at rank i as:

$$\mathsf{Disc}(i) = \frac{1}{\log_2(1+i)}. \tag{2.5}$$

Considering with the position and the relevance score of documents in the run, DCG at evaluation depth k is defined as the sum of weighted relevance gain:

$$\mathsf{DCG}(\boldsymbol{r_{s,t}}, k) = \sum_{i=1}^{k} \mathsf{Gain}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) \cdot \mathsf{Disc}(i). \tag{2.6}$$

If we assume that the documents shown in Table 2.2 are instead judged using a ordinal 4-level relevance scale with numeric relevance score from 0 (not relevant) to 3 (highly relevant), the weighted relevance gain computed via Equation 2.4 and DCG score at each depth can be computed as shown in Table 2.3. Note that DCG is non-decreasing with depth, and can grow without limit.

Because the weight assigned by the discount function (Equation 2.5) decreases with the depth k, DCG is a top-weighted metric (that is focuses more on top ranks rather than deep ranks). However the DCG value that might be attained is dependent on the total number of relevant documents for the query and does not have any upper limit, which

k	1	2	3	4	5	6	7	8	9	10	...
Disc(k)	1	0.63	0.50	0.43	0.39	0.36	0.33	0.32	0.30	0.29	...
Rel($\boldsymbol{r}_{G,t}(k), \boldsymbol{t}$)	3	2	3	1	0	0	3	1	1	0	...
Gain($\boldsymbol{r}_{G,t}$, k)	7/7	3/7	7/7	1/7	0	0	7/7	1/7	1/7	0	...
DCG($\boldsymbol{r}_{G,t}$, k)	1.00	1.27	1.77	1.83	1.83	1.83	2.17	2.21	2.25	2.25	...
Rel($\boldsymbol{r}_{B,t}(k), \boldsymbol{t}$)	3	3	2	0	1	0	0	0	1	0	...
Gain($\boldsymbol{r}_{B,t}$, k)	7/7	7/7	3/7	0	1/7	0	0	0	1/7	0	...
DCG($\boldsymbol{r}_{B,t}$, k)	1.00	1.63	1.85	1.85	1.90	1.90	1.90	1.90	1.94	1.94	...

Table 2.3: The relevance score of document in 4-level scale (0, 1, 2, 3), Disc, Rel($\boldsymbol{r}_{s,t}(k), \boldsymbol{t}$), Gain (computed using Equation 2.4) and DCG at depth k for ranked lists retrieved by Google ($\boldsymbol{r}_{G,t}$) and Bing ($\boldsymbol{r}_{B,t}$) respectively, for the query $\boldsymbol{t}$ = *"phd graduate jobs australia"* mentioned in Chapter 1.

may become a problem when DCG scores for a system are averaged across multiple topics with different numbers of relevant documents. Thus, *Normalized Discounted Cumulative Gain* (NDCG) [13] is proposed to normalize the DCG score by dividing the DCG value of the *perfect* document ranking for the same topic. For instance, the relevance scores of top-10 documents retrieved by Google for the query shown in Table 2.3 are:

$$3, 2, 3, 1, 0, 0, 3, 1, 1, 0.$$

The perfect ranking for the query might be (assuming there are two more relevant documents with relevance score of 3 and 2 respectively after rank 10 of $\boldsymbol{r}_{G,t}$, an assumption that might or might not be upheld if further judgments were undertaken):

$$3, 3, 3, 3, 2, 2, 1, 1, 1, 0$$

whose DCG(k) score, also known as IDCG(k) score, at each rank would be:

$$1.00, 1.63, 2.13, 2.56, 2.73, 2.88, 2.93, 2.97, 3.02, 3.02.$$

The NDCG with evaluation depth k is defined as:

$$\mathsf{NDCG}(\boldsymbol{r}_{s,t}, k) = \frac{\mathsf{DCG}(\boldsymbol{r}_{s,t}, k)}{\mathsf{IDCG}(\boldsymbol{r}_{s,t}, k)}. \tag{2.7}$$

Thus NDCG scores for $\boldsymbol{r}_{G,t}$ at each rank from 1 to 10 are:

$$1.00, 0.78, 0.83, 0.72, 0.67, 0.64, 0.74, 0.74, 0.75, 0.75.$$

However, similar to AP, the problem associated with NDCG(k) is that finding the perfect ranking to rank k requires relevance scores of top-k relevant documents, which means that the ranked list needs to be checked until the most relevant k documents are found, rather than stopping checking at rank k of the run. For example in Table 2.3, if we keep scanning the ranked list $\boldsymbol{r}_{G,t}$ after rank 10, we might find as many as 10 documents with relevance score of 3 and so the perfect ranking would change to be:

$$3, 3, 3, 3, 3, 3, 3, 3, 3, 3.$$

Unlike when computing AP, we can stop scanning once 10 highly relevant documents ($\mathsf{Rel}(\boldsymbol{r}_{G,t}(d), \boldsymbol{t}) = 3$) are found. Otherwise (for example if $|\{\boldsymbol{r}_{G,t}(d) \mid \mathsf{Rel}(\boldsymbol{r}_{G,t}(d), \boldsymbol{t}) = 3\}| < 10$) the run needs to be scanned though to the end same as with the computation of AP, which requires the total number of relevant documents. In general, NDCG(k) assumes the numbers of documents in different relevance levels are known, which may not be realistic when k is large.

Rank-Biased Precision Rank-Biased Precision (RBP) proposed by Moffat and Zobel [73] simulates the behavior of IR system users with a patience parameter ϕ, the probability that they keep going to examine the document at rank $i+1$ after checking the document at rank i. The RBP score is calculated as the weighted sum of utility (relevance) gained at ranks. The weight of rank i (beginning from $i = 1$) is given by $(1-\phi)\phi^{i-1}$, a function that serves a similar role to the Disc that assigns decreasing factors for ranked documents in DCG.

With the persistence parameter ϕ, the expected viewing depth can be computed as $k_{\text{exp}} = 1/(1-\phi)$. Thus, to discover an appropriate value of ϕ for RBP when using it in a particular case, an experiment for finding the expected depth where users stop examining can be carried out. Assuming the expected searching depth of users is k_{exp}, the persistence parameter ϕ for RBP can be obtained by:

$$\phi = 1 - \frac{1}{k_{\text{exp}}}.$$

Thus, the greater the value of ϕ, the deeper the expected evaluation depth.

For the ranked list $\boldsymbol{r}_{s,t}$ retrieved by system $\boldsymbol{s}$ for some topic $\boldsymbol{t}$, the expected relevance value gained per document checked by users with patience ϕ, RBP can be calculated as:

$$\mathsf{RBP}(\boldsymbol{r}_{s,t}, \phi) = (1-\phi) \cdot \sum_{i=1}^{\infty} \mathsf{Gain}(\boldsymbol{r}_{s,t}(i), \boldsymbol{t}) \cdot \phi^{i-1}.$$

This definition of RBP assumes that relevance scores of all documents for topic t are known and the length of ranked list is infinite, which are impossible in practice. If we denote J_t as the set of judged documents in $r_{s,t}$, that is $J_t = \mathbf{H} \cup \mathbf{G}$ referring to Figure 2.3, the RBP can be instead computed as:

$$\mathsf{RBP}(r_{s,t}, \phi) = (1-\phi) \cdot \sum_{r_{s,t}(i) \in J_t} \mathsf{Gain}(r_{s,t}(i), t) \cdot \phi^{i-1}.$$

Once this is done, each RBP score can be associated with a *residual* which tells the maximum score that could be added on as a result of unjudged documents (if all unjudged documents are maximally relevant), computed as the weighted sum of unjudged documents:

$$\mathsf{RBPres}(r_{s,t}, \phi) = (1-\phi) \cdot \sum_{r_{s,t}(i) \notin J_t} \phi^{i-1},$$

assuming that the maximum gain is 1. In particular, if J_t is only pooled to the depth k of the run, the upper bound of the RBPres can be given according to the properties of the geometric sequence as ϕ^k [73].

If RBP is used to evaluate the ranked list shown in Table 2.3, the effectiveness scores for $r_{G,t}$ and $r_{B,t}$ with evaluation depth $k = 10$ are (assuming $\phi = 0.9$, and so the $k_{\text{exp}} = 10$):

$$\begin{aligned}
\mathsf{RBP}(r_{G,t}, \phi = 0.9, k = 10) =& (1-0.9) \cdot (1 \cdot 0.9^0 + 3/7 \cdot 0.9^1 + 1 \cdot 0.9^2 + 1/7 \cdot 0.9^3 + 1 \cdot 0.9^6 \\
& + 1/7 \cdot 0.9^7 + 1/7 \cdot 0.9^8) \\
=& 0.296 \\
\mathsf{RBP}(r_{B,t}, \phi = 0.9, k = 10) =& (1-0.9) \cdot (1 \cdot 0.9^0 + 1 \cdot 0.9^1 + 3/7 \cdot 0.9^2 + 1/7 \cdot 0.9^4 \\
& + 1/7 \cdot 0.9^8) \\
=& 0.240.
\end{aligned}$$

The RBPres for $r_{G,t}$ and $r_{B,t}$ are both $0.9^{10} = 0.349$.

When ϕ drops to 0.5, RBP becomes a shallower metric whose expected evaluation depth is smaller, $k_{\text{exp}} = 2$. The scores of $r_{G,t}$ and $r_{B,t}$ are:

$$\begin{aligned}
\mathsf{RBP}(r_{G,t}, \phi = 0.5, k = 10) =& (1-0.5) \cdot (1 \cdot 0.5^0 + 3/7 \cdot 0.5^1 + 1 \cdot 0.5^2 + 1/7 \cdot 0.5^3 + 1 \cdot 0.5^6 \\
& + 1/7 \cdot 0.5^7 + 1/7 \cdot 0.5^8) \\
=& 0.750 \\
\mathsf{RBP}(r_{B,t}, \phi = 0.5, k = 10) =& (1-0.5) \cdot (1 \cdot 0.5^0 + 1 \cdot 0.5^1 + 3/7 \cdot 0.5^2 + 1/7 \cdot 0.5^4 \\
& + 1/7 \cdot 0.5^8) \\
=& 0.808.
\end{aligned}$$

The RBPres of $\boldsymbol{r}_{G,t}$ and $\boldsymbol{r}_{B,t}$ both decrease to $0.5^{10} = 0.001$. As can be seen, the smaller that ϕ is, the greater weights the top ranks have, and the lower the residual is at any given judgment depth.

When focusing on the top two ranks, the retrieved result of $\boldsymbol{r}_{B,t}$ (two fully relevant documents) is better than the result of $\boldsymbol{r}_{G,t}$ (one fully relevant document, and one partially relevant document). Thus the RBP score of $\boldsymbol{r}_{B,t}$ is greater than the RBP score of $\boldsymbol{r}_{G,t}$ when ϕ is low, that is 0.5. However, the system $\boldsymbol{G}$ performs better than system $\boldsymbol{B}$ at later ranks. Thus, when ϕ increases to 0.9, and RBP becomes a deeper metric, $\mathsf{RBP}(\boldsymbol{r}_{B,t}, \phi = 0.9, k = 10)$ is lower than $\mathsf{RBP}(\boldsymbol{r}_{G,t}, \phi = 0.9, k = 10)$.

The residual is an upper bound on the *uncertainty* of the RBP score and can be used to quantify the measurement fidelity. Moffat et al. [74] summarize the assumptions made by different metrics about user behaviors when sequentially scanning a ranked list (also known as *user models*, and described in more detail in Section 2.3.2). These assumptions implicitly suggest the expected user examining depth modeled by metrics, and which could then be used to connect RBP with other recall-based metrics such as AP and NDCG. Using Formula 2.8, the "best" ϕ whose corresponding d_{exp} is the closest to the expected examining depth of the estimated metric can be computed. The RBP with the obtained ϕ can then be used to compute residuals (upper bounds on the uncertainty) for the estimating metric. See Chapter 4 for a detailed exploration of these ideas.

2.3.2 Metric User Models and Properties

One way of considering the effectiveness metrics is to regard them as measuring the rate of relevance gain obtained by users as they scan the ranked result list. The *user model* behind a metric simulates the user behaviors and expectations, which is helpful for selecting appropriate metrics to assess systems for a given task.

Weight Function Moffat, Thomas, and Scholer [71] introduce the idea of weight functions to assign the value of $\mathsf{W}_\mathsf{M}(i)$ as the weight associated with the relevance gain value $\mathsf{Gain}(\boldsymbol{r}_{s,t}(i), \boldsymbol{t})$ at rank i, with $\sum_{i=1}^{\infty} \mathsf{W}_\mathsf{M}(i) = 1$. The metric score (also the weighted relevance gain obtained by the user) is then expressed as [71]:

$$\mathsf{M}(\boldsymbol{r}_{s,t}) = \sum_{i=1}^{\infty} \mathsf{W}_\mathsf{M}(i) \cdot \mathsf{Gain}(\boldsymbol{r}_{s,t}(i), \boldsymbol{t}). \tag{2.8}$$

In the case of $\mathsf{Prec}(k)$, which is not a top-weighted metric, the top k documents receive non-zero and equal weights, $1/k$. Documents ranked after the depth k are modeled as never being examined by users, so they receive a weight of zero. In general, the weight

function of Prec(k) can be specified as [70]:

$$\mathsf{W}_{\mathsf{Prec}}(i,k) = \begin{cases} \frac{1}{k} & 0 < i \leq k \\ 0 & i > k. \end{cases} \tag{2.9}$$

As was described in Section 2.3, AP averages the sum of all Prec(k) where the relevant document is founded at rank k. The weight function of AP is given as:

$$\mathsf{W}_{\mathsf{AP}}(i, \boldsymbol{r}_{\boldsymbol{r}_{s,t}}) = \frac{1}{\boldsymbol{R_t}} \cdot \sum_{j=i}^{\infty} \frac{\mathsf{Gain}(\boldsymbol{r}_{\boldsymbol{s,t}}(j), \boldsymbol{t})}{j}, \tag{2.10}$$

where $\boldsymbol{r}_{\boldsymbol{s,t}}$ is the evaluated run, $\boldsymbol{t}$ is the topic, and $\boldsymbol{R_t}$ is the total number of relevant documents for topic $\boldsymbol{t}$. As shown in Equation 2.10, the weight function of AP not only changes with rank i, but also depends on the evaluated run $\boldsymbol{r}_{\boldsymbol{s,t}}$. The relevance gain value $\mathsf{Gain}(\boldsymbol{r}_{\boldsymbol{s,t}}(i), \boldsymbol{t})$ of rank i depends on the relevance of the document ranked at position i in $\boldsymbol{r}_{\boldsymbol{s,t}}$. Note that because $\mathsf{W}_{\mathsf{AP}}(i, \boldsymbol{r}_{\boldsymbol{r}_{s,t}})$ is a non-increasing function, AP is a top-weighted metric. In addition, $\mathsf{W}_{\mathsf{AP}}(i, \boldsymbol{r}_{\boldsymbol{r}_{s,t}})$ requires knowledge of the total number of relevant documents for the given topic, that is $\boldsymbol{R_t}$.

The metric RR simulates users who only need one fully relevant document and stop checking when they find one. Thus the weight function of RR, assuming the first relevant document is found at rank j in the evaluated run $\boldsymbol{r}_{\boldsymbol{s,t}}$, is:

$$\mathsf{W}_{\mathsf{RR}}(i, \boldsymbol{r}_{\boldsymbol{r}_{s,t}}) = \begin{cases} \frac{1}{j} & 0 < i \leq j \\ 0 & i > j \end{cases} \tag{2.11}$$

where j is not a fixed number, but related to the given run $\boldsymbol{r}_{\boldsymbol{r}_{s,t}}$. Similar to the weight function of AP, weights of RR are dependent in part on the run being viewed. In this sense, AP and RR are *adaptive* metrics, since the user is assumed to modify their behavior based on the documents inspected on the ranking is viewed (In the case of AP, its adaptivity is based on the entirety of the run, in RR it is based on a prefix of the run).

According to the definition of DCG, its weight function is the same as the discount function:

$$\mathsf{W}_{\mathsf{DCG}}(i) = \mathsf{Disc}(i) = \frac{1}{\log_2(i+1)}, \tag{2.12}$$

which assigns decreasing weight to the relevance gain at rank i to achieve top-weightedness. But note that $\sum_{i=1}^{\infty} \mathsf{W}_{\mathsf{DCG}}(i) > 1$ and no upper limit, as a refinement *scaled discounted cumulated gain* (SDCG) is proposed to normalize the $\mathsf{W}_{\mathsf{DCG}}(i)$ within range $[0, 1]$. The weight

function of $\mathsf{SDCG}(k)$, whose evaluation depth is k, is given as [71]:

$$\mathsf{W}_{\mathsf{SDCG}}(i,k) = \begin{cases} \frac{1}{\log_2(i+1)} / \sum_{i=1}^{k} \frac{1}{\log_2(i+1)} & 0 < i \leq k \\ 0 & i > k \end{cases} \tag{2.13}$$

with $\sum_{i=1}^{k} \mathsf{W}_{\mathsf{SDCG}}(i,k) = 1$, by definition.

The weight function of $\mathsf{RBP}(\phi)$ is also obvious to be acquired from the definition as (with the patience parameter ϕ):

$$\mathsf{W}_{\mathsf{RBP}}(i,\phi) = (1-\phi) \cdot \phi^{i-1}, \tag{2.14}$$

which implies that the lower the ϕ value is, the greater the weights assigned to the top-ranked documents. In other words, $\mathsf{RBP}(\phi)$ models that users with less patience pay more attention to the top ranks and have reduce interest in examining document at deeper ranks.

Continuation Probability Function Weight functions can be interpreted and converted into conditional continuation probability, denoted as $\mathsf{C}_{\mathsf{M}}(i)$, whose relationship with $\mathsf{W}_{\mathsf{M}}(i)$ is [71]:

$$\mathsf{C}_{\mathsf{M}}(i) = \frac{\mathsf{W}_{\mathsf{M}}(i+1)}{\mathsf{W}_{\mathsf{M}}(i)}, \tag{2.15}$$

that is the conditional probability that the user, having checked the document at rank i, continues examining the ranking and moves to rank $i+1$. In the case of $\mathsf{RBP}(\phi)$, its continuation probability function could be simply defined as:

$$\mathsf{C}_{\mathsf{RBP}}(i,\phi) = \phi,$$

and does not change with the rank i.

The RR simulates users who only need one fully relevant document and stop checking once they find one (such as users carrying out question answering tasks):

$$\mathsf{C}_{\mathsf{RR}}(i) = \begin{cases} 1 & 0 < i < \min\{\, j \mid \mathsf{Rel}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(j), \boldsymbol{t}) = 1\} \\ 0 & i \geq \min\{\, j \mid \mathsf{Rel}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(j), \boldsymbol{t}) = 1\}. \end{cases}$$

where as before, $\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}$ is the evaluated run.

For other metrics with a fixed evaluation depth k such $\mathsf{Prec}(k)$, the continuation probability before depth k is one, and becomes to zero at and after depth k (keep checking

and quit at depth k):

$$\mathsf{C}_{\mathsf{Prec}}(i,k) = \mathsf{C}_{\mathsf{SDCG}}(i,k) = \begin{cases} 1 & 0 < i < k \\ 0 & i \geq k. \end{cases}$$

The continuation probability function of most metrics can be computed according to Equation 2.15. But a special case is needed for the metric AP. As we can see from the weight function of AP (Equation 2.10), the weight of relevance value at rank i is dependent on how many relevant documents in total, how many relevant document have (or have not) been found, and the ranks at which they would be found of the entire ranking was viewed. User behavior as simulated by AP keeps checking documents until a relevant document is found. At ranks of these relevant documents, users have probabilities to quit or continue the scanning which are dependent on the ranks of relevant document that have not been found yet (if all relevant documents have been found, users will definitely exit). In other words, AP assumes that both the $\boldsymbol{R_t}$ and ranks of relevant documents are known before scanning, which is inconsistent with any assumptions in regard to top-down viewing. The continuation probability function of AP can be summarized by [71]:

$$\mathsf{C}_{\mathsf{AP}}(i) = \frac{\sum_{j=i+1}^{\infty} \mathsf{Gain}(\boldsymbol{r_{s,t}}(j), \boldsymbol{t})/j}{\sum_{j=i}^{\infty} \mathsf{Gain}(\boldsymbol{r_{s,t}}(j), \boldsymbol{t})/j}$$

where, as a special case $0/0$ is deemed to be zero.

Metric AP is usually deemed as being system-oriented, rather than user-oriented [78], because it is not based on user models but only considers the evaluated runs and test collections. Robertson [78] interprets AP in terms of a user model in order to improve the understanding of what exactly is measured by AP. The model assumes that relevance is binary, and the relevance of a document considered by the user is independent of other documents viewed by the user. Then, the contribution of the document at rank i to the final AP score of the evaluated run $\boldsymbol{r_{s,t}}$, denoted as $\mathsf{P}_i(\boldsymbol{r_{s,t}})$, could be defined as:

$$\mathsf{P}_i(\boldsymbol{r_{s,t}}) = \begin{cases} \frac{1}{i} \sum_{j=1}^{i} \mathsf{Rel}(\boldsymbol{r_{s,t}}(j), \boldsymbol{t}) & \mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) = 1 \\ 0 & \mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t}) = 0 \end{cases}$$

$$= \frac{\mathsf{Rel}(\boldsymbol{r_{s,t}}(i), \boldsymbol{t})}{i} \cdot \sum_{j=1}^{i} \mathsf{Rel}(\boldsymbol{r_{s,t}}(j), \boldsymbol{t}),$$

that is, when the document at rank i is relevant, P_i is equal to the precision at depth i, otherwise the contribution is zero. Normally, AP could be obtained by:

$$\mathsf{AP}(\boldsymbol{r_{s,t}}) = \frac{1}{\boldsymbol{R_t}} \sum_{i=1}^{\infty} \mathsf{P}_i(\boldsymbol{r_{s,t}}).$$

In the probabilistic user model defined for AP, Robertson [78] further denotes $p(i)$ as the probability that the user is satisfied (alternatively, has been frustrated, exhausted, or for other reasons) with the viewed results so far, and stops scanning the ranking at depth i. Specially, $p(i) = 0$ when $\mathsf{Rel}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(i), \boldsymbol{t}) = 0$, and $\sum_{i=1}^{\infty} p(i) = 1$. A new proposed metric called Normalized Cumulative Precision (NCP) which computes the expected precision viewed by the user [78]:

$$\mathsf{NCP}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}) = \sum_{i=1}^{\infty} p(i) \cdot \mathsf{P}_i(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}).$$

If all the non-zero $p(i)$ (the document at rank i is relevant) are set as $1/\boldsymbol{R}_{\boldsymbol{t}}$, the original AP could be obtained. Thus Robertson concludes that AP is the expected precision with an assumption that the probabilities of users stopping inspection where they observe a relevant document are equal for all the positions with relevant documents.

Using this model, RR could be interpreted as the precision that users stop scanning when they find the first relevant document, that is $p(j) = 1$, where j is the rank where the first relevant document is found in the evaluated run.

Goal-Sensitive Metrics As can be seen from the continuation probability function of different metrics listed above, $\mathsf{C}_{\mathsf{M}}(i)$ of metrics such as $\mathsf{Prec}(k)$ and $\mathsf{SDCG}(k)$ only change with rank i. Similarly, the value of $\mathsf{C}_{\mathsf{RBP}}(i, \phi)$ is always ϕ. The user models of these metrics whose continuation probability functions are only related to rank, or metric parameter, without considering the relevance gained (documents viewed) by users till the rank i, are known as *static* user models. In contract, metrics such as AP and RR, are sensitive to the accumulated relevance to rank i, that is $\sum_{j=1}^{i} \mathsf{Gain}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(j), \boldsymbol{t})$, the user model of which is *adaptive*.

The $\mathsf{C}_{\mathsf{M}}(i)$ could have connection with other factors, such as the total number of relevant documents, or total value of relevance, expected by the user. Moffat, Scholer, and Thomas [70] proposed a RBP-like metric, called $\mathsf{INSQ}(T)$, the user of which is assumed to expect T relevant documents in total, and its weight function is given as:

$$\mathsf{W}_{\mathsf{INSQ}}(i, T) = \frac{1}{S_{2T-1}} \cdot \frac{1}{(i + 2T - 1)^2},$$

where S_{2T-1} is a constant related to T, that is, $S_{2T-1} = (\pi^2/6) - (\sum_{i=1}^{2T-1} 1/i^2)$. And hence, the conditional continuation probability at rank i is:

$$\mathsf{C}_{\mathsf{INSQ}}(i, T) = \frac{(i + 2T - 1)^2}{(i + 2T)^2}.$$

As can be seen, $\mathsf{W}_{\mathsf{INSQ}}(i, T)$ is inversely proportion to T, that is, the smaller the T is, the

more important the top ranks are treated by metric INSQ(T). The $\mathsf{C}_{\mathsf{INSQ}}(i, T)$ is an increasing function of rank i, and it is distinct for different value of T. Users with greater relevance expectation T, modeled by INSQ(T), are more patient when examining the ranking. The expected examining depth associated with INSQ(T) is around $2T + 0.5$ [70], which solely depends on T rather than what have been viewed by the user.

To improve INSQ(T) to an adaptive metric, INST($T, \boldsymbol{r}_{s,t}$) proposed by Moffat, Thomas, and Scholer [71] define T_i as an estimation of remaining relevance expected by the user till depth i, which is computed as:

$$T_i = T - \sum_{j=1}^{i} \mathsf{Gain}(\boldsymbol{r}_{s,t}(j), \boldsymbol{t})$$

where $\boldsymbol{r}_{s,t}$ is the evaluated run, and $\mathsf{Gain}(\boldsymbol{r}_{s,t}(j), \boldsymbol{t})$ is the fractional relevance gain provided by the document at rank j. In general, $\sum_{j=1}^{i} \mathsf{Gain}(\boldsymbol{r}_{s,t}(j), \boldsymbol{t})$ is the relevance accumulated till depth i when the user examine the run $\boldsymbol{r}_{s,t}$. The weighted precision metric INST(T) is then defined using the continuation probability function:

$$\begin{aligned}\mathsf{C}_{\mathsf{INST}}(i, T) &= \left(\frac{i + T + T_i - 1}{i + T + T_i}\right)^2 \\ &= \left(\frac{i - 1 + 2T - \sum_{j=1}^{i} \mathsf{Gain}(\boldsymbol{r}_{s,t}(j), \boldsymbol{t})}{i + 2T - \sum_{j=1}^{i} \mathsf{Gain}(\boldsymbol{r}_{s,t}(j), \boldsymbol{t})}\right)^2.\end{aligned}$$

Moffat et al. [75] explore the properties of INST(T), and demonstrate that the weight assigned by INST(T) is smaller when the rank goes deeper (top-weighted). Similar to INSQ(T), when T is smaller, INST(T) would focus more on documents at top ranks. More importantly, if there is little relevance found at top ranks (that is T_i stays large when rank i increases), the user would be more patient, and so the metric will be less top-focused. The expected examining depth of INST(T) is bounded between about $T + 0.25$ (when all viewed documents are relevant) and $2T + 0.5$ (when all encountered documents are irrelevant) [6]. Note that, the $\mathsf{C}_{\mathsf{M}}(i)$ function of INSQ(T) and INST(T) is always smaller than 1, that is, the user modeled by these two metrics have non-zero probability to quit the examining at all ranks i. Therefore, the weight of document at any rank is greater than 0. Like RBP before, there is no fixed evaluation depth for INSQ(T) and INST(T), and users may reach arbitrary depth as they wish. Documents at every rank in the run either contributes to the final score, or are considered as part of the residual score. Overall, the described properties of INST(T) correspond with the observed user behavior [71].

Properties of Desired Metrics According to the analysis of user behaviors in practice, Moffat, Scholer, and Thomas [70] summarize five desirable properties of metrics which reflect user behaviors when examining result ranking:

P1. the model computation should not require properties of the collection (for example, $\boldsymbol{R_t}$), but only depend on the user (searcher) and the part of ranking viewed by the user;

P2. the model should be top-weighted, that is $\mathsf{W}_\mathsf{M}(i) \geq \mathsf{W}_\mathsf{M}(i+1)$, and $\mathsf{W}_\mathsf{M}(i) > 0$ for any rank i as deep as possible; hence the conditional continuation probability $\mathsf{C}_\mathsf{M}(i)$ is always greater than zero;

P3. assuming other factors being equal, the conditional continuation probability of user is non-decreasing with rank i, that is $\mathsf{C}_\mathsf{M}(i) \leq \mathsf{C}_\mathsf{M}(i+1)$;

P4. the model should be adaptive to relevance encountered till rank i, that is, assuming other factors being equal, $\mathsf{C}_\mathsf{M}(i)$ should decrease when relevance is accumulated;

P5. the model should be sensitive to the volume of relevance expected by the searcher in total; assuming other factors being equal, if the user expect to gain more relevance in the search task (that is, when T is larger), then $\mathsf{C}_\mathsf{M}(i)$ should be higher.

For the described current metrics, recall-based metrics such as AP do not satisfy ***P1***. The users modeled by the top-weighted metrics RBP(ϕ), INSQ(T) and INST(T) could examine the run as far as they like (***P2***), and their continuation probabilities do not decrease with rank i (***P3***). The metrics RR, AP and INST(T) are adaptive metrics (***P4***) which consider the accumulated relevance gained by users. The INSQ(T) and INST(T) are designed based on ***P5***, which are goal-sensitive. The Prec(k), SDCG(k) and RBP(ϕ) also could be adjusted for regarding the expectations of users (for example, k could be set to $2T$, the value of ϕ could be chosen as $1 - 1/(2T)$) for each query, but in practice, k and ϕ are usually set for the whole evaluation, rather than change for different queries [70].

As described above, the continuation probability simulated by different metrics could be: a constant through to some depth k, which does not change with rank (such as Prec), and is zero thereafter; or an increasing function of rank (such as SDCG), still zero after some depth k; or can depend on the evaluated run (such as RR). In other words, different metrics measure different aspects of run (or system) performance. What kinds of metrics should be chosen for evaluation is dependent on the expectations and behaviors of real users.

User behaviors in tasks with different complexity levels are also different. For example, users processing question answering (simple) tasks may want to find the answer as soon as possible, which implies that users pay more attention to top ranked documents (that is, the initial patience of these users may be quite low). Therefore, how soon the first relevant document is found might be considered as one of the most important aspects, which could be measured by RR. If the continuation probabilities are varied from user to user, and do not change with documents viewed, RBP might be more helpful to simulate different users. If users expect to find multiple relevant documents ***T***, and the initial patience of users are almost same but might change with rank, INSQ would be a

good choice. However, if the patience of user changes and depends on the relevance of the examined documents and the total expectation $\boldsymbol{T}$, metrics with adaptive user models, such as INST, might be more useful.

Zhang et al. [119] propose a new metric, the Bejeweled Player Model (BPM), which considers the upper limit of both benefit ($\boldsymbol{T}$) and search cost. The BPM simulates users who quit scanning results when either they have found sufficiently many useful documents to meet the total expectation $\boldsymbol{T}$, or have no patience to continue. Moreover, in *Dynamic* BPM, $\boldsymbol{T}$ and $\mathsf{C}_{\mathsf{BPM}}(i)$ may change with what has been viewed by the user, that is, users are assumed to expect more useful information when relevant documents are found, and to lose patience when irrelevant documents are found. Zhang et al. [119] conclude that considering dynamic $\boldsymbol{T}$ and $\mathsf{C}_{\mathsf{BPM}}(i)$ can improve the performance of BPM, and has a better correlation with user satisfaction.

Wicaksono and Moffat [116] analyze the interaction logs of users when they search job advertisements on the `Seek.com` Website[11] to understand a real user model in the practical job search task. The log data of a user includes *impressions* (job summaries are fully displayed to the user above 0.5 second), *clicks* (the links of job advertisements are clicked by the user), and *applications* (jobs are applied by the user). The authors find that job summaries ranked at earlier positions are more likely to be clicked by users (top-weighted), and that users tended to view a few more job summaries after their last clicks (that is, the continuation probability of a user usually does not drop to zero after the rank of the "last" relevant documents considered by the user). Moreover, the probability that a user makes a job application increases with the number of distinct job summaries that the user have clicked. And the continuation probabilities of users generally increase with the rank. Thus, metrics with increasing $\mathsf{C}_{\mathsf{M}}(i)$ function seem to be better when evaluating the search system in this application.

As an application log is always accompanied by a click and an impression log (similarly, before a click log, there is always an impression log), Wicaksono, Moffat, and Zobel [117] use practical user behavior logs to estimate the coefficients in the proposed impression model, which is based on the conditional continuation probability for predicting which documents have high probability to have been examined by the user according to the observed click sequence, and therefore estimate parameters of continuation function $\mathsf{C}_{\mathsf{M}}(i)$ for metrics such as RBP and INSQ.

2.4 Measurements for Data Analysis

2.4.1 Aggregation

Section 2.3 summarized a range of metrics that can be used to examine the performance of a single run $\boldsymbol{r}_{s,t}$ retrieved by system $\boldsymbol{s}$ for topic $\boldsymbol{t}$. In the batch evaluation technique

[11] http://seek.com

described in Section 2.1.2, a system is usually tested with multiple topics and the aggregated value of all per-topic scores $\mathsf{M}(\boldsymbol{r}_{s,t})$ is deemed to be the overall effectiveness of the system.

The most common aggregation technique, and also the technique used in the experiments described in this thesis, is to compute the arithmetic mean of all run effectiveness scores of system $\boldsymbol{s}$ across all of the given topics:

$$\mathsf{AM\text{-}M}(\boldsymbol{s}) = \frac{1}{\boldsymbol{N_t}} \sum_{t} \mathsf{M}(\boldsymbol{r}_{s,t}), \tag{2.16}$$

where $\boldsymbol{N_t}$ is the number of topics.

As an alternative, Robertson [79] suggests to compute the geometric mean of run scores across topics, since the impact of topics with bad performance would therefore be emphasized. The geometric mean of run scores given by system $\boldsymbol{s}$ is defined as:

$$\mathsf{GM\text{-}M}(\boldsymbol{s}) = \sqrt[N_t]{\prod_{t} \mathsf{M}(\boldsymbol{r}_{s,t})} \tag{2.17}$$

$$= \exp\left[\frac{1}{\boldsymbol{N_t}} \sum_{t} \log \mathsf{M}(\boldsymbol{r}_{s,t})\right]. \tag{2.18}$$

However, using the formula above, we face a problem if any $\mathsf{M}(\boldsymbol{r}_{s,t})$ is zero; the geometric mean computed by Equation 2.17 is also zero. To resolve this problem associated with the original equations, a refined version which employs a positive adjustment number ϵ is defined as:

$$\mathsf{GM\text{-}M}_{\epsilon}(\boldsymbol{s}) = \exp\left[\frac{1}{\boldsymbol{N_t}} \sum_{t} \log\Big(\epsilon + \mathsf{M}(\boldsymbol{r}_{s,t})\Big)\right] - \epsilon. \tag{2.19}$$

As $\mathsf{M}(\boldsymbol{r}_{s,t})$ is non-negative, the $\mathsf{GM\text{-}M}_{\epsilon}(\boldsymbol{s})$ computed by Equation 2.19 can only be a positive number.

Instead of applying ϵ to every run score, a variant of $\mathsf{GM\text{-}AP}_{\epsilon}$ (which deals with AP scores in the program of `trec_eval`) only adjust scores smaller than $\epsilon = 10^{-5}$ [108]:

$$\mathsf{GM\text{-}AP}_{\epsilon,\texttt{trec}}(\boldsymbol{s}) = \exp\left[\frac{1}{\boldsymbol{N_t}} \sum_{t} \log \max\Big(\epsilon, \mathsf{M}(\boldsymbol{r}_{s,t})\Big)\right].$$

Ravana and Moffat [76] compare multiple aggregation technologies for TREC evaluations using AP. They conclude that $\mathsf{GM\text{-}AP}_{\epsilon}$ is more appropriate than AM-AP for TREC evaluations, because there are great numbers of low AP scores (zero, or close to zero) across systems and topics in TREC. When the fraction of low scores is small, AM-AP "performs well", defined as system orderings derived from different sets of topics are similar. The authors also carried out similar analysis for other metrics such as $\mathsf{Prec}(k = 10)$, NDCG and $\mathsf{RBP}(\phi = 0.95)$. The conclusions are similar, that is, the number of low effectiveness

scores generated by the given metric is low, AM-M is a appropriate aggregation technology, otherwise, GM-M$_\epsilon$ is a better choice.

Not only metric scores, but also results such as document relevance scores of different judgment sets can be aggregated by the described methodologies. In Chapter 5, we average assessors' preferences of different judgment collection methods by computing the arithmetic mean, and combine relevance ratio scores assigned by distinct assessors for each document pair by calculating their geometric mean. In our experiments, the choice between these two averaging methods usually depends on the scale (binary, numeral and so on) of the data to be aggregated.

2.4.2 Statistical Testing

Systems can be compared in relative effectiveness by analyzing per-topic performance scores of their returned runs tested on the same set of topics. As described in Section 2.4.1, for each system $\boldsymbol{s}$, run scores $\mathsf{M}(\boldsymbol{r}_{s,t})$, evaluated by some particular metric M, can be aggregated into a single overall effectiveness score which can be viewed as the general retrieval quality of the system and so can be used to select systems with "better" general performance.

Moreover, the mean effectiveness scores of two paired systems can be further analyzed using a *statistical significance test* to discover how much confidence we may have to distinguish these two systems.

There are various statistical significance tests used in IR research such as the Student's paired t-test, the Wilcoxon signed rank test and the sign test, which have been debated and compared for suitability in IR experiments [98]. The paired t-test has been reported by Sakai [87] as the most commonly used statistical test in IR (used by 66% and 61% of papers in SIGIR and TOIS, respectively) followed by the Wilcoxon signed rank test.

The paired t-test takes two lists of sample data from two independent populations as input and determines if they are likely (or not) to be the result of the same underlying process [97]. The p-value resulting from the t-test procedure indicates the confidence of rejecting the null hypothesis: the two examined populations have identical expected (mean) values. If the paired t-test is used in collection-based IR experiments, it takes $\mathsf{M}(\boldsymbol{r}_{s_1,t})$ and $\mathsf{M}(\boldsymbol{r}_{s_2,t})$ (processing the same set of topics $\boldsymbol{t}$ and evaluated by some chosen metric M) of two systems $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, the p-value of the test indicates the likelihood of the observed relationship having occurred by chance alone, assuming the null hypothesis. Small p-value suggest high confidence of rejecting the null hypothesis, that is the observed distinction between systems is real rather than an artifact of the observed data. If the p-value is less than or equal to the pre-determined significance level, denoted as α, the outcome can be viewed as being statistically significant.

In the significance test for each system pair, the effectiveness scores of runs are given by some chosen IR metric. If runs are re-evaluated by another metric, the p-value of the t-test is also likely to be different. As stated in Section 2.3 and 2.3.2, distinct metrics

measure different aspects of system effectiveness. Taking the run scores computed by M as inputs, the significance tests examine if the paired systems are significantly different in aspects measured by the metric M.

Sakai [85, 84] describes and compares statistical significance tests in the context of comparing IR systems using test collections. In addition, Sakai [86] states that reporting the effect size (magnitude of the distinction between systems) and the sample size (number of topics) along with the p-value would make the results more informative. Comparing the results with other researchers' works to imply the practical significance is also necessary.

2.4.3 Correlation and Agreement Measurements

A set of systems can be ordered from "best" to "worst" according to their aggregated run scores. As different metrics, say M_1 and M_2, may measure distinct aspects of relevance, the system rankings given by M_1 and M_2 may be diverse. The two system ranking lists can be compared and so answer the question: whether metric M_1 and M_2 are relatively correlated or not (that is, whether M_1 and M_2 are measuring similar aspects of retrieval performance).

Kendall's τ This conventional and well-known rank correlation measure assesses the statistical association of two orderings of a same set of items. The similarity of two methods (for example, two effectiveness metrics) which generate the two orderings can be qualified using this approach.

Suppose metric M_1 and M_2 score a same set of systems with size of n, and generate system ranked lists $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ respectively, with all systems sorted in decreasing metric score order. Denote the ranks of system $\boldsymbol{s}_i$, ranked by M_1 and M_2 in $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ respectively, as $x_{\boldsymbol{s}_i}$ and $y_{\boldsymbol{s}_i}$. Kendall's τ makes use of the rank tuples of all the systems, that is $\langle x_{\boldsymbol{s}_i}, y_{\boldsymbol{s}_i} \rangle$ (i from 1 to n), to compute the correlation coefficient of two system rankings given by M_1 and M_2. For each pair of tuples, say $\langle x_{\boldsymbol{s}_i}, y_{\boldsymbol{s}_i} \rangle$ and $\langle x_{\boldsymbol{s}_j}, y_{\boldsymbol{s}_j} \rangle$ where $i \neq j$, if:

- $x_{\boldsymbol{s}_i} > x_{\boldsymbol{s}_j}$ and $y_{\boldsymbol{s}_i} > y_{\boldsymbol{s}_j}$, or
- $x_{\boldsymbol{s}_i} < x_{\boldsymbol{s}_j}$ and $y_{\boldsymbol{s}_i} < y_{\boldsymbol{s}_j}$

it will be counted as *concordant* pair, otherwise as a *discordant* pair (assuming no ties). Among totally $n(n-1)/2$ pairs, suppose there are ***ConcPairs*** concordant pairs and ***DiscPairs*** discordant pairs. The Kendall's τ is obtained by:

$$\begin{aligned} \tau &= \frac{\boldsymbol{ConcPairs} - \boldsymbol{DiscPairs}}{\boldsymbol{ConcPairs} + \boldsymbol{DiscPairs}} \\ &= \frac{\boldsymbol{ConcPairs} - \boldsymbol{DiscPairs}}{n(n-1)/2} \end{aligned}$$

System	Score			Rank			Rank Distance	
	M_0	M_1	M_2	M_0	M_1	M_2	(M_0, M_1)	(M_0, M_2)
s_1	9.0	8.5	9.7	1	2	1	-1	0
s_2	8.0	9.3	8.1	2	1	2	1	0
s_3	7.0	8.0	5.5	3	3	5	0	-2
s_4	6.0	7.5	6.0	4	4	4	0	0
s_5	5.0	7.0	6.9	5	5	3	0	2

Table 2.4: An example of systems s_1, s_2, s_3, s_4 and s_5 evaluated by metrics M_0, M_1 and M_2. Mean effectiveness scores of systems given by three distinct metrics are shown in the left. Based on scores, system competition ranks are shown in the middle columns. For each system, the distances between its ranks given by different metrics are shown in the right.

which is a numeric value between -1 (perfect negative association, one ranking is the reverse of the other) and $+1$ (the two rankings are exactly the same). Values close to zero indicate little or no correlation between the methods being measured.

Kendall's τ is only concerned with how much the two rankings agree and disagree. It does not consider the position of the agreement or the disagreement, and disorder that arises at the bottom of two rankings is identically weighted as disorder that arises at the top. In addition, only competition ranks, rather than numeric scores, are considered by Kendall's τ. The magnitude of the difference between scores of adjoining items has no effect to the correlation coefficient.

A detailed example of three distinct metrics evaluating five systems is shown in Table 2.4. For example, when computing Kendall's τ of M_0 and M_1, in total there are $5(5-1)/2 = 10$ pairs of systems, and nine concordant pairs and one discordant pair (that is s_1 and s_2, whose rank tuples are $\langle 1, 2\rangle$ and $\langle 2, 1\rangle$), thus the Kendall's $\tau(M_0, M_1) = (9-1)/(9+1) = 0.8$.

As already noted, Kendall's τ is not a top-weighted measure. In Table 2.4 (middle), M_0 and M_1 are different in ordering of top systems, while the orderings of M_0 and M_2 are different in systems at deeper ranks (but there is still only one discordant pair in this case, s_3 and s_5). But when computing $\tau(M_0, M_1)$ and $\tau(M_0, M_2)$, as their ***ConcPairs*** and ***DiscPairs*** are both nine and one respectively, they both receive Kendall's τ score of 0.8. However, some other top-weighted measures such as Rank-Biased Overlap, described later, will produce different scores to these two comparisons, because disagreements at early ranks give rise to greater score penalties if such measures are used.

In addition, the rank difference between 1 and 2 (when comparing M_0 and M_1) is smaller than 3 and 5 (when comparing M_0 and M_2). As Kendall's τ does not consider the magnitude of the difference, $\tau(M_0, M_1)$ and $\tau(M_0, M_2)$ are the same. But some other correlation measurements, such as Spearman's ρ [102], takes difference magnitudes into

System	Score			Competition Rank			Fractional Rank		
	M_1	M_3	M_4	M_1	M_3	M_4	M_1	M_3	M_4
s_1	8.5	8.3	9.1	2	1	1	2	1	1
s_2	9.3	7.8	8.2	1	2	2	1	2	2
s_3	8.0	6.5	7.4	3	3 =	3	3	3.5	3
s_4	7.5	6.5	6.5	4	3 =	4 =	4	3.5	4.5
s_5	7.0	5.0	6.5	5	5	4 =	5	5	4.5

Table 2.5: Effectiveness scores (left), competition ranks (middle) and fractional ranks (right) of systems s_1, s_2, s_3, s_4 and s_5 evaluated by metrics M_1, M_3 and M_4. Metric M_3 assigns the same scores to systems s_3 and s_4, and so they receive the same competition ranks (the highest rank of the tied systems) and fractional ranks (the average of ranks occupied by the tied systems).

account and the computed correlation coefficients for these two comparisons will be distinct.

Kendall's τ_b The previous Kendall's τ computation does not consider ties in input lists. If $x_{s_i} = x_{s_j}$ or $y_{s_i} = y_{s_j}$, the pair is neither concordant nor discordant, but called a tied pair. This situation happens in our project, described in Chapter 3, when the scores of two systems, given by some evaluation metric, are the same. To deal with this added challenge, Kendall's τ_b adjusts the calculation for tied pairs, given as [50]:

$$\tau_b = \frac{\textbf{\textit{ConcPairs}} - \textbf{\textit{DiscPairs}}}{\sqrt{(\textbf{\textit{ConcPairs}} + \textbf{\textit{DiscPairs}} + X_0)(\textbf{\textit{ConcPairs}} + \textbf{\textit{DiscPairs}} + Y_0)}} \tag{2.20}$$

$$= \frac{\textbf{\textit{ConcPairs}} - \textbf{\textit{DiscPairs}}}{\sqrt{(n(n-1)/2 + X_0)(n(n-1)/2 + Y_0)}} \tag{2.21}$$

where X_0 is the number of pairs tied on x, that is $x_{s_i} = x_{s_j}$, and Y_0 is the number of pairs tied on y.

When all the systems have been scored by a metric, they will be ordered by their scores. Each system will receive a compositional ranking number. From rank p, q systems with equal scores receive the same highest ranking number p, and the ranking numbers of following systems start from $p + q$.

A detailed example of three metrics evaluating five systems and generating ties is given in Table 2.5. In the example, systems s_3 and s_4 both receive a score of 6.5 from M_3, thus their competition ranks are same as 3, and the next rank (of system s_5) is 5. The full ranking results are shown in middle columns of the table.

For instance, computing Kendall's τ of M_1 and M_4 using competition ranks, for each system, compare its rank tuple with the rank tuples of following systems in order to

identify the total numbers of concordant, discordant and tied pairs. In this process, when rank tuple of $\boldsymbol{s}_1(\langle 2,1\rangle)$ compares with rank tuple of $\boldsymbol{s}_2(\langle 1,2\rangle)$, as $x_{\boldsymbol{s}_1} > x_{\boldsymbol{s}_2}$ whereas $y_{\boldsymbol{s}_1} < y_{\boldsymbol{s}_2}$, thus it is a discordant pair. And for the pair of $\boldsymbol{s}_4$ and $\boldsymbol{s}_5$ ($\langle 4,4\rangle$ and $\langle 5,4\rangle$), it is tied on y due to $y_{\boldsymbol{s}_4} = y_{\boldsymbol{s}_5}$. In general, ***ConcPairs*** $= 8$, ***DiscPairs*** $= 1$, $X_0 = 0$ and $Y_0 = 1$. According to Equation 2.21, Kendall's τ_b of M_1 and M_4 can be calculated as:

$$\begin{aligned}\tau_b(\mathsf{M}_1, \mathsf{M}_4) &= \frac{8-1}{\sqrt{(8+1+0)(8+1+1)}} \\ &= 0.738.\end{aligned}$$

The comparison is similar when calculating the Kendall's τ_b of M_3 and M_4. In this instance, $\langle 3,3\rangle$ and $\langle 3,4\rangle$ is a pair tied on x. And $\langle 3,4\rangle$ paired with $\langle 5,4\rangle$ is tied on y. There is no discordant pairs but only ties penalize the Kendall's τ_b in this case.

$$\begin{aligned}\tau_b(\mathsf{M}_3, \mathsf{M}_4) &= \frac{8-0}{\sqrt{(8+0+1)(8+0+1)}} \\ &= 0.889.\end{aligned}$$

Spearman's ρ Another well-known statistical rank correlation measure is Spearman's ρ [102], defined as the Pearson correlation coefficient between two variables.

To measure the correlation of, for example M_1 and M_2, the Spearman's ρ can be calculated from:

$$\rho(\mathsf{M}_1, \mathsf{M}_2) = 1 - 6 \cdot \sum_{\boldsymbol{s}} \mathsf{dist}(\mathsf{M}_1, \mathsf{M}_2, \boldsymbol{s})^2/(n(n^2-1))$$

where n is the total number of systems, and $\mathsf{dist}(\mathsf{M}_1, \mathsf{M}_2, \boldsymbol{s})$ for system $\boldsymbol{s}$ is computed as the difference of its ranks in system ranking lists, $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$, given by metrics M_1 and M_2 respectively.

For the example in Table 2.4, systems are ordered by score and receive ranking numbers, shown in the middle columns. The rank distances $\mathsf{dist}(\mathsf{M}_0, \mathsf{M}_1, \boldsymbol{s})$ and $\mathsf{dist}(\mathsf{M}_0, \mathsf{M}_2, \boldsymbol{s})$ for each system $\boldsymbol{s}$ are shown in the right hand side of the table. In this example, with $n = 5$, Spearman's ρ of M_0 and M_1 can be computed as:

$$\begin{aligned}\rho(\mathsf{M}_0, \mathsf{M}_1) &= 1 - 6 \cdot \sum \mathsf{dist}(\mathsf{M}_0, \mathsf{M}_1, \boldsymbol{s})^2/n(n^2-1) \\ &= 1 - 6 \cdot ((-1)^2 + 1^2 + 0^2 + 0^2 + 0^2)/5 \cdot (25-1) \\ &= 0.9.\end{aligned}$$

As can be seen, when computing $\rho(\mathsf{M}_0, \mathsf{M}_1)$, the sum of rank distance squared between the only discordant pair, $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, is 2. When comparing M_0 and M_2, there is also only one discordant pair (thus $\tau(\mathsf{M}_0, \mathsf{M}_1) = \tau(\mathsf{M}_0, \mathsf{M}_2)$). However, as the magnitude of

the difference between $\boldsymbol{s}_3$ and $\boldsymbol{s}_5$ is greater than it between $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, $\sum \mathsf{dist}(\mathsf{M}_0, \mathsf{M}_2, \boldsymbol{s})^2 = (-2)^2 + 2^2 = 8$ is larger than $\sum \mathsf{dist}(\mathsf{M}_0, \mathsf{M}_1, \boldsymbol{s})^2$. And we can compute:

$$\begin{aligned}\rho(\mathsf{M}_0, \mathsf{M}_2) &= 1 - 6 \cdot 8/5 \cdot (25 - 1) \\ &= 0.6\end{aligned}$$

which shows that M_2 is less correlated to M_0 than M_1.

For handling ties, in contrast to Kendall's τ, Spearman's ρ measures the association of system fractional ranks (from rank p, q systems with equal scores receive the same ranking number which is the average of their ordinal rank range, $(2p + q - 1)/2$, and the ranking numbers of following systems start from $p + q$), while Kendall's τ applies on competition ranks. For the example in Table 2.5, systems are ordered by score and receive fractional ranking numbers shown in right columns of the table. In the example, the Spearman's ρ of M_3 and M_4 is computed as:

$$\begin{aligned}\rho(\mathsf{M}_3, \mathsf{M}_4) &= 1 - \frac{6 \times ((1-1)^2 + (2-2)^2 + (3-3.5)^2 + (4.5-3.5)^2 + (4.5-5)^2)}{5 \times (25-1)} \\ &= 1 - \frac{6 \times (0.25 + 1 + 0.25)}{5 \times (25-1)} \\ &= 0.925.\end{aligned}$$

Like Kendall's τ, Spearman's ρ is not top-weighted. No matter where the rank distances appear in the ranked lists, if the sum of differences is the same, the score given by Spearman's ρ will not change. In addition, both Kendall's τ and Spearman's ρ require that the lengths of two input lists are equal, and both variables (metrics) being measured should evaluate the same set of systems. These measures cannot be applied to the IR cases if two metrics evaluate different sets of systems, or when comparing two runs but only a head part of the ranking results is available (known as indefinite rankings) such as in Web search.

In our project of collecting relevance judgments using distinct methodologies (see Chapter 5), we use correlation measures to compare judgments of a same set of document-topic combinations collected using different methods, and then answer how similar the compared methods are.

Table 2.6 shows an example of relevance scores of ten documents in Binary and 6-level ordinal relevance scales. For convenience, while discussing which factors may affect the Kendall's τ and Spearman's ρ scores, the documents are labeled in decreasing relevance score order. Of course, this would not normally be the case in general.

In order to find which factors (such as magnitude of scores, range of scores, and distribution of scores) may affect the Kendall's τ and Spearman's ρ, the correlation coefficients of relevance scores (in Table 2.6), rank numbers (ranked by relevance scores, documents tied on scores receive the same rank numbers), and mapped scores (2 to the power of

Doc	d_0	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
Binary	1	1	1	1	1	1	0	0	0	0
6-level	5	5	4	3	3	3	2	2	1	0

Table 2.6: An example of relevance scores of ten documents in Binary (0 or 1) and 6-level (from 0 to 5) relevance judgments respectively, sorted in decreasing relevance order.

original scores) according to the Binary and 6-level judgments are shown in Table 2.7. The input lists of each computation are shown in the second column.

The results in Table 2.7 illustrate the way in which Kendall's τ and Spearman's ρ coefficients are unaffected by whether the magnitude, range, and distribution of scores are used as inputs. That is, the correlation coefficients between judgments collected using different relevance scales are not related to relevance score magnitudes, ranges, and distributions in the judgments, but only have the relationship with rank agreements of every pair of documents that two judgment sets have.

Next, we explore whether the number of categories (levels), or the aggregation (mapping) function change the correlation coefficients, and the extent to which they may be affected. For the 6-level relevance judgments, we define (for example) the following five mapping function to combine some relevance levels:

$$\mathsf{MF}_1(\mathsf{Rel}(\boldsymbol{d},\boldsymbol{t})) = \begin{cases} \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 0,1,2,3 \\ 5 & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 4,5 \end{cases}$$

$$\mathsf{MF}_2(\mathsf{Rel}(\boldsymbol{d},\boldsymbol{t})) = \begin{cases} \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 0,1,4,5 \\ 3 & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 2,3 \end{cases}$$

$$\mathsf{MF}_3(\mathsf{Rel}(\boldsymbol{d},\boldsymbol{t})) = \begin{cases} 1 & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 0,1 \\ 3 & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 2,3 \\ 5 & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 4,5 \end{cases}$$

$$\mathsf{MF}_4(\mathsf{Rel}(\boldsymbol{d},\boldsymbol{t})) = \begin{cases} 1 & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 0,1,2 \\ 4 & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 3,4,5 \end{cases}$$

	Compared Lists	Kendall's τ	Spearman's ρ
Score	$[1,1,1,1,1,1,0,0,0,0]$ $[5,5,4,3,3,3,2,2,1,0]$	0.775	0.869
Rank	$[1,1,1,1,1,1,7,7,7,7]$ $[1,1,3,4,4,4,7,7,9,10]$	0.775	0.869
Power of Score	$[1,1,1,1,1,1,0,0,0,0]$ $[32,32,16,8,8,8,4,4,2,1]$	0.775	0.869

Table 2.7: The Kendall's τ and Spearman's ρ when comparing Binary and 6-level relevance judgments via numeric relevance scores (shown in Table 2.6), ranks (tied scores receive the same rank numbers), and 2 to the power of original scores.

$$\mathsf{MF}_5(\mathsf{Rel}(\boldsymbol{d},\boldsymbol{t})) = \begin{cases} 0 & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 0 \\ 1 & \text{if } \mathsf{Rel}(\boldsymbol{d},\boldsymbol{t}) = 1,2,3,4,5 \end{cases}$$

The relevance order of each pair of documents in the mapped judgments either agree with the original judgments, or become equal, with tied relevance scores.

Table 2.8 summarizes the mapping function applied to the 6-level relevance judgments (first column), the input lists of document relevance scores in Binary and the mapped judgments respectively (second column), and the computed correlation coefficients (last two columns). As can be seen, Kendall's τ and Spearman's ρ both have no regular relationship with the number of relevance levels (categories), and vary considerably according to the exact choice of the mapping function applied. That is, the strategy for combining relevance levels is a key factor that affects rank correlation coefficients. Choosing the strategy to reduce the number of relevance levels or aggregate judgments need to be careful if then using Kendall's τ and Spearman's ρ to measure the correlation between judgments.

Drawing the example in Table 2.6, 2.7, and 2.8 together, we note that care needs to be exercised when seeking the method of combining relevance levels, or reducing the fidelity of relevance scores in judgments. In Chapter 5, as choosing different normalization strategies might lead the number of relevance levels (or the precision of relevance scores) varying, Kendall's τ or Spearman's ρ is not the main measurement when analyzing the agreement between judgments.

Rank-Biased Overlap Rank-Biased Overlap (RBO) proposed by Webber, Moffat, and Zobel [115] models behaviors of a user examining two rankings $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ in decreasing metric score order. As RBO is a top-weighted measurement, it will be employed (see

Mapping Function	Number of relevance levels	Compared Lists	Kendall's τ	Spearman's ρ
MF_1	2 5	$[1,1,1,1,1,1,0,0,0,0]$ $[5,5,5,3,3,3,2,2,1,0]$	0.795	0.877
MF_2	2 5	$[1,1,1,1,1,1,0,0,0,0]$ $[5,5,4,3,3,3,3,3,1,0]$	0.630	0.685
MF_3	2 3	$[1,1,1,1,1,1,0,0,0,0]$ $[5,5,5,3,3,3,3,3,1,1]$	0.660	0.694
MF_4	2 2	$[1,1,1,1,1,1,0,0,0,0]$ $[4,4,4,4,4,4,1,1,1,1]$	1.000	1.000
MF_5	2 2	$[1,1,1,1,1,1,0,0,0,0]$ $[1,1,1,1,1,1,1,1,1,0]$	0.408	0.408

Table 2.8: The Kendall's τ and Spearman's ρ when comparing Binary and mapped 6-level relevance judgments. The first column shows the mapping function used to combine relevance levels in the 6-level relevance judgments. The document relevance ordering of the mapped judgments does not conflict with the ordering given by original judgments. Only level combination methods affect the correlation coefficient scores.

Chapter 4) to compute the agreement of two ranking lists when the agreements at top ranks are more important than at bottom ranks.

Denote $\boldsymbol{L}(j)$ as the system ranked at j in the list $\boldsymbol{L}$ and its metric score evaluated by metric $\boldsymbol{M}$ as $\boldsymbol{M}(\boldsymbol{L}(j))$. Then we can represent the set containing all the systems ranked from depth 1 to i in $\boldsymbol{L}$ as:

$$S(\boldsymbol{L}, i) = \{\, \boldsymbol{L}(j) \mid j \leq i \,\} \tag{2.22}$$

The user is assumed to examine the system rankings $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ from the top to the bottom ranks again (as was assumed in RBP in terms of a persistence parameter ϕ). At each depth i, the user checks the *overlap* of the two list:

$$O(\boldsymbol{L}_1, \boldsymbol{L}_2, i) = S(\boldsymbol{L}_1, i) \cap S(\boldsymbol{L}_2, i). \tag{2.23}$$

If there are no ties in either rankings, we have $0 \leq |O(\boldsymbol{L}_1, \boldsymbol{L}_2, i)| \leq i$. The agreement of two rankings at depth i observed by the user can be calculated as:

$$A(\boldsymbol{L}_1, \boldsymbol{L}_2, i) = \left| \frac{O(\boldsymbol{L}_1, \boldsymbol{L}_2, i)}{i} \right|, \tag{2.24}$$

where $0 \leq A(\boldsymbol{L}_1, \boldsymbol{L}_2, i) \leq 1$.

i	$W(\phi=0.8,i)$	$O(\boldsymbol{M}_0,\boldsymbol{M}_1,i)$	$A(\boldsymbol{M}_0,\boldsymbol{M}_1,i)$	$O(\boldsymbol{M}_0,\boldsymbol{M}_2,i)$	$A(\boldsymbol{M}_0,\boldsymbol{M}_2,i)$
1	0.200	$\{\,\}$	$0/1$	$\{\boldsymbol{s}_1\}$	$1/1$
2	0.160	$\{\boldsymbol{s}_1,\boldsymbol{s}_2\}$	$2/2$	$\{\boldsymbol{s}_1,\boldsymbol{s}_2\}$	$2/2$
3	0.128	$\{\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{s}_3\}$	$3/3$	$\{\boldsymbol{s}_1,\boldsymbol{s}_2\}$	$2/3$
4	0.102	$\{\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{s}_3,\boldsymbol{s}_4\}$	$4/4$	$\{\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{s}_4\}$	$3/4$
5	0.082	$\{\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{s}_3,\boldsymbol{s}_4,\boldsymbol{s}_5\}$	$5/5$	$\{\boldsymbol{s}_1,\boldsymbol{s}_2,\boldsymbol{s}_3,\boldsymbol{s}_4,\boldsymbol{s}_5\}$	$5/5$

Table 2.9: RBO weight (W), overlap (O) and agreement (A) of $\boldsymbol{M}_0$ with $\boldsymbol{M}_1$, and $\boldsymbol{M}_0$ with $\boldsymbol{M}_2$ at each depth, assuming $\phi = 0.8$.

Similar to RBP, RBO is a top-weighted measure. The agreement at each depth i is assigned a geometrically decreasing weight:

$$W(\phi, i) = (1-\phi)\phi^{i-1} \tag{2.25}$$

where ϕ $(0 < \phi < 1)$ represents the user persistence, that is the probability that the user continues examining from depth i to $i+1$. Thus swaps occurring near top ranks affect the RBO score more with the lower ϕ.

Combining these ideas, RBO is defined as the expected average agreement of the two rankings $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$, examined by a user with patience ϕ [115]:

$$\mathsf{RBO}(\boldsymbol{L}_1,\boldsymbol{L}_2,\phi) = \sum_{i=1}^{\infty} W(\phi,i)\cdot A(\boldsymbol{L}_1,\boldsymbol{L}_2,i) \tag{2.26}$$

$$= (1-\phi)\sum_{i=1}^{\infty}\phi^{i-1}\cdot A(\boldsymbol{L}_1,\boldsymbol{L}_2,i). \tag{2.27}$$

However in practice, the list lengths are fixed (say k, the user can at most examine to depth $i = k$) instead of infinite. The residuals after depth k cannot be ignored otherwise RBO will be affected by the ranking length. An extended equation which extrapolates from seen k items in the list and adds the true minimum value of the residual to the base part becomes a solution [115]:

$$\mathsf{RBO}(\boldsymbol{L}_1,\boldsymbol{L}_2,\phi)_{EXT} = \frac{1-\phi}{\phi}\sum_{i=1}^{k}\phi^i\cdot A(\boldsymbol{L}_1,\boldsymbol{L}_2,i) + \phi^k\cdot A(\boldsymbol{L}_1,\boldsymbol{L}_2,k) \tag{2.28}$$

where k is the evaluation depth, typically taken to be the length of the lists.

The RBO score of two disjoint rankings is zero, and of two identical (till the evaluation depth k) rankings is $+1$.

As we saw in the previous example (Table 2.4) of comparing the orderings induced by M_0, M_1 and M_2, since Kendall's τ is not a top-weighted measure, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are evaluated as equally correlated to $\boldsymbol{M}_0$ (that is $\tau(\boldsymbol{M}_0,\boldsymbol{M}_1) = \tau(\boldsymbol{M}_0,\boldsymbol{M}_2)$). However, due to the

decreasing weights assigned to the RBO agreements at each rank, differences of ordering at earlier ranks generate greater penalties to the RBO score than differences at later ranks. Table 2.9 gives the overlaps and agreements at each depth when comparing $\boldsymbol{M}_0$ with $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ respectively. According to the Equation 2.28, the values of $\mathsf{RBO}(\boldsymbol{M}_0, \boldsymbol{M}_1, \phi)$ and $\mathsf{RBO}(\boldsymbol{M}_0, \boldsymbol{M}_3, \phi)$ can be calculated as (assuming $\phi = 0.8$):

$$\begin{aligned}
\mathsf{RBO}(\boldsymbol{M}_0, \boldsymbol{M}_1, 0.8) &= \sum_{i=1}^{5} W(i) \cdot A(\boldsymbol{M}_0, \boldsymbol{M}_1, i) + \phi^5 \cdot A(\boldsymbol{M}_0, \boldsymbol{M}_1, 5) \\
&= 0 \cdot 0.200 + 1 \cdot 0.160 + 1 \cdot 0.128 + 1 \cdot 0.102 + 1 \cdot 0.082 + 0.8^5 \cdot 1 \\
&= 0.800 \\
\mathsf{RBO}(\boldsymbol{M}_0, \boldsymbol{M}_2, 0.8) &= \sum_{i=1}^{5} W(i) \cdot A(\boldsymbol{M}_0, \boldsymbol{M}_2, i) + \phi^5 \cdot A(\boldsymbol{M}_0, \boldsymbol{M}_2, 5) \\
&= 1 \cdot 0.200 + 1 \cdot 0.160 + 2/3 \cdot 0.128 + 3/4 \cdot 0.102 + 1 \cdot 0.082 + 0.8^5 \cdot 1 \\
&= 0.846.
\end{aligned}$$

Regardless the value of ϕ, as $0 < \phi < 1$, $\mathsf{RBO}(\boldsymbol{M}_0, \boldsymbol{M}_1, \phi)$ is always strictly less than $\mathsf{RBO}(\boldsymbol{M}_0, \boldsymbol{M}_2, \phi)$ due to the disagreement at top ranks.

Comparing the correlation measurements described above with the same example, Kendall's τ, which is not top-weighted and does not consider the magnitude of the difference, gives the same values to $\tau(\boldsymbol{M}_0, \boldsymbol{M}_1)$ and $\tau(\boldsymbol{M}_0, \boldsymbol{M}_2)$; Spearman's ρ, which is also not top-weighted but takes the magnitude of disagreements in the given lists into account, shows that $\rho(\boldsymbol{M}_0, \boldsymbol{M}_1) > \rho(\boldsymbol{M}_0, \boldsymbol{M}_2)$; the results of the top-weighted measure, RBO, are $\mathsf{RBO}(\boldsymbol{M}_0, \boldsymbol{M}_1, \phi) < \mathsf{RBO}(\boldsymbol{M}_0, \boldsymbol{M}_2, \phi)$. The choice of correlation measurements usually depends on which aspects characteristics of the association between the measured variables need to be considered.

Krippendorff's α Unlike the correlation measures just discussed, Krippendorff's α measures the agreement of values for a set of *units* given by any number of *raters* [53]. Each rater assigns a value (could be binary, ordinal, ratio etc.) for each unit, or leaves nothing for it (incomplete data is accepted). Krippendorff's α takes the unit values from all raters as inputs and computes the inter-rater reliability. The score calculated by Krippendorff's α can be up to 1.0 (perfectly reliable) and down to 0.0 (not reliable). In the social sciences, values of Krippendorff's α greater than 0.8 are taken to indicate strong inter-rater reliability, and values of α less than values of 0.67 suggest low reliability [53].

Krippendorff's α is computed as:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{2.29}$$

	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$	$\boldsymbol{d}_3$	$\boldsymbol{d}_4$
$\boldsymbol{Rater}_1$	R	N	H	R
$\boldsymbol{Rater}_2$	R	N	R	H
$\boldsymbol{Rater}_3$	H	N	H	R

Table 2.10: An Example for computing Krippendorff's α agreement of workers (***Raters***) who assess the relevance of four documents $\boldsymbol{d}_1$, $\boldsymbol{d}_2$, $\boldsymbol{d}_3$, and $\boldsymbol{d}_4$ using an ordinal relevance scale (H–highly relevant, R–relevant, N–non-relevant).

where D_o is the observed disagreement and D_e is the expected disagreement. Denote A as the set containing all possible responses from raters, and U as the set of all units. Then D_o is given by:

$$D_o = \frac{1}{n} \sum_{i \in A} \sum_{j \in A} \delta(i,j) \sum_{u \in U} m_u \frac{n_{ij,u}}{\mathsf{P}(m_u, 2)} \tag{2.30}$$

where n is the number of pairable responses across all units, δ is a metric function, m_u is the number of answers in the unit u, $n_{ij,u}$ is the number of (i,j) pairs (where i and j are responses, that is $i, j \in A$) in the unit u, and $\mathsf{P}(m_u, 2)$ is the permutation function:

$$\mathsf{P}(m_u, 2) = \frac{m_u!}{(m_u - 2)!}$$

With these definitions, D_e is computed as:

$$D_e = \frac{1}{\mathsf{P}(n, 2)} \sum_{i \in A} \sum_{j \in A} \delta(i,j) P_{ij} \tag{2.31}$$

where P_{ij} is the number of all possible (i,j) pairs that can be made. That is:

$$P_{ij} = \begin{cases} n_i n_j & i \neq j \\ n_i(n_i - 1) & i = j \end{cases} \tag{2.32}$$

A coincidence $\boldsymbol{v}$-by-$\boldsymbol{v}$ ($\boldsymbol{v} = |A|$) square matrix containing numbers of occurrences for response pairs according to the reliable (no missing data) answers from raters can be used to simplify the equations above. That is, each cell $o_{vv'}$ in the matrix records the number of times that responses v and v' are given by two different raters respectively. Having the Equation 2.29, 2.30, 2.31 and 2.32, Krippendorff's α can be calculated as:

$$\alpha = \frac{(n-1) \sum o_{vv} - \sum n_v(n_v - 1)}{n(n-1) - \sum n_v(n_v - 1)}. \tag{2.33}$$

Table 2.10 shows an example of collecting relevance judgments (see Chapter 5) for

	N	R	H	n_v
N	3	0	0	3
R	0	2	3	5
H	0	3	1	4
n'_v	3	5	4	12

Table 2.11: The coincidence square matrix o for the example in Table 2.10.

documents $\boldsymbol{d}_1$, $\boldsymbol{d}_2$, $\boldsymbol{d}_3$ and $\boldsymbol{d}_4$ from three different assessors (raters), and we want to use the Krippendorff's α to measure the agreement of assessors. With three categories in the relevance scale used, $A = \{ \mathtt{H}, \mathtt{R}, \mathtt{N} \}$, and 3 assessors $\times$ 4 documents $= 12$ pairable responses, that is $n = 12$. A 3-by-3 matrix o is built for counting the number of each response pair (v, v'), shown in Table 2.11.

To calculate numbers in the matrix, for instance, based on the judgments of $\boldsymbol{d}_1$, three pairs: one (R, R), one (R, H) and one (H, R), are counted by matrix o. The value of $o_{\mathtt{RR}}$, $o_{\mathtt{RH}}$ and $o_{\mathtt{HR}}$ increase by 1. According to the judgments of $\boldsymbol{d}_2$, the number of response pair (N, N) is 3. There is no pair (N, N) in judgments of other documents, thus $o_{\mathtt{NN}} = 3$. The counting is similar for other pairs. The value of n'_v is the sum of values in column v', and n_v records the sum of values in row v. The sum of values in the 3-by-3 matrix, that is n, is shown in the last cell.

With the matrix, the α of three raters in this example can be computed by:

$$\alpha = \frac{(n-1)(o_{\mathtt{NN}} + o_{\mathtt{RR}} + o_{\mathtt{HH}}) - (n_{\mathtt{N}}(n_{\mathtt{N}}-1) + n_{\mathtt{R}}(n_{\mathtt{R}}-1) + n_{\mathtt{H}}(n_{\mathtt{H}}-1))}{n(n-1) - (n_{\mathtt{N}}(n_{\mathtt{N}}-1) + n_{\mathtt{R}}(n_{\mathtt{R}}-1) + n_{\mathtt{H}}(n_{\mathtt{H}}-1))}$$

$$= \frac{(12-1)(3+2+1) - (3\times 2 + 5\times 4 + 4\times 3)}{12(12-1) - (3\times 2 + 5\times 4 + 4\times 3)}$$

$$= 0.298$$

If assessors use numeric scores instead of relevance categories to examine the relevance of documents, for example, the relevance scores of documents assigned by assessors are shown in the left hand side of the table. Taking these numeric values as inputs directly, the computed α is -0.080, because numbers used by the assessors are treated as different categories. Each document has a low probability of receiving identical numeric scores from different assessors. However, the assessors' agreement on document relevance ordering can be measured instead. For each pair of documents shown in the right half of Table 2.12, for each worker, the relevance preference choice between the paired documents (according to relevance scores shown in the left) can be either of: *L*–left document is more relevant; *R*-right document is more relevant; *T*–tied.

The o matrix for the example in Table 2.12 is shown in Table 2.13. As there are three

	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$	$\boldsymbol{d}_3$	$\boldsymbol{d}_4$	($\boldsymbol{d}_1$, $\boldsymbol{d}_2$)	($\boldsymbol{d}_1$, $\boldsymbol{d}_3$)	($\boldsymbol{d}_1$, $\boldsymbol{d}_4$)	($\boldsymbol{d}_2$, $\boldsymbol{d}_3$)	($\boldsymbol{d}_2$, $\boldsymbol{d}_4$)	($\boldsymbol{d}_3$, $\boldsymbol{d}_4$)
Rater$_1$	0.5	0.1	0.7	0.5	L	R	T/?	R	R	L
Rater$_2$	0.4	0.2	0.5	0.7	L	R	R	R	R	R
Rater$_3$	0.9	0.5	0.9	0.7	L	T/?	L	R	R	L

Table 2.12: An Example for computing Krippendorff's α agreement of workers (***Raters***) who assess the relevance of four documents. For every pair of documents, the rater votes one of them (L-left, R-right) according to their assigned scores. If the paired documents have the same score, the categoric value will be T and deemed as a tie. In this example, $n = 3 \times 6 = 18$, $A = \{ L, R, T \}$. The Krippendorff's α taking the categoric values in the right half table as input is 0.353, with the tie rate $2/18 = 0.11$. If all T values are treated as unknown (missing values, represented by "?" in the table), the α of three raters will be 0.500.

	L	R	T	n_v
L	4	1	1	6
R	2	7	1	10
T	0	2	0	2
n'_v	6	10	2	18

Table 2.13: The coincidence square matrix for the example in Table 2.12. For the pair (L, L), there are three in ($\boldsymbol{d}_1$, $\boldsymbol{d}_2$) and one in ($\boldsymbol{d}_3$, $\boldsymbol{d}_4$), so the o_{LL} is $3 + 1 = 4$. For the pair (L, R), there is only one in ($\boldsymbol{d}_3$, $\boldsymbol{d}_4$). The n_v and n'_v record the sum of each row and column. The n for this example is $6 + 10 + 2 = 18$, shown in the last cell.

possible responses when the documents are mapped to pairwise relationships, it is a 3-by-3 matrix. With the matrix in Table 2.13 and Equation 2.33, the α of the example shown in Table 2.12 is:

$$\alpha = \frac{(n-1)(o_{LL} + o_{RR} + o_{TT}) - (n_L(n_L-1) + n_R(n_R-1) + n_T(n_T-1))}{n(n-1) - (n_L(n_L-1) + n_R(n_R-1) + n_T(n_T-1))}$$

$$= \frac{(18-1)(4+7+0) - (6\times 5 + 10 \times 9 + 2 \times 1)}{18(18-1) - (6\times 5 + 10 \times 9 + 2 \times 1)}$$

$$= 0.353$$

with tie rate $2/18 = 0.11$.

As Krippendorff's α is applicable for any number of raters as well as incomplete data, it will be used to measure the agreement of assessors and different sets of judgments collected using our pairwise combined techniques in Chapter 5.

2.4.4 Mixed Effects Models

As was described in Section 2.4.2, the t-test takes the sample data from two variables as inputs and computes a p-value that suggests if the means of two variables are significantly different. Compared to the t-test, a statistical model called *analysis of variance* (ANOVA) [32] can analyze more than two factors (also called groups) in one test by comparing variations of means among factors and finding where the significant differences are.

When we propose new methods to collect data such as gathering relevance judgments from human assessors, if the data is normally distributed, ANOVA or other mixed effects models such as *generalized linear mixed model* (GLMM) [9] can be used to analyze how the factors of the data (for example, topic, collection method, assessor group) affect the response variable, such as the relevance score assigned to each document. GLMM is also suitable for handling incomplete data. In the experiment described in Chapter 5, as it is too expensive to collect relevance judgments for every pair of documents, we randomly pair documents and ensure each document is in the same number of pairs. Thus, the collected results of incomplete data are better analyzed using GLMM rather than ANOVA.

In GLMM, there are two kinds of factor effects in the model: (1) fixed effects, independent factors manipulated in the study, such as topic and method used; and (2) random effects, factors whose values (or levels) are randomly sampled from a large population, and the single values of random effects are not of interest, such as subject (for example, user's ID).

The formula of linear mixed model is:

$$\mathbf{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\varepsilon} \tag{2.34}$$

where $\mathbf{y}$ is a $N \times 1$ vector variable containing responses (outcomes), $\boldsymbol{X}$ is a $N \times p$ matrix of p predictor variables (fixed effects), $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown regression coefficients for fixed effects, $\boldsymbol{Z}$ is a $N \times r$ matrix of r random effects, and $\boldsymbol{u}$ is a $r \times 1$ unknown vector of coefficients for random effects. The $\boldsymbol{\varepsilon}$ is a $N \times 1$ vector of residuals. Note that the p fixed effects include a fixed intercept whose value in $\boldsymbol{X}$ is always 1 for each subject.

The $\boldsymbol{\beta}$, $\boldsymbol{u}$ and $\boldsymbol{\varepsilon}$ can be estimated by the LMM when it takes $\mathbf{y}, \boldsymbol{X}$ and $\boldsymbol{Z}$ as inputs. Each estimated coefficients in $\boldsymbol{\beta}$ shows how much the corresponding fixed effect in $\boldsymbol{X}$ influences the outcome.

The coefficients chosen by the model are the ones which best fit the given data when considering all variables simultaneously. Thus adding or removing variables will affect all of the other coefficients too. Selecting different factors as predictor variables in the formula will result in different estimates and related p-values. Furthermore, models which include two correlated factors will perform worse than a model including only one of them, because the effect of one underlying "measurement" is split between two

worker	topic	units	consistency
1001	401	2	0.8
1002	401	1	0.5
1003	402	4	1.0
1004	402	3	0.3
...	...	...	...
1100	410	5	0.6

Table 2.14: An example of workers' assessment consistency (from 0 to 1.0) when using a new proposed method to collect relevance judgments. Each row shows the worker's ID, topic ID assessed by the worker, the number of units completed by the worker (workload) and the worker's overall assessment consistency. The GLMM can be employed to estimate how the factors of topic and workload affect worker's consistency.

variables, which is referred to as "multicollinearity" and can be a big problem in social sciences.

Table 2.14 shows an example of the experiment results of one hundred workers who assessed the relevance of documents using the new method described in Chapter 5. GLMM can be used to analyze how the factors of topic and number of assessments completed by the worker affect worker's assessment consistency. In this example, the fixed effects are topic and unit number ($\boldsymbol{X}$ is a 100×2 matrix) and the random effect $\boldsymbol{Z}$ is worker's ID (first column). The response variable $\mathbf{y}$ is the consistency (last column).

We use `lme4` and `car` packages in R[12], and the formula given to the R program is:

```
my_model <- lmer(consistency ~
                 topic + units + (1 | worker),
                 data = table_1.14, REML=FALSE)
summary(my_model)
Anova(my_model)
```

where topic IDs are treated as categories instead of numeric values. In the results computed by the R program, the estimated impact coefficient, β and $\boldsymbol{u}$, will show how each factor affects the response variable, and each estimate is companied with a significance p-value provided by `Anova` function. If the p-value is less than the significance level such as 0.05, we can conclude that the impact (positively or negatively, depends on the factor's estimate value) of the factor is significant.

[12]https://www.r-project.org

2.5 Summary

With a new system design, its performance is often examined by batch evaluation techniques. Conventionally, IR evaluation metrics score the system runs by comparing the retrieved ranked results using relevance judgments, then the generated evaluation values of systems are compared in significance tests so that the superior systems could be identified.

In this chapter, we summarized and compared several commonly-used IR evaluation metrics and techniques that have been employed for comparing runs or systems, or collecting relevance judgments. Some correlation measurements and data analysis models for IR evaluations were also described. The next three chapters build on this foundation, and are briefly summarized next.

In the retrieved results (runs), IR systems may assign the same similarity scores to documents when processing a given topic. The role of tied similarity scores in the past TREC runs and possible strategies to handle them in IR evaluations will be described in Chapter 3. Tied similarity scores can be caused by score rounding within similarity metrics, and might be undertaken for efficiency. With that in mind, we also deliberately group documents as ties in order to discover which level of similarity score rounding can be tolerated so that systems can process queries faster without losing effectiveness, and without affecting the ability of IR metrics to distinguish systems.

In Chapter 4, the uncertainty of effectiveness metric scores, which is often associated with unjudged documents in relevance judgments, will be explored. In this chapter, we estimate the reliability of IR system effectiveness scores evaluated by recall-based metrics, and compare the effectiveness scores given by recall-based metrics with the residual scores (uncertainty) computed by utility-based metrics. We show that the uncertainty of recall-based metrics could be very high when the number of unjudged documents is large, and suggest that researchers report uncertainties for system effectiveness scores, which could be computed by weighted-precision metrics (such as RBP), in addition to statistical test results for examining metric score consistency.

We also reviewed the potential problems of currently widely-used relevance judgment collection methodologies, such as the variation of assessors' perceptions of relevance, and ties in categorical judgments. Conventional relevance judgments are usually collected via ordinal relevance scales with two or more relevance categories, and assessed by small numbers of trained experts. Judgments collected by such scales often contain ties: documents in the same category cannot be separated by relevance. Collecting judgments on a scale with higher fidelity can help us to better understand user's perception of relevance so that the gain functions used for mapping relevance categories to numeric scores might be refined. Moreover, the similarity score rounding may not be greatly tolerated (described in Chapter 3) anymore when the number of ties in relevance judgments decreases.

In Chapter 5, we propose the use of pairwise forced-choice to collect relevance judgments using three different scales: preference, absolute relevance, and relevance ratio via crowd-sourcing platform which provides a large number of non-specialist assessors. We investigate the variation of the normalized relevance judgments generated by answers associated with these three methods, and compare them with previous judgments: NIST Binary, Sormunen and Magnitude Estimation. We measure the number of assessed documents, average assessing speed, average document length, assessing inconsistency, accuracy and method preferences of workers, and analysed which factors might affect the quality of relevance assessments.

Chapter 3

Ties in Evaluation

To answer the question "is System s_1 demonstratively better than s_2", the batch evaluation techniques described in Section 2.1.2 are often employed. For each of the IR systems s and for each of a set of given topics t, the system computes a similarity score $\mathsf{Sim}(d, s)$ for each document d related to the topic t, and returns a ranked list (or a run) $r_{s,t}$ containing documents in non-increasing similarity score order. In the similarity computations, documents d_1 and d_2 are deemed as *ties* if they receive the same similarity scores from the system, that is $\mathsf{Sim}(d_1, s) = \mathsf{Sim}(d_2, s)$.

The effectiveness of the run $r_{s,t}$ is then evaluated by a metric M which computes a numeric run score, $\mathsf{M}(r_{s,t})$, using a set of relevance judgments. A suitable statistical test, such as paired t-test, takes the run scores of s_1 and s_2 over several topics as the input, and asks whether if one of them is significantly better than the other according to the generated p-value. Alternatively, the run scores of a system can be aggregated into a single score using some methodology across topics (for example, averaged into an geometric mean) to get an overall retrieving quality score which will then be used in system comparisons.

In this chapter, the first research question is:

RQ1: What are the consequences of allowing ties in runs?

We explore the impact of ties on IR system evaluation, and possible strategies for handling ties when assessing the effectiveness of systems. As documents having the same similarity scores could be presented in any permutations that is consistent with "non-increasing" ordering, the ranked list $r_{s,t}$ could be ordered in different ways if the run contains ties, which might possibly cause ambiguity in the effectiveness run score, $\mathsf{M}(r_{s,t})$, and hence affect system comparison outcomes. We count the proportion of ties in past TREC runs, and quantifying extent to which the system evaluations had been affected by the similarity score ties. Then we explore whether, if the method of ordering ties in a run is different, the system comparison results change? In this part of the project, we compare the system scores computed using a range of tie-breaking regimes, including the strategy employed by the well-known `trec_eval`[1] program used in TREC evaluations, and conclude that even though similarity score ties might bring potential risks to the outcomes of IR evaluations, in practice the influence is minor.

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

This conclusion then takes us to the second research question. If we deliberately introduce ties, for example reduce the precision of similarity scores in the ranking so that the computation speed and space occupied can therefore be optimized, even then we may end up with great numbers of ties in the runs, but

RQ2: to which extent the tied similarity scores could be tolerated without affecting system comparisons?

We propose a method which allows controlled grouping of scores in runs. We test the methodology deliberately introducing ties to past submitted TREC runs in distinct extends, and show that markedly high levels of similarity score approximation can be tolerated without significant changes in the results of system comparisons, and hence that strategically reducing the precision of scoring might be possible in practice for search efficiency boosting.

3.1 Methods for Dealing with Ties

Suppose that documents are sorted by similarity score in non-increasing order in the run which is divided into *groups* – documents in each group have the same similarity score. Denote b_g (when $g = 1$, $b_g = 1$) as the rank in a run where the g th ($g \geq 1$) equi-score group starts, and e_g is the rank of the last document in the g th group. Thus, $b_{g+1} = e_g + 1$, and the g th group contains $b_{g+1} - b_g$ documents. Table 3.1 shows an example of a ten-item ranking given by system $\boldsymbol{s}$ for topic $\boldsymbol{t}$, with each document labeled by a single letter for convenience, and with five groups. As Table 3.1 shows, the first group begins at rank 1 ($b_g = 1$). As $\mathsf{Sim}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(1), \boldsymbol{s}) \neq \mathsf{Sim}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(2), \boldsymbol{s})$, the first group only contains one document and $b_2 = 2$. Documents H, A and C are presumed to have the same similarity score of 9.3, so they are ties in the second group. The third group begins at rank 5.

The third row of Table 3.1 shows the presumed Binary gain value (0 for irrelevant, 1 for relevant) for the corresponding document at each rank. Without considering similarity scores, if only the given ordering of documents and their relevance gains are employed to compute (for example) the Prec at depth 5 for run $\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}$, as there are two documents with $\mathsf{Gain}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d), \boldsymbol{t}) = 1$ when $1 \leq d \leq 5$, so $\mathsf{Prec}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}, 5) = 2/5 = 0.4$. As the first relevant document appears at rank $d = 3$, the reciprocal rank score is $\mathsf{RR}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}) = 1/3 = 0.333$. Similarly, scores of other metrics such as Average Precision (AP), Rank-Biased Precision (RBP), and Normalized Discounted Cumulative Gain (NDCG), can also be computed, solely on the gain value $\mathsf{Gain}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d), \boldsymbol{t})$ at each rank without considering document labels and their similarity scores $\mathsf{Sim}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d), \boldsymbol{s})$.

However if the similarity scores are included in the consideration, the situation becomes complicated. For example in the third group, document M and S are assigned the same similarity score of 8.4 by the system. Moreover, using the given run list or reordering S before M within the group does not break the non-increasing ranking. However, if S is ranked at $d = 5$ and M is placed after it, $\mathsf{Prec}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}, 5)$ will change from 0.4

d	1	2	3	4	5	6	7	8	9	10
$\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d)$	D	H	A	C	M	S	W	B	E	J
$\mathsf{Gain}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d), \boldsymbol{t})$	0	0	1	1	0	1	1	0	0	1
$\mathsf{Sim}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d), \boldsymbol{s})$	9.8	9.3	9.3	9.3	8.4	8.4	8.2	8.0	8.0	8.0
groups	b_1=1	b_2=2			b_3=5		b_4=7	b_5=8		

Table 3.1: Example run showing five equi-score groups. The document $\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d)$ at each rank d of the run given by system $\boldsymbol{s}$ for topic $\boldsymbol{t}$, has the relevance gain $\mathsf{Gain}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d), \boldsymbol{t})$ in Binary (`1` and `0`), and receives the similarity score $\mathsf{Sim}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d), \boldsymbol{s})$. The last row shows the beginning rank b_g of each group.

to $3/5 = 0.6$. Similarly, the RR score be might either $1/2$ or $1/3$ depending on the tie-breaking rule applied to the ordering of the tied documents (H, A and C) in the second group (that is, among six ordering permutations of the second group, either A or C being ranked at $d = 2$ brings the $\mathsf{RR}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}) = 1/2$, otherwise $\mathsf{RR}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}) = 1/3$).

Overall, as documents need to be sequentially presented in the run, documents having the same similarity scores (ties) need to be ordered according some default or explicit mechanism. We now consider a range of options to do this.

Run Order As described in the example of Table 3.1, the first and the most obvious mechanism is to ignore the similarity scores of documents and directly process the run using the ordering in which the documents are sequentially presented by the system. This approach presumes the retrieval system has employed more information (for example, the scores computed during the internal computations may have higher precision than the final similarity scores passed to the evaluation regime, or the system may have sorted the tied documents by a secondary-key which does not involve the scores at all) than what is captured in the final score. Using this mechanism, the retrieval system takes all the responsibility for tie-breaking, and while we may have no idea how the system has handled the ties, we choose to accept and trust the ordering of documents represented in the run provided by the system. Although the similarity scores are disregarded, this method is a plausible default way of handling ties in the system evaluation.

Using this approach to handle ties in the example shown in Table 3.1, the run score of (for example) $\mathsf{RBP}(\phi = 0.9)$ can be computed as $(1-0.9)(0.9^2+0.9^3+0.9^5+0.9^6+0.9^9) = 0.305$.

External Tie-Break Rule A second way is to re-sort the documents within each tied group using some external fixed ordering criterion such as document identifier, document length, URL or filename. A specific example employing this type of approach is employed in the widely-used `trec_eval` program which re-sorts documents in each

tied group into *decreasing* document identifier order before passing the runs to the computations performed by the various effectiveness metrics.

If this method is employed for handling ties in Table 3.1 and re-orders documents in each tied group by decreasing document identifier (as `trec_eval` does), the sorted run will be (from rank 1 to 10): D, H, C, A, S, M, W, J, E, B. The RBP($\phi = 0.9$) score is therefore computed as: $(1 - 0.9)(0.9^2 + 0.9^3 + 0.9^4 + 0.9^6 + 0.9^7) = 0.320$.

The external tie-breaking rule can be varied based on the type of documents and search tasks. For example, Wang, Wu, and Fang [114] involved the number of followers into ranking models to break ties in tweets retrieval. They showed that the ranking method of combining tf-idf, document length, and number of followers can break document ties, and archive the greatest effectiveness in microblog search.

Optimistic and Pessimistic Limits A third method of handling ties in the run is to compute the best and worst scores that might arise across all the permutations in all tied groups. Instead of presenting a single run score, this method provides a score range which suggests the potential ambiguity of the run score. A wide score range then indicates large run score uncertainty caused by ties.

Denote there are l_g documents in the g th group, that is, $l_g = e_g - b_g + 1$. And suppose that there are R_g ($0 \leq R_g \leq l_g$) relevant documents in the g th group. To compute the upper bound of the run score, in each group, all R_g relevant documents need to appear at the first R_g ranks within the group (that is, from rank b_g to rank $b_g + R_g - 1$). Using this approach, in the example shown in Table 3.1, the gain values of documents in the re-ordered run which has the best score across all the permutations is (from rank 1 to 10):

$$0, 1, 1, 0, 1, 0, 1, 1, 0, 0.$$

Thus the best RBP($\phi = 0.9$) is computed as

$$\begin{aligned} \text{RBP}(\phi = 0.9) &= (1 - 0.9)(0.9^1 + 0.9^2 + 0.9^4 + 0.9^6 + 0.9^7) \\ &= 0.338. \end{aligned}$$

Similarly, assume all relevant documents in each group appear at ranks as deep as possible, that is for each g, all R_g relevant documents are in the last R_g ranks in the g th group. The gain values of the "worst" run are (from rank 1 to 10):

$$0, 0, 1, 1, 0, 1, 1, 0, 0, 1$$

and the RBP($\phi = 0.9$) is therefore calculated as:

$$\begin{aligned} \text{RBP}(\phi = 0.9) &= (1 - 0.9)(0.9^2 + 0.9^3 + 0.9^5 + 0.9^6 + 0.9^9) \\ &= 0.305. \end{aligned}$$

Hence, the possible RBP($\phi = 0.9$) score range of the run in Table 3.1 is $[0.305, 0.338]$.

Using this strategy, if there are unjudged documents in the run, they are assumed to be relevant documents for the purpose of establishing the upper bound of the run score, and to be non-relevant when computing the lower bound (but notably, not for recall-based metrics such as AP and NDCG).

Averaging Across Permutations Providing worst–best score ranges of runs can be informative but not an ideal method for system comparisons. The score ranges of runs could be deemed as the potential uncertainties of run scores affected by similarity score ties, but they are not comparable, and thus could not be used when clear conclusions are needed in system comparisons (such as when answering the yes–no question "is System s_1 demonstratively better than s_2"). Thus computing the average, or expected, value of the metric across all possible permutations of documents and all of the tied groups is a further possible mechanism for providing a single score. Assuming each permutation of documents in each group has equal chance to be presented, the expected score is then simply computed by averaging scores of all permutations given by the metric. The number of permutations of this brute-force approach is $\prod_g (l_g!)$, which is feasible when numbers of documents and groups of the run are small. If there are any large groups, or the length of run is great, this method may be expensive.

Fortunately, some tractable computations which calculated the expected (average) score over all permutations for most metrics based on the probability, were described by McSherry and Najork [69]. Denote the group number for the document at rank k in the run as $\mathsf{grp}(k)$. For the example in Table 3.1, the group number of the first document is 1, documents at rank $2, 3$ and 4 are in the second group, document at rank 5 and 6 are tied in the third group:

$$\begin{aligned}
&\mathsf{grp}(1) = 1,\\
&\mathsf{grp}(2) = \mathsf{grp}(3) = \mathsf{grp}(4) = 2,\\
&\mathsf{grp}(5) = \mathsf{grp}(6) = 3\\
&\ldots
\end{aligned}$$

As can be seen, the group $[b_{\mathsf{grp}(k)}, b_{\mathsf{grp}(k)}+1, \ldots, e_{\mathsf{grp}(k)}]$ contains all documents having the same similarity score with the document at rank k. As denoted before, the length of the $\mathsf{grp}(k)$ th group is $l_{\mathsf{grp}(k)}$, and there are $R_{\mathsf{grp}(k)}$ relevant documents in the group.

We define function $\mathsf{enr}(k)$ as the expected number of relevant documents till depth k in the run, which is the sum of two parts: (1) the number of relevant documents ranked before the $\mathsf{grp}(k)$ th group, that is $\sum_{i=1}^{\mathsf{grp}(k)-1} R_i$; and (2) the expected number of relevant documents ranked from $b_{\mathsf{grp}(k)}$ till k inside of the $\mathsf{grp}(k)$ th group. Assuming the expected relevance of documents in the same group is equal, the expected relevance at rank i is therefore equivalent to the proportion of relevant documents in the corresponding $\mathsf{grp}(i)$

th group, that is, $R_{\mathsf{grp}(i)}/l_{\mathsf{grp}(i)}$. Overall, $\mathsf{enr}(k)$ can be given as:

$$\mathsf{enr}(k) = \begin{cases} R_{\mathsf{grp}(k)}/l_{\mathsf{grp}(k)} \cdot (k - b_{\mathsf{grp}(k)} + 1) & \text{if } \mathsf{grp}(k) = 1 \\ \sum_{i=1}^{\mathsf{grp}(k)-1} R_i + R_{\mathsf{grp}(k)}/l_{\mathsf{grp}(k)} \cdot (k - b_{\mathsf{grp}(k)} + 1) & \text{if } \mathsf{grp}(k) > 1. \end{cases} \tag{3.1}$$

For the example in Table 3.1, $\mathsf{enr}(1) = 0$, because there is no relevant document in the first group. There are three documents in the second group, and two of them are relevant. As the beginning rank of the second group is 2, we have:

$$\begin{aligned} \mathsf{enr}(2) &= 0 + \frac{2}{3}(2 - 2 + 1) \\ &= \frac{2}{3} \\ \mathsf{enr}(3) &= 0 + \frac{2}{3}(3 - 2 + 1) \\ &= \frac{4}{3} \\ \mathsf{enr}(4) &= 0 + \frac{2}{3}(4 - 2 + 1) \\ &= 2. \end{aligned}$$

When $k = 5$, there are two relevant documents before rank 5, and there is half a chance that the only relevant document in the third group (that is, document S) is placed at rank 5. Thus $\mathsf{enr}(5)$ could be calculated as:

$$\begin{aligned} \mathsf{enr}(5) &= (0 + 2) + \frac{1}{2}(5 - 5 + 1) \\ &= \frac{5}{2}. \end{aligned}$$

The computations of $\mathsf{enr}(k)$ for later ranks are similar.

According to Equation 3.1, Prec at k (evaluation stops at rank k) score across all permutations of all the groups, denoted as $\overline{\mathsf{Prec}(k)}$, can therefore be computed as:

$$\overline{\mathsf{Prec}(k)} = \frac{\mathsf{enr}(k)}{k}.$$

The expected $\mathsf{AP}(k)$ score across all the permutations of all the groups can be given as:

$$\overline{\mathsf{AP}(k)} = \frac{1}{\boldsymbol{R}} \sum_{i=1}^{k} \overline{\mathsf{Prec}(i)} \cdot R_{\mathsf{grp}(i)}/l_{\mathsf{grp}(i)} \tag{3.2}$$

$$= \frac{1}{\boldsymbol{R}} \sum_{i=1}^{k} \frac{\mathsf{enr}(i)}{i} \cdot \frac{R_{\mathsf{grp}(i)}}{l_{\mathsf{grp}(i)}}. \tag{3.3}$$

where $\boldsymbol{R}$ is the total number of relevant documents for this query. In our example,

$\overline{\mathsf{AP}(1)} = 0$ as there is no relevant document in the first group. Referring to the computed values of $\mathsf{enr}(k)$ established above, then $\overline{\mathsf{AP}(2)}$ is computed as:

$$\begin{aligned}\overline{\mathsf{AP}(2)} &= \frac{1}{\boldsymbol{R}} \cdot (0 + \mathsf{enr}(2)/2 \cdot 2/3) \\ &= \frac{1}{\boldsymbol{R}} \cdot (1/3 \cdot 2/3) \\ &= \frac{2}{9\boldsymbol{R}},\end{aligned}$$

and the computations of $\overline{\mathsf{AP}(k)}$ when $k > 2$ are similar.

For weighted-precision metrics, such as RBP, a similar process can be adopted to obtain the average of metric scores across all the permutations and all the groups. The expected gain value applied to rank i (in the $\mathsf{grp}(i)$ th group) is the expected relevance of document at rank i, that is, $R_{\mathsf{grp}(i)}/l_{\mathsf{grp}(i)}$.

The expected gain value of each rank i is then weighted by $(1-\phi)\phi^{i-1}$, and summed to the RBP score. Thus, the averaged RBP with the parameter ϕ across all permutations is computed as:

$$\overline{\mathsf{RBP}(\phi, k)} = (1-\phi) \sum_{i=1}^{k} \frac{R_{\mathsf{grp}(i)}}{l_{\mathsf{grp}(i)}} \cdot \phi^{i-1}.$$

When $\phi = 0.9$, the $\overline{\mathsf{RBP}(\phi = 0.9, k = 10)}$ of the run shown in Table 3.1 is computed as:

$$\begin{aligned}\overline{\mathsf{RBP}(\phi = 0.9, k = 10)} = (1-0.9) \cdot (&0 \cdot 0.9^0 + 2/3 \cdot 0.9^1 + 2/3 \cdot 0.9^2 + 2/3 \cdot 0.9^3 + 1/2 \cdot 0.9^4 \\ &+ 1/2 \cdot 0.9^5 + 1 \cdot 0.9^6 + 1/3 \cdot 0.9^7 + 1/3 \cdot 0.9^8 + +1/3 \cdot 0.9^9) \\ = 0.321\end{aligned}$$

For RR, assume that the first relevant document is in the v th group. Define $\mathsf{notr}(x, r, n)$ as the probability that in a list of n documents with totally r relevant documents, none of the first x documents is relevant:

$$\mathsf{notr}(x, r, n) = \begin{cases} 1 & \text{if } x = 0, \\ (1 - \frac{r}{n-x+1}) \cdot \mathsf{notr}(x-1, r, n) & \text{otherwise.} \end{cases}$$

For each document at rank i in the v th group, the probability that all the documents ranked before i are non-relevant is $\mathsf{notr}(i - b_v, R_v, l_v)$, and the probability that the document at rank i is the first relevant document can therefore be computed by:

$$\mathsf{firstr}(i) = \frac{R_v}{l_v - (i - b_v)} \cdot \mathsf{notr}(i - b_v, R_v, l_v).$$

The expected RR across all permutations, $\overline{\mathsf{RR}}$, is the sum of $1/i$ weighted by the probability $\mathsf{firstr}(i)$:

$$\overline{\mathsf{RR}} = \sum_{i=b_v}^{e_v} \frac{1}{i} \cdot \frac{R_v}{l_v - (i - b_v)} \cdot \mathsf{notr}(i - b_v, R_v, l_v).$$

3.2 Ties in TREC Experimentation

3.2.1 TREC Resources

Firstly, we explore the number of similarity score ties that occur in some standard past TREC test environment and how they might affect the evaluation. Our hypothesis is that applying different orderings strategies for handling ties will alter TREC run scores, and maybe even alter the system rankings.

The 103 submitted runs for the 1998 TREC-7 Ad-Hoc experimentation round [113] was the primary resource we made use of. Harman [37] was also referred as a broad overview.

Each file submitted by a TREC-7 participant contains 50 rankings (runs) for 50 topics respectively. We partitioned the file by topics so that each run file only included a list of (up to) $1{,}000$ documents for one run, $\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}$, which was retrieved by the system $\boldsymbol{s}$ for the topic $\boldsymbol{t}$. Each row in the run file is for one system-topic combination, corresponding to an unique line number, and includes fields of document number $\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d)$, rank d and score $\mathsf{Sim}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d), \boldsymbol{s})$. Thus documents in each run can be sorted in three possible ways:

1. by (increasing) values in line numbers, that is, the ordering of documents presented in the run file;
2. by (non-decreasing) values in rank d field;
3. by (non-increasing) values in the score $\mathsf{Sim}(\boldsymbol{r}_{\boldsymbol{s},\boldsymbol{t}}(d), \boldsymbol{s})$ field.

The increasing line numbers have no ties, but both rank and score might generate ties in runs, depending on how the system implementors chose to develop that software.

When we analyzed the TREC-7 Ad-Hoc runs, we surprisingly found that there were 254 instances in the archived runs where similarity scores were increasing with the line number, and there were five system affected. The primary reason caused this inconsistency was incorrectly sorting of similarity scores when scientific notation was being used.

An example of the run (extracted from the submitted file) returned by system `bbn1` for topic 355 is shown in Figure 3.1. The column of each entry represents line number, topic, Q0, document (identifier), rank, similarity score, and system respectively. The line number, increasing from 1, is assigned to each row of the run. As can be seen, the run is primarily sorted by the similarity score in non-increasing order, except the last row whose similarity score is represented in scientific notation, and has been treated incorrectly in

1	355	Q0	FBIS4-20436	0	18.34	bbn1
2	355	Q0	FBIS4-49691	1	16.13	bbn1
3	355	Q0	FBIS3-40224	2	15.98	bbn1
...	...	...	...	...	...	...
304	355	Q0	FBIS3-2363	303	0.005601	bbn1
305	355	Q0	FBIS3-25845	305	-0.0006148	bbn1
...	...	...	...	...	...	...
955	355	Q0	FBIS3-27456	956	-1.33	bbn1
956	355	Q0	FBIS3-33460	957	-1.33	bbn1
957	355	Q0	FBIS4-4125	955	-1.33	bbn1
958	355	Q0	FBIS4-45831	958	-1.33	bbn1
959	355	Q0	FT943-11853	959	-1.331	bbn1
...	...	...	...	...	...	...
992	355	Q0	LA101490-0067	992	-1.366	bbn1
993	355	Q0	FBIS4-3483	993	-1.367	bbn1
994	355	Q0	FBIS3-55378	995	-1.368	bbn1
995	355	Q0	LA111089-0164	994	-1.368	bbn1
996	355	Q0	FBIS3-59596	997	-1.369	bbn1
997	355	Q0	FR940810-2-00026	996	-1.369	bbn1
998	355	Q0	FBIS3-21021	999	-1.37	bbn1
999	355	Q0	FBIS3-46125	998	-1.37	bbn1
1000	355	Q0	FT931-2004	304	-7.763e-05	bbn1

Figure 3.1: The run retrieved by system `bbn1` for topic 355.

this sorting. For documents tied on the similarity score, for example, the documents from line number `955` to `958` having the same score value, `-1.33`, are sorted by the secondary key – document identifier – in increasing order (but would be reversed by `trec_eval`. The rank numbers of the tied documents shown in the given ordering are sometimes discordant with their line numbers. Thus we may assume that, the rank number is assigned by the system program when sorting the runs based on some other rules, before the run is re-sorted (for some reason) in the file retained in the NIST archive. As can be seen from the example, even the document `FT931-2004` in last entry is placed in the wrong row, its rank number is correct.

If we sort the run by increasing line numbers, the score of the second-to-last document (whose rank d is labeled as 998) in the run is `-1.37` but the score of the final document (rank $d = 304$) is `-7.763e-05`. Instead of sorting by line numbers, if we sort the run by the system-specified rank d, the situation is more confused: 7.3% of documents in the submitted runs (totally 358,631 entries) were not correctly ordered (similarity score is not non-increasing when rank d grows) according to their labeled ranks. For example, Figure 3.2 shows the run retrieved by system `LIAClass` for topic 351 after sorted by similarity score. The documents with the rank numbers `536` and `537` have been assigned either the wrong rank labels, or incorrect similarity scores.

```
1    351  Q0  FT932-16710  1    20.179836  LIAClass
2    351  Q0  FT941-9999   2    19.717664  LIAClass
3    351  Q0  FT934-4848   3    19.574823  LIAClass
4    351  Q0  FT932-6577   536  19.548333  LIAClass
5    351  Q0  FT944-10523  4    19.345366  LIAClass
6    351  Q0  FBIS4-66185  537  19.086538  LIAClass
7    351  Q0  FT941-12534  5    17.601315  LIAClass
...  ...  ...  ...          ...  ...        ...
```

Figure 3.2: The `LIAClass` run for topic 351, sorted by similarity score.

In general, the ranking of documents in the runs held at the NIST archive in non-increasing similarity score order corresponds to neither increasing rank nor to growing line number over all the runs. We can only assume that this mislabeling issue was caused by programming errors when the runs were generated by the corresponding participants (research groups).

To ensure that results of our experiments in this work would not be affected by the mislabeling problem in the runs, we re-ordered documents for all TREC-7 submitted runs using decreasing numeric similarity score as the primary key (treating scores in scientific format with great care) and then increasing rank d as the secondary key (in TREC-7 runs, there was no ties on values of rank). There were no score-based out-of-order items in the sorted runs, and documents tied on similarity scores were ordered by their rank label. That is, we took the set of available TREC runs and reordered them into canonical representations.

After performing the re-sorting using the sorting command

```
sort -k1,1 -k2,2 -k4gr,4 -k3,3,
```

the run retrieved by system `bbn1` for topic 355 described before, is now shown in Figure 3.3, which also appears to be correct according to the system authors intentions.

Then we counted the frequencies that similarity score ties appeared at levels of document, run (system-topic combination), and system. We also discovered the occurrences of rank contradictions (where the rank labels and similarity scores of two adjacent documents indicate opposite orderings) in the sorted runs. The results are listed in Table 3.2.

The tie rates of TREC-7 and TREC-8 shown in Table 3.2 are generally quite high and provide the motivation for our work here. There are 14% of documents having the same similarity scores with other documents in that run, generated by 98 out of 103 systems. And 1.3% of documents (in 7 distinct systems) are affected by the mislabeling and therefore there is no ordering for them that is consistent with both their similarity scores and their rank labels. The geometric average rank of the first tie in TREC-7 runs (not including runs which did not generate tie) is 66. Some systems did not generate ties. But ties of some systems started from very early ranks, around 5.

1	355	Q0	FBIS4-20436	0	18.34	bbn1
2	355	Q0	FBIS4-49691	1	16.13	bbn1
3	355	Q0	FBIS3-40224	2	15.98	bbn1
...	...	...	...	...	...	...
304	355	Q0	FBIS3-2363	303	0.005601	bbn1
305	355	Q0	FT931-2004	304	-7.763e-05	bbn1
306	355	Q0	FBIS3-25845	305	-0.0006148	bbn1
...	...	...	...	...	...	...
956	355	Q0	FBIS4-4125	955	-1.33	bbn1
957	355	Q0	FBIS3-27456	956	-1.33	bbn1
958	355	Q0	FBIS3-33460	957	-1.33	bbn1
959	355	Q0	FBIS4-45831	958	-1.33	bbn1
960	355	Q0	FT943-11853	959	-1.331	bbn1
...	...	...	...	...	...	...
993	355	Q0	LA101490-0067	992	-1.366	bbn1
994	355	Q0	FBIS4-3483	993	-1.367	bbn1
995	355	Q0	LA111089-0164	994	-1.368	bbn1
996	355	Q0	FBIS3-55378	995	-1.368	bbn1
997	355	Q0	FR940810-2-00026	996	-1.369	bbn1
998	355	Q0	FBIS3-59596	997	-1.369	bbn1
999	355	Q0	FBIS3-46125	998	-1.37	bbn1
1000	355	Q0	FBIS3-21021	999	-1.37	bbn1

Figure 3.3: The re-ordered run of `bbn1` for topic 355.

We used the re-sorted runs, and took the top 80 systems (ordered by mean AP score across the 50 topics, the bottom 23 systems were disregarded) in our next experiments to ensure that conclusions of our further evaluations would not be affected by the mislabeling issue, or the TREC-7 systems with low effectiveness performance (similar restrictions were also applied by other authors when considering the TREC-7 experimental submissions).

3.2.2 Ties in TREC-7 and TREC-8

The metric AP is one of the most widely-used evaluation metrics implemented in the program `trec_eval`, and also the primary effectiveness measurement employed in TREC-7 and TREC-8. In our next experiment, we explore the extent to which the similarity score ties might affect the AP score of systems.

For each of the 80 TREC-7 systems with all the runs sorted by the similarity score and rank (decried in Section 3.2.1), the mean AP score across the 50 topics computed by the `trec_eval` program is shown in the horizontal axis in Figure 3.4. Note that the `trec_eval` program sorts documents in non-increasing similarity score order (treats exponential number formats correctly) and orders tied documents by decreasing document identifier in the input runs without considering the line ordering or the supplied rank

		Percentage affected			First tie
		systems	system-topics	docs	
TREC-7	Tied scores	95.2	91.0	14.0	66
	Rank/score contradictions	6.8	4.2	1.3	
TREC-8	Tied scores	93.8	91.8	11.0	71
	Rank/score contradictions	6.2	5.6	1.8	

Table 3.2: Similarity score ties, rank contradictions, and geometric average rank of the first tie in TREC-7 and TREC-8 Ad-Hoc runs when re-sorted using score as a primary key and embedded rank number as the secondary key) runs. The first two rows show the percentage of 103 systems, 103×50 runs (system-topics) and $4{,}900{,}952$ documents that have tied similarity scores; the percentage of score-rank contradictions; and the average rank of the first tied scores in TREC-7. The last two rows shows the similar results for TREC-8 which has 129 systems, 129×50 runs and $6{,}295{,}843$ documents. Note that some runs returned by some systems contained less than $1{,}000$ documents.

field. As described in Section 3.1, using different methods to handle ties in the runs may result in different metric scores. In Figure 3.4, the `trec_eval` score is plotted as a circle for each system. We also calculated the average across permutations (plotted as a triangle), as well as the optimistic and pessimistic limits (shown as crosses), which are described in Section 3.1. Each system is plotted as a segment whose right and left ends reflect the best and worst AP scores that this system could receive from the optimistic and pessimistic orderings for each of the tied groups respectively. Systems determined by AP as having better performance are closer to the right-hand-side of the graph.

The vertical axis in Figure 3.4 shows the system's AP score range. For each system, the system score range is computed as the average of the differences between the optimal and pessimal run scores for each of the 50 topics. Note that the vertical axis is in log-scale and truncated at 10^{-6}. Systems with score range of 10^{-6} or below are all plotted along the line $y = 10^{-6}$. As can be seen in the graph, the higher up the vertical axis each system is plotted, the wider its score range (that is, larger score uncertainty) it has.

The color of each system point reflects the number of document ties generated by this system. Systems plotted by red points generated more than one thousand ties acrosss the fifty topics, thus they generally have wider score ranges than systems plotted in other colors, and so they are mostly plotted at the top of the graph. As can be seen from the graph, systems generated fewer similarity score ties have narrower potential score ranges, and are plotted lower in the graph overall. System scores computed by `trec_eval` are usually not too far away from the average across permutations. The amount of ties generated by the system has no obvious relationship with the retrieval quality of the system.

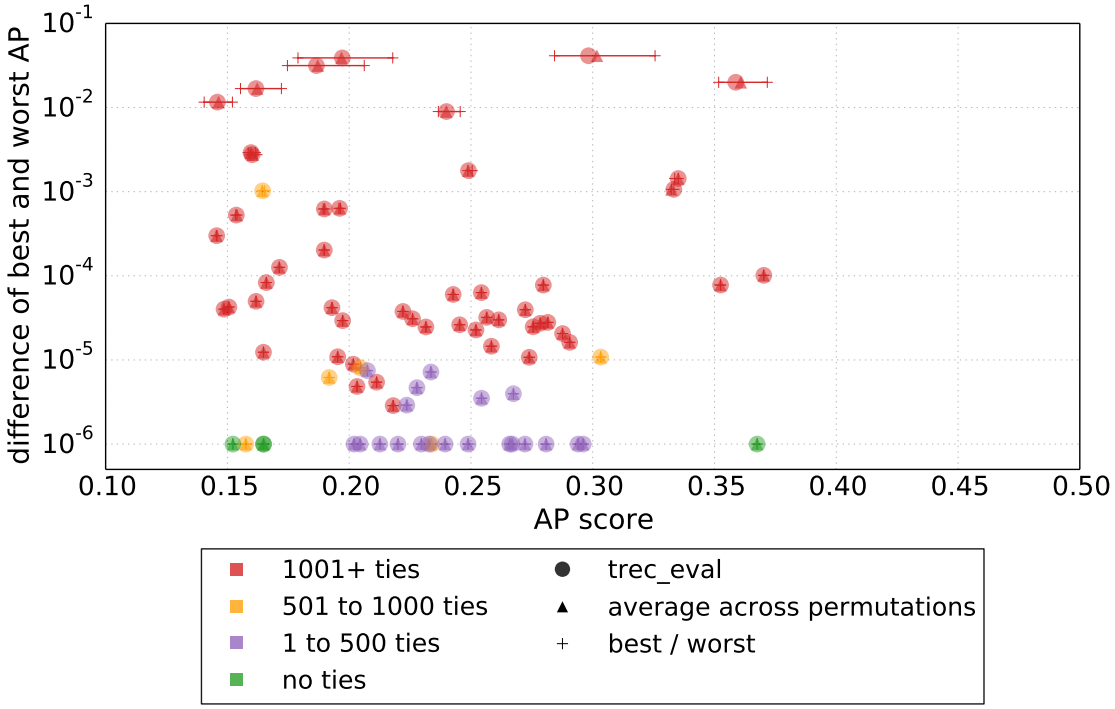


Figure 3.4: Imprecision in AP scores caused by ties in a set of 80 TREC-7 systems.

As Figure 3.4 illustrates, the best-to-worst spread of some systems is appreciable and overlaps with the range of other systems. In the top right corner of the graph, the system evaluated as the third best by `trec_eval` also has a wide score range caused by ties, whose best score is the largest AP scores that all TREC-7 systems could receive, and whose worst score is lower than the fourth best system. This best-to-worst spread overlap happens to other top systems (with AP score larger than 0.3) as well. Thus we conclude that ties may have affected the relative ordering of the top few systems.

At the bottom of the graph, only a small number of systems did not generate any tied scores. And fortunately, even when ties existed in most system evaluations, they did not lead to any discernible wide score range, instead, the topic-averaged optimal-to-pessimal ranges of most systems are less than 10^{-4}.

3.2.3 Ties in Other Years

We also explored several other TREC rounds and carried out the same analysis. For example, the tie rates of systems in TREC-8 were similar to those of TREC-7 systems, and are also shown in Table 3.2. Figure 3.5 illustrates the AP scores of systems in TREC-8 using different tie-breaking rules, similar to Figure 3.4. In TREC-8, there are also some systems with wide potential score ranges caused by similarity score ties, but the ordering of top few systems is not affected. Similar to the pattern shown in TREC-7, systems generating more than one thousand ties (in red) are plotted higher in the graph. There are three

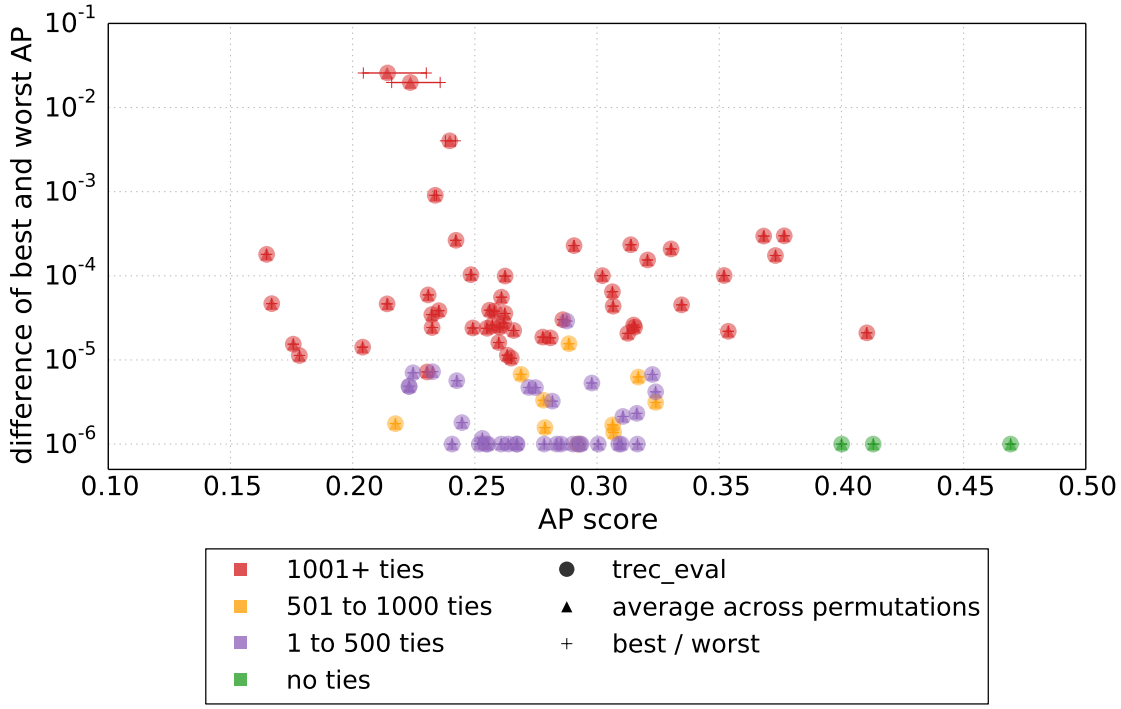


Figure 3.5: Imprecision in AP scores caused by ties in a set of 99 TREC-8 systems.

TREC-8 systems that did not generate any ties (in green) in their runs, and their average AP retrieval performances are amongst the best for this round of experimentations.

For TREC rounds in later years, the results of analysis are similar to those of TREC-8, and tie-breaking rules have no substantial effect on system evaluations.

3.3 Deliberate Score Grouping

In previous sections we explored the ties occurring in TREC runs. Although the rates of ties were not small overall, evaluations for TREC systems were largely unaffected. Thus in this section, we consider deliberately introducing ties to similarity scores and discover to which extent ties could be tolerated without having discernible effect on system effectiveness evaluations.

3.3.1 Score Approximation

Computing similarity scores for documents usually involves a great number of arithmetic, especially when using phrase components or term proximity components. Scores with high decimal precision may require longer computation time and that more index information be stored. Thus some regimes such as WAND [10] look for approaches that minimize the number of different scores assigned to documents without changing the ranking for the top-k documents. This approach only ensures the top-k documents are

in the right positions in the ranking but no guarantee is made for documents after rank k, which is deemed as being *rank-safe to depth k*. There are also some other computation-pruning techniques for providing flexible trade-offs between the precision of ranking and resources used in score computing.

In this project, we particularly consider a relatively weaker requirement: that each document has to be in the correct *band* of the ranking, where documents in each band have the same similarity scores and document ordering within the band could be arbitrary. In particular, suppose that the length of each band, l_g, the length of the g th ($g \geq 1$) band, is defined geometrically based on a parameter $\rho > 1$:

$$l_g = \lceil \rho^{g-1} \rceil.$$

More precisely, for the first band, $b_1 = 1$, and thereafter let $b_{g+1} = \lceil \rho \cdot b_g \rceil$.

For example, if $\rho = 2$, then the ranks of bands are $[1 \ldots 1]$, $[2 \ldots 3]$, $[4 \ldots 7]$, and so on; and if $\rho = 1.62$ (the golden ratio) the bands are $[1 \ldots 1]$, $[2 \ldots 3]$, $[4 \ldots 6]$, $[7 \ldots 11]$, and so on, with band lengths of numbers in the Fibonacci sequence. Documents in the same band are assumed to be assigned the same similarity scores, so they becomes ties. The score assigned to documents in the $g + 1$ th band is strictly smaller than the score of documents in the g th band.

Using this approach, the g th band will contain fewer documents if the controlling parameter ρ is smaller. The number of deliberately generated ties decreases with ρ as well. When the value of ρ approaches 1, the ranking is closer to the exact "true" ranking in which the relationships of all the documents are finalized. In this case, the system is required to perform a "full" computation without approximation and place each document at its final position without ambiguity. But when $\rho > 1$, a system is allowed to return groups of tied documents $[b_g \ldots e_g]$ (where $e_g = b_{g+1} - 1$), with the same scores assumed within each band, and by doing so, perhaps saving computational and spatial costs.

3.3.2 Worst-Case Bounds

As described before, when $\rho > 1$, ties are allowed to be deliberately introduced into the run, and the start rank of the first band that contains multiple documents (that is, documents tied) is given by $v = b_v = 1 + \lfloor 1/(\rho - 1) \rfloor$. It allows bounds on the imprecision in similarity score computation, and ensures that the score approximation mechanism is rank-safe to depth $v - 1$.

If we consider a run with the first relevant document in the v th group (the first group

ρ	Metric		
	ΔRR	ΔRBP($\phi = 0.5$)	ΔRBP($\phi = 0.85$)
1.1	0.0038	0.0002	0.0087
1.2	0.0119	0.0052	0.0231
1.4	0.0417	0.0429	0.0482
1.7	0.0833	0.0945	0.0777
2.0	0.0833	0.1016	0.0971

Table 3.3: Worst-case metric score differences for runs in which documents are geometrically grouped by parameter ρ. It is not possible to derive equivalent bounds for AP.

that has more than one document), the greatest loss of score that can arise when computing Reciprocal Rank (RR) across permutations is:

$$\Delta\text{RR} = \frac{1}{b_v} - \frac{1}{e_v - b_v + 1} \sum_{k=b_v}^{e_v} \frac{1}{k},$$

where the bound arises for the worst case when there is only one relevant document in the vth group and it is at rank b_v in the original run. Table 3.3 gives some ΔRR values; when $\rho \leq 2$, all are less than 0.1.

Similarly for the metric RBP, its worst-case differences could be computed by the score of the "best" ranking where relevant documents are placed at the start of each group followed by non-relevant documents for the rest of each group, minus the average across permutations. As denoted before (at the page 70), R_g is the number of relevant documents in the g th group ($0 \leq R_g \leq l_g$). The maximum difference of RBP (with the parameter ϕ) score of each group g is:

$$\Delta\text{RBP}_g = \left(\sum_{i=b_g}^{b_g+R_g-1} (1-\phi)\phi^{i-1} \right) - \left(\frac{R_g \cdot w_g}{e_g - b_g + 1} \right),$$

where $w_g = \sum_{i=b_g}^{e_g} (1-\phi)\phi^{i-1}$ is the sum of the RBP weights associated with that gth group. The overall bound on the RBP score difference is the sum of ΔRBP_g, that is

$$\Delta\text{RBP} = \sum_g \Delta\text{RBP}_g.$$

Table 3.3 shows ΔRBP for two different values of the RBP parameter ϕ. The choice of ϕ values is based on the expected evaluation depth of the metric. RBP($\phi = 0.5$) is a shallow metric with expected evaluation depth of 2. As results of RBP are similar when $\phi \geq 0.85$, we choose RBP($\phi = 0.85$) to represent deep metrics which model users with

high patience.

As Table 3.3 shows, for utility-based metrics, RR and RBP, the worst-case bounds (the uncertainties of score across permutations) increase with the value of ρ. That is, when the number of documents in each group grows, the number of ties in the run increases, and the worst-case bound of metric score becomes greater.

Recall-based metrics such as AP cannot be analyzed as readily. The ΔAP may not grow with ρ. Because when the value of ρ increases, the proportion of relevant documents in each group g (that is, R_g/l_g) might grow, which might decrease rather than increase the score of $\overline{\text{AP}}$ as the division base $\boldsymbol{R}$ (the total number of relevant documents) does not change, according to the Equation 3.3. The reason is similar for other recall-based metrics.

3.3.3 Effectiveness Score Difference in Practice

As the results shown in Table 3.3, the worst-case bounds do not decrease with the value of ρ, that is, the more ties are assumed to be deliberately introduced in the run, the greater the effect they have on effectiveness metric scores. So our next research question is, to what extent does an allowance of imprecision in similarity scores affect metric scores in practice?

We again used the systems and runs of the TREC-7 resources, as were examined in Section 3.2. As already stated in Section 3.2, 23 TREC-7 systems with low AP scores were removed from our experiments. For each run, we applied the grouping technique described in Section 3.3.1 with a set of ρ values. The similarity scores of documents given by the systems in the original runs were ignored as the grouping operation was being carried out. Documents ranked in the g th band were assigned a synthetic score of $1/g$. That is, for example, the score of $1/1$ was assigned to document(s) in the first group, and $1/2$ was assigned to documents in the second group. The assumed similarity scores of documents in later groups are always strictly smaller than scores of documents in previous groups.

The banded runs with deliberately introduced ties, and the original runs with orderings in the submitted files (used as reference points), were then evaluated by four metrics: RR, RBP($\phi = 0.5$), RBP($\phi = 0.85$) and AP respectively. The scores of the banded run were the average across all permutations; we followed the standard protocols and treated all unjudged documents as non-relevant when scoring the runs.

For each run and each ρ, the banded run score was compared with the original run score, and their difference was plotted in Figure 3.6. Each graph in Figure 3.6 is for one metric, and contains a sequence of box-whisker elements for different values of ρ. Each column (for each ρ) shows the distribution of score differences for 4000 runs (80 systems $\times$ 50 topics).

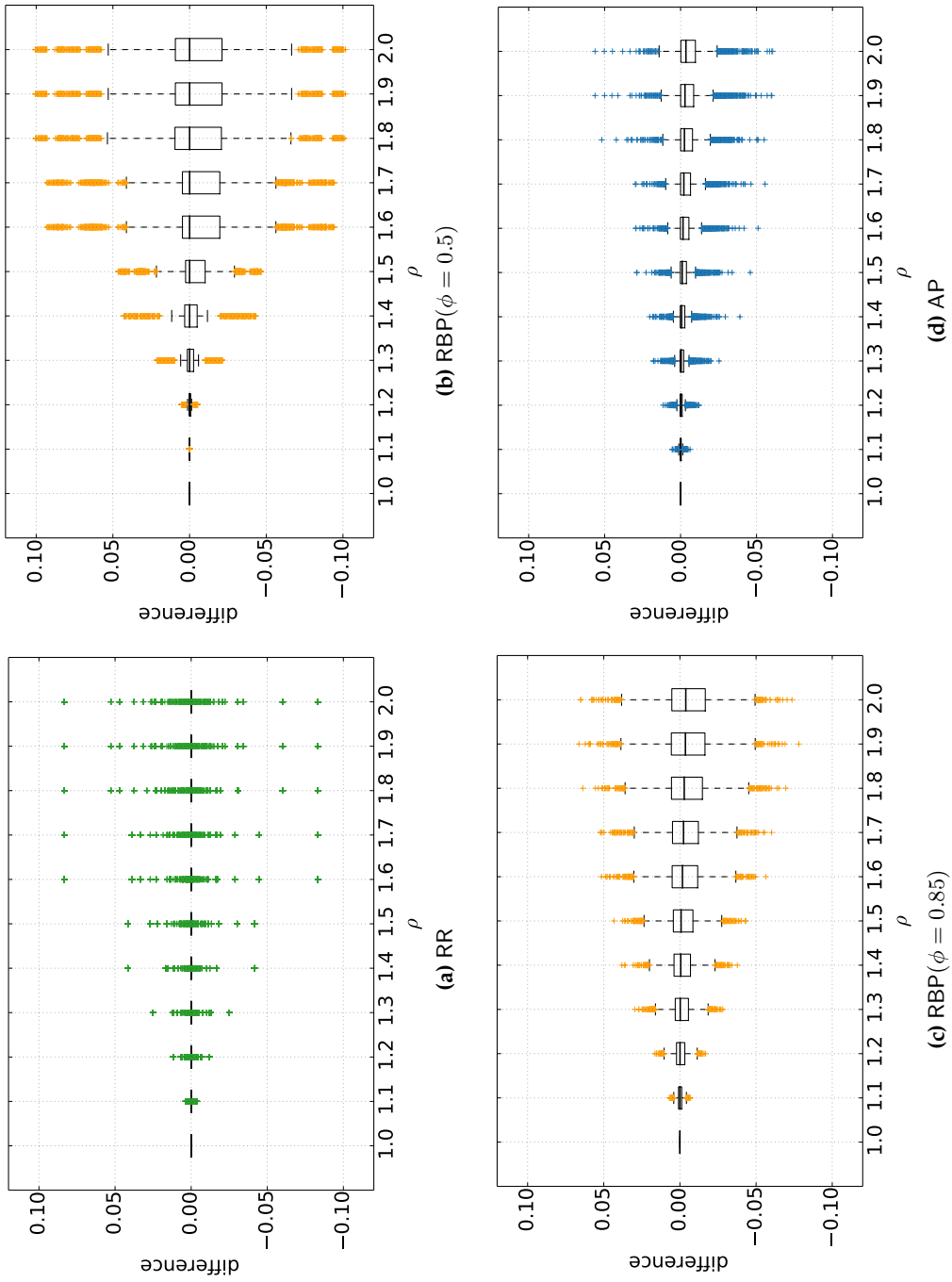


Figure 3.6: Variation in metric effectiveness score across a set of 80 systems and 50 topics (that is, 50×80 points are plotted in each column), as a function of ρ from 1.0 to 2.0, for four different retrieval effectiveness metrics. The whiskers indicate the last outlier still within 1.5 times of the inter-quartile range from the corresponding quartile (the limits of the boxes).

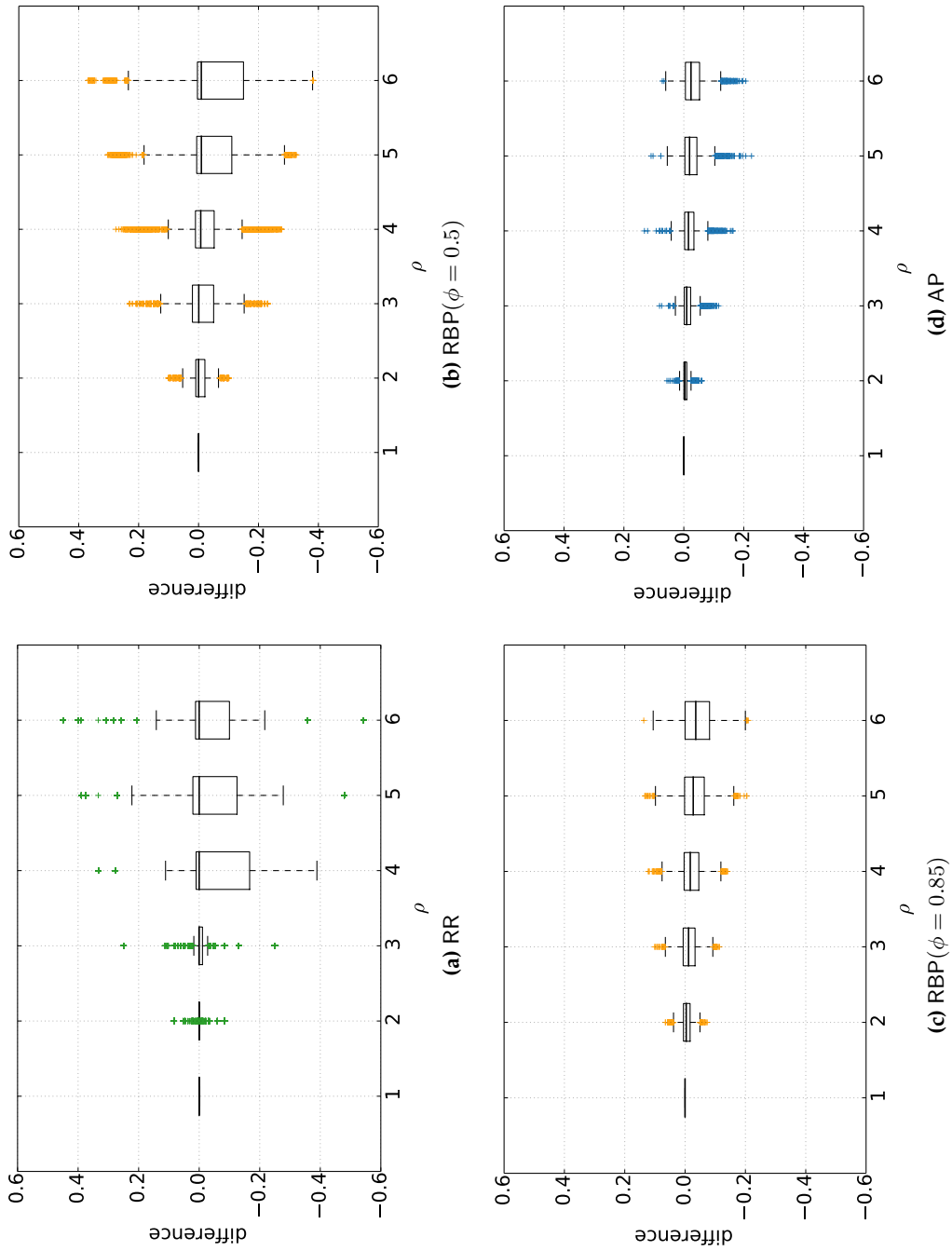


Figure 3.7: Variation in metric effectiveness score of 50×80 runs, as a function of $\rho \in \{1, 2, 3, 4, 5, 6\}$, for four different retrieval effectiveness metrics.

Figure 3.6 shows that the variation of effectiveness run scores arising from the grouping process is relatively small. Most scores of RR (the most shallow metric) are not affected by all of the ρ value tested. The inter-quartile ranges of score difference boxes for two deep metrics, RBP($\phi = 0.85$) and AP, are also small. The original metric scores averaged over all runs for RR, RBP($\phi = 0.5$), RBP($\phi = 0.85$) and AP are, respectively, 0.6939, 0.5556, 0.4677, and 0.2311.

As can be seen, scores of AP are generally smaller than other metrics, and hence so are the AP score differences shown in Figure 3.6. The metric of RBP($\phi = 0.85$) suffers the most from the deliberately generated ties, but only when $\rho > 1.5$, the first value for which documents at rank 2 and 3 are grouped into the same band, so the scores changes a lot.

When $\rho \leq 2$, there is always only one document in the first band. We also explored the score differences arising when ρ becomes bigger than 2, and carried out the same analysis, shown in Figure 3.7.

The pattern is similar shown in Figure 3.7, the score differences generally increase with the value of ρ. For deep metrics such as RBP($\phi = 0.85$) and AP, the effects of ties on the run scores are on average still modest. But for shallow metrics, the grouping process affects the run scores heavily when $\rho \geq 4$. The variation of RR score differences becomes obvious when $\rho > 3$. We found that around 60% of runs receive score of 1.0 by RR when using the original orderings, and about 76% runs having at least one relevant documents in top-3. Moreover, the Prec(k) score of TREC-7 runs generally decreased with k, that is, the number of runs whose Prec($k+1$) $<$ Prec(k) was bigger than the number of runs with Prec($k+1$) $>$ Prec(k). This decreasing is about 5% when $k = 1$, but grows to 10% when $k = 3$, and then keeps steady when $k > 3$. The RR and Prec(k) scores suggest that, the proportions of relevant documents in the first two positions of the runs are generally higher than the proportions of relevant documents ranked at the third and the fourth positions. And that is why the RR score might be affected greatly if the first band contains more than three documents. Thus we conclude that the number of documents the first band should not be greater than three (that is, when $\rho \leq 3$) if RR is employed for the evaluations.

To further test if the score differences shown in Figure 3.6 can be regarded as being significant, for each system, its run scores computed using the original runs were multiplied by 0.99 and compared to the banded run scores using a one-tail paired t-test. If the generated p-value was less than or equal to $\alpha = 0.05$ (the significance level), it yields confidence that the grouping process did not degrade the system score by 1% or less. The t-test was applied for each of 80 systems, and we counted the number of systems that were significantly affected by the introduced ties and summarized in Table 3.4. The closer the number of such systems is to 80 in the table, the more confidence we have that ties deliberately given by the grouping process will not cause "notable inferior" (defined as being δ lower than the original scores) to the system scores. The δ is defined as 1% in

ρ	Relative to 99% of original score				Relative to 97% of original score			
	RR	RBP0.5	RBP0.85	AP	RR	RBP0.5	RBP0.85	AP
1.1	80	80	80	80	80	80	80	80
1.2	80	80	80	80	80	80	80	80
1.4	77	44	65	44	80	80	80	80
1.7	37	11	14	0	80	67	80	77
2.0	38	10	3	0	80	61	71	20
3.0	2	3	0	0	14	11	6	1
4.0	0	0	0	0	3	2	1	0
5.0	1	0	0	0	1	0	0	0
6.0	1	0	0	0	2	0	0	0

Table 3.4: Number of systems (out of 80) for which a one-tail paired t-test across 50 topics yields confidence at the $p \leq 0.05$ level that the banded runs yield a metric score greater than or equal to 99% (left) and 97% (right) of the original run scores for that system.

the left half, and $\delta = 3\%$ in the right half of Table 3.4. The possible implications of score changes increase when the fidelity of similarity score drops (that is, the value of ρ grows), indicated by the decreasing numbers in this column of the table. If we further reduce the tolerable degradation limit to 95% (that is, $\delta = 5\%$), none of systems was affected when $\rho \leq 2$ for all four metrics.

3.3.4 System Comparison Sensitivity

Systems can be compared in pairwise manner using their effectiveness run scores. In the last experiment, we explore the effect that similarity score ties, which might be caused by score rounding, have on the ability of effectiveness metrics to distinguish systems in pairs.

In the normal process of comparing two systems, a paired t-test is carried out by taking the run scores, over a set of topics, of these two systems, then the generated p-value indicates whether the systems are significantly different. The smaller the p-value is, the more confidence we have that the outcomes of the two systems being compared on the data used are distinct. When the resulted p-value is less than or equal to the significance level α, often $\alpha = 0.05$, the pair of systems are deemed as being significantly different.

To measure how the similarity score rounding might affect the system comparisons, we again used the runs of 80 systems and 50 topics from TREC-7. For each of four evaluation metrics (shown as a group of graphs in Figure 3.8, 3.9, 3.10 and 3.11 respectively) and each $\rho \in \{1.1, 1.4, 1.7, 2.0\}$ (shown as a pane in each Figure), each of the generated $80 \times 79/2 = 3{,}160$ system pairs is plotted as one point, whose value on horizontal axis

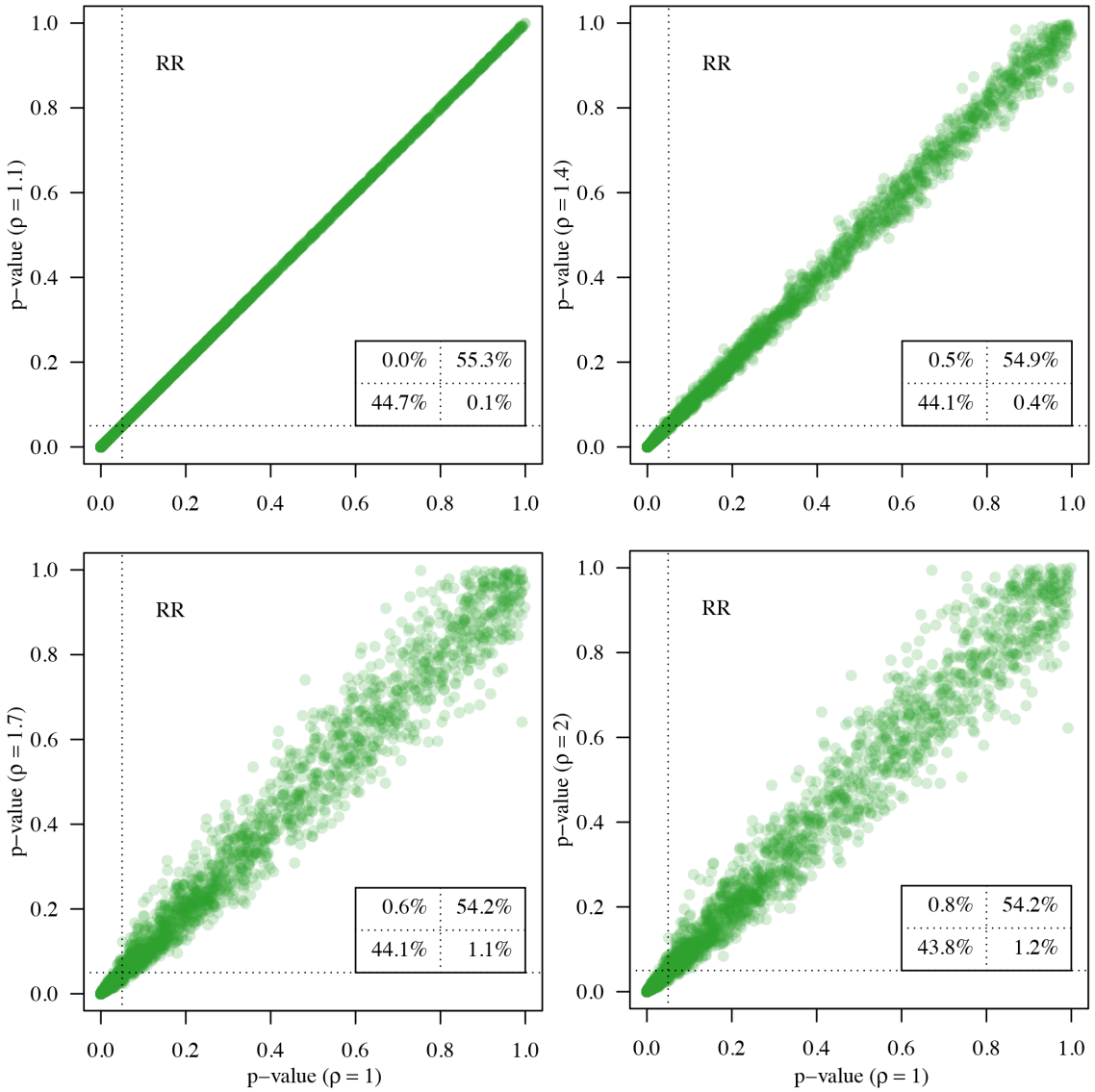


Figure 3.8: Correlation of p values for all pairs of systems ($80 \times 79/2 = 3{,}160$ points per pane), with the p value from a paired t-test using the original system RR scores across 50 topics plotted on the horizontal axis, and the p value for the corresponding system pair with banded runs ($\rho \in \{1.1, 1.4, 1.7, 2.0\}$ in the four panes) on the vertical axis. The dotted lines at are p-value$= 0.05$, with the grid showing the percentage of data points in each quadrant, in each of the four panes.

is the p-value given by a paired t-test taking the original run ($\rho = 1$) scores of the paired systems, and with the p-value on the vertical axis generated using the banded run scores (applied the grouping technology using $\rho > 1$). Similar to previous experiments in this project, all banded run scores were computed using the averaging across permutations process described in Section 3.1, and the original run scores were given by the metric using the sorted-by-score order without considering the similarity scores.

In each pane, there is an additional table scoring lines the percentage of points in each

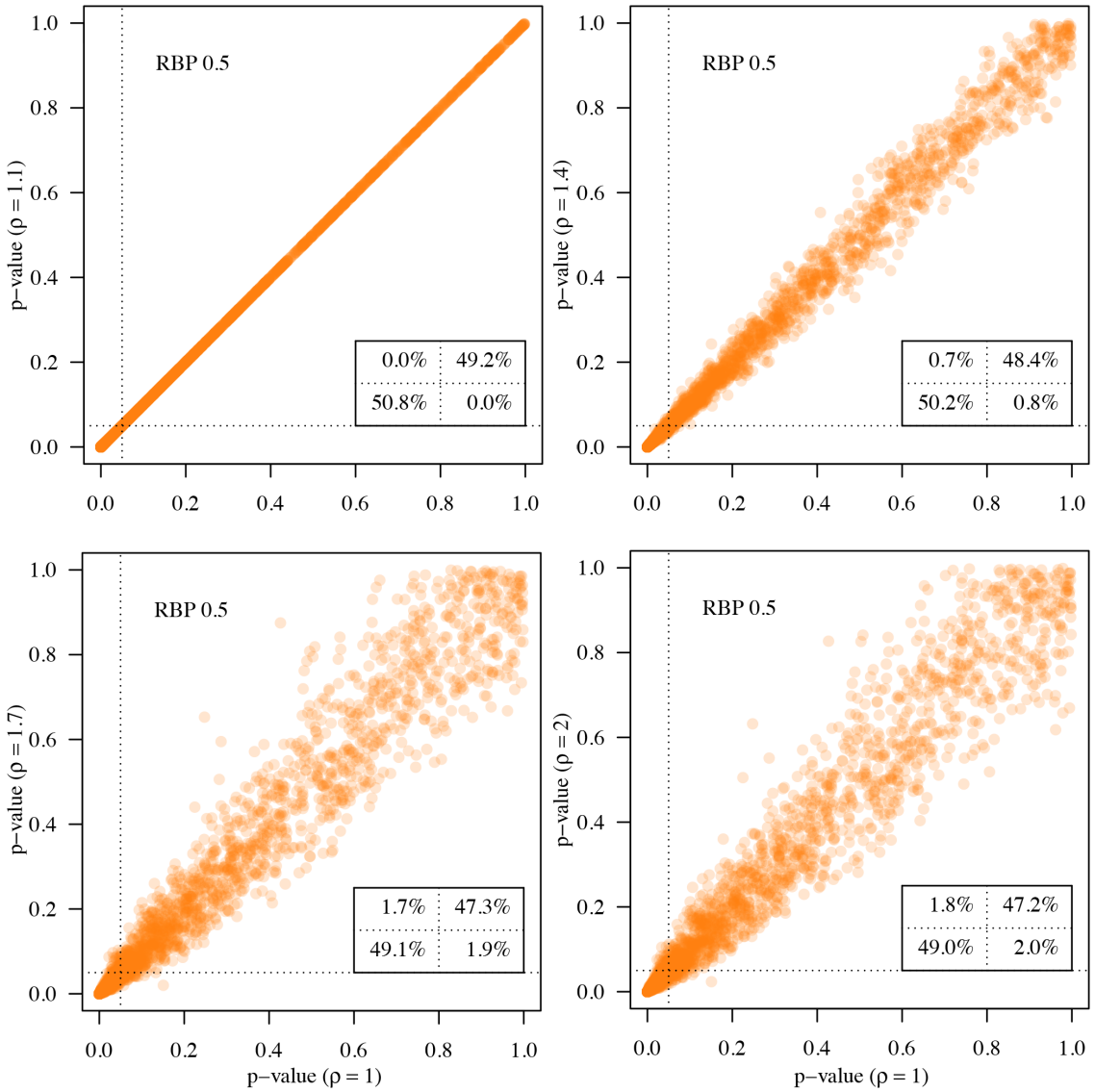


Figure 3.9: Similar to Figure 3.8, the correlation of p values for all pairs of systems generated by paired t-tests using the RBP($\phi = 0.5$) scores of the original runs and banded runs respectively.

quadrant when divided by the two p-value $= 0.05$. For example, the right pane of the last row in Figure 3.8 for RR shows when $\rho = 1.0$ (no grouping process applied), 44.6% of the system pairs could be distinguished by RR for their p-value≤ 0.05; when $\rho = 2.0$, that fraction is 45.0%.

As Figure 3.8, 3.9, 3.10 and 3.11 show, when the ρ of the grouping rule increases, the correlation of p-values generated using original runs and banded runs (with the grouping rule applied) decreases, and system pair points spread out. For all the metrics, the percentages of system pairs in each quadrant (the sum of percentages in four quadrants is 100%), shown in the grid box of each pane, indicate that the sum of true positive (systems can be distinguished by the metric using the original runs can also be distinguished using

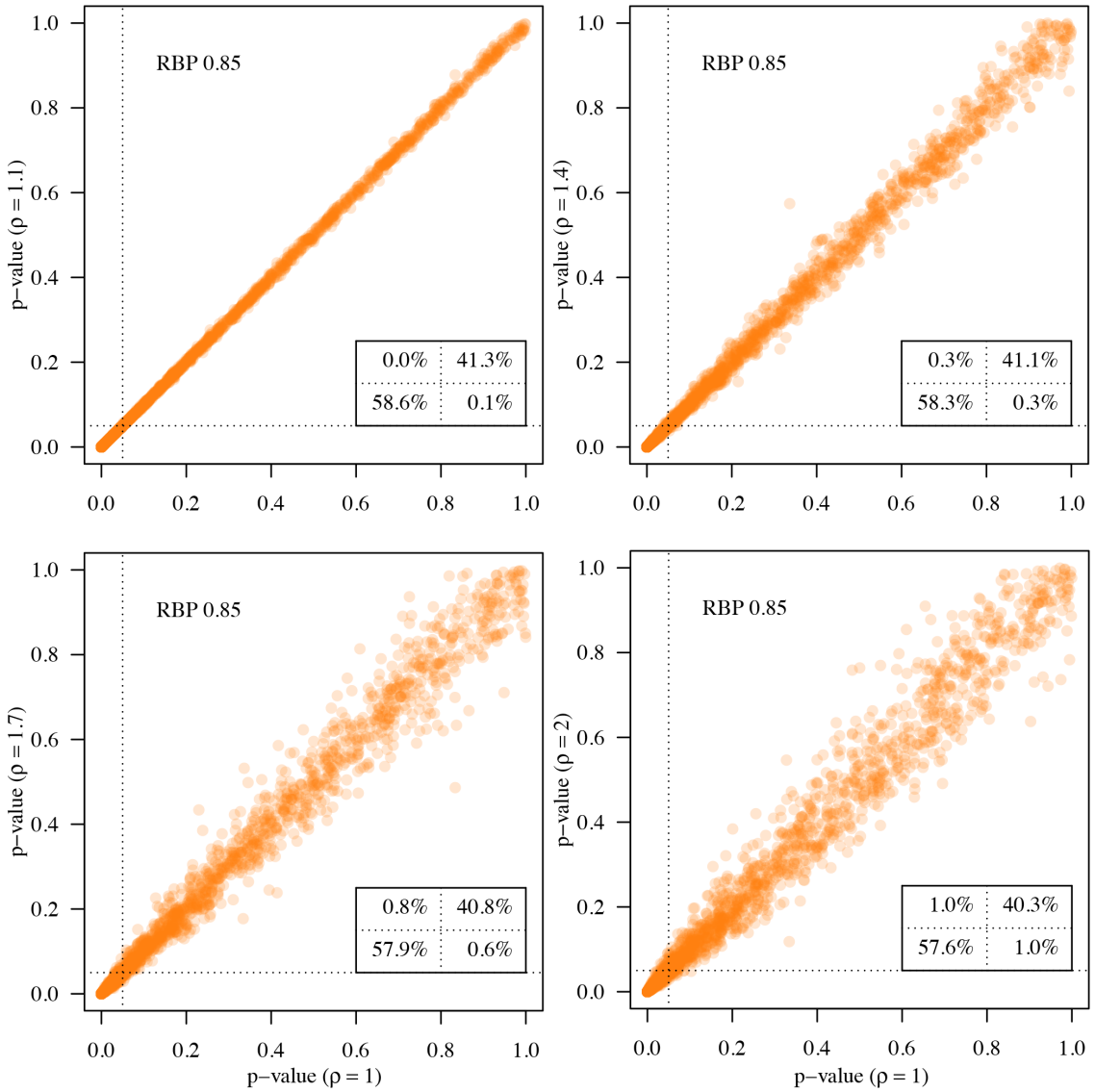


Figure 3.10: Similar to Figure 3.8, the correlation of p values for all pairs of systems generated by paired t-tests using the RBP($\phi = 0.85$) scores of the original runs and banded runs respectively.

the banded runs) pairs and the true negative (neither original runs nor banded runs can be used to distinguish the paired systems) pairs is over 96% even when $\rho = 2.0$.

For example, in Figure 3.8, there are 44.7% true positive pairs (their p-values on horizontal axis and vertical axis are both not greater than 0.05) when $\rho = 1.1$. Even when ρ increases to 2, the rate of true positive pairs only drops 0.9%. The number of true negative pairs also declines when ρ grows from 1.1 to 2.0, but only for 1.1%.

The decrease rates of true positives and negatives fractions are similar for other metrics (RBP($\phi = 0.5$), RBP($\phi = 0.85$) and AP). Moreover, by comparing the results in Figure 3.8, 3.9, 3.10 and 3.11, the decreases of true positives and negatives are found to be smaller for deeper metrics such as RBP($\phi = 0.85$) and AP. The fractions of false positives

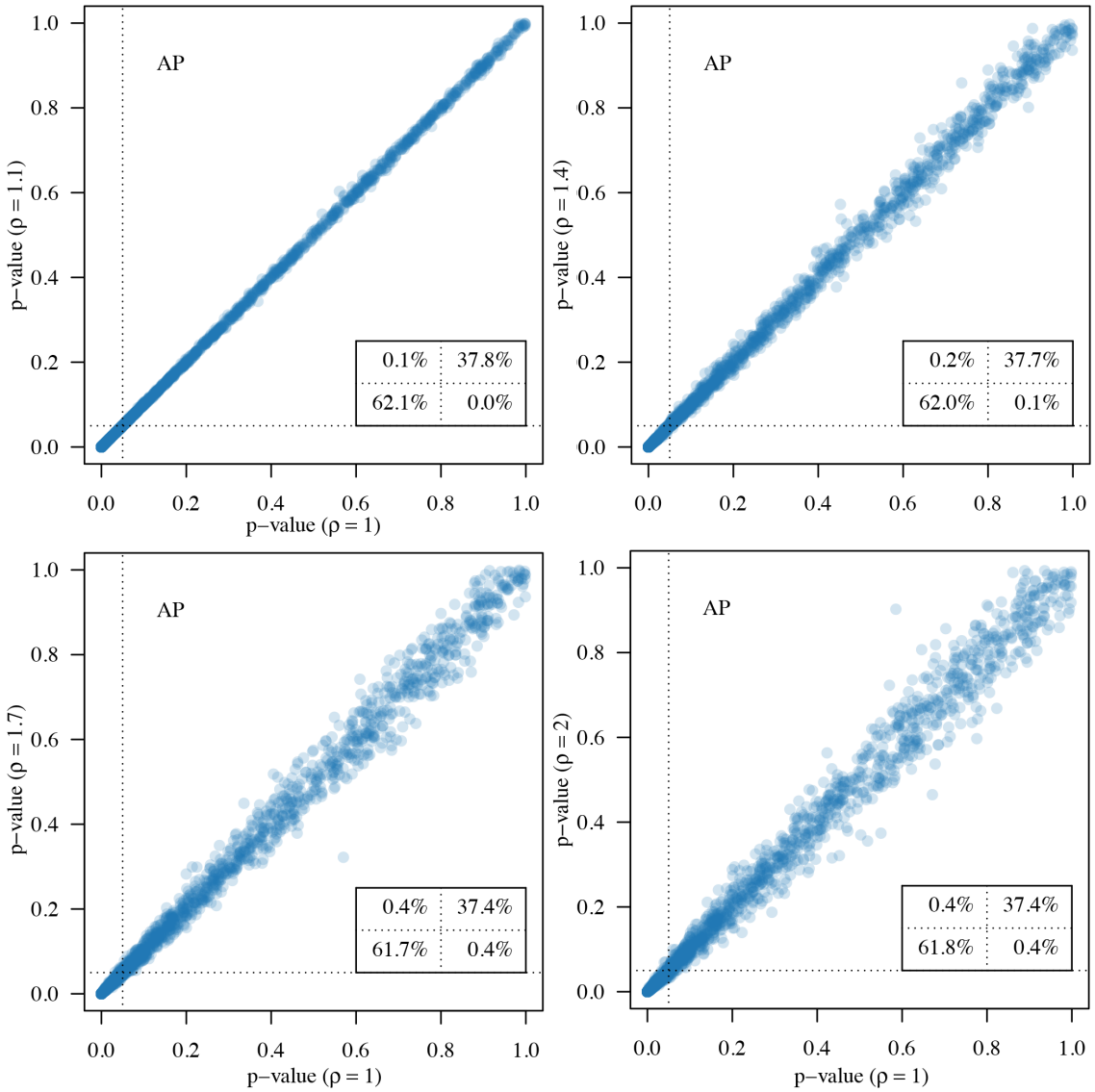


Figure 3.11: Similar to Figure 3.8, the correlation of p values for all pairs of systems generated by paired t-tests using the AP scores of the original runs and banded runs respectively.

and negatives are small for all the metrics with all values of ρ.

In general, we conclude that the similarity score ties deliberately introduced by the grouping process have almost no effect (even when ρ grows to 2) on the system discrimination of the metrics used in these experiments.

3.4 Summary

In this Chapter, we described our explorations into how similarity score ties might affect the evaluation of retrieval systems effectiveness, measured using the NIST binary relevance judgments and the established metrics of RR, RBP($\phi = 0.5$), RBP($\phi = 0.85$) and AP. Documents receiving the same similarity score values from the system are deemed as ties. We described four strategies of ordering tied documents in the ranked list. Although neither of them would break the "non-increasing" rule when sorting retrieved documents by similarity scores in the run, document orderings in the final runs employing different tie-breaking strategies may not be the same. Thus, their effectiveness scores may be distinct (we demonstrated detailed examples in Section 3.1), and hence may affect system comparisons.

We illustrated the mislabeling mistakes associated with ties, or caused by programming errors that failed to handle scientific notation correctly, in the original TREC runs in Section 3.2.1. We resolved these mistakes by resorting the runs to make sure that there were no score-based out-of-order documents, and then reported the rates of ties occurring in TREC-7 and TREC-8. We found that the numbers of ties generated by TREC systems were conspicuous, and which might be caused by similarity score rounding. We explored system AP scores using different tie-breaking strategies, and the potential AP score range of each system in Section 3.2.2. As Figure 3.4 and 3.5 illustrated, a non-trivial number of systems did generate ties in the runs, and which in same cases then led to ambiguous score outcomes. Fortunately, most system comparisons were not affected, and the overall conclusions of system evaluations for TREC-7, TREC-8 and other TREC rounds were unlikely to have been compromised.

In Section 3.3, we proposed the use of a hypothetical "controlled grouping" rule to deliberately introduce ties into TREC runs, and explored to what extent the ties could be allowed without affecting system comparisons. The parameter ρ in the grouping process was used to control the extent of deliberately introduced ties. For each of four metrics: RR, RBP($\phi = 0.5$), RBP($\phi = 0.85$) and AP, we illustrated the system effectiveness score variations when using the original (re-sorted) runs and the banded runs (with the grouping rule applied). We found that the average system scores generally changed by more as ρ increased (that is, as more ties were deliberately generated). The shallowest metric RR was nearly unaffected by the score grouping process for all of the tested ρ values from 1.0 to 2.0. The deeper metrics suffered more from the introduced ties, but the variations were still minor. We further used statistical tests to confirm if the grouping process significantly degraded the original system scores. As expected, the grouping process gave notably inferior (degradation of δ in system scores) sooner for deep metrics (such as AP) than shallow metrics (such as RR) when ρ increased. This pattern was similar for RBP, but not always true for all ρ values. Thus it is worthy to perform the same analysis for other metrics such as NDCG, Prec(10), Prec(100), INST and other metrics, to further determine whether this grouping process affect deep metrics greater than shallow metrics in

regards of system scores, even the user models of metrics are distinct.

In our final experiment described in Section 3.3.4, we explored how the score rounding (which leads similarity score ties) might affect the ability of metrics RR, RBP and AP to discriminate systems in pairs. Similar to previous experiments, for each of four metrics, and for each system pair, a t-test took the run scores (evaluated by the given metric) of two systems, and computed a p-value to indicate if the given paired systems were significantly different. For each system pair, its p-value generated using the original runs ($\rho = 1$) was compared with the p-value computed using the banded runs ($\rho > 1$). The patterns shown in Figure 3.8, 3.9, 3.10 and 3.11 demonstrate that, for all of the tested metrics, their ability of discriminations slightly decreased when the ρ grew (that is, the level of score rounding increased, and more ties were introduced), but the changes were small. This result opens the possibility of retrieval efficiency being improved by reducing the precision of score computations, without affecting system comparisons.

In general, we explored the ties arising in similarity scoring in TREC runs, and showed potential AP score ranges (associated with ties) that TREC systems might have. We concluded that even though a great number of ties were generated by systems, and that they did have potential to affect system comparison, in practice, they had only minor impact on system evaluations. Then, we proposed a controlled grouping rule that deliberately introduced ties to the runs. We further demonstrated that permitting ties in the TREC runs resulted in only small variations in the ability to compare systems. Reducing the accuracy of similarity scores to improve the search speed and reduce the space used is feasible.

We have not yet implemented the approach to achieve the efficiency gains by reducing the precision of scoring. But a clear direction for this method is to explore the computation embedded in similarity scoring regimes employed by the systems, and existing dynamic pruning heuristics (such as WAND [10], MAXSCORE [106], Score-at-a-time [64], and so on), and discover how the inexact scoring could be used to gain the efficiency without degrading the effectiveness of IR systems.

The conclusions we have made in this part of the investigation may be correct only when the NIST Binary relevance judgments are used. As there are only two relevance levels (relevant, and irrelevant) in the binary judgments, if the tied documents are in the same relevance level, ordering them in different ways will not alter the run score. However, if relevance judgments with higher fidelities (such as Sormunen [99] with four relevance categories) are used, the probability that relevance levels of the tied documents are the identical may be smaller, which may lead to larger changes to the run scores, and alter the conclusions made.

Chapter 4

Uncertainty In Recall-Based Effectiveness Metrics

As described in Chapter 2, the effectiveness of IR systems is commonly assessed using batch evaluation technique. For each given topic, the prefix of runs returned by the selected (contributed) systems are pooled to create relevance judgments. As Depth@k is employed in TREC experimentations, the number of documents contributed by each selected systems is the same. The chosen metrics use those judgments to compute an effectiveness score for each run and aggregate (usually arithmetic mean) scores into a single value which is deemed to be the retrieval quality of the tested system.

However for documents returned by the system but not in the pool, metrics conventionally treat them as not relevant documents and so the performance of the system may be under-estimated or even over-estimated. If utility-based metrics are used, system effectiveness can be under-estimated when treating unjudged documents as irrelevant. However, as system scores evaluated by recall-based metrics are usually normalized by the total number of relevant documents, they may be over-estimated when the total number of relevant documents decreases. The upper limit of this measurement uncertainty (that is when all unjudged documents are relevant) can be provided by residual given by utility-based metric such as RBP. In this chapter, we therefore consider the following research questions:

RQ3: Is there a connection between the utility-based metrics and the recall-based metrics such as AP and NDCG?

RQ4: If there is, what do the utility-based metric residuals indicate when comparing the effectiveness of paired systems over a set of topics?

4.1 Reliability of Pooling and Measurement Uncertainty

As described in Section 2.2, it is impractical to obtain comprehensive human relevance judgments covering all documents in large collections for even a small number of topics. Thus in the evaluation of collections such as TREC, uniform pooling strategy Depth@d

(see detailed descriptions in Section 2.2.1) is typically employed to collect relevance judgments for top-d documents retrieved by a subset of participating systems. The pooling depth d is constrained by the available budget for judging and is fixed for all selected systems so that they have equal chance to contribute to the pool [22].

However, when the same document collection and relevance judgments are used to evaluate a new system which does not contribute the pool, the runs returned by this new system may include previously unjudged documents even at depths prior to d. These unjudged documents are generally treated as not relevant by IR experiments, which may cause potential bias against the new system and therefore triggers extensive researches about the reusability of test collections.

Zobel [120] carried out "leave one out" experiments that re-evaluated a system using judgments without documents uniquely contributed to the pool by that system. Zobel [120] concluded that the performance of a new system could be fairly evaluated by existing TREC collections but might be underestimated. In addition, the number of unjudged documents retrieved by the new system might need to be considered.

The impact of incomplete relevance judgments for newer TREC-8, TREC-10 and TREC-12 was analyzed by Büttcher et al. [15]. They randomly selected a percentage of judgments from the full qrels and progressively removed them when evaluating systems. Their analysis leaded to the conclusion that the Kendall's τ between system orderings generated using full judgments and reduced judgments respectively decreased when the reduced judgments became more incomplete. A new metric BPref proposed by Buckley and Voorhees [11] was shown to retain higher correlation than other metrics such as AP when more judgments were removed.

Tonon, Demartini, and Cudré-Mauroux [103] used residual scores to measure the impact of unjudged documents in TREC evaluations, and therefore decided the new pooling strategy (collect new judgments for high-impact documents, and merge them into existing judgments) to improve the evaluation reliability.

Condensed versions of metrics AP, NDCG and Q-Measure were proposed by Sakai [81], where all unjudged documents were firstly removed from the ranked list before calculating the metric score. This approach was demonstrated to be more effective than BPref regarding Kendall's τ and discriminative power. Sakai and Kando [88] further extended this method to the test collections TREC and NTCIR. They randomly removed 10% of judgments and analysis the impact of the reduction on correlations of system effectiveness scores, and on metric discrimination power. The authors concluded that the condensed versions of metrics perform better than the standard ones. The random judgment reduction method was then compared with depth-based reduction by Sakai [83, 82] which analyzed two kinds of bias introduced by random reduction.

In addition to the problem of unjudged documents when evaluating system re-using the test collection, many other factors such as judgment variations made by assessors (described in Section 2.2.3) which may reduce the reliability of relevance judgments have

been studied. Buckley et al. [12] state that systems selected for pooling usually prefer documents containing query terms in the title. Thus new systems employing wholly different retrieval methods may not be fairly evaluated using such test collection and judgments. Moreover, as the size of the test collection increases, the pool may not be representative anymore.

For utility-based metrics such as RBP, when the pool size is enlarged and more relevance judgments are added, the RBP score will increase if any previously unjudged document is assessed as relevant according to the new judgments. As newly found relevance gain would be weighted (which is always positive) and added to the RBP score, the RBP score can only increase, and the RBP residual decreases when new relevant documents are found. However for recall-based metrics such as AP and NDCG, their scores are normalized by the total number of relevant documents in the judgments, thus there is no guarantee that the score of recall-based metrics will increase, or decrease, when previously unjudged documents are found as relevant. There is no exist method to compute the uncertainty (the upper bound of metric score changes because of the unjudged documents) of AP or NDCG scores. The brute force strategy is to compute metric scores of all the combinations across all of the unjudged documents and relevance levels, and then take the maximum of score differences.

For example, suppose that some run has these relevance scores for the top-5 documents in the ranking (1 for relevant, 0 for non-relevant, $?$ for unjudged):

$$1, 0, ?, 1, ?,$$

and assume that there are only these five documents in the test collection, and three documents in the relevance judgments (and hence that, $R = 2$). The AP score of this ranking using the current judgments is: $(1/1 + 2/4)/2 = 0.75$. If the third document is in fact relevant, and the fifth document is non-relevant, then $R = 3$, and the AP score increases to $(1/1 + 2/3 + 3/4)/3 = 0.81$. If the third document is not relevant, but the fifth document is relevant, even though the value of R is still 3, the AP score drops to: $(1/1 + 2/4 + 3/5)/3 = 0.70$. If the unjudged documents are both relevant, the AP score of this ranking would be $(1/1+2/3+3/4+4/5)/4 = 0.80$. As including more relevant documents (which were outside the previous pool) might not lead increment of the metric score, to compute the score uncertainty, we need to compute metric scores for all of the situations covering all the unjudged documents, which is very expensive when the ranking is long, or the test collection is large. In this example, we need to consider four situations for only two unjudged documents at first five ranks. But in TREC evaluations, most runs contains about one thousand documents. And the pool depth of NIST Binary judgments for TREC test collections is usually only 100. The number of unjudged documents is even larger in practical searches.

To help understand these complex relationships, this chapter considers how to apply a RBP-like residual to the computation of recall-based metrics such as AP and NDCG. We

discover the "best" ϕ parameter to make RBP(ϕ) the most similar to each recall-based metric, and use RBP residual with that ϕ to estimate the uncertainty of the recall-based metric. We also explore the relationship between the pool size and RBP residual in regards of system comparisons.

4.2 Datasets and Methodology

We make extensive use of the TREC-developed experimental materials already described in Chapter 2. In each round of TREC, the participating systems developed by university research group and commercial organizations were tested on a set of topics and documents. The generated runs were then submitted for pooling and later evaluation.

We use three of the newswire collections: the TREC-7 Ad-Hoc Track [113], the TREC-8 Ad-Hoc Track [112] and the TREC-13 Robust Track [107]. Parameters of each TREC round that we used in our experimentations are summarized in the Table 4.1. The first five rows show the year the topics were used, the tested topic IDs, the total number of tested topics, the total number of participating systems, the number of systems contributing to the pool, and the pooling depth of each TREC round. Note that when we used the dataset of TREC-13, we removed topic 672 from the original topic set because no relevant documents were identified by the pooling, which means that recall-based metrics cannot be used in the evaluation of this topic. Note also that because TREC-13 combined and reused parts of judgments from previous years [107, 110], the pooling parameters are shown as "n/a" in the table. The last four rows shows the computed results that: the average number of documents judged per-topic, the average number of relevant documents (about 5–6% of all judged documents) per-topic, the average (across systems and topics) rank at which the first unjudged document appears in the run, and the number of *deeply-judged systems*, defined as the set of systems for which the lengths of the generated runs were all at least 50 and, all documents in the runs were judged down to at least rank 50. We found that some pooled systems generated a least one *short run* whose length is less than 50 even the pooling depth is reported as 100 by track overviews [112, 113]. So we report the number of deeply-judged systems whose runs for every topic contained at least 50 documents and all top 50 documents were judged, in the last row of Table 4.1.

As TREC-13 re-used the qrels from previous years, there is no sense counting the number of TREC-13 runs contributed to prior-year pools. If only the 49 new topics are considered, pooled and judged topics in TREC-13, there are 52 deeply judged systems and 42 systems contributed to the pool.

In a run with all documents ranked from 1 to k judged (including the case of short runs) and the document at rank $k+1$ is the first unjudged, $k+1$ is deemed as the depth of the first unjuged document of that run. The second-to-last row of Table 4.1 reports the average depth of the first unjudged document for all the deeply-judged topic-system combinations in each collection.

Dimension	Collection		
	TREC-7	TREC-8	TREC-13
Year	1998	1999	2004
Topics	351–400	401–450	301–450 601–671 673–700
Number of topics	50	50	249
Number of systems	103	129	110
Number of systems pooled	77	71	*n/a*
Pooling depth	100	100	*n/a*
Number of documents judged (avg.)	1606.9	1736.6	1250.6
Number of relevant documents (avg.)	93.5	94.6	74.1
Depth of first unjudged document (avg.)	101.8	95.8	69.6
Number of deeply-judged systems	65	67	0

Table 4.1: TREC collections and relevance judgments used in experimentation in this chapter.

As the issue of ties in TREC rounds described in Chapter 3, we re-sorted documents in all the submitted runs of these three tracks into decreasing similarity score order, and also correct orderings of documents with scores using exponential notation. If documents had ties on similarity scores (column five in the submitted file of each run), the assigned rank (column four) was used as the secondary key to re-order documents in the run. We used these re-sorted runs where documents are definitely in order of decreasing similarity score calculated by systems, rather than original run files submitted by system developers. The results shown in Table 4.1 are all based on these re-sorted runs.

4.3 Behavior of RBP

Given that both RBP scores and residuals are functions of ϕ, we plot general patterns of RBP scores (blue) and RBP residuals (red) over all systems and all topics of TREC-7, TREC-8 and TREC-13 in Figure 4.1. Each box element in Figure 4.1 represents a set of RBP scores or residuals over all system-topic runs for the x-axis value of ϕ. As the trend curves shows in Figure 4.1, when ϕ is small (that is the evaluation is shallow, only a few top ranked documents are examined on average), the RBP residuals are also small because most top ranked documents are pooled and assessed by judges. The measured RBP scores are typically quite high because almost all systems are able to retrieve and rank relevant documents to the top positions in the ranking.

However when the value of ϕ increases, the RBP residuals grow because the expected evaluation depth becomes deeper so more documents in the ranking are checked, and hence documents which are not in the pool and do not have relevance judgments receive non-negligible weights. The uncertainty in the measurements increases with the expected

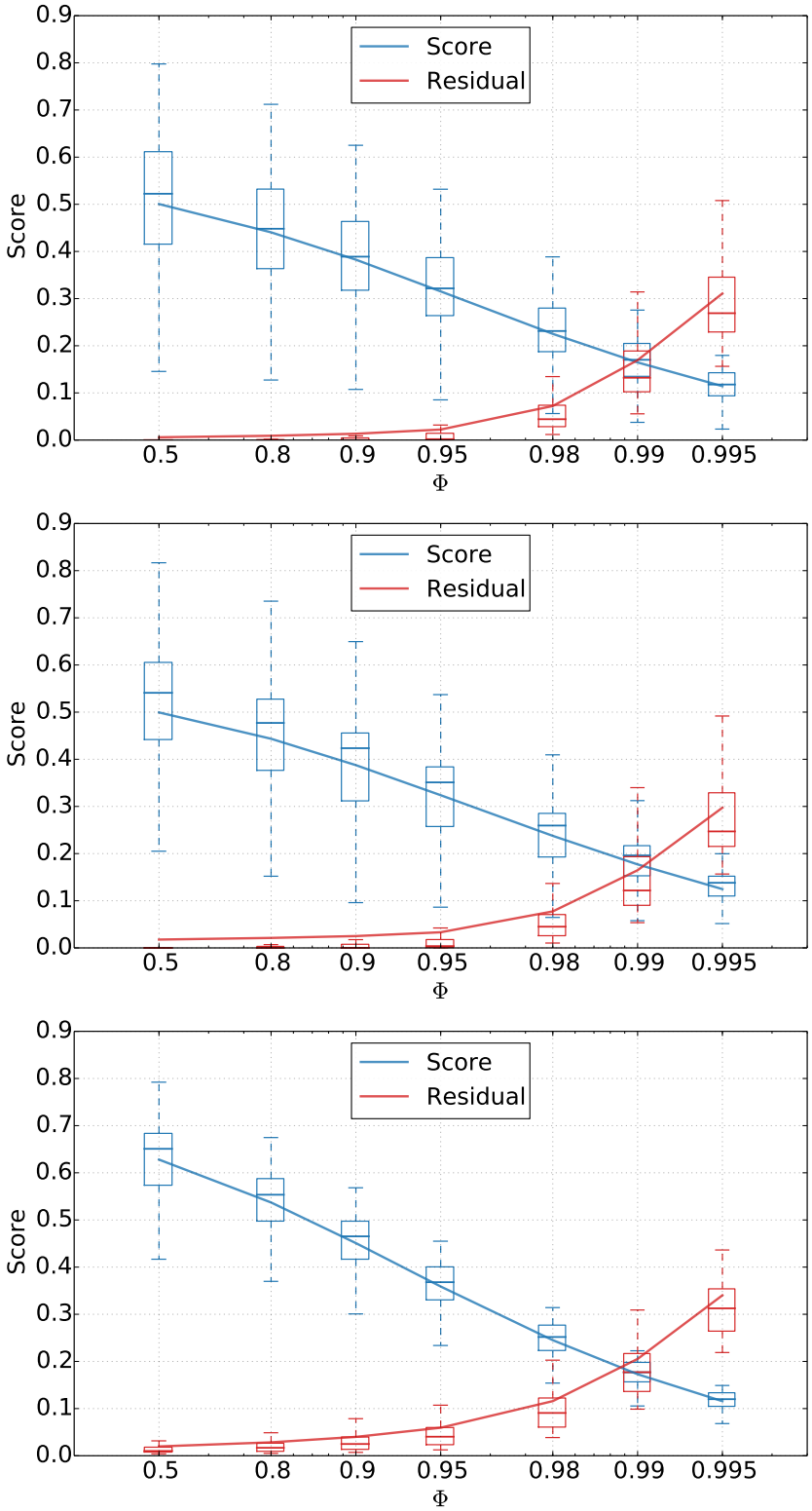


Figure 4.1: TREC-7 (top), TREC-8 (middle) and TREC-13 (bottom), the RBP scores and residuals over all system-topic runs for different values of ϕ which are determined by the expected evaluation depths ($d' = 1/(1-\phi)$) of $2, 5, 10, 20, 50, 100$ and 200.

evaluation depth of RBP, which is positively correlated to the value of ϕ. Meanwhile, the RBP scores decrease with ϕ because systems are not as good at placing relevant documents into lower positions (say 50) as they are for top positions (say 1). And it also because there are more documents whose relevance judgments are unavailable when the evaluation depth becomes deeper, so their relevance gains may not add to the final RBP score.

As Figure 4.1 shows, in the TREC-7, TREC-8 and TREC-13 collections, the RBP residual exceeds the RBP score once the value of ϕ is about 0.99 and the expected evaluation depth is 100, which is, of course, the pooling depth.

The RBP residual is an optimistic estimate of the upper bound that the RBP score could increase if all unjudged documents were relevant, and should not be thought of as a "confidence interval". With a more measured estimate, where if all the unjudged documents were pooled and judged, there would be roughly 5%–6% of them being judged as relevant, the "true" RBP score would therefore increase 2.5%–3% when $\phi = 0.99$. Lu, Moffat, and Culpepper [63] propose mechanisms of estimating the relevance for unjudged documents according to their ranks and relevance scores of judged documents in the same runs.

4.4 Estimating RBP ϕ for Other Metrics

If the RBP parameter ϕ (or the expected evaluation depth) and the pooling depth are both known, the RBP residual of a run can be bounded above, which can be extended for measuring the uncertainty of the effectiveness score given by another metric $\boldsymbol{M}$ if a particular value of ϕ, saying $\phi_{\boldsymbol{M}}$, can be found that makes RBP($\phi_{\boldsymbol{M}}$) generate similar system orderings to metric $\boldsymbol{M}$.

The graphs in Figure 4.2 plot the Kendall's τ of comparing system orderings evaluated by a set of widely used metrics, and the orderings given by RBP across a range of the parameter ϕ. The x-axis value of each point in the graph is the average RBP residual over systems and topics, computed using the value of ϕ whose corresponding expected evaluation depth is reflected on the top horizontal axis. The chosen values of ϕ correspond to the expected depths of $2, 5, 10, 20, 50, 100$ respectively, the same set of ϕ values as those used for Figure 4.1. For each metric $\boldsymbol{M}$ compared with RBP, represented by a colored curve in the graph, the peak point with the highest Kendall's τ in the curve indicates the ϕ value that makes the system ordering given by RBP closest to that generated by the reference metric $\boldsymbol{M}$. For example, RBP is most similar to Prec(10) when ϕ is around 0.9, where the expected evaluation depth is about 10. As the expected evaluation depths of other four recall-based metric Recall(1000), R-Prec, AP(1000) and NDCG(1000) are all deeper, the values of ϕ are higher when their correlation with RBP are maximized, and so their corresponding RBP residuals are greater.

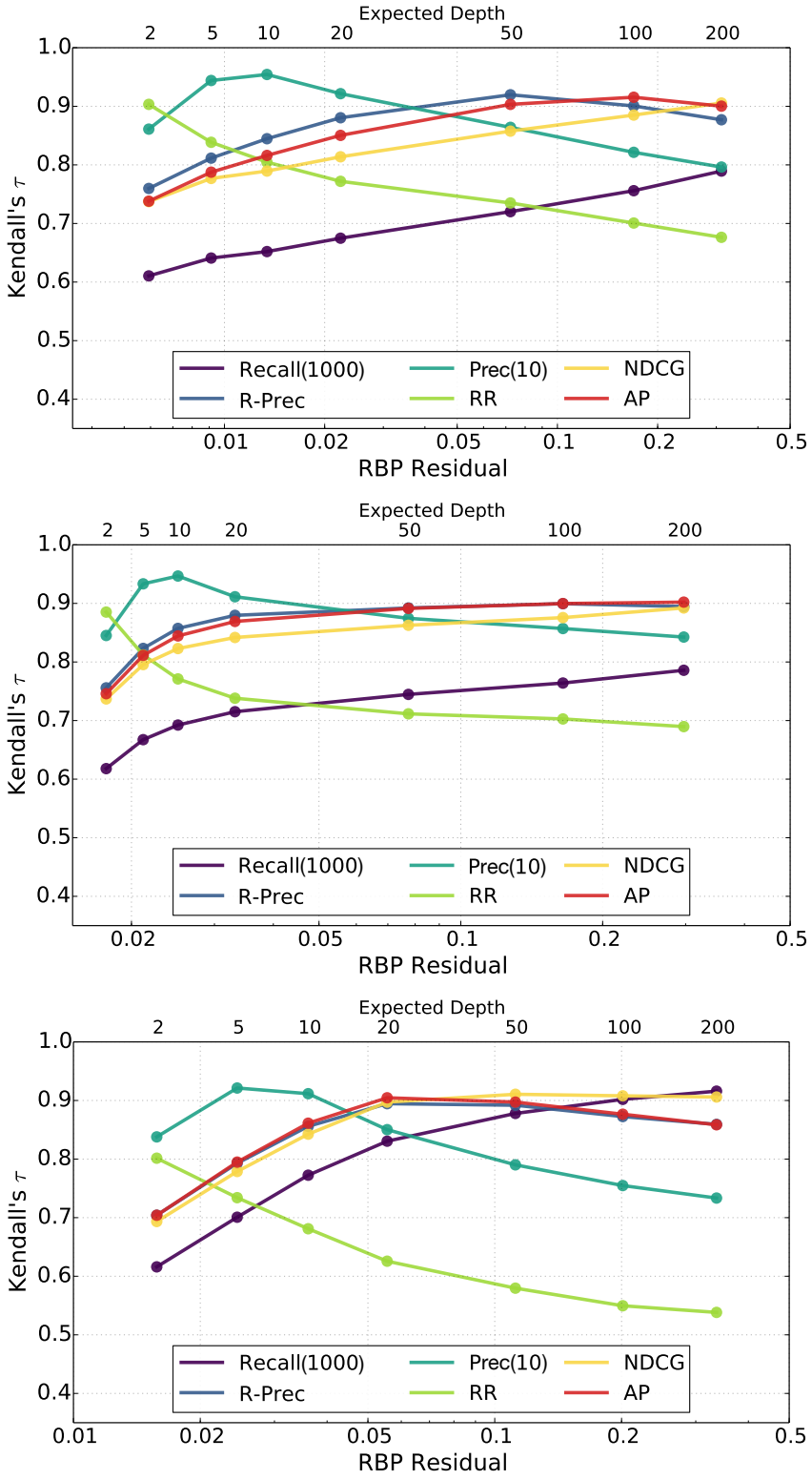


Figure 4.2: TREC-7 (top), TREC-8 (middle) and TREC-13 (bottom), Kendall's τ between system orderings induced by RBP (with a set of ϕ parameters, the corresponding RBP residuals are plotted as the bottom x-axis values) and six other metrics, indicated by the colored lines. The expected evaluation depth corresponding to the ϕ value of each point is shown on the top x-axis.

As expected, since RR and Prec(10) are shallow metrics, they are highly correlated to RBP when ϕ is small, corresponding to shallow expected evaluation depth and low residuals. Figure 4.2 shows that deep metrics such as AP, NDCG and R-Prec commonly correspond to ϕ values of 0.98 or more (expected depth greater than 50) with RBP residuals of 0.1 or more.

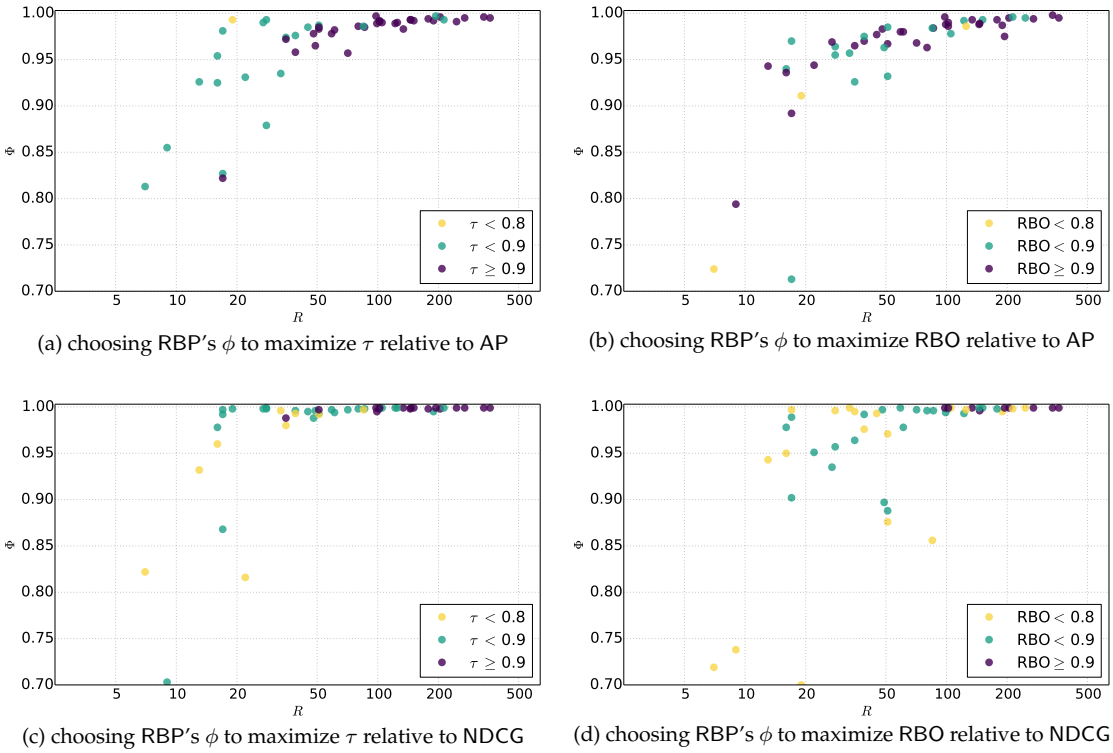
Fairthorne [30] explored the extent to which different human relevance judgments may affect system orderings. The Kendall's τ correlations between TREC-6 system rankings using different combination of judgments assessed by TREC experts and university students are in the range from 0.87 to 0.95, which can help to conclude that the correlations observed in our experiments are in high degree. Any remaining discrepancies are likely to be less than what might be observed when considering variations in relevance judgments given by human assessors.

4.5 RBP ϕ Variations Related to R for Each Topic

As Figure 4.2 illustrates, the systems rankings generated by RBP change as ϕ is varied. When comparing different metrics with RBP using system orderings based on the average of all per-topic run scores, the parameter ϕ can be tuned to maximize the correlation of RBP and the reference metric, and so the expected evaluation depth can therefore be inferred. In Figure 4.2, the system rankings used to compute the correlation between RBP and the reference metric were based on their average performances over all the topics. However, each topic $\boldsymbol{t}$ may have different number of known relevant documents (previously denoted as $\boldsymbol{R_t}$ in Chapter 2), which could lead variations of system orderings even using the same metric. Figure 4.3 and 4.4 explore the value of ϕ, on a per-topic basis, again maximizing the correlation of system rankings given by RBP and the reference metric.

In each of the scatter plots shown in Figure 4.3 and 4.4, a recall-based reference metric, AP or NDCG, is selected. Each colored dot is for a single topic, whose x-value is the total number of relevant documents for this topic in the TREC corresponding binary judgments, denoted as $\boldsymbol{R}$. The y-value of each dot is the discovered ϕ value, searched from 0.500 to 0.999 in 0.001 increments, which makes the RBP most like the reference metric for this topic (that is, the correlation of system orderings for this topic given by the reference metric and RBP with the found ϕ is highest). The color of each dot represents the maximized correlation score strength, measured using different correlation coefficients, Kendall's τ (in the panes in left column) and RBO (in the panes in right column). In Figure 4.3, the four panes cover two reference metrics, AP and NDCG, two correlation coefficients. Figure 4.4 is the same but for the test collection of TREC-13.

As the patterns shown in panes (a) and (b) with reference metric AP in the top row of Figure 4.3 and 4.4, when the total number of relevant documents for a topic, $\boldsymbol{R}$, is small, the value of ϕ discovered for the most AP-like RBP is also relatively small. On the



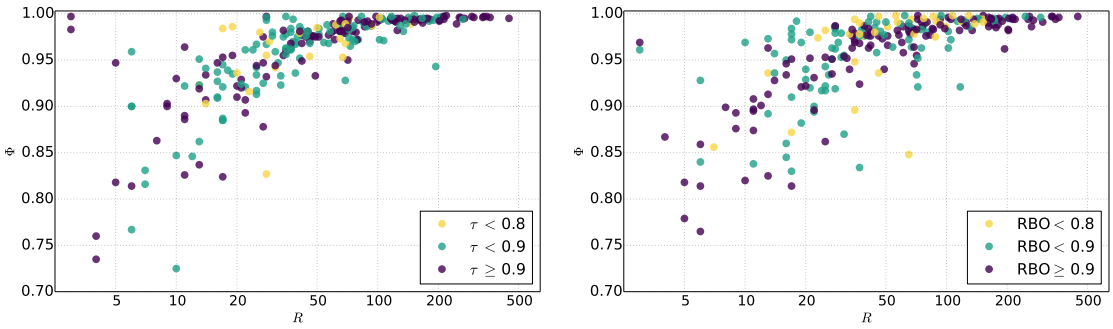
(a) choosing RBP's ϕ to maximize τ relative to AP

(b) choosing RBP's ϕ to maximize RBO relative to AP

(c) choosing RBP's ϕ to maximize τ relative to NDCG

(d) choosing RBP's ϕ to maximize RBO relative to NDCG

Figure 4.3: TREC-7, the relationship between the number of known relevant documents (R, shown as x-axis) for each topic and the value of ϕ which maximizes the Kendall's τ (left two panes), and maximizes RBO (described in Chapter 2, right two panes) correlation coefficients between per-topic system rankings given by RBP and two recall-based metrics respectively. In the first row, the reference metric is AP; in the second row, it is NDCG. There are 50 points (topics) plotted in each of four panes. The color scale represents the maximized correlation coefficient for that topic.



(a) choosing RBP's ϕ to maximize τ relative to AP

(b) choosing RBP's ϕ to maximize RBO relative to AP

(c) choosing RBP's ϕ to maximize τ relative to NDCG

(d) choosing RBP's ϕ to maximize RBO relative to NDCG

Figure 4.4: Same as Figure 4.3, except that the dataset is TREC-13. There are 249 points (topics) in each of four panes.

Metric	Correlation	Count of cases			Kendall's τ	p	Average ϕ
		< 0.8	< 0.9	≥ 0.9			
AP	Kendall's τ	23	110	116	0.659	< 0.0001	0.958
AP	RBO	36	90	123	0.595	< 0.0001	0.953
NDCG	Kendall's τ	58	131	60	0.509	< 0.0001	0.956
NDCG	RBO	99	95	55	0.459	< 0.0001	0.943

Table 4.2: TREC-13 Robust, strength of correlations, measured by Kendall's τ and RBO respectively, between two reference metrics (AP and NDCG) and RBP with best values of ϕ which maximize per-topic system orderings evaluated by RBP and the reference metrics. The last column shows the average of obtained values of ϕ over all 249 topics in TREC-13. another.

other hand when $\boldsymbol{R}$ is large for a topic, the RBP parameter ϕ for generating the greatest correlated system ranking with which ordered by AP is large as well. These outcome patterns are similar when the reference metric is changed to NDCG, shown in panes (c) and (d). The Kendall's τ correlation coefficient together with the significance p-value for the points in each pane in Figure 4.4 are summarized in Table 4.2. The counts of dots (topics) in each color and the average of ϕ across all 249 topics for each of the four graphs are also listed in the table.

The purpose of this section is not determining the value ϕ for RBP for each topic based on the pre-knowledge of $\boldsymbol{R}$, because it is impossible for users to be aware of $\boldsymbol{R}$ before examining the ranked list, and hence $\boldsymbol{R}$ should not in anyway influence their behaviors. One of primary factors that influence their behavior during the examining is the total amount of relevance that they have gained from the ranked lists [71, 74]. Rather this experiment is intended to show that if we want to find the best value of ϕ for RBP whose behaviors closely match the recall-based ones so the bounds of residual-like uncertainty for recall-based metrics can be well estimated, then knowledge of $\boldsymbol{R}$ is required.

4.6 Reducing Qrels to Add Uncertainty

If only a subset of the pooled judgments is available for the evaluation, the residual of the system effectiveness scores will increase. In our next experiment, we explored the relationship between residuals caused by reduced incomplete relevance judgments and the ability of evaluation metrics to distinguish systems, which was quantified by p-values generated by the statistical test.

To ensure that each system received equal disadvantages from the reduced judgments, we artificially pooled the deeply-judged systems to depths of $d' \in \{5, 10, 15, 20, 25\}$ respectively and used the judgments pooled to $d' = 50$ as a reference baseline.

Dimension	Collection		
	TREC-7	TREC-8	TREC-13
Topics	50	50	49
Systems	65	67	52
Documents, judged	33870	40238	17509
Documents, relevant	3121	3175	1576
Single-vote documents, judged	15943	24149	5597
Single-vote documents, relevant	639	731	147

Table 4.3: Reduced qrels files for three collections when a pooling depth of $d' = 50$ is employed, contributed by the set of deeply-judged runs described in Table 4.1. The third row shows the total number of pooled documents across all topics, followed by the number of relevant documents of those in the fourth row. The last two rows provide the number of documents which were nominated (ranked to top-50) by only one deeply-judged systems and of those, how many were judged as relevant.

For instance, in TREC-7 there were 65 deeply-judged systems and 50 topics. When pooling depth $d' = 5$, only the judgments (extracted from NIST qrels file) of top-5 documents retrieved by the deeply-judged systems (at most $65 \times 50 \times 5$) were filtered into a reduced qrels file. For each of the six cases of d', we repeated these steps that select judgments of top-d' documents ranked by deeply-judged systems only from the original NIST qrels file. Thus the result was a set of qrels files in which all pooled system had equal opportunity to provide documents and contribute to the new reduced pool.

Table 4.3 summarizes the information about reduced qrels files when $d' = 50$. For example, in TREC-7 there are 33,870 distinct documents pooled when $d' = 50$, resulting from 50×65 topic-system combinations, and filtered into the reduced qrels; 3121 of those were previously judged as relevant by NIST binary; 15,943 of those are ranked to top-50 by one sole deeply-judged system (named as single-voted document) and 639 single-voted documents are judged as relevant in TREC-7. The same rules are applied to generate sets of reduced qrels for TREC-8 and TREC-13 as well. As there is not relevant document for topic 672 in TREC-13, topic 672 is not included in tested topic set for this experiment. Since TREC-13 directly re-used judgments from previous years, there are fewer systems deemed as deeply-judged for this test collection, which results in a smaller number of pooled and judged documents. Note that all data in Table 4.3 (and also Table 4.4) is based on extracting subsets of the official qrels files for those TREC rounds, available from the NIST TREC web site.

Table 4.4 provides decomposed information about reduced qrels files with $d' = 20$ (top table) and $d' = 50$ (bottom table). Each table shows the number of different systems which nominated each document in to the top-d' (the multiplicity) and, the proportion of the observed relevance. The patterns of all three test collections confirm that documents

Multiplicity	TREC-7 Ad-Hoc		TREC-8 Ad-Hoc		TREC-13 Robust	
	Count	% Rel.	Count	% Rel.	Count	% Rel.
1	6988	7.8	10266	5.3	2329	4.7
2	2791	9.5	2538	11.7	1147	8.0
3–4	1651	17.4	1548	16.3	1132	9.2
5–8	1178	23.1	1017	26.6	1010	14.2
9–16	780	29.5	732	34.0	654	25.5
17–32	567	44.3	544	43.6	552	38.9
33+	375	65.1	416	61.3	369	61.8
Total	14330	14.6	17061	12.3	7193	14.7

(a) $d' = 20$

Multiplicity	TREC-7 Ad-Hoc		TREC-8 Ad-Hoc		TREC-13 Robust	
	Count	% Rel.	Count	% Rel.	Count	% Rel.
1	10905	5.6	22685	3.1	5610	2.6
2	3860	9.3	5510	6.2	2628	3.7
3–4	2978	13.7	3587	10.7	2922	5.1
5–8	2315	16.3	2406	15.5	2466	7.7
9–16	1770	21.7	1724	20.6	1612	13.1
17–32	1266	32.0	1317	30.9	1278	22.9
33+	809	54.5	1123	49.6	1001	48.8
Total	23903	12.5	38352	8.1	17517	9.0

(b) $d' = 50$

Table 4.4: Reduced qrels files with $d' = 20$ (upper table) and $d' = 50$ (bottom table), showing the count of pooled documents nominated by different number of systems (document's multiplicity), and of those, the proportion that were judged relevant by the NIST assessment process. All numbers are totals over all topics. The TREC-13 columns refer to topics 651–700 only.

nominated by more systems have a higher chance to be judged as relevant by NIST assessors. When $d' = 20$, if documents are solely voted by one of the systems, the conditional probability that they are judged as relevant is only around 5%. However for documents voted by more than 33 systems, the conditional probability increases to around 60%. According to similar data for reduced qrels files with other values of d', we can conclude that the smaller the d' value is, the higher the conditional probabilities of being judged as relevant the documents have.

Figure 4.5 describes the ranges of pool sizes and relevant documents counts over all topics at each of the $d' \in \{5, 10, 15, 20, 25\}$. Each purple whisker element together with a red whisker represent the distributions of the document size and the number of documents judged as relevant in the reduced pool across all the topics at each pooling depth d'. The patterns in all test collections show that both numbers of documents pooled and documents judged as relevant increase with d' overall, though the size of relevant documents increases in slower rate which can be used to estimate the total number of relevant document, $\boldsymbol{R}$ [120].

Figure 4.6 compares how effectiveness scores evaluated by different metrics are affected by the pooling depth ($d' \in \{5, 10, 15, 20, 25\}$) of judgments. The experiment is performed with three test collections, TREC-7, TREC-8 and TREC-13, and each pane in Figure 4.6 is for one collection. Each box/whisker element in the pane shows the distribution of metric score differences when using reduced judgments with $d' \in \{5, 10, 15, 20, 25\}$ compared to the scores that arise when judgments created with $d' = 50$ are used. Each box/whisker element is plotted over effectiveness scores of all system-topic combinations evaluated by the given metric.

As the patterns in Figure 4.6 illustrate, when d' increases (and so more judgments are added to the reduced pool, more relevant documents are encountered in runs), effectiveness scores given by RBP (as well as other weighted-precision metrics) are non-decreasing, since more relevance is found and added to the metric score. But AP and NDCG (and also other recall-based metrics) scores generally decline when the d' increases, that is when the pool is extended. This is because recall-based metrics are normalized by (or related to) $\boldsymbol{R}$, the total number of identified relevant documents for the topic, which normally increases with the pool size. As $\boldsymbol{R}$ is the denominator factor in calculations of recall-based metrics, even more relevant documents are discovered in the extended pool, and so the total relevance gained (or for AP, the sum of the Prec scores) increases, but the growth they bring to the final metric score is not strong enough to overcome the denominator $\boldsymbol{R}$.

Table 4.5 shows how the change of d' affects the ability of metrics to distinguish systems in pairs. For each test collection, the set of deeply-judged systems were selected and then paired with each other (systems submitted by the same research groups or organizations were not paired). For two "compared" systems in each pair, their (paired) run effectiveness scores evaluated by one of three tested metrics(AP, NDCG and RBP with

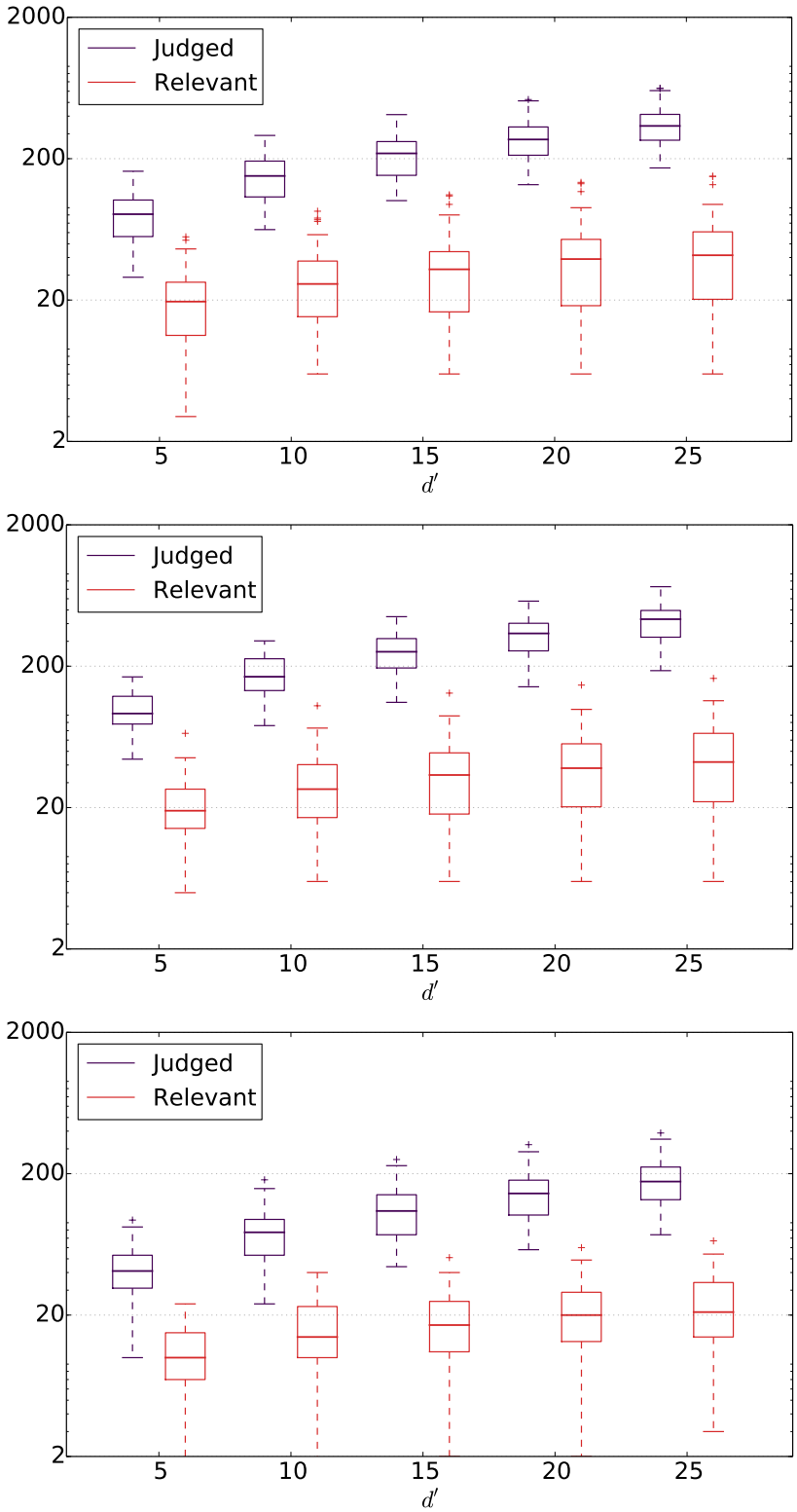


Figure 4.5: TREC-7 (top), TREC-8 (middle) and TREC-13 Robust (bottom, topics 651–671, 673–700), the number of documents and the number of relevant documents in the reduced pools averaged over all topics, for depths $d' = 5, 10, 15, 20$ and 25.

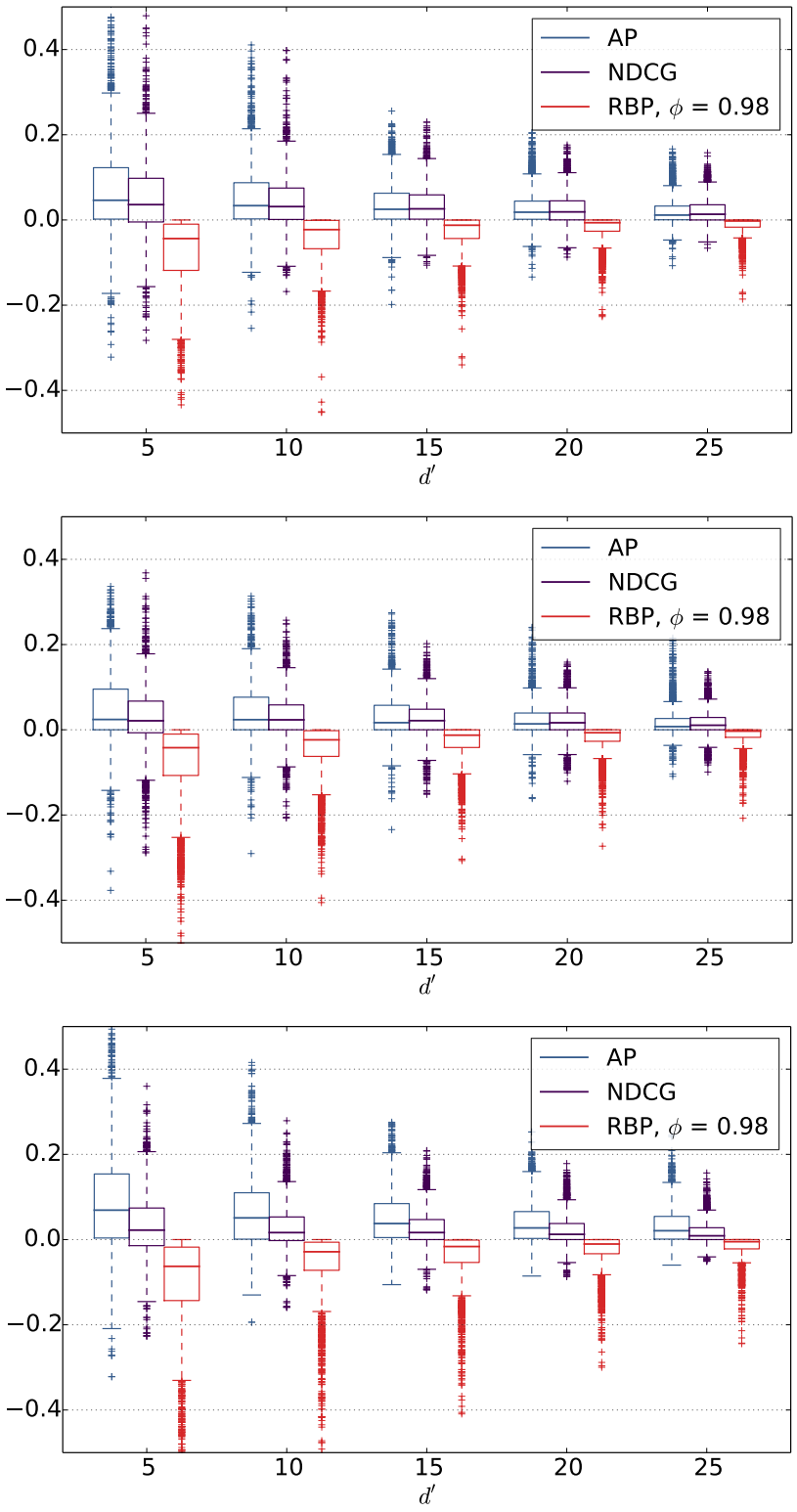


Figure 4.6: TREC-7 (top), TREC-8 (middle) and TREC-13 Robust, system-topic score differences when scored using AP, NDCG, and RBP ($\phi = 0.98$), using reduced judgments for each of pooling depths $d' \in \{5, 10, 15, 20, 25\}$ and a reference judgment set created using $d' = 50$. Only the deeply-judged systems are used.

Collection	Metric	d'					
		5	10	15	20	25	50
TREC-7	AP	68.0	70.9	71.7	73.2	73.3	73.0
	NDCG	71.2	73.5	74.1	74.4	74.8	75.0
	RBP, $\phi = 0.98$	68.4	69.7	70.6	70.6	70.4	69.5
TREC-8	AP	71.1	72.6	73.0	73.6	73.5	73.8
	NDCG	68.7	70.2	70.2	70.8	71.3	71.6
	RBP, $\phi = 0.98$	69.6	71.2	71.7	72.1	72.2	72.5
TREC-13	AP	63.4	63.1	63.7	64.5	66.1	66.7
	NDCG	58.1	59.3	61.1	60.9	62.1	64.7
	RBP, $\phi = 0.98$	57.0	55.5	56.0	54.3	56.0	56.9

Table 4.5: Discrimination ratios as a function of d': the percentage of deeply-judged system pairs in which systems evaluated by a particular metric, using reduced qrels files with $d' \in \{5, 10, 15, 20, 25, 50\}$, are deemed as significantly different (p-value is less than $\alpha = 0.05$) by the paired two-tailed t-test.

$\phi = 0.98$) over all the topics, using the reduced judgments with $d' \in \{5, 10, 15, 20, 25, 50\}$, were treated as inputs of a student t-test which generated a p-value to indicate the discrimination ratio of these two paired systems. The fractions of these computed p-values less than the significance level $\alpha = 0.05$ for all system pairs for different value of d' and for distinct metrics are shown in Table 4.5.

In Table 4.5, for each collection, the discrimination ratios of two recall-based metrics AP and NDCG are slightly greater than those of RBP with $\phi = 0.98$, even though they have similar expected evaluation depth. The discrimination ratios in each row of the table do not always increase with pooling depth d', but mostly grow a little bit when the d' increases from 5 to 25. The further growth of d' from 25 to 50 does not cause many changes of discrimination ratios. So at face value at least, we can conclude that it is not necessary to further increase the pooling depth from 25 and 50 because the statistical confidence of system comparisons outcome in t-test is not greatly affected by the added judgments from deeper pool with d' larger than 25.

4.7 Consistent Discrimination

The conclusion made in Section 4.6 is correct only if it is the same set of system pairs being found to be significantly different. So in this section, we perform another experiment to explore if the separable pairs identified by the $d = 50$ judgments are the same pairs found by reduced judgments.

One of key purposes of IR evaluation is to distinguish systems in pairs and conclude whether one system is more effective than the other. In the next experiment, we perform

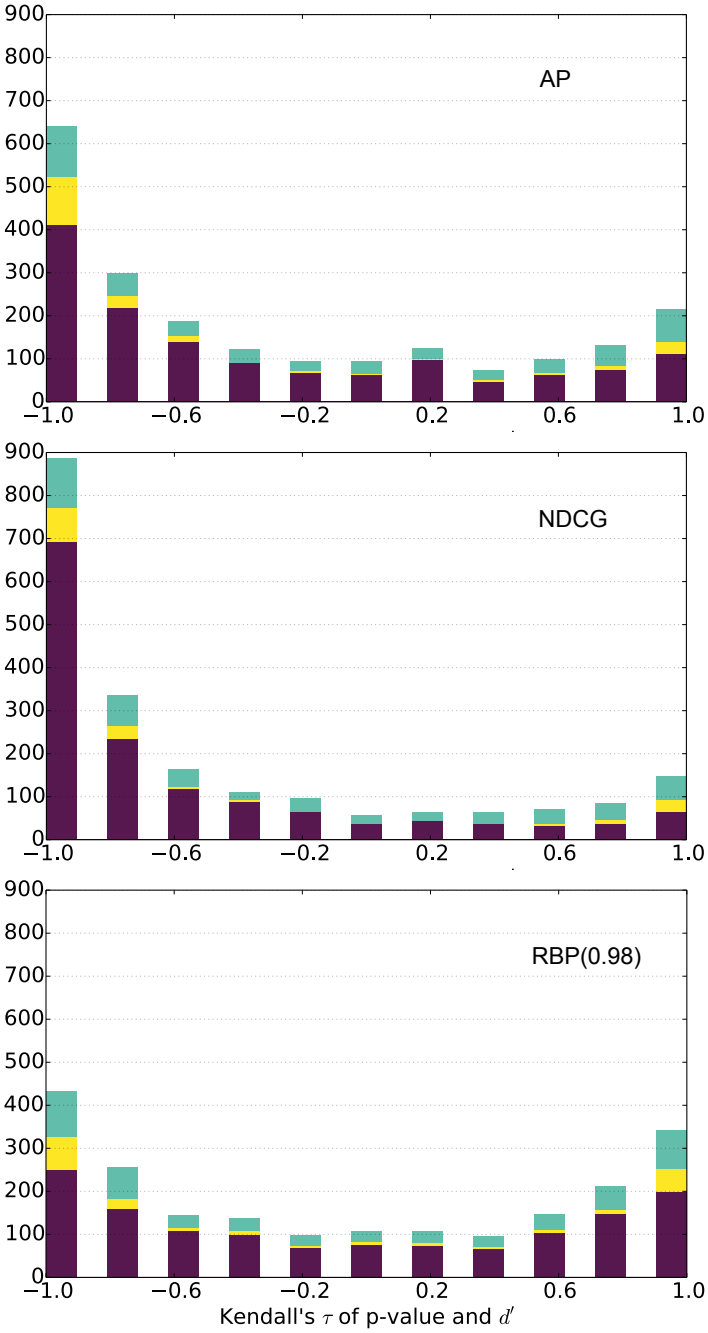


Figure 4.7: Kendall's τ scores, showing the relationship between the pooling depths $d' \in \{5, 10, 15, 20, 25\}$ and the p-values generated by the t-test taking metric scores of the paired systems evaluated via the reduced judgments with d'. Each pane is plotted for $65 \times 64/2 = 2080$ Kendall's τ scores calculated across 2080 system pair comparisons for 65 deeply-judged TREC-7 systems. The evaluation metrics are AP (top), NDCG (middle) and RBP with $\phi = 0.98$ (bottom). In each bar, the green section counts system pairs whose five p-values are all greater than $\alpha = 0.05$; the yellow section shows the number of system pairs for which the generated five p-values straddle $\alpha = 0.05$; the purple section indicates the count of system pairs for which the five values are all less than $\alpha = 0.05$.

statistical tests for each system pair, which take the metric scores of the paired systems evaluated using the reference (the set of judgments created using $d = 50$) and the reduced judgments with varied pooling depth $d' \in \{5, 10, 15, 20, 25\}$.

For each system-versus-system pair, we employ Kendall's τ to explore the relationship between the generated p-value using the reduced judgments, and the corresponding pooling depth d'. That is, for each system pair, the Kendall's τ takes two lists of five-point sequences:

- the values of d': $5, 10, 15, 20, 25$;
- the p-values of five t-tests taking the metric scores (computed using the reduced judgments with $d' \in \{5, 10, 15, 20, 25\}$ respectively) of the paired systems.

We compute Kendall's τ scores for 2080 system pairs (covering 65 deeply-judged TREC-7 systems), and assume that other factors except for pooling depth are held constant during this analysis. For each 5-element sequence, the obtained Kendall's τ values can be $-1.0, -0.8, -0.6, \ldots, 0.8, 1.0$. There are (only) $5! = 120$ possible permutations involved, one of which yields $\tau = 1.0$ (and another one with the opposite ordering in one sequence yields $\tau = -1.0$); four of which yield $\tau = 0.8$ (similarly, another four yield $\tau = -0.8$); nine of which give a τ value of 0.6; fifteen give $\tau = 0.4$; twenty give $\tau = 0.2$; and 22 give a value of zero. We determine trends of the p-value (the smaller the p-value is, the more confidence we have the system pair can be distinguished by the metric) when the pool size increases.

For example, in the system pair `CLARIT98CLUS` and `INQ501`, each system has five sets of 50 AP scores, evaluated using the reduced judgments with $d' \in \{5, 10, 15, 20, 25\}$ respectively, for 50 given topics. A t-test taking the AP score sets of the paired systems is performed for each of five d'. The generated five p-values are: 0.0836, 0.0128, 0.0076, 0.0070 and 0.0067, which indicate the confidence of distinguishing these two systems when evaluate using reduced judgments with pool depth $5, 10, 15, 20, 25$ respectively. The Kendall's τ of the p-values of the system pair `CLARIT98CLUS` and `INQ501`, and the pool depth sequence $[5, 10, 15, 20, 25]$ is -1.0. For another pair `bbn1` and `mds98t`, the Kendall's τ takes the generated p-values $[0.0044, 0.0080, 0.0071, 0.0083, 0.0095]$ and pool depths $[5, 10, 15, 20, 25]$ as the input, and compute their correlation coefficient as 0.8.

Figure 4.7 illustrates the distribution of Kendall's τ values, computed using the approach described above for TREC-7 (results for TREC-8 and TREC-13 are similar), for AP, NDCG and RBP with $\phi = 0.98$ respectively. For the two recall-based metrics, the distribution of τ values are unimodal, with the peak at -1.0. The distribution of RBP is bimodal, with the maximum values at -1.0 and 1.0. A Kendall's τ value of -1.0 indicates that the discrimination of the tested metric increases with the pooling depth d', that is when the size of the relevance judgment sets grows. The bar of $\tau = -1.0$ covers around 30% system pairs for AP, 45% pairs for NDCG, and 20% pairs for RBP with $\phi = 0.98$. The system pairs counted by this bar have a strict pattern of p-values decreasing as d' grows, which

is regarded as a plausible outcome – it makes sense that increasing the pool size leads to greater confidence in distinguishing systems. Note that, if the p-values are random, the computed τ values will be in normal distribution, and the two extreme bars ($\tau = -1.0$ and $\tau = 1.0$) would be expected to cover less than 1% of system pairs. For RBP, there are around 17% system pairs covered by the bar of $\tau = 1.0$, which suggests the exact reverse – more judgments being involved results in reducing confidence of separating systems. We perform the same analysis for RBP with other values of ϕ (results are not shown), and obtained a similar pattern of performance. It is probably because when the pool size increases, and more judgments are involved, the RBP scores never decrease, but increase once new relevant documents (which are unjudged) are found. Although it reduces the uncertainty of RBP scores, adding judgments may bring ambiguous outcomes when separating systems.

Of particular interest in Table 4.5 and Figure 4.7 is to determine whether choosing different d' will lead to conflict system comparison outcomes in regard to statistical significance. In the graph of RBP, the bars of $\tau = 1.0$ and $\tau = -1.0$ have approximately the same fraction of pairs which are "statistically different" (the purple session of each bar). But for AP and NDCG, the number of pairs with all the p-values less than 0.05 is largest in the bar of $\tau = -1.0$, and the proportion of these pairs in bars of negative τ scores is generally greater than which in bars for positive τ values. That is, for AP and NDCG, the confidence of pairs which can be distinguished by judgments with all pool depths (that is, whose comparison outcome is not alerted by d') usually increases when the pool is deepened.

For each system pair, there are five p-values generated using five sets of judgments with $d' \in \{5, 10, 15, 20, 25\}$ respectively. If the p-value *straddles* the fixed significance level $\alpha = 0.05$ (that is, the minimum p-value is less than α, and the maximum p-value is greater than α), it implies that one choice of pool depth might conclude that the compared two systems are significantly different, but another choice of d' could lead to an opposite conclusion. The system pairs in which the p-value straddles $\alpha = 0.05$ are counted, and plotted as yellow segment in each of the bars in Figure 4.7. As can be seen, straddling pairs occur primarily at the two extremes ($\tau = -1.0$ and $\tau = 1.0$). In the graph of AP, 17.2% of pairs in the bar of $\tau = -1.0$ straddle the $\alpha = 0.05$, and the proportion of straddling pairs with $\tau = 1.0$ is 13.0%. The occurrence frequency of pairs straddling the significance level in the bar of other τ values is under 10%. Overall, the larger the absolute value of τ, the greater the proportion of straddling pairs. The results are also similar for other metrics and test collections.

4.8 Consistently Consistent Discrimination

In the experiment above, by comparing the generated p-values, whether the discrimination of the metric (which scored the systems) is affected by the pooling depth could

therefore be found. For each system pair, if the p-values generated using the reference and the reduced judgments are both smaller, or both greater, than $\alpha = 0.05$, the metric discrimination for this pair will be deemed as consistent, otherwise affected. In our next experiment, we focus on the "straddling" pairs found in the previous experiment, and explore how the d' affect the discrimination consistency of metrics.

Table 4.6 shows the significance results measured by two-tailed paired t-test ($\alpha = 0.05$) for 2080 system pairs generated by 65 deeply-judged systems in TREC-7 collection. For each shallow pooling depth $d' \in \{5, 20, 15, 20, 25\}$ and each metric, the 2080 system pairs are categorized into four classes:

- TP (true positive): evaluations using shallow d' judgments and $d = 50$ judgments both indicate statistical significance;
- FP (false positive): the evaluation using shallow d' judgments indicates statistical significance, but evaluation using $d = 50$ judgments does not;
- FN (false negative): the evaluation using $d = 50$ judgments indicates statistical significance, but evaluation using shallow d' judgments does not;
- TN (true negative): neither evaluations using shallow d' judgments nor evaluation using $d = 50$ judgments indicates statistical significance.

For all system pairs categorized into TP, we also checked that if both evaluations using shallow and deeper judgments respectively favor the same system in the pair. No such "contradictory" situation was found for TP system pairs in TREC-7 and TREC-13 but there were three pairs in TREC-8 that both $d' = 5$ and $d = 50$ evaluations using RBP indicate significance but preferring opposite systems in the pair. These three pairs were therefore counted as FP instead of TP.

The discrimination ratio, shown already in Table 4.5, can be calculated as the proportion of system pairs receiving concordant conclusions of separability from the t-tests using the d' judgments and $d = 50$ judgments respectively: (TP+FP)/(TP+FP+FN+TN). As Table 4.6 states, for all metrics, numbers of system pairs in TP are the largest and much greater than numbers of pairs in other three categories. Another important column that affect the discrimination ratio computation is the FP, in which the paired systems are assessed as significantly different using the shallow d' judgments but would not be if judgments with deeper pooling depth $d = 50$ were used. The FP counts for each metric are the smallest and uniformly decrease when the pooling depth d' grows. The decreasing speed of FP counts, as d' increases, is generally slower than the increasing speed of TP counts so that the obtained discrimination ratio typically increases with the size of judgments (and also d'). However focusing on the declining of FP counts, we noticed that, as the d' increases, system pairs might not only moving out from the FP, but might also shifting to the FP from other categories. For example, for the metric AP, when d' increase from 5 to 10, the dropping of FP counts from 60 to 58 involving 13 system pairs

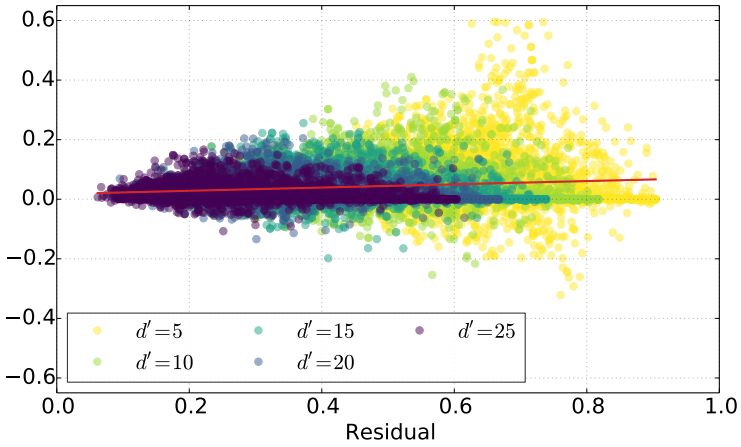
d'	AP				NDCG				RBP, $\phi = 0.98$			
	TP	FP	FN	TN	TP	FP	FN	TN	TP	FP	FN	TN
5	1355	60	164	501	1428	52	133	467	1320	103	125	532
10	1416	58	103	503	1488	40	73	479	1371	78	74	557
15	1452	40	67	521	1510	32	51	487	1395	73	50	562
20	1484	39	35	522	1522	25	39	494	1416	53	29	582
25	1494	30	25	531	1534	21	27	498	1425	39	20	596

Table 4.6: Taking the two-tailed paired t-test (significance level $\alpha = 0.05$) results of comparing all 65 TREC-7 deeply-judged systems in pairs (totally 2080 pairs) using the judgments to depth $d = 50$ as the reference, count the differences when comparing systems using judgments with pooling depth $d' \in \{5, 10, 15, 20, 25\}$. For each metric and each row of d', the evaluated system pair was categorized into: TP, if the assessments using shallow judgments and deeper judgments ($d = 50$) both indicate significance; FP, if the assessment using shallow judgments indicates significant but does not when using deeper judgments; FN, the deeper assessment indicates significant but shallow assessment does not; TN, neither the shallow assessment nor the deeper assessment indicates significant.

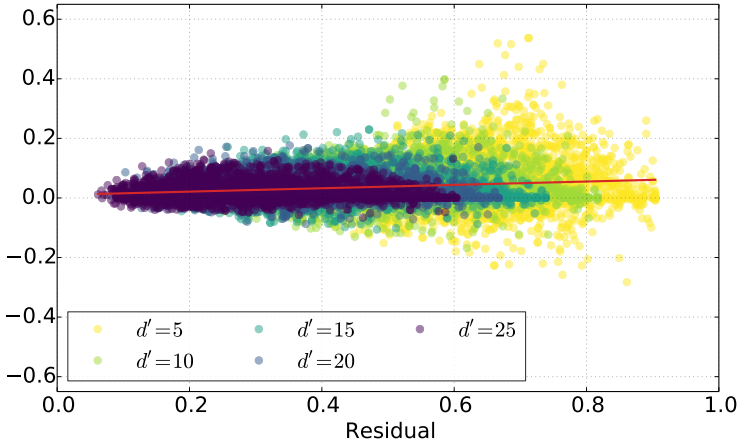
leaving and further 11 system pairs shifting in. In the same column, when d' rises from 20 to 25, there are 12 pairs moving out and 3 pairs joining in the category of FP. This extent of drifting is to be expected given the nature of statistical testing with a small value of significance level $\alpha = 0.05$.

4.9 Relationships between RBP Residuals and Run Scores

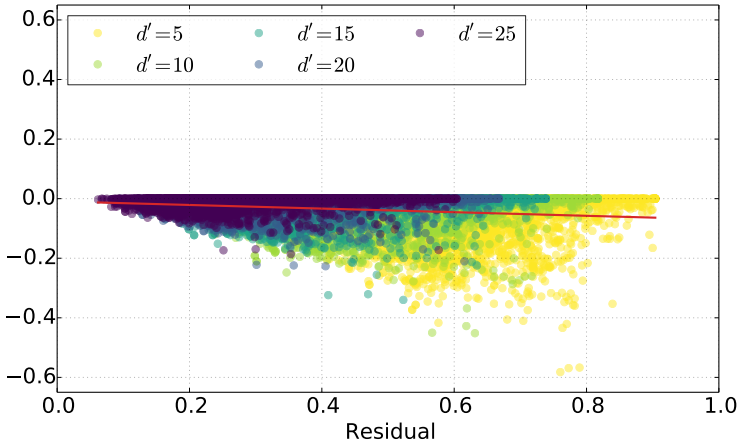
We have observed how run scores evaluated by recall-based metrics are affected by judgments to different pooling depths in previous sections. And we also discovered the shifts of outcomes when comparing systems in pairs when more and more judgments were added (that is, as pooling depth increases). Now we turn back to the research question: for each run, what is the strength between its RBP-based residual and the score movement when the pooling depth for collecting relevance judgments increases? In this experiment, the run scores assessed using judgments to the depth $d = 50$ are regarded as the reference point (that is, assumed to be the outcome closest to the final score which are evaluated with full judgments). For each run, we compute the differences between run scores evaluated by shallow judgments (for each $d' \in \{5, 10, 15, 20, 25\}$) and $d = 50$ judgments respectively. These run score differences are the same as those used for plotting boxes graphs shown in Figure 4.6. But now we plot them as a function of RBP-based residual in Figure 4.8 and Figure 4.9 for TREC-7 and TREC-13 respectively. For each $d' \in \{5, 10, 15, 20, 25\}$ (represented by distinct colors), each run is plotted as a dot, whose



(a) TREC-7, AP score movements, $\tau = 0.04$

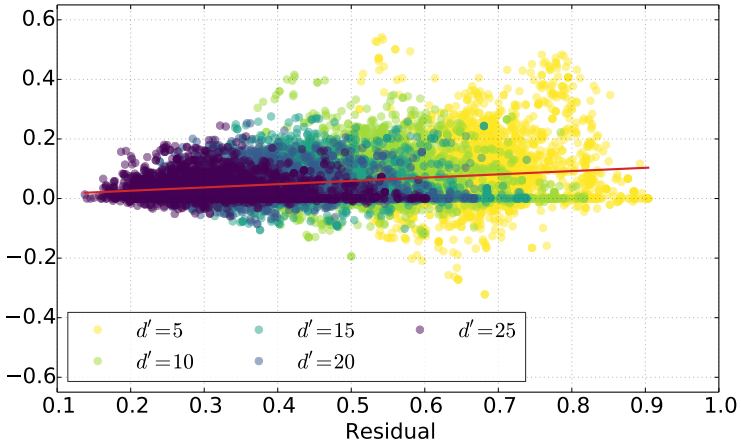


(b) TREC-7, NDCG score movements, $\tau = 0.09$

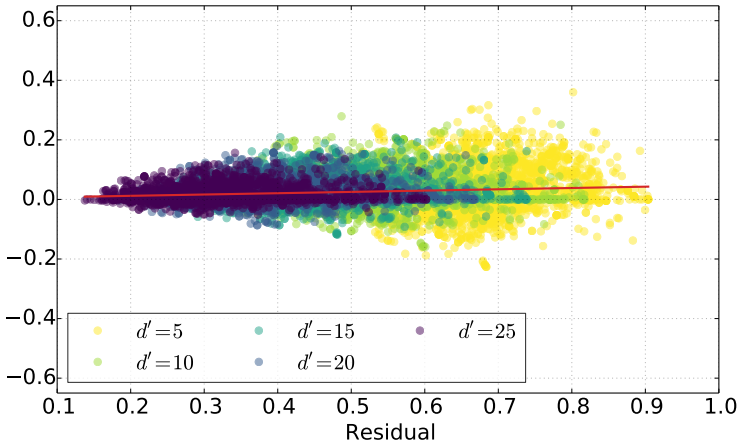


(c) TREC-7, RBP score movements, $\tau = -0.06$

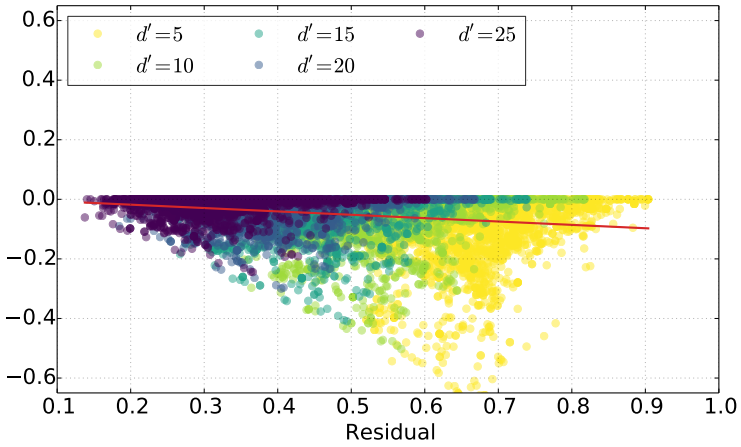
Figure 4.8: The movement of TREC-7 run (per system-topic) scores evaluated by shallow judgments and reference judgments ($d = 50$), as a function of RBP residual ($\phi = 0.98$) for $d' \in \{5, 10, 15, 20, 25\}$ in five colors respectively, for three metrics. The y-axis value of each dot (run) is computed as the run score assessed using shallow d' judgments minus the run score computed using $d = 50$ judgments. Positive values correspond to metric scores that decreases as judgments are added.



(a) TREC-13 Robust, AP score movements, $\tau = 0.09$



(b) TREC-13 Robust, NDCG score movements, $\tau = 0.07$



(c) TREC-13 Robust, RBP score movements, $\tau = -0.12$

Figure 4.9: Same as Figure 4.8, except that the dataset is TREC-13.

RBP residual ($\phi = 0.98$, the pooling depth of judgments is d') is the value on x-axis and the y value is the difference between scores assessed using d' judgments and $d = 50$ judgments. Figure 4.8 and Figure 4.9 show the scatter graphs, together with the Kendall's τ of all of the dots in each pane, for three metrics and two test collections.

In Figure 4.8 and Figure 4.9, all six graphs indicate that both RBP residuals (computed using judgments pooled to d') and run score differences (relative to judgments with pooling depth $d = 50$) generally grow larger when the pooling depth d' becomes shallower. For the two recall-based metrics, AP and NDCG, most run score changes are positive, that is the majority of run scores evaluated by judgments with shallower pooling depths are greater than reference run scores, and only a minority number of run scores increase with the pooling depth. The percentages of runs whose score movements are below, equal, and above zero for each metric, d', and test collection, are summarized in Table 4.7. As can be seen, for AP and NDCG, the number of run scores increasing when the pooling depth increase from $d' < 50$ to $d = 50$ (shown in each < 0 column) is smaller than the number of run scores decreasing (shown in each > 0 column) for any d', but it tends to drop when d' becomes closer to the reference ($d = 50$).

The difference of run scores given by RBP is strictly non-positive, as shown in the RBP row of Table 4.7, as well as the last graph of Figure 4.8 and Figure 4.9. In other words, enlarging the pool and adding judgments to the evaluation generally decreases the run score measured by recall-based metrics (so the Kendall's τ scores of dots in the first two graphs of Figure 4.8 and Figure 4.9 are all positive), and never decrease the run score evaluated by weighted-precision metric RBP (Kendall's τ scores are negative). But as shown in Table 4.7, for all of three metrics, the number of runs whose score movements do not change (shown in $= 0$ column of each metric and d') increases with d', that is, the stability of run scores is greater when d' is closer to $d = 50$. Moreover, the increment of runs with score movement of zero is quick when d' increases from 5 to 10, and from 15 to 25, but slow (even negative for NDCG in TREC-7) when d' rises from 10 to 15. Deepening the pool from 10 to 15 seems have fewer effect on system evaluation than other pool depth range.

The Kendall's τ scores of points in the graphs of Figure 4.8 and Figure 4.9 are summarized in third column of Table 4.8, for two test collections and three metrics. As the small values of Kendall's τ indicate, the correlations between RBP residuals and run score differences for AP, NDCG and RBP are very weak. The possible reasons that the τ values are not large could be that the magnitudes of effectiveness run scores are generally low, and the runs with high RBP residuals are special ones whose top-ranked documents in most cases are not selected by other systems. Referring back to Table 4.4, most pooled documents are labeled by one or two systems and only a small number of those documents are judged as relevant. This indicates that runs with greater residuals may have high probabilities to be scored lower, and so large score changes are unlikely to happen when evaluated using deeper pooled judgments.

Collection	Metric	$d' = 5$			$d' = 10$			$d' = 15$			$d' = 20$			$d' = 25$		
		< 0	$= 0$	> 0	< 0	$= 0$	> 0	< 0	$= 0$	> 0	< 0	$= 0$	> 0	< 0	$= 0$	> 0
TREC-7	AP	20.0	1.1	78.9	13.9	5.0	81.1	12.3	5.0	82.7	11.5	6.9	81.6	11.1	10.9	78.0
	NDCG	27.5	1.1	71.4	19.0	5.2	75.9	18.2	5.1	76.7	17.9	6.9	75.2	18.2	11.0	70.8
	RBP	95.7	4.3	0.0	89.3	10.7	0.0	87.4	12.6	0.0	82.6	17.4	0.0	77.6	22.4	0.0
TREC-13	AP	19.3	2.6	78.1	14.7	8.7	76.6	10.5	8.7	80.8	9.5	10.7	79.8	7.6	14.7	77.7
	NDCG	32.2	2.6	65.2	26.9	8.8	64.3	22.0	8.8	69.2	22.9	10.8	66.3	19.4	14.8	65.7
	RBP	94.8	5.2	0.0	88.7	11.3	0.0	85.5	14.5	0.0	81.9	18.1	0.0	76.8	23.2	0.0

Table 4.7: When the pool depth increases from d' to $d = 50$ (the reference), the percentages of runs whose metric score grows (< 0), stay the same ($= 0$), and decreases (> 0) respectively. That is, the percentage of points shown in each graph of Figure 4.8 and Figure 4.9 which are below (< 0), on ($= 0$), and above (> 0) the horizontal line of $y = 0$.

Collection	Metric	τ of differences	τ of ratios
TREC-7	AP	0.04	0.18
	NDCG	0.09	0.16
	RBP, $\phi = 0.98$	-0.06	-0.19
TREC-13	AP	0.09	0.14
	NDCG	0.07	0.08
	RBP, $\phi = 0.98$	-0.12	-0.23

Table 4.8: Kendall's τ between RBP residual scores and metric score differences (the third column), and metric score ratios (the fourth column) using the reduced judgments.

To verify this assumption, we computed the Kendall's τ of ratios, instead of arithmetic score differences, between run scores given by shallow judgments and $d = 50$ judgments. For each of the three situations in Figure 4.8, the Kendall's τ values of score ratios (in TREC-7, 35 runs were removed because their run score was 0 even when pooling depth is 50) are summarized in the last column of Table 4.8. And for three situations in Figure 4.9 (runs having no relevant document before rank 50 in TREC-13 were removed), the Kendall's τ scores are shown in the fourth column of last three rows in Table 4.8. The Kendall's τ of score ratios indicate marginal stronger relationships between RBP residuals and run score changes, compared to Kendall's τ computed using arithmetic score differences.

An ANCOVA (analysis of covariance) analysis of IR effectiveness scores evaluated by four metrics: RBP, AP, NDCG and RR, was carried for investigate the relative effects that could be caused by factors of judgment pool to size (`d`), topic difficulty (`topic`) and system diversity (`sys`). With the null hypothesis that none of factors of pooling depth, topic and system brings variation to the metric scores and no interactions between them, the ANCOVA took the input of 65 deeply-pooled systems from TREC-7 Ad Hoc and 50 corresponding topics (67 deeply-pooled systems and 50 corresponding topics when testing for TREC-8 Ad Hoc, and 52 deeply-pooled systems and 49 topics (651–671, 673–700) when testing for TREC-13 Robust), and 10 pooling depths $d' \in \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. Reciprocal rank (RR) was included in this experiment as a reference point because it is a shallow metric, so that changing of pooling depth d' should have little effect on it.

We used the `car` package in R, and the formula of computing main effects for factors given to the R program after reading and factorizing data into `dat`, for example, when investigating how AP scores (`AP`) of systems depend on factors judgment pool depth (`d`), topic (`topic`), and system diversity (`sys`), is expressed as:

```
Anova(aov(AP ~ d + topic + sys, data=dat), type="III").
```

factor/interaction	TREC-7				TREC-13			
	AP	NDCG	RBP	RR	AP	NDCG	RBP	RR
d	***	***	***	0.860	***	***	***	0.875
topic	***	***	***	***	***	***	***	***
sys	***	***	***	***	***	***	***	***
sys	***	***	***	***	***	***	***	***
d:topic	***	***	***	***	***	***	***	***
d:sys	***	***	***	***	***	***	***	***
topic:sys	***	***	***	***	***	***	***	***
d:topic:sys	***	***	***	***	***	***	***	***

Table 4.9: The p-values generated in ANCOVA tests for four metrics, which investigate the relative effects that factors judgment pool depth (d), topic (topic), system diversity (sys) and their interactions might have on the metric scores. The *** represents the p-value less than the significance level $\alpha = 0.05$, that is the metric scores are significantly affected by the tested factor (or the interaction) shown in the first column.

The formula for fitting a model including interactions with factors is:

```
Anova(aov(AP ~ d * topic * sys, data=dat), type="III").
```

The ANCOVA analysis was performed for all four metrics and three test collections. The generated p-values for TREC-7 and TREC-13 (the results for TREC-8 are similar) are summarized in Table 4.9. The first column shows the tested factors (d, topic, and sys), and their possible interaction combinations (d:topic, d:sys, topic:sys, and d:topic:sys). If the generated p-value is less than the significance level ($\alpha = 0.05$), represented by *** in the table, the test hypothesis should be rejected (that is, the tested factor, or interaction, does significantly affect the metric scores). As the results show in Table 4.9, RR scores do not significantly depend on the pool depth, but the other three deeper metrics do. The outcomes also show that the tested four metrics are all significantly affected by the topic, system diversity, and interactions for all of the three factors. Results for four metrics across TREC-7, TREC-8, and TREC-13 test collections are all similar.

4.10 Summary

We have estimated the reliability of IR system effectiveness scores evaluated by recall-based metrics using distinct methods, a typical approach of which is comparing systems in pairs over all the topics using paired two-tail t-test with the null hypothesis that lower

p-value indicates higher confidence in the experiment conclusion that the paired systems are different. A presumption was made that if the scores developed by metrics were imprecise, it would be more difficult to distinguish systems. However, the results shown in this chapter state that system discrimination ratios were relatively unaffected by the pool depth of judgments (shallow pooling depth increases the unreliability of relevance judgments and hence adds uncertainty to metric scores used for comparing systems) or the imprecision of metric scores. The shallow pooling depths neither reduce the confidence in t-test outcomes. So our presumption may need to be re-considered.

We explored the relationships between residuals computed by utility-based metrics such as RBP and effectiveness scores evaluated by recall-based metrics. System orderings generated by RBP and some different recall-based metrics were compared and found to be quite close when the RBP parameter $\phi \approx 0.98$. If the pooling depth is $d = 50$, RBP residual will be at least $\phi^d = 0.36$. And hence if using RBP to evaluate a run, saying the metric score summed over the pooled documents is 0.60, it may be raised to $0.60 + 0.36 = 0.96$ by unjudged documents. Such big score jumps are quite unusual in practice, but as we have demonstrated in previous sections, the situation that the metric score shifts greatly when the pooling depth is extended from shallow to deeper can definitely occur.

In the absence of explicit method for computing residuals for recall-based metrics, we have explored the behaviors of recall-based metrics when uncertainty is varied, and discovered that the discrimination of recall-based metrics typically increases with the pool depth (and also the number of judgments). The confidence of distinguishing systems which are highly different also mostly increased with the pool size. We also found that the scores developed by AP and NDCG tend to decrease as the pool size increases. As we had previously found that RBP residuals generally decrease with judgment uncertainty, we then sought for the connection between RBP residuals and score movements of recall-based metrics as the pooling depth increases. However, the detected correlation was not strong, which might be due to factors that we did not consider in our experiments, or it might be that metric consistency (as we tested to compare systems in pairs before using t-test) has no clear relationship to residuals.

In general, we have demonstrated that even though current IR evaluations may have important potential risks on reproducibility, they do tend to work well in practice. The extent of uncertainty in per-topic evaluation, which is caused by unjudged documents, cannot be reflected in statistical significance tests. But as we discussed in previous sections, the behaviors of RBP with high values of parameter ϕ could be quite close to recall-based metrics AP and NDCG, whose corresponding uncertainties were not minor. Thus we suggest that residuals developed by weighted-precision metrics such as RBP for indicating uncertainties of scores should be presented in addition to statistical test results for examining metric score consistency. When recall-based metrics are considered as preferred metrics in IR evaluations, with no direct relationship between metric scores and residuals, we recommend that researchers report high ϕ RBP residuals companied with

the statistical significance test results to quantify the reliability of assessments. If such residuals cannot be calculated for some reason, we would encourage researchers to provide information about the fraction of unjudged documents before rank k across the set of topics as part of their presentation of experimental results.

Chapter 5

Pairwise Judgments

Information retrieval (IR) system performance is often assessed by the batch evaluation techniques described in Section 2.1.2. For a set of given topics (information needs), each system s that is to be evaluated computes the similarity score relative to the given query t for each document using some ranking function and then returns a ranked list $r_{s,t}$, also called a *run*, containing documents in decreasing similarity score order. An IR evaluation *metric* M such as Average Precision (AP) or Rank-Biased Precision [73] is then employed to compute the effectiveness of each system based on a set of *relevance judgments*. Each system s will receive a numeric value $\mathsf{M}(r_{s,t})$ from metric M as an assessment of the retrieval quality. Relevance judgments are typically formed using ordinal scales at binary level or multiple levels [22]. They are recorded in the form of a *qrels* file which is composed of a list of tuples, each containing a topic number, a document number and a relevance grade indicating how much overlap there is between the document and the topic.

Conventionally, relevance judgments were assessed by small numbers of expensive trained experts using ordinal relevance scales. In recent years, researchers have explored using different relevance scales such as *pairwise preference* (PP) [18], *Magnitude Estimation* (ME) [105] and *fine-gained* scales (S100) [80] to collect relevance judgments via *crowd-sourcing* platforms, with good quality and low cost.

Using PP and ME, instead of assigning an absolute relevance score for each document, assessors only need to compare the relevance between two (or paired) documents, and answer questions such as which document is more relevant, or what is the relevance ratio between the paired documents. Assessors' perceptions of relevance can be naturally adapted without interpreting the relevance gain of each level expected by different assessors. But the preference or ratio answers need to be converted into a number for each document to allow computations of metrics. In contract, the S100 method or binary judging does not require assessors consider the relevance of a document relative to one another, and the collected relevance grades can be directly used to form qrels without requiring score conversion or normalization.

In order to collect relevance judgments with high fidelity and hence have a better understanding of user perceptions of relevance, we investigate the variation of relevance judgments collected using three different relevance scales: pairwise preference, absolute relevance, and relevance ratio; all using a crowd-sourcing platform which provides a

large number of non-specialist assessors. We take advantage of previously proposed methods and combine them in our experimentation, including forced choice answers and embedded quality control processes. The following research questions are considered:

RQ5: Does the combination of relevance judging techniques collected by via crowd-sourcing give similar relevance scores to previous methods?

RQ6: Is it possible to collect those judgments at lower cost than previous collection schemes?

RQ7: What factors might affect the quality of relevance assessments?

RQ8: Do our crowd-sourced judgments alter IR system evaluation compared to previous judgment schemes?

5.1 Experimental Design

We selected nine topics from the TREC-8 [112] Ad Hoc collection, as this set has already been used in experiments by other researchers for collecting ME, Sormunen and NIST Binary judgments, and the top-10 documents returned by the contributing TREC systems as the pool of each topic (so we have the same dataset with ME). We denote the pool size as $\boldsymbol{NumDocs}_{pool}$ which varies across topics. The title query and $\boldsymbol{NumDocs}_{pool}$ of each selected topic are listed in Table 5.1.

5.1.1 Pairing Documents

Figure 5.1 shows the process of generating document pairs for each topic. For each topic, the $\boldsymbol{NumDocs}_{pool}$ pooled documents were randomly and equally partitioned into different groups a total of $\boldsymbol{X}$ times. Each group contained ***DocsPerGroup*** documents with two special ones – `HR` (highly relevant) and `NR` (not relevant), whose ordinal relevance values are known according to the Sormunen judgments. The pair list generated for each group contained a pair of `HR` and `NR`, called the *gold standard*, used for assessing the worker's accuracy (workers are expected to choose the `HR` rather than `NR` in the gold standard). The other ***DocsPerGroup*** documents in each group were randomly selected from the pool. If $\boldsymbol{NumDocs}_{pool}$ cannot be integer divided by ***DocsPerGroup*** for any particular topic, some further randomly selected documents (outside the pool) were included into the pool to make it so. The column ***NumDocs*** in Table 5.1 shows the sum of $\boldsymbol{NumDocs}_{pool}$ and the number of documents outside the pool added to obtain the integer multiple.

Topic	Title Query	$\textbf{\textit{NumDocs}}_{pool}$	***NumDocs***	***NumParts***	***NumGroups***	***JudgPerDoc***	Date
402	behavioral genetics	278	285	14	532	126	2017.12
403	osteoporosis	111	120	6	96	54	2017.07
405	cosmic events	214	216	11	297	99	2018.05
407	poaching, wildlife preserves	212	225	11	330	99	2017.11
408	tropical storms	188	192	10	240	90	2018.05
415	drugs, Golden Triangle	179	184	9	207	81	2017.12
416	Three Gorges Project	174	176	9	198	81	2017.12
431	robotic technology	203	208	10	260	90	2017.12
440	child labor	264	270	13	468	117	2017.12

Table 5.1: Data and parameters for different topics in the experiment as used in Equation 5.2. $\textbf{\textit{NumDocs}}_{pool}$ is the number of documents in the pool of each topic. ***NumDocs*** includes any additional documents (which are outside the pool) required to allow the integer division in Equation 5.2. ***JudgPerDoc*** is the number of judgments received by each document for the given topic. The "Date" column shows the month in which the relevance judgments were collected.

Notation	Description
NumDocs	The number of documents (including `HR` and `NR` documents) in the pool.
`HR`	Documents whose relevance level is high in Sormunen, used to construct gold standard pairs for accuracy checking.
`NR`	Documents which were judged as irrelevant by Sormunen, used to construct gold standard pairs.
NumHR	The number of gold standard `HR` documents.
NumNR	The number of gold standard `NR` documents.
DocsPerGroup	The number of documents in each group, for which a pair list is generated.
X, NumParts	The number of partitionings.
NumGroups	The total number of pair lists (or groups) for the given topic.
K	The number of documents that each document is paired with in a pair list (or group).
PairsPerGroup	The number of document pairs in a pair list (or group).
NumPairs	The total number of document pairs for the given topic.
Y	The number of distinct assessments collected for each pair list.

Table 5.2: Glossary for parameters in the process of generating pair lists.

The fidelity of pairwise judgments might be affected by the number of documents (***NumDocs***) and the number the generated pairs. For each topic, the repetition multiplier ***X*** was chosen so that each document would be paired with around 15% of the other documents in the pool for that topic. The total number of groups across all partitions for each topic is thus:

$$\begin{aligned} \textbf{\textit{NumGroups}} &= \textbf{\textit{NumGroups Per Partitioning}} \times \textbf{\textit{X}} && (5.1)\\ &= \textbf{\textit{NumDocs}}/\textbf{\textit{DocsPerGroup}} \times \textbf{\textit{X}} && (5.2) \end{aligned}$$

For each group, a list of document pairs were generated in which each document was compared with ***K*** other documents from the same group. Thus the number of pairs in a group can be given as

$$\textbf{\textit{PairsPerGroup}} = \textbf{\textit{DocsPerGroup}} \times \textbf{\textit{K}}/2. \quad (5.3)$$

To further reduce the assessment complexity, one of the documents in the current pair was randomly chosen and retained as the next pair was formed.

As in the pair list example shown in Figure 5.1, the crowd worker starts with reading document B and C and examining the pair (B, C). In the next pair where document B stays, the worker only need to read document A to answer questions for the pair (B, A). In other words, except for the first pair, participants read one new document at each pair assessment. Moreover, as each pair list is generated by documents in a group, only ***DocsPerGroup*** documents need to be viewed by participants to complete assessments

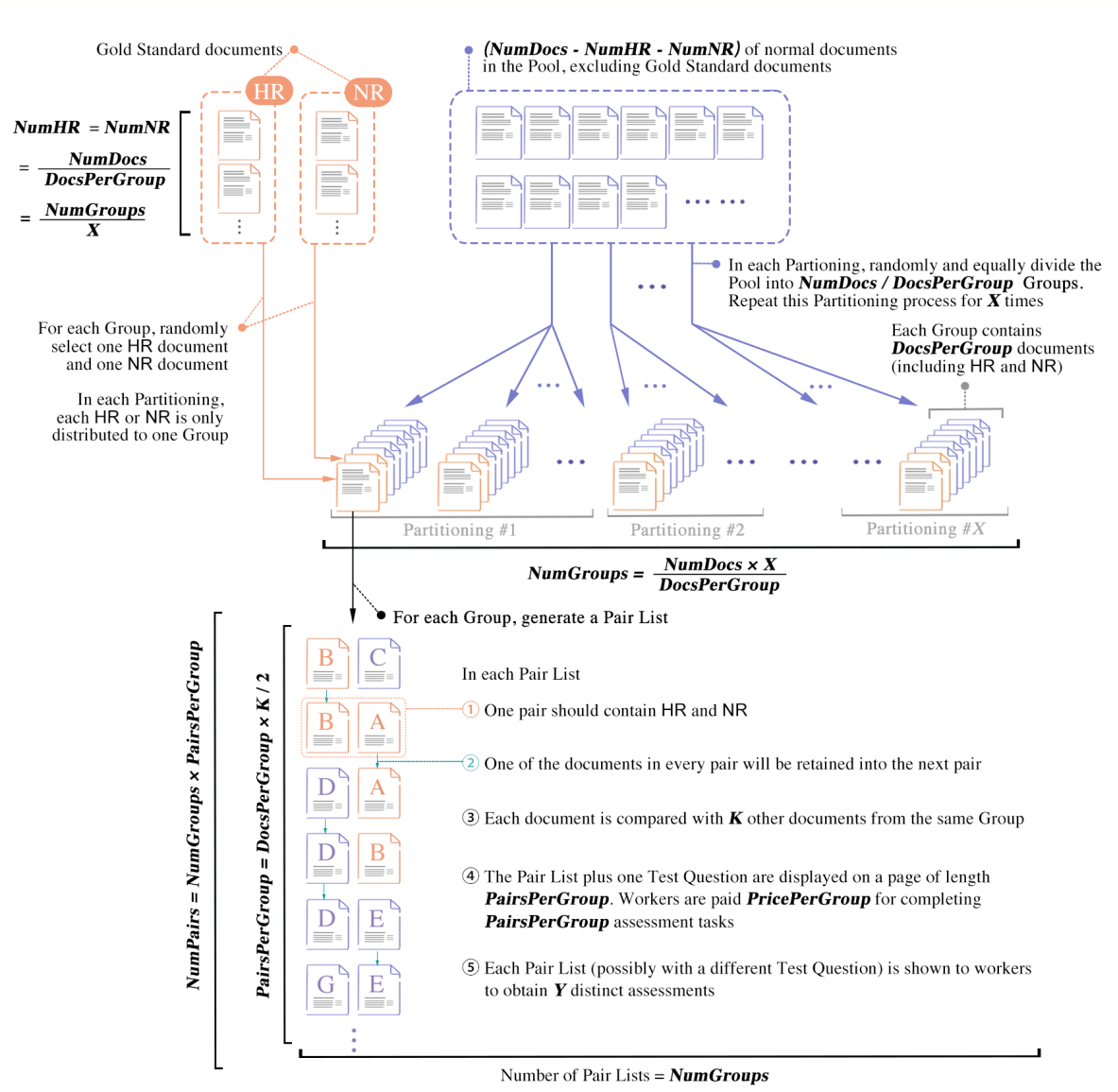


Figure 5.1: The workflow of generating pairs for each topic offline before uploading data to the crowd-sourcing platform. The parameters shown in the diagram are take on different values for each of the topics.

for ***DocsPerGroup*** $\times$ $\boldsymbol{K}/2$ (that is, ***PairsPerGroup***) different pairs.

In the pair list, each document is paired with exact $\boldsymbol{K}$ other documents in the group, and one of the pairs has to be the gold standard. For example, assuming *A*, *B*, *C* and *D* are documents in a group, *A* is HR, *B* is NR, and $\boldsymbol{K} = 2$, a possible randomly generated pair list might be $[A, C], [A, B], [D, B], [D, C]$. Each document is in $\boldsymbol{K} = 2$ pairs, and the gold standard pair in the generated list is $[A, B]$.

The process of generating such a pair list for a group is descried in Algorithm 1. The starting point, PAIRING_FOR_GROUP function, takes the a group of documents and the value of $\boldsymbol{K}$ as parameters, and calls the FIND_PAIR_LIST function to find all the pairs in the list.

The recursive function FIND_PAIR_LIST takes the *pair_list* which records pairs generated till here from previous nested calls (on line 29 and 34), and finds the remaining pairs which can only be constructed by documents in *remains* and the pair generated in the current recursion should contain *doc1*. The function PICK_DOC from line 45 is called by FIND_PAIR_LIST to randomly pair *doc1* with another document in *remains*. It ensures that each document in the group is paired with exactly $\boldsymbol{K}$ other documents in the list and that there is no repeated pair.

The parameters for the nine selected topics are listed in Table 5.1. In our experiment, we grouped eight documents (including a HR and a NR, that is, ***DocsPerGroup*** $= 8$), and randomly pair each document with three other documents in the same group ($\boldsymbol{K} = 3$, thus ***PairsPerGroup*** $= 12$). Each assessment task was deemed to be finished when it successfully received valid judgments from three workers ($\boldsymbol{Y} = 3$).

For example, the ***NumDocs*** of topic 405 is 216. As we set ***DocsPerGroup*** $= 8$, the group number of each partitioning is $216/8 = 27$, and so we need 27 HR and 27 NR documents determined according to the Sormunen judgments. The remaining $216 - 27 \times 2 = 162$ documents are randomly divided into 27 groups.

The HR documents were primarily selected from documents in the categories of H and R by Sormunen. For Topics 402, 403, 407 and 440, there were not enough H and R documents, thus some M documents were also used. For Topics 402, 403, 407 and 440, H, R and M documents are too few relative to ***NumDocs***, thus HR documents for these topics were reused to make up the required number.

As already noted, we took ***DocsPerGroup*** $= 8$ and $\boldsymbol{K} = 3$, the pair list for each group contains ***PairsPerGroup*** $= 12$ pairs. For topic 405, for example, the partitioning process was repeated $\boldsymbol{X} = 11$ times and so there were $11 \times 27 = 297$ groups in total, with each document paired with $\boldsymbol{K} \times \boldsymbol{X} = 3 \times 11 = 33$ other documents (15.3% of the pool) in the assessment of this topic. Each group was assigned to a task unit which required $\boldsymbol{Y} = 3$ different workers to provide valid judgments. That is, in total, each document for Topic 405 received $\boldsymbol{K} \times \boldsymbol{X} \times \boldsymbol{Y} = 99$ judgments.

An example of a pair list for a group of eight documents for topic 408 is shown below:

Algorithm 1 Algorithm to randomly generate a pair list for a group of documents. A list of document ID (*a_group*) and the value of ***K*** (*K_value*) are passed as parameters to the PAIRING_FOR_GROUP function which collaborates with two other functions and returns the pair list.

```
1: function PAIRING_FOR_GROUP(K_value, a_group)
2:     for all doc in a_group do
3:         K_of_doc(doc) = K_value
4:     start_doc ← Randomly choose one from a_group
5:     pair_list ← FIND_PAIR_LIST(K_of_doc, a_group, start_doc, [ ])
6:     return pair_list
7: function FIND_PAIR_LIST(K_of_doc, remains, doc1, pair_list)
8:     if doc1 is Nothing then return pair_list                    ▷ No solution.
9:     else if length of remains is 1 then
10:        return pair_list                    ▷ No solution. doc1 is the only left.
11:    loop
12:        doc2 ← PICK_DOC(K_of_doc, remains, doc1, pair_list)
13:        if doc2 is not Nothing then
14:            Add pair (doc1, doc2) into pair_list
15:            K_of_doc(doc1) ← K_of_doc(doc1) - 1
16:            K_of_doc(doc2) ← K_of_doc(doc2) - 1
17:            if this is the last pair and K_of_doc of every doc is 0 then    ▷ Last pair
18:                Remove doc1 and doc2 from remains
19:                return pair_list          ▷ Last pair has been added. Task finished.
20:            else                                                ▷ Not the last pair
21:                if either K_of_doc(doc1) or K_of_doc(doc2) is 0 then
22:                    Remove the one that is 0 from remains
23:                    Assign another to stayed
24:                    replaced ← Nothing
25:                else                                    ▷ Randomly pick one to stay
26:                    stayed ← Randomly pick doc2 or doc1
27:                    replaced ← The element that was not picked
28:
29:                new_pair_list ← FIND_PAIR_LIST(K_of_doc, remains, stayed, pair_list)
30:                if a new pair was found and stored in new_pair_list then
31:                    return new_pair_list          ▷ Solution is found and returned.
32:                else                              ▷ No solution if stayed is stayed.
33:                    stayed ← replaced          ▷ How about keeping the other document?
34:                    new_pair_list ← FIND_PAIR_LIST(K, remains, stayed, pair_list)
35:                    if a new pair was found and stored in new_pair_list then
36:                        return new_pair_list      ▷ Solution is found and returned.
37:                    else                  ▷ (doc1, doc2) causes no further solution.
38:                        Roll back every change made in this interation
39:                        if ele1 is the only left one in remains then
40:                                ▷ No other option can be tried for doc2. Stop looping
41:                            return pair_list                        ▷ No solution.
42:
43:        else                                ▷ Cannot pick any element in remains for doc2.
44:            return pair_list          ▷ No solution. The original pair_list is returned.
```

```
45: function PICK_DOC(K_of_doc, remains, doc1, pair_list)
46:     Remove doc1 from remains                    ▷ Cannot be paired with itself.
47:     loop
48:         if remains is empty then
49:             return Nothing
50:         doc2 ← Randomly pick an element from remains
51:         if K_of_doc(doc1) is 1 and K_of_doc(doc2) is 1 then
52:             if only doc2 is in remains then
53:                 if pair (doc1,doc2) is already in pair_list then
54:                     return Nothing
55:                 if any document whose K_of_doc is greater than 1 then
56:                     return Nothing
57:                 return doc2
58:             Remove doc2 from remains   ▷ Try another document in the next interation.
59:         else if (doc1,doc2) is in pair_list then
60:             Remove doc2 from remains   ▷ Try another document in the next interation.
61:         else
62:             return doc2
```

```
LA101589-0180     FR940728-2-00132
LA101589-0180     FT943-2257
LA102389-0075     FT943-2257
LA102389-0075     FR940803-2-00061
LA102389-0075     FT911-124
FR940803-2-00061  FT911-124
FR940803-2-00061  LA031190-0095
LA101589-0180     LA031190-0095
FR940728-2-00132  LA031190-0095
FR940728-2-00132  LA072990-0060
FT943-2257        LA072990-0060
FT911-124         LA072990-0060
```

in which each document is paired with $K = 3$ other documents, and one of the paired documents stays in place at each pair transition.

5.1.2 Data on Figure8

The generated pair lists for each topic are stored in a file, and each row contains the topic number and identifiers of two paired documents, such as:

UNIT ID	STATE	JUDGMENTS	AGREEMENT	TOPIC	DOC1	DOC2
2238486573	new	0		408	LA101589-0180	FR940728-2-00132
2238486574	new	0		408	LA101589-0180	FT943-2257
2238486575	new	0		408	LA102389-0075	FT943-2257

Figure 5.2: A screen shot of three uploaded rows (document pairs) on `Figure8`.

```
408  LA101589-0180  FR940728-2-00132
408  LA101589-0180  FT943-2257
408  LA102389-0075  FT943-2257
...  ...            ...
```

As ***PairsPerGroup*** $= 12$, each twelve rows in the file are for the pairs in one group. The file is then uploaded to `Figure8`, with a screen shot of three data rows is shown in Figure 5.2. On the `Figure8`, we tick the box of *rows should be completed in order*, and set the number of rows per page to be twelve, so that pairs in each pair list are shown on one page, and judged by the same workers. As stated before, each pair list is required to be assessed by $\boldsymbol{Y} = 3$ different workers. Thus in `Figure8`, we set *judgments per row* to be three, and remove the tick from the box of *rows remain finalized* (that is, the three judgments obtained for each pair list have to be valid and provided by "trusted" workers whose overall judging accuracies are above ***MinAccuracy*** $= 84\%$, which will be described in Section 5.1.4). If a pair list has received $\boldsymbol{Y} = 3$ judgments, and is, finalized, but the assessment accuracy of two of its assessors drop below 84%, then the judgments provided by these untrusted workers will be removed, and the pair list will not remain finalized but re-open to other workers to collect two sets of new judgments. The crowd task of each topic finishes only after every pair list receiving judgments from $\boldsymbol{Y} = 3$ trusted workers.

When workers complete the assessment of one page, they get paid USD$0.12 for twelve pairs if they reach the requirements of quality control processes (described in Section 5.1.4). Note that `Figure8` takes 20% transaction fee for each launched task from the task owner, which is also included as cost of our experiments. In general, we paid USD$0.01 + 0.002 to obtain answers of three questions for each row (document pair).

5.1.3 Interface

The instruction and assessing steps that workers should follow and complete, shown on the top of the task page before workers start assessing the group, are (for each pair of documents):

1. *Read the description, narrative, and query at the top of each pair.*

2. *Read and analyze both documents.*

3. *In* QUESTION1*, decide which document is more relevant to the given description and click the button below the preferred document. If you think the documents are equally relevant to the description, choose one of them as best as you can.*

4. *In* QUESTION2*,*

   - *if you think both documents are relevant to the topic, click "Both documents are relevant".*
   - *if you think only one document is relevant, that is the document that you chose in* QUESTION1*, click "Only the document I selected is relevant".*
   - *if you think neither of the documents is relevant, click "Both documents are non-relevant".*

5. *In* QUESTION3*, judge each document in relation to the paired one by assigning each document a relevance score. You may use any positive number that seems appropriate to you – whole numbers or decimals. If you think the document is NOT relevant, assign 0. Make sure the relevance score of the document that you selected in* QUESTION1 *is larger.*

There is some additional information that workers needed to notice before starting the task, which helps workers get familiar with the interface, and understand the requirements of our tasks:

- *the window frame for each document can be scrolled; please read the whole content of each document;*
- *please make the preference choice carefully; there are checks in each task to assess your accuracy;*
- *you are required to answer ALL of the shown questions for ALL of the document pairs before you can submit the task;*
- *one of the documents in each pair will be shown again in the next pair; make sure you analyze BOTH documents in every pair before making your preference choice.*

The interface employed to assess each pair is shown in Figure 5.3. For each document pair, workers needed to make a preference choice of which document is more relevant to the topic than the other by clicking the button for the preferred document in the first question (QUESTION1). If both documents in the pair are considered as irrelevant and cannot be distinguished, workers were instructed to "choose one of them as best as you can". Workers were required to give the absolute relevance (relevant or irrelevant) for both documents in QUESTION2 using three further buttons (see the screen shot). In QUESTION3, workers were asked to assign a numeric score (any positive number for a relevant document and zero for irrelevant documents) to each document to indicate its relevance

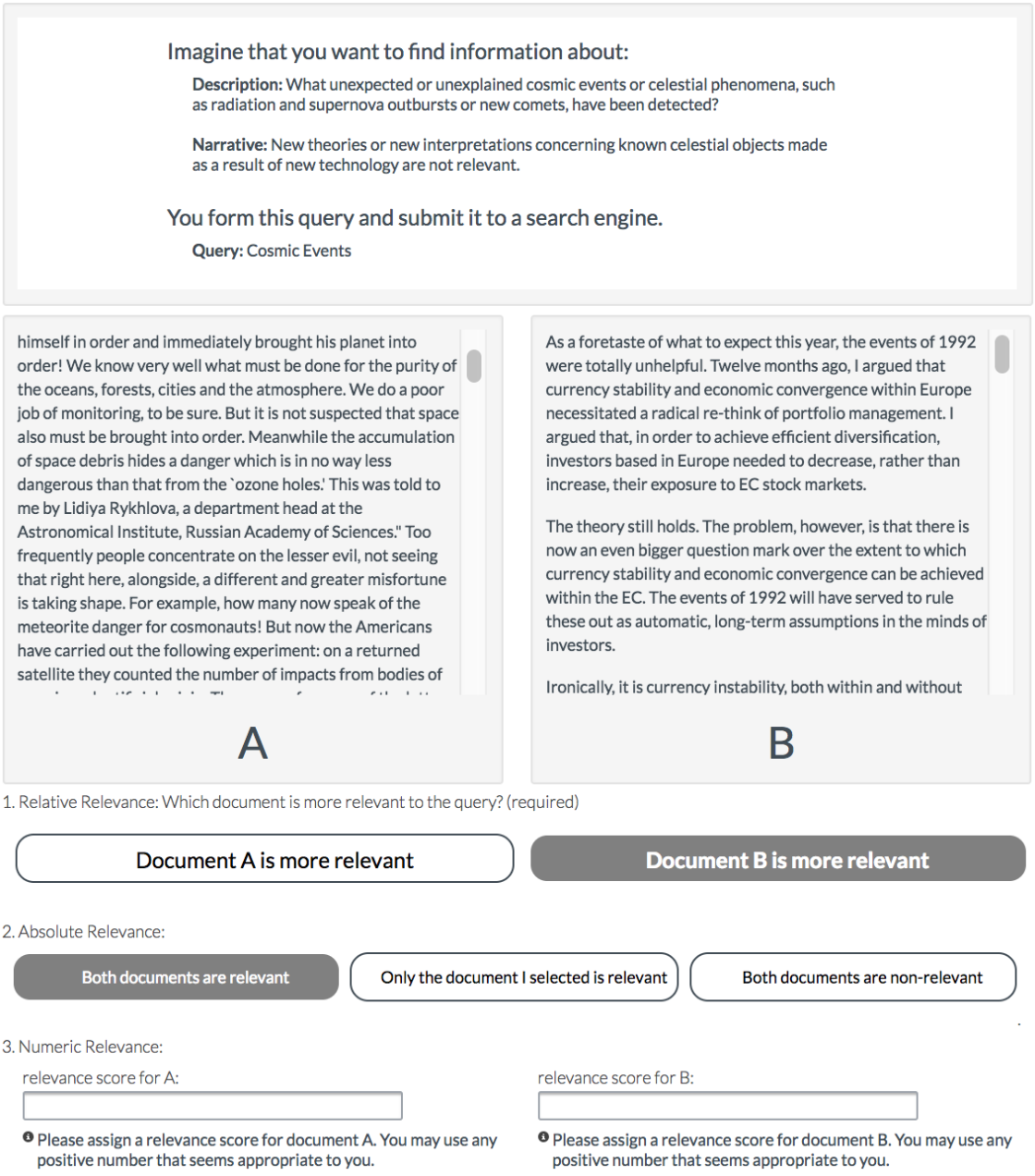


Figure 5.3: Screen shot of a document pair assessment on `Figure8`. The description of the topic is shown first with the document pair below. The two documents are displayed side-by-side (the ordering was randomly chosen when pairs were generated) in a scrolling box, followed by three questions. Workers are required to answer all of QUESTION1, QUESTION2 and QUESTION3 for each displayed pair. Workers get paid when they have provided valid judgments for all twelve pairs associated with a single group. Each of eight documents in the group is presented three times in the sequence of pairs.

in relation to the paired one. This last question was intended to capture the relevance ratio between the paired documents.

After workers finished the assessments of a group (a page of ***PairsPerGroup*** $= 12$ pairs), they saw a multiple-choice survey question which asks "which assessing method(s) do you like". Workers are allowed to choose one or more answers from following options:

- *pairwise preference (in* QUESTION1*);*
- *giving absolute relevance (in* QUESTION2*);*
- *assigning numeric scores (in* QUESTION3*).*

We gathered the answers of this survey, and analyze assessor's preference of the three methods in Section 5.3.5.

5.1.4 Quality Control

At the beginning of each task, workers were asked to read the task specification and the topic description carefully. As Figure 5.4 shows, we employed the `Figure8`'s *quiz mode* to filter out low-quality workers as well as training assessors about the task, such as which documents should be considered as relevant according to the topic description and how to use the interface, before they could start the real assessing task in the *work mode*. To enter work mode, workers needed to complete ***PairsPerGroup*** $+ 1$ (the same number of pairs per page in the work mode; as shown in Figure 5.4, workers needed to answers questions of ***PairsPerGroup*** document pairs, plus one test question on one page in the work mode) of randomly picked known-answer test questions in the quiz mode with at least accuracy of ***MinAccuracy***. In our experiment, ***PairsPerGroup*** $= 12$, thus workers judged 13 document pairs per page. The ***MinAccuracy*** was set to be $11/13 = 84\%$, that is, workers needed to correctly answer 11 out of 13 test questions to enter the work mode.

The test questions were constructed by a pair of pseudo documents, hand-crafted short summaries relative to the topic, with known and distinct relevance levels (`H`–highly relevant, `M`–marginally relevant, `N`–irrelevant). Two academics created ten pseudo documents for each topic. For example, the standard NIST query, description and narrative of topic 405 are:

- query: cosmic events
- description: what unexpected or unexplained cosmic events or celestial phenomena, such as radiation and supernova outbursts or new comets, have been detected?
- narrative: new theories or new interpretations concerning known celestial objects made as a result of new technology are not relevant.

One instance of hand-crafted pseudo documents in each relevance level created for topic 405 is:

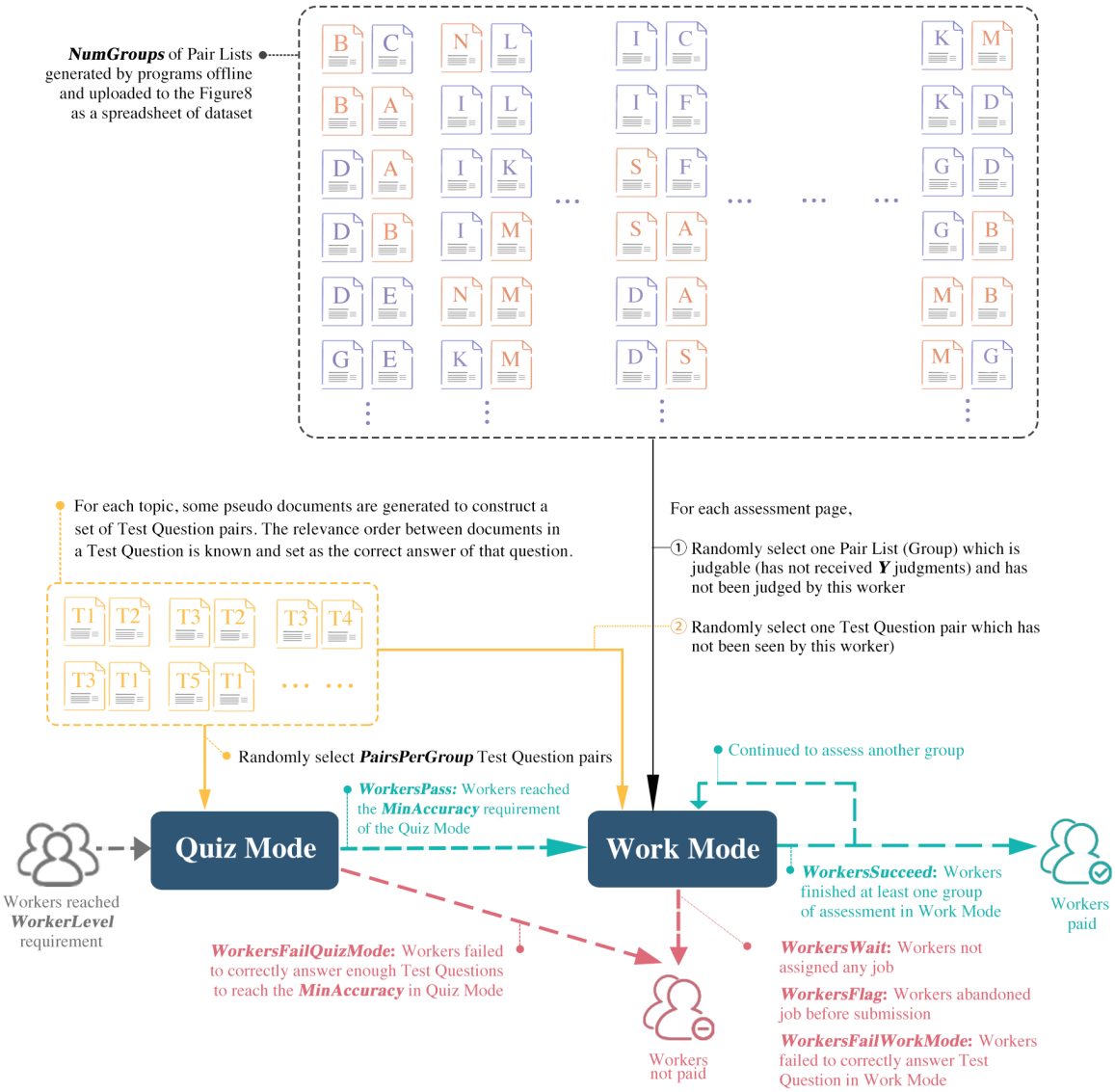


Figure 5.4: The workflow of our experiments running on the crowd-sourcing platform, `Figure8`. Pair lists shown at the top of the diagram are generated offline, as illustrated in Figure 5.1. Workers were non-expert assessors from `Figure8`. The parameters of Y, *WorkerLevel*, *MinAccuracy* and the money paid for one valid group of assessments could be set on the website of `Figure8`. We used $Y = 3$, *WorkerLevel* $= 1$ and *MinAccuracy* $= 0.84$.

Doc1	Doc2	QUESTION1		QUESTION2		
		Doc1	Doc2	OPTION1	OPTION2	OPTION3
H	H	✓	✓	✓		
H	M	✓		✓	✓	
M	H		✓	✓	✓	
H	N	✓			✓	
N	H		✓		✓	
M	N	✓			✓	✓
N	M		✓		✓	✓
N	N	✓	✓			✓

Table 5.3: The correct answers of QUESTION1 and QUESTION2 for test question pairs. The first two columns show the relevance levels of the paired pseudo documents. As all of the options for pair (M, M) are not wrong, and hence (M, M) is avoided when generating test question pairs. In QUESTION2, the three options are: OPTION1 – both documents are relevant; OPTION2 – only the document selected in QUESTION1 is relevant; OPTION3 – both documents are non-relevant. The ✓ mark represents the correct answer for the question. If the question has multiple correct answers, workers only need to choose one of them.

- H – A document that reports the recent discovery of a new comet that will pass near Earth in 2031.
- M – A document that summarizes new measurements of supernova SN 1052.
- N – A document that describes the film series Star Wars.

In test mode, workers viewed 13 randomly selected pairs of pseudo documents, and answer QUESTION1, QUESTION2 and QUESTION3 for each pair. The correct answers (might be more than one for each question) of QUESTION1 and QUESTION2 for test question pairs were provided to Figure8 before launching the job. As the answer of QUESTION3 was not unique, the worker's answer was deemed to be correct if the score of the preferred document in QUESTION1 was larger. The acceptable answers for each test question pair are decided based on Table 5.3.

We also employed forced-choice testing and embedded quality control processes in the work mode HITs to reduce the assessment criteria drift. For example, for each document pair, each worker's answer to QUESTION3 was required to be in agreement with the preference choice made in QUESTION1. In each group of assessments in the work mode, one test question which had not been viewed by the worker was randomly selected and included in the assessment page. If the worker failed to correctly answer the test question in the work mode, the worker's assessments of this group would not be deemed as trusted answers by Figure8.

Topic	Workers	Costs (USD)	Trusted	Untrusted
402	152	296.78	19152	5772
403	47	43.63	3456	192
405	126	130.61	10692	240
407	103	157.10	11880	1272
408	125	121.25	8640	1512
415	72	95.76	7452	540
416	74	94.32	7128	774
431	79	138.10	9360	2232
440	133	225.79	16848	2064
Total	911	1303.34	94608	14598

Table 5.4: The number of workers, overall cost, the number of trusted judgments and the number of untrusted judgments for each topic. The overall money paid to the crowd-workers (shown in the third column) includes payments for trusted judgments (the assessments we used for building judgments), untrusted judgments (bad assessments filtered out by `Figure8`) and transaction fees (paid to `Figure8`).

As Figure 5.4 shows, workers who failed to reach the ***MinAccuracy*** in the quiz mode, or gave up the task in work mode, or incorrectly answered the test question in the work mode, or had no assigned tasks, did not get paid. Workers who provided trusted answers of a group were paid USD$0.12 (that is, USD$0.01 per pair). After completing a group of assessments, they could choose to judge another group, or quit.

5.2 Overall Outcomes

We launched jobs for nine TREC-8 topics on `Figure8`, and collected pairwise relevance judgments for 1876 topic-document combinations (see Table 5.1 for number of documents for each topic) using three methods: pairwise preference (QUESTION1), absolute relevance (QUESTION2), and relevance ratio (QUESTION3). Documents for each topic were partitioned into groups, and the partitioning process was repeated for $\boldsymbol{X}$ times. Documents in the group were paired using the strategy described in Section 5.1.1. For each group, one list of ***PairsPerGroup*** $= 12$ document pairs was randomly generated. In total, we generated 2628 pair lists and $2628 \times 12 = 31536$ document pairs for eight topics in our experiment. For each pair, we asked $\boldsymbol{Y} = 3$ different trusted crowd workers three questions, thus we collected $2628 \times 12 \times 3 \times 3 = 283824$ trusted answers on the `Figure8` crowd-sourcing platform.

As shown in Table 5.4, we spent a total of USD$1303 to collect pairwise judgments for nine topics and 1876 topic-document combinations on the `Figure8` crowd-sourcing

platform. After aggregating and normalizing the answers of three questions of each document pair, each topic-document combination has three judgments collected by three methods respectively. That is, each of three sets of judgments for 1876 topic-document combinations only cost USD$\$1303/3 = \$434.3$ on average for collection.

There were 806 different workers (in total of 911 worker-topic combinations) who completed 2628 pair groups and provided $2628 \times 12 \times 3 = 94608$ trusted judgments. Note that each judgment included answers of three questions. There were about 11% judgments considered as untrusted by `Figure8` because the accuracy of test question answers given by some workers dropped below the $\boldsymbol{MinAccuracy} = 84\%$. All judgments provided by these untrusted workers (even they were paid) were automatically removed from the trusted judgments set, and `Figure8` would collect new judgments until all pairs had $\boldsymbol{Y} = 3$ assessments from three different trusted workers.

5.3 Aggregated Judgments

5.3.1 Normalization

In order to measure the validity and accuracy of our method which jointly collects pairwise preference (in QUESTION1), absolute relevance (in QUESTION2) and relevance ratio (in QUESTION3) judgments in our experiment design for gathering document relevance data for IR, the collected judgments are compared with TREC binary, Sormunen and ME judgments.

The collected answers of each question from different workers were aggregated and normalized into numeric score in the range $[0, 1]$. That is, there are three sets of judgments for each topic-document combination, generated by answers of three questions respectively.

The frequency that a document is preferred by workers in QUESTION1 (preference) is denoted as *preference frequency*. The upper bound of preference frequency for each topic is computed as $\boldsymbol{K} \times \boldsymbol{X} \times \boldsymbol{Y}$, where $\boldsymbol{K} = 3$, and $\boldsymbol{Y} = 3$ in our experiments (as $\boldsymbol{X}$ is diverse for different topics, the upper bound for each topic is distinct). The normalized preference frequency of each document is therefore computed as the fraction of the actual times that the document is preferred in the assessments divided by the $\boldsymbol{K} \times \boldsymbol{X} \times \boldsymbol{Y}$ upper bound frequency of the topic. For example, for topic 405, the number of partitioning is 11, that is, $\boldsymbol{X} = 11$. Each document was in 11 different groups, and paired with $\boldsymbol{K} = 3$ other documents in each group. As each pair was judged by $\boldsymbol{Y} = 3$ distinct workers, each document for topic 405 was totally presented $3 \times 11 \times 3 = 99$ times. If a document, for example, was preferred 6 of those times, its normalized preference frequency would be computed as $6/99 = 0.061$. The normalized relevance score of documents in the judgments of QUESTION2 were similarly computed.

The third question style required a more complex approach. For each topic, we converted and normalized the relevance ratios of document pairs, collected in QUESTION3, into numeric relevance scores for all documents using the following steps:

1. A ***NumDoc*** $\times$ ***NumDoc*** pairwise comparison matrix M recording relevance ratios between pairs of pooled documents is built. The cell $M_{i,j}$ stores the geometric mean of collected relevance ratios between the document i and j. The ratio is calculated by $(s_i+\epsilon)/(s_j+\epsilon)$ where s_i and s_j are non-negative numerical answers given by the assessor in QUESTION3, $\epsilon = 1$. As each document was not paired with every other pooled documents, the evidence collected is partial (and almost certainly inconsistent), the matrix is incomplete in the initial stage.

2. Calculate the geometric mean of column i as the relevance score S_i for document i. Repeat for every document.

3. For any two documents which were not paired in the experiment (say document u and v), the value of $M_{u,v}$ is filled by S_u/S_v, which have been computed in step 2.

4. Repeat step 2 and 3 until all document relevance scores are stable.

5. For each stable S_i, normalize the score by dividing the maximum obtained score S_{max}.

The optimization of the original incomplete matrix M via this approach has been proven to be solvable and to have a unique optimal solution using the *Logarithmic Least Squares Method* [8].

In the normalization process, we chose $\epsilon = 1$ in step 1; repeating steps 2 and 3 a fixed 100 times made the matrix stable (any S_i score change between iterations was less than 10^{-5}).

The normalized scores of all document-topic combinations, compared with Sormunen and Binary, are plotted in the bottom graph of Figure 5.5.

5.3.2 Relevance Frequencies and Scores

To answer the first research question, whether our judgments are similar to the previous judgments, for all topic-document combinations across nine TREC-8 topics, we compared document normalized scores in the three sets of judgments built by answers to the three questions with the Sormunen and Binary judgments. Figure 5.5 shows the normalized relevance scores of 1876 documents for nine tested topics, using the answers of each of three methods. In each sub-graph, the collected judgments are compared with two sets of judgments: Sormunen and Binary. Each topic-document combination has one colored circle in the left five columns and one in the right three columns. The x-axis shows the relevance categories that documents were classified using Sormunen (`H`, `M`, `R`, `N`) and

Binary (`1`, `0`) respectively. Documents which were not assessed by Sormunen or Binary are labeled as `U`.

In the top graph of the Figure 5.5, each circle represents a topic-document combination, whose score in our preference judgments is shown on the vertical axis, and the x-axis shows its relevance categories in Sormunen (left five boxes) and NIST Binary (right three boxes). As is shown in Figure 5.5, documents which are considered as relevant by Sormunen and Binary generally receive higher relevance scores in our three sets of pairwise judgments. Overall the medians of relevance scores align with the ordinal categories. The relevance distances between `H` and `M` in Sormunen are very close according to relevance score distributions of our judgments. Irrelevant documents are probably easy to be recognized and classified by workers, but ranking and scoring of relevant documents is more complicated.

As the boxes shown in Figure 5.5, QUESTION2 and QUESTION3 are good at recognizing non-relevant documents (medians of `N` and `0` are very low). Judgments generated by QUESTION3 do not perform well when ordering relevant documents considered by Sormunen, but judgments built by QUESTION1 and QUESTION2 usually have higher agreements with Sormunen when classifying top relevant documents. Note that relevance score distributions are be affected by the normalization processes for distinct question answers. In QUESTION1, documents ranked very low may still have scores of 0.4 on average, because workers were required to make a preference choice even when the two documents in the pair were both non-relevant.

As there are only two categories in Binary scale, and four in Sormunen, there would be a large number of ties when computing correlation coefficients for graphs in Figure 5.5. Thus, instead of measuring score correlations using Kendall's τ, we computed the agreements between judgments when ordering documents in pairs in Section 5.3.3, and when ranking systems in pairs in Section 5.3.4.

To compare the judgments of our methods with ME, as document relevance scores in all four of these schemes are numerical rather than categorical, we firstly compared their relevance score distributions. For each set of judgments collected using pairwise preference (QUESTION1), absolute relevance (QUESTION2), pairwise ratio (QUESTION3) and Magnitude Estimation, we sorted documents by their relevance scores in decreasing order. Figure 5.6 illustrates the distributions of relevance scores collected by these four methods and for nine TREC-8 topics respectively. In each sub-figure, the score distribution of each methods is shown in different colors (note that, the document orderings given by four judgments are different). The normalized relevance scores in our judgments are within score range $[0, 1]$, represented by the left y-axis in each sub-figure. The score range of ME is $[0, 19]$, represented by the right y-axis. The relevance scores in ME judgments are not normalized into score range $[0, 1]$, and thus y-axes for ME and our judgments are different. As the score range and distribution of ME judgments are distinct for each topic, which may depend on the topic difficulty and assessors, dividing ME relevance scores

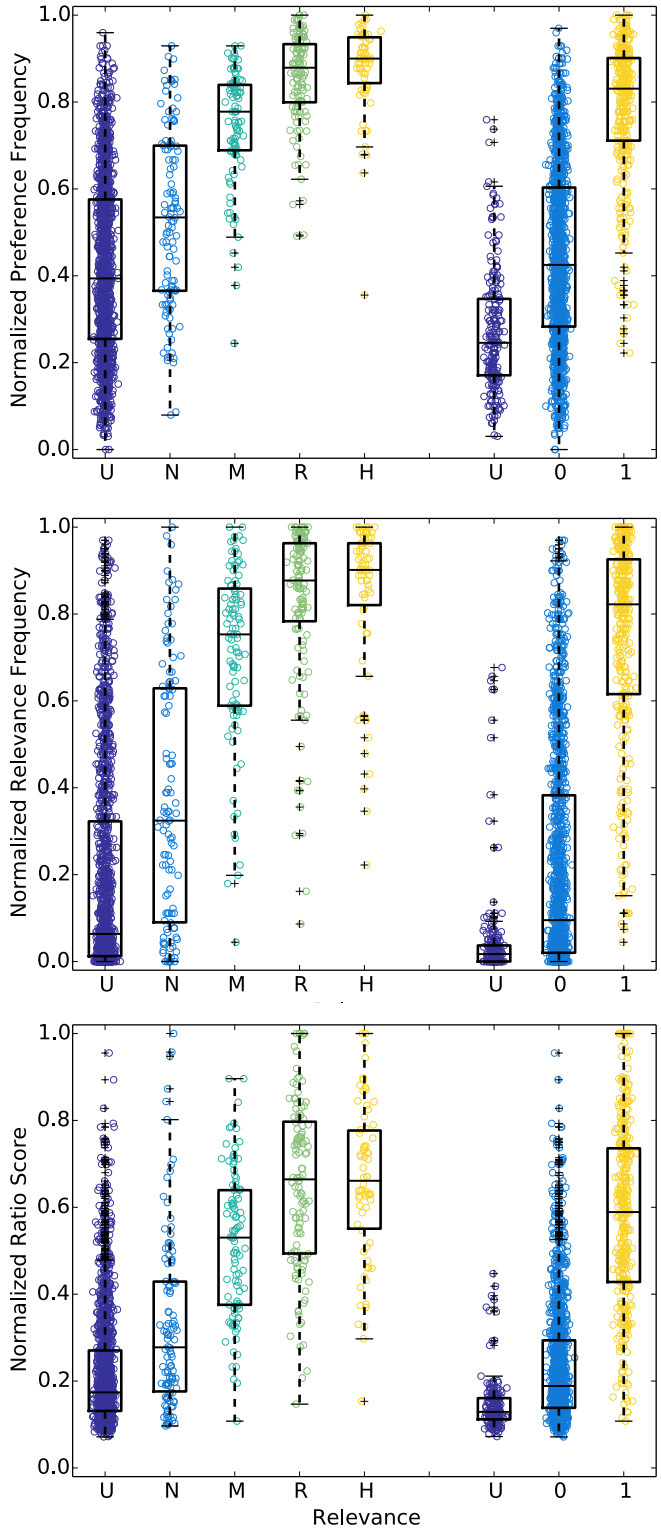


Figure 5.5: Normalized judgment frequencies and scores over nine TREC-8 topics collected using three methods: pairwise preference (top), absolute relevance (middle) and relevance ratio (bottom), compared with relevance labels of Sormunen (left five columns) and NIST Binary (right three columns). Each document-topic combination is represented as a circle the whisker box. Documents having no judgments in Sormunen or NIST Binary are categorized into column `U` (for *unjudged*).

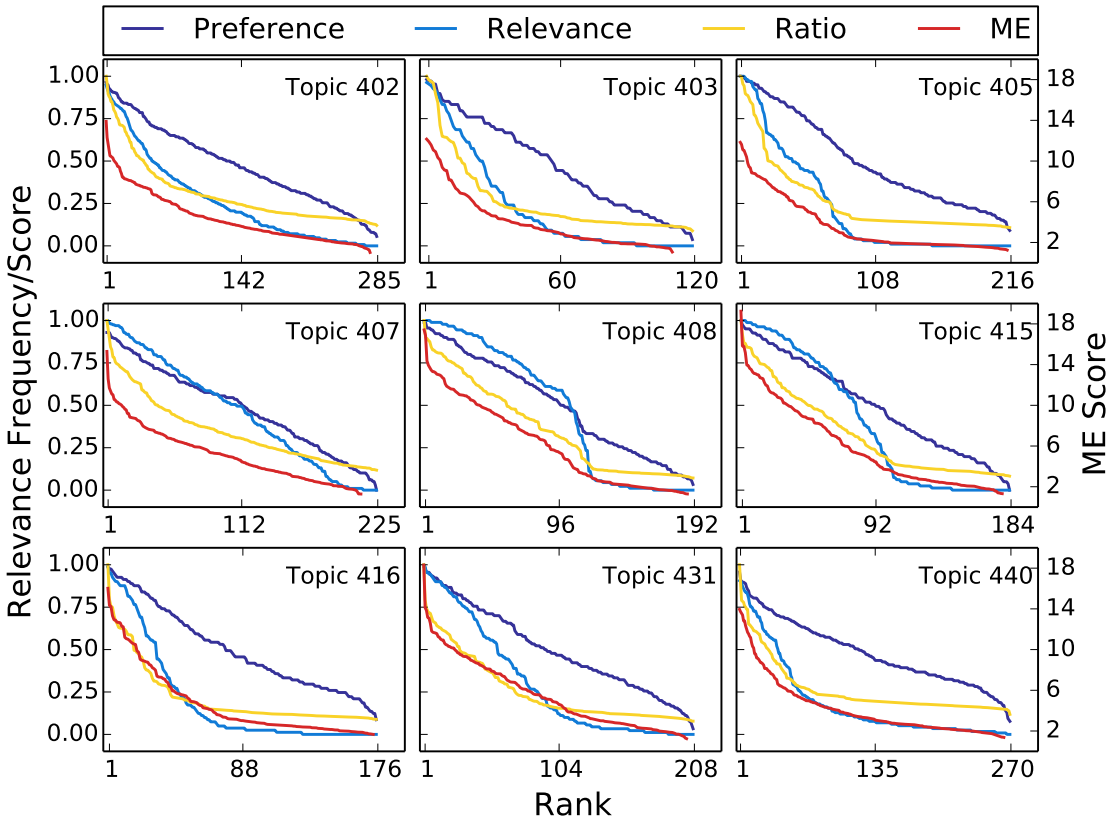


Figure 5.6: The distribution of document relevance scores, collected using methods of pairwise preference (dark purple), absolute relevance (blue), pairwise ratio (yellow) and Magnitude Estimation (red), as a function of document ranks on relevance. For each of nine topics, documents are sorted by relevance scores given by each of four methods in decreasing order. The left vertical axis is shared by preference, absolute relevance and ratio scores, the right y-axis is for ME scores.

by the maximum score in each topic is not a fair approach. If we normalize ME relevance scores using a *magic* number which is larger than relevance scores of all the tested topics, if new topics join into the test, the ME judgments need to be re-normalized, and scores may therefore change. Thus, ME relevance scores are illustrated as they are.

As Figure 5.6 shows, relevance levels and document number in each level of different topics are distinct. Some topics, such as 408 (ME score average 6.10) and 415 (ME score average 5.90), have more relevant documents in the pool, but document relevance scores of topics such as 403 (ME score average 4.38) and 405 (score average 4.01) are generally smaller. The shape of blue lines for QUESTION2 judgments also indicates that topic 408 and 415 have more highly relevant documents in top ranks.

In our QUESTION3, workers directly assigned relevance ratios for randomly generated document pairs. In ME, from the second document, each document was compared with

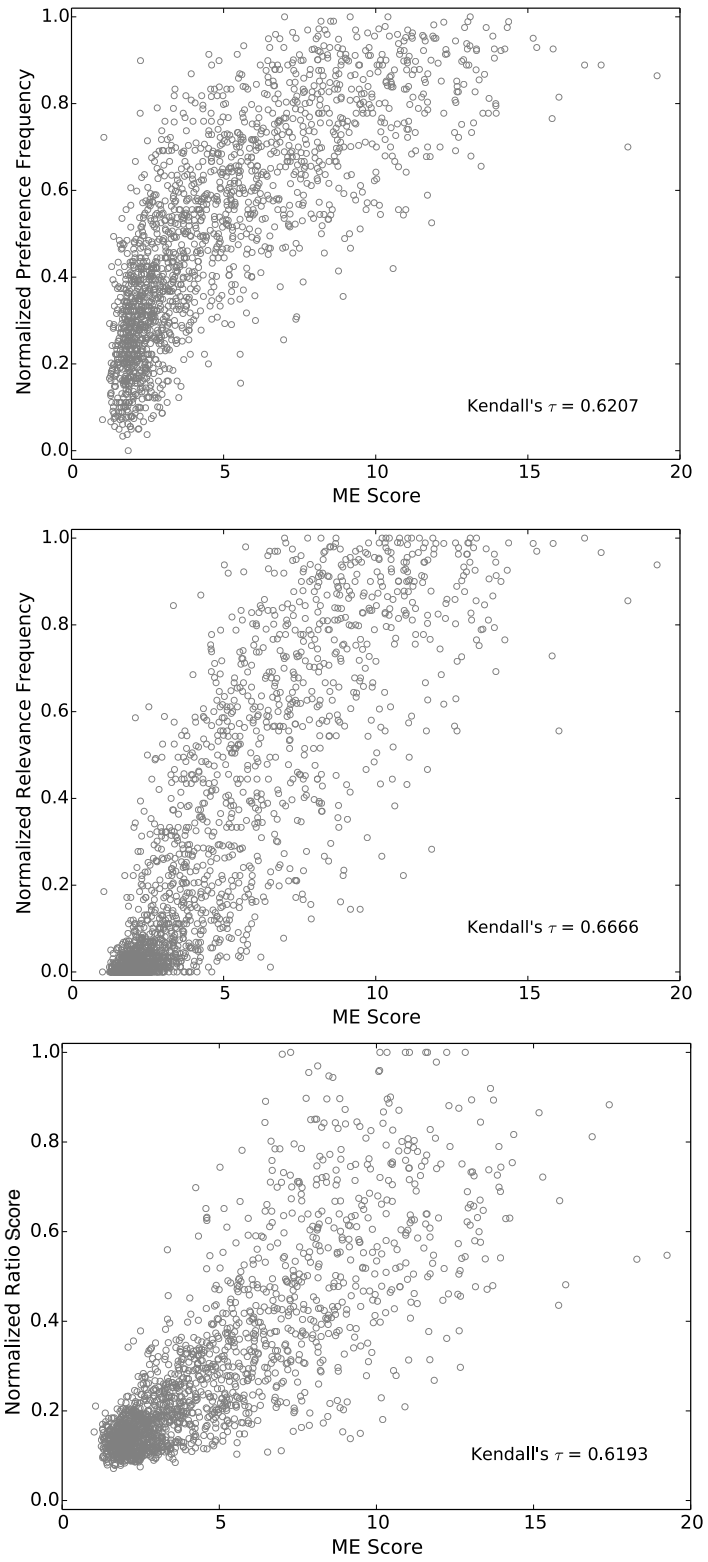


Figure 5.7: Normalized relevance scores of nine topics, collected using pairwise preference (top), absolute relevance (middle) and relevance ratio (bottom), compared with scores in Magnitude Estimation judgments.

the previous one. Relevance judgments collected by both of these two methods were based on pairwise comparisons and relevance ratios. Thus for each topic shown in Figure 5.6, the yellow lines (for QUESTION3) are the most similar to the ME lines.

By looking into the relevance score distribution of each question in Figure 5.6, we found that the ability to distinguish documents on relevance using QUESTION1 judgments was almost equal across all the ranks. The scores of irrelevant and slightly relevant documents in QUESTION2 judgments were extremely close, and only the highly relevant documents could be clearly ranked by relevance. The judgments of QUESTION3 could show relevance distance between any two documents in any relevance level, and the relevance score distribution here was the one that was the most similar to that of the ME judgments.

To compare the values of relevance scores in our judgments and ME for each topic-document combination, Figure 5.7 illustrates the numeric relevance scores of all topic-document combinations in judgments of QUESTION1, QUESTION2 and QUESTION3, compared to the scores assigned to the same documents by ME. The Kendall's τ score of each set of crowd-sourced judgments and ME is shown in the lower right corner of each subgraph. The relevance scores generated from answers of QUESTION2 are the ones most correlated to ME, but the difference between the three is small.

5.3.3 Document Orderings

To answer the first research question, we compared judgments collected using our methods with NIST Binary, Sormunen, and ME by exploring the relevance score distributions over nine topics in Section 5.3.2. As noted in Section 2.4.3, rank correlation measures such as Kendall's τ and Spearman's ρ are affected by the mapping function for combining relevance levels of judgments. That is, Kendall's τ and Spearman's ρ which directly take document relevance scores in compared judgments may be affected by the parameters used in the normalization process for converting pairwise judgments into numeric relevance scores. Thus, in this section, we compute the agreement between different judgments when ordering documents in pairs based on the relevance scores of the paired documents in the judgments.

To compare the judgments of QUESTION1, QUESTION2, and QUESTION3 with the NIST Binary judgments in terms of document ordering, for every unordered $\{0, 1\}$ documents pair (in which one document is relevant, and another is irrelevant according to the Binary judgments), their normalized relevance scores in the pairwise judgments are compared. The pairwise judgments are deemed as "agree" with Binary judgments if the normalized relevance score of the relevant document (whose score is 1 in Binary) is higher than the score of the paired irrelevant document. If the relevance order of documents in a pair is discordant in Binary and pairwise judgments, judgments will be deemed as "disagree" on this pair. If the relevance scores of the paired documents are equal in the pairwise judgments, but different in Binary, the document pair will be deemed as a "tie". Note that this evaluation is only considering document pairs that have different relevance values

Topic	QUESTION1			QUESTION2			QUESTION3		
	agree	disagree	tie	agree	disagree	tie	agree	disagree	tie
402	0.901	0.094	0.005	0.903	0.093	0.003	0.908	0.092	0.000
403	0.939	0.048	0.013	0.984	0.013	0.003	0.991	0.009	0.000
405	0.949	0.046	0.005	0.939	0.057	0.003	0.941	0.059	0.000
407	0.841	0.152	0.007	0.848	0.147	0.006	0.859	0.141	0.000
408	0.923	0.071	0.007	0.931	0.063	0.006	0.925	0.075	0.000
415	0.953	0.041	0.005	0.963	0.033	0.004	0.952	0.048	0.000
416	0.935	0.059	0.006	0.950	0.047	0.003	0.936	0.064	0.000
431	0.847	0.146	0.007	0.921	0.076	0.003	0.904	0.096	0.000
440	0.901	0.095	0.003	0.922	0.076	0.002	0.924	0.076	0.000

Table 5.5: Agreement of NIST Binary and judgments of preference (QUESTION1), absolute relevance (QUESTION2), and relevance ratio (QUESTION3) on relevance ordering of $\{0, 1\}$ (unordered) document pairs in which one document is judged as relevant (1) by Binary, and another is irrelevant (0). If the relevance order of the paired documents is the same according to their relevance scores in Binary and pairwise judgments, judgments will be deemed as "agree" on this pair. The fractions of $\{0, 1\}$ pairs that Binary and pairwise judgments agree on the relevance order of the paired documents are shown in the "agree" column of each set of pairwise judgments. The "disagree" columns show the fractions of pairs in which the normalized relevance score of relevant document is lower than irrelevant document in pairwise judgments. If the paired documents have the same relevance score in pairwise judgments, it will be deemed as a "tie".

in the NIST Binary qrels files. Table 5.5 shows the fractions of $\{0, 1\}$ document pairs that pairwise judgments of QUESTION1, QUESTION2, and QUESTION3 agree, disagree, and tied on their relevance order in Binary judgments for nine tested topics respectively.

Table 5.5 shows that, for all $\{0, 1\}$ documents pairs, the pairwise judgments of the three methods agree with NIST Binary judgments on document relevance ordering in most pairs. The average agreement with Binary is 0.910 for judgments of QUESTION1, 0.929 for judgments of QUESTION2, and 0.927 for judgments of QUESTION3. As can be seen, QUESTION2 has the highest agreement with Binary judgments on $\{0, 1\}$ pairs. The judgments of QUESTION3 also agree in a high level, and more than judgments of QUESTION1. It may be caused by ties in preference judgments. The ties neither agree nor disagree with Binary judgments. The tie fraction of QUESTION1 is higher than QUESTION2 and QUESTION3 for each topic, but its average is very low, under 0.010. Meanwhile, scores of the paired documents are never tied in relevance ratio judgments (QUESTION3).

For the document pairs that NIST Binary cannot separate in terms of relevance, that is, the pairs of $\{0, 0\}$ and $\{1, 1\}$, the percentages of these pairs in which documents have different relevance scores in pairwise judgments are summarized in Table 5.6. As can be seen, as most documents judged by NIST Binary are non-relevant, the number of $\{0, 0\}$ pairs is usually much greater than the number of $\{1, 1\}$ pairs for each of the tested nine topics. The preference judgments (QUESTION1) can separate over 95% of pairs that are tied in Binary judgments. For $\{1, 1\}$ pairs, the discriminations of preference judgments and absolute relevance judgments (QUESTION2) are at a comparable level, but for $\{0, 0\}$ pairs, judgments of QUESTION2 never perform better than judgments of QUESTION1. The percentages of $\{0, 0\}$ pairs that can be distinguished by preference judgments (on average 98.5%) are generally higher than the percentages of $\{1, 1\}$ that are separable in preference judgments (on average 97.1%). However, documents in 6.4% (averaged over topics) of $\{0, 0\}$ pairs are still tied in judgments of QUESTION2. The average tie rate is only 2.2% when using judgments of QUESTION2 to separate documents in $\{1, 1\}$ pairs. The discrimination of relevance ratio judgments for distinguishing all kinds of document pairs is the best in pairwise judgments. As it turns out, there is no document that has the same normalized relevance score with other documents in our relevance ratio (QUESTION3) judgments. This is not a guaranteed nature of the iteration process, but it is a likely outcome because of the holistic nature of the computation.

To further quantify the relationships shown in Figure 5.5 and Figure 5.7, and analyze the combined results of Table 5.5 and Table 5.6, the fractions of pairs that any two judgment qrels files agree ("A"), disagree ("D") and tie ("U") on relative document order are shown in Table 5.7. For example, when calculating the agreement of Binary and Sormunen, for a document pair $\{\boldsymbol{d}_1, \boldsymbol{d}_2\}$, if the relevance score of one document is higher than score of another in both of Sormunen and Binary judgments, the relevance ordering of $\{\boldsymbol{d}_1, \boldsymbol{d}_2\}$ will be deemed as agreed ("A") by two compared judgments. If their relevance orderings in two judgments conflicts, it will be deemed as a disagree ("D") pair.

Topic	{1,1}	QUESTION1		QUESTION2		QUESTION3	
		separable	tied	separable	tied	separable	tied
402	861	0.985	0.015	0.999	0.001	1.000	0.000
403	190	0.953	0.047	0.989	0.011	1.000	0.000
405	378	0.960	0.040	0.974	0.026	1.000	0.000
407	703	0.984	0.016	0.984	0.016	1.000	0.000
408	1711	0.976	0.024	0.954	0.046	1.000	0.000
415	1035	0.963	0.037	0.955	0.045	1.000	0.000
416	406	0.966	0.034	0.975	0.025	1.000	0.000
431	2080	0.979	0.021	0.987	0.013	1.000	0.000
440	253	0.976	0.024	0.988	0.012	1.000	0.000

(a)

Topic	{0,0}	QUESTION1		QUESTION2		QUESTION3	
		separable	tied	separable	tied	separable	tied
402	23 871	0.990	0.010	0.984	0.016	1.000	0.000
403	2850	0.974	0.026	0.924	0.076	1.000	0.000
405	13 530	0.986	0.014	0.872	0.128	1.000	0.000
407	6786	0.987	0.013	0.987	0.013	1.000	0.000
408	6786	0.987	0.013	0.949	0.051	1.000	0.000
415	6670	0.986	0.014	0.931	0.069	1.000	0.000
416	6786	0.984	0.016	0.878	0.122	1.000	0.000
431	7381	0.986	0.014	0.940	0.060	1.000	0.000
440	24 976	0.985	0.015	0.963	0.037	1.000	0.000

(b)

Table 5.6: Percentages of (a) $\{1,1\}$ pairs and (b) $\{0,0\}$ pairs in which documents cannot be distinguished by Binary judgments in terms of relevance, that can be separated by pairwise judgments (shown in the column of "separable" for each of the pairwise judgments: preference (QUESTION1), absolute relevance (QUESTION2), and relevance ratio (QUESTION3), and that documents are "tied" in both compared judgments. For each of nine topics, the second column shows the total number of (a) $\{1,1\}$, or (b) $\{0,0\}$ pairs, based on NIST Binary judgments.

Judgments	Sormunen			ME			Pref			Rele			Ratio		
	A	D	U	A	D	U	A	D	U	A	D	U	A	D	U
Binary	.334	.005	.661	.301	.030	.669	.300	.028	.672	.307	.022	.671	.306	.024	.670
Sormunen		–		.522	.181	.297	.579	.112	.309	.561	.129	.309	.551	.152	.297
ME					–		.810	.177	.013	.796	.152	.052	.821	.179	.000
Pref								–		.824	.109	.067	.867	.119	.013
Rele											–		.858	.087	.055

Table 5.7: Judgment agreements of relative document ordering in pairs over nine topics. Documents assessed by both compared judgments sets were paired and for each of them, if both judgments indicate that one is more relevant than the other in the pair, judgments will be deemed as agree (A) on this pair, otherwise disagree (D). If either judgment set has the same relevance scores for the paired documents, that is a relevance score tie, the pair will be counted as unknown (U).

If the relevance scores of $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are the same in at least one of Sormunen and Binary, the pair will be counted as a tie ("U").

In Table 5.7, any two of six judgments are compared, and the fractions of pairs in each of three categories are computed. When the judgments are compared with Binary (shown in the first row of Table 5.7), the numbers of pairs in "U" are generally quite high because Binary has only two relevance categories (`0` and `1`) thus it gives a lot of ties. Agreements within our crowd-sourced judgments of QUESTION1 (Pref), QUESTION2 (Rele) and QUESTION3 (Ratio) are very high, and they also greatly correlated with judgments collected using ME. As the relevance score fidelities of our judgments and ME are higher than Sormunen and Binary, the tie rates resulted form comparisons between these four judgments are much lower, that is judgments collected by our methods and ME can distinguish more documents on relevance than Sormunen and Binary.

In QUESTION3, workers were required to provide the relevance ratio of the paired documents by assigning a positive number for each document. For example, for the document pair $\{\boldsymbol{d}_1, \boldsymbol{d}_2\}$, the worker may assign 10 to $\boldsymbol{d}_1$ and 1 for $\boldsymbol{d}_2$ and so the relevance ratio is $10/1 = 10$. Although workers could use any positive numbers they liked to express ratios, it is also interesting to examine the absolute values of numbers they entered for documents in different relevance levels. To explore this, we explored the raw scores of documents, used by workers in the interface to indicate relevance ratios between documents in QUESTION3. Documents in our judgments can be in categories of relevant (`1`), irrelevant (`0`) or unjudged (outside the pool) in Binary judgments. In Figure 5.8, for each topic, the raw scores of documents in each category were averaged into a single value and shown as the circle beside the whisker box.

As the pattern shows, the raw scores of documents which were not judged in NIST Binary are generally the lowest, which suggests that the pooling process for building the NIST Binary qrels files was effective. For different topics, the average raw scores of documents in the same category are also distinct, and the variance becomes greater for categories in higher relevance levels. But generally, documents judged as relevant (`1`) in Binary received higher raw scores than irrelevant documents (`0`).

Overall, the results presented in this subsection indicate that judgments generated by answers of three questions all have high agreements to previous judgments in terms of scores and orderings of documents in pairs. Moreover, we repeated our experiments of randomly generating pair lists and collecting answers of three questions for each document pair on `Figure8` for four (405, 407, 408, and 415) of the tested nine topics. The obtained similar results indicate that our methodologies are repeatable, and the conclusions made do not depend on `Figure8` workers, or the date when crowd jobs launch.

5.3.4 System Rankings

To investigate whether the results of system evaluations using our crowd-sourced judgments and previous judgments are similar, we computed the agreements of different

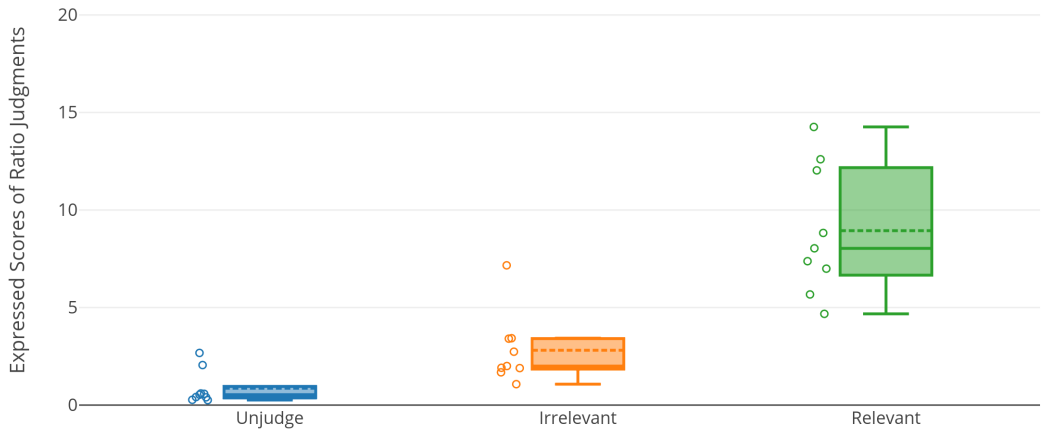


Figure 5.8: The average raw scores inputed by workers to express relevance ratios of documents in QUESTION3, compared with relevance categories that documents were classified in NIST Binary judgments over nine topics. The raw scores are averaged for each topic, shown as circles in the left hand side of whisker boxes. The dashed line in each box is the average of nine circles in each category, and the solid line is the median.

judgments on system orderings, shown in Table 5.8. Table 5.8 summarizes the agreement ("A"), disagreement ("D") and tie rates ("U") on orderings of pairs of systems (evaluated by RBP with $\phi = 0.9$) using each possible combination of available relevance judgments. Similar to the results of comparing judgments in document ordering, judgments of QUESTION1, QUESTION2, QUESTION3 and ME have high correlations with each other. Binary and Sormunen also greatly agree with each other.

The RBP scores of systems using different judgments are illustrated in Figure 5.9. The Kendall's τ of dots (systems) is shown in the lower right corner of each sub-graph. The patterns agree with the conclusions made in connection with Table 5.8. As can be seen, judgments in ordinal scales (Binary and Sormunen) have high agreement on ordering of top systems, and so do the ME and crowd-sourced judgments. The correlations of top system rankings given by categorical and numerical judgments respectively are generally not large.

In Figure 5.10, for each topic, judgments having numeric relevance scores (using methodologies of ME and those associated QUESTION1, QUESTION2 and QUESTION3) were mapped to Sormunen categories according to proportions of documents in each category in Sormunen judgments. For each topic $\boldsymbol{t}$, denote the proportions of documents in each category of Sormunen are: $\boldsymbol{P}_{H,t}, \boldsymbol{P}_{R,t}, \boldsymbol{P}_{M,t}$ and $\boldsymbol{P}_{N,t}$. If there are $\boldsymbol{NumDocs}_t$ documents in ME judgments, then the most $\boldsymbol{NumDocs}_t \times \boldsymbol{P}_{H,t}$ relevant documents in ME will be mapped and classified into category H. A Linear gain function was then applied to all judgments to evaluate systems.

Judgments	Binary			Sormunen			ME			Pref			Rele		
	A	D	U	A	D	U	A	D	U	A	D	U	A	D	U
Binary		–													
Sormunen	.938	.062	.000		–										
ME	.898	.102	.000	.873	.127	.000		–							
Pref	.874	.126	.000	.858	.142	.000	.945	.055	.000		–				
Rele	.886	.114	.000	.858	.142	.000	.951	.049	.000	.955	.045	.000		–	
Ratio	.887	.113	.000	.863	.137	.000	.942	.058	.000	.956	.044	.000	.970	.030	.000

Table 5.8: Judgment agreements of relative system ordering in pairs over nine topics. For any two compared judgments, their agreement (A), disagreement (D) and the tie rate (U) on TREC-8 systems ordering in pairs ($123 \times 122/2 = 7503$ system pairs), based on system scores evaluated by RBP ($\phi = 0.9$) across nine topics using the compared judgment qrels files respectively, are computed.

Comparing Figure 5.9 and Figure 5.10, when numeric relevance scores in judgments are mapped to categories based on relevance distributions of categorical judgments, system scores evaluated by Sormunen and the mapped judgments (ME, preference, absolute relevance and ratio) are more correlated than before. The Kendall's τ values of system scores given by Sormunen-based mapped judgments and Binary become greater as well, which is probably because Binary are originally more correlated with Sormenen than numerical judgments (before mapping). As the first row of Figure 5.9 and Figure 5.10 show, the correlation of system rankings evaluated using numerical judgments is stronger (even better than the correlation between Sormunen and Binary) than with the mapping. Thus, we suggest to apply category mapping to numerical judgments only when aiming to build a set of judgments that is similar to the categorical judgments. Otherwise, it would be preferred to keep the high fidelity of the numerical judgments.

After comparing judgments in terms of system rankings, we found that although the categorical judgments (Binary and Sormunen) are self-correlated well and their agreements with numerical judgments (ME, preference, absolute relevance and ratio) are lower, the overall agreements of six sets of judgments are still in high levels. Thus we concludes that our crowd-sourced judgments are similar to previous judgments and valid to be used by evaluation metrics to examine the performance of IR systems.

5.3.5 Workers

Inconsistency We also explored the inconsistency of crowd-workers when judging the relevance of documents in QUESTION1 (which of QUESTION3 is exactly the same) and QUESTION2.

Worker's consistency in QUESTION1 was measured by the transitivity of the preferences made: a list of eight documents in a group would be generated to maximize the agreement with preference judgments given by the worker, the proportion of documents related to "bad" judgments (disagree with the built list) counted. For example, denote $\boldsymbol{d}_1 \succ \boldsymbol{d}_2$ as $\boldsymbol{d}_1$ is preferred by the worker as more relevant than $\boldsymbol{d}_2$. The worker made preference choices $\boldsymbol{d}_1 \succ \boldsymbol{d}_2$, $\boldsymbol{d}_2 \succ \boldsymbol{d}_3$, $\boldsymbol{d}_3 \succ \boldsymbol{d}_4$ and $\boldsymbol{d}_4 \succ \boldsymbol{d}_1$ for four document pairs. As the worker made a intransitive decision, one possible list generated from these preference judgments containing documents in decreasing relevance order can be: $\boldsymbol{d}_1 \succ \boldsymbol{d}_2 \succ \boldsymbol{d}_3 \succ \boldsymbol{d}_4$. The list disagrees with the preference judgment $\boldsymbol{d}_4 \succ \boldsymbol{d}_1$ and two out of four documents ($\boldsymbol{d}_1$ and $\boldsymbol{d}_4$) are affected. Thus the inconsistency of this worker is computed as $2/4 = 0.5$.

The top graph in Figure 5.11 illustrates the QUESTION1 inconsistency of worker across the nine topics. The QUESTION1 inconsistency score (y-value of dot) has a relatively weak relationship (Kendall's $\tau = 0.214$) with the number of documents assessed by the worker.

As each document was paired with three other documents in the same group, each document would receive answers of QUESTION2 three times from the same worker. The worker would be deemed as making one inconsistent judgment if they gave opposite

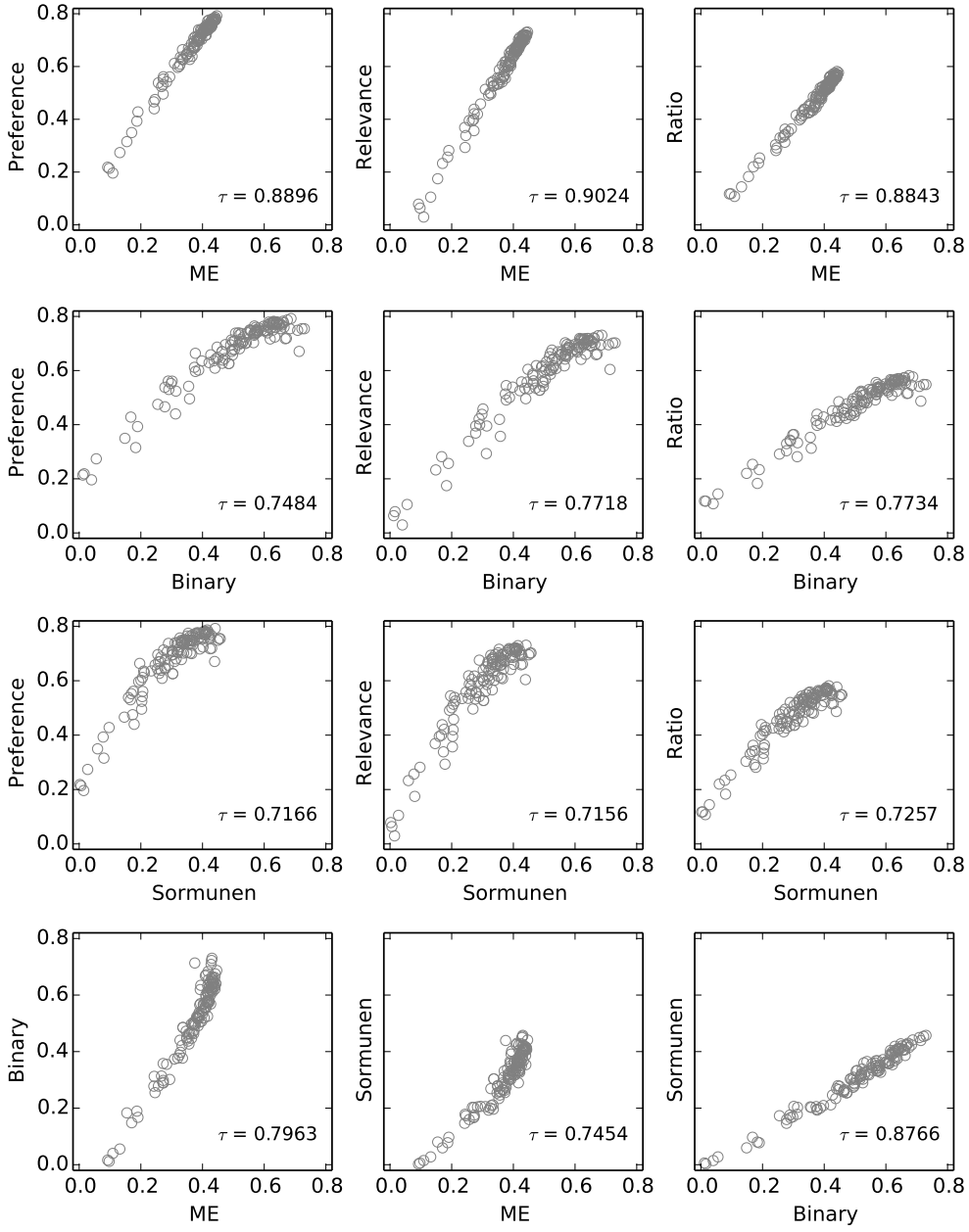


Figure 5.9: TREC-8 system scores evaluated by RBP ($\phi = 0.9$) using TREC Binary, Sormunen, ME and crowdsourced judgments. The Kendall's τ of system scores in each sub-graph is shown in the lower right corner.

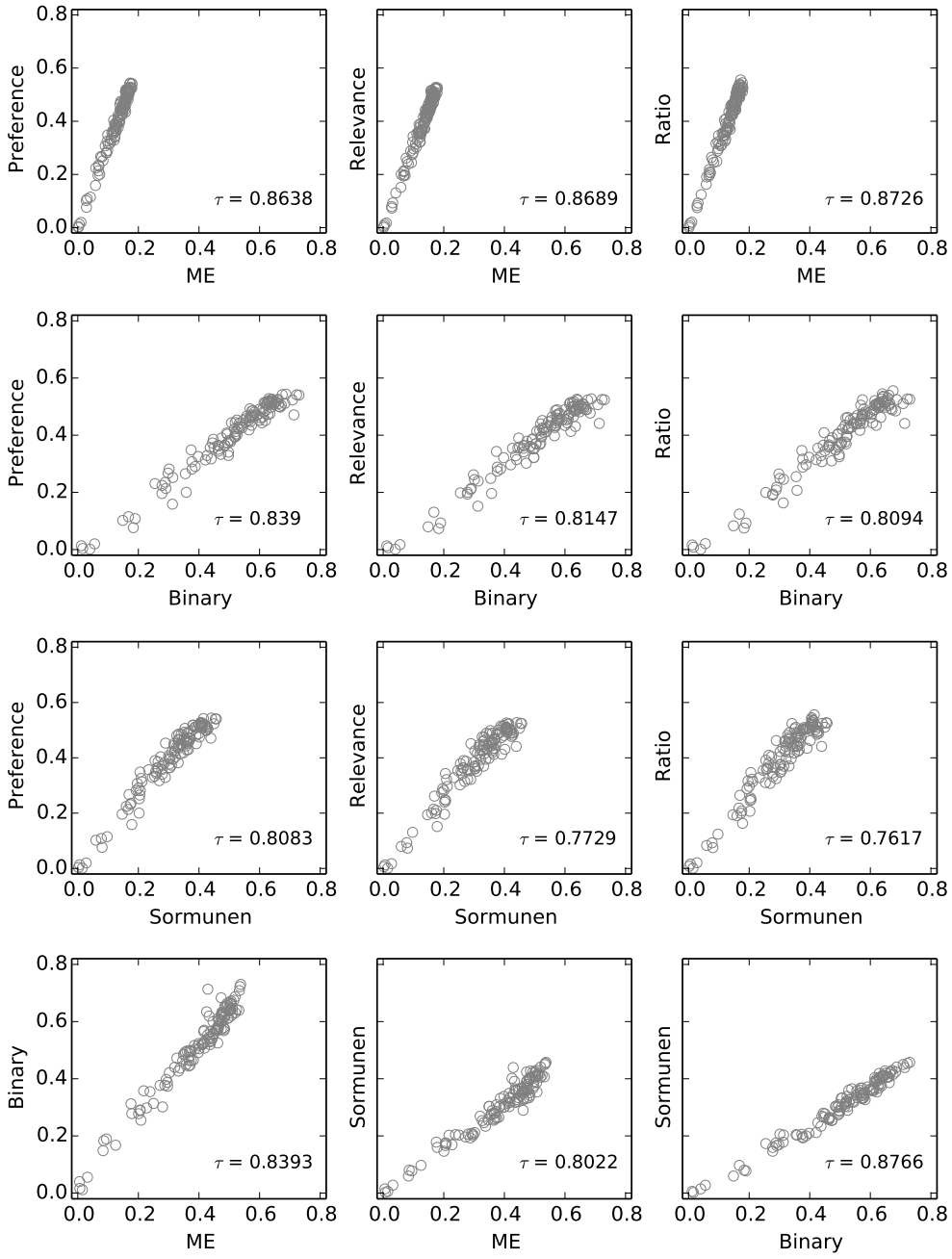


Figure 5.10: TREC-8 system scores evaluated by RBP ($\phi = 0.9$) using categorical judgments: TREC Binary, Sormunen, mapped-ME and mapped crowd-sourced judgments. The ME, preference, absolute relevance and ratio judgments were firstly mapped to Sormunen categories (`H`, `R`, `M` and `N`) according to proportions of documents in each category and topic.

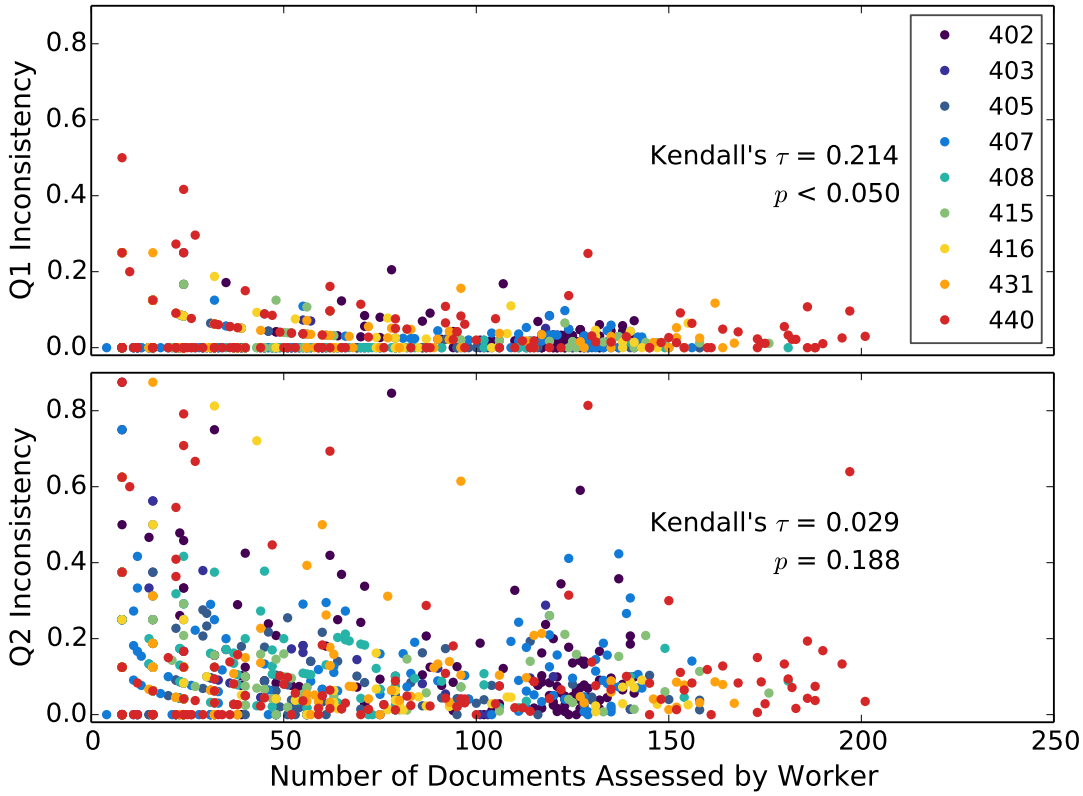


Figure 5.11: Fraction of documents affected by intransitive preference judgments in QUESTION1 (top); and receiving at least one inconsistent judgment from assessors in QUESTION2 (bottom). Each dot in the graph represents an assessor, whose x-value is the total number of documents that this assessor judged for the topic, with topics shown in distinct colors. The low Kendall's τ values of all dots in each graph indicate that the assessment consistency of workers has almost no relationship with the total number of documents judged by the worker. On average, each worker judged 60.2 documents for each topic that they contributed to with an inconsistency score of 0.02 in QUESTION1, and of 0.11 in QUESTION2 (related paired t-test, $p < 0.0001$).

answers for a document in QUESTION2. For each topic and each worker, the fraction of judged documents receiving inconsistent answers in QUESTION2, as a function of the total number of documents judged by the worker, is plotted in the bottom graph of Figure 5.11. The average QUESTION2 inconsistency of all topic-worker combinations is 0.11, higher than QUESTION1. The Kendall's $\tau = 0.029$ states that the inconsistency of worker's assessment of absolute assessment may have no significant relationship with the number of documents assessed by the worker.

We also used generalized linear mixed model (GLMM, see Section 2.4.4) to test if other factors, such as assessing time and topics, affected the consistency of workers. However, they all seemed to have no significant relationship with worker's consistency.

Topic	Q1	Q2	Q3	Workers	AP	Accuracy (std.)
402	.397	**.635**	.217	152	0.204	87% (20%)
403	**.716**	.423	.245	47	0.708	89% (27%)
405	.493	**.584**	.210	126	0.182	94% (11%)
407	.530	**.573**	.324	103	0.279	83% (21%)
408	**.569**	.531	.188	125	0.198	91% (20%)
415	.456	**.529**	.223	72	0.400	89% (18%)
416	**.482**	.466	.294	74	0.295	88% (22%)
431	.513	**.549**	.240	79	0.329	81% (19%)
440	.408	**.642**	.328	133	0.154	76% (28%)

Table 5.9: The results of the survey that workers completed when they finished the assessment of a unit, about which method (more than one could be chosen) they prefered to use for judging document relevance in pairs. The average rates that methods were liked by workers were shown in column 2 to 4. The bold score is the greatest one of three for each topic. The "Workers" column shows the number of workers who contributed to the assessment of the topic. The "AP" column includes the average of TREC-8 systems' mean AP scores using binary judgments, which may indicate the difficulty of the topic. The last column shows the mean and standard deviation of workers' gold standard accuracies.

Accuracy As each group contains a gold standard pair which is constructed by a HR and a NR, we can report the proportion that HR is preferred than NR in QUESTION1 of each worker. This result is shown in the last column of Table 5.9 and states that the assessment accuracies for different topics are varied but generally very high. Note we did not exclude groups when NR is preferred to HR in our analysis.

Preference After completing the assessment of a group, workers were asked to complete a survey question about which method(s) they liked. As a worker might examine several groups and complete the survey for each of them, answers of each worker were averaged by the total number of group(s) that the worker completed. For example, if a worker completed 10 groups and voted QUESTION1 in all of the 10 surveys, QUESTION2 5 times and QUESTION3 5 times, the raw results would be normalized to 1.0 (QUESTION1), 0.5 (QUESTION2) and 0.5 (QUESTION3) respectively. The preference rates of each method shown in each row of Table 5.9 are the macro average across all workers who contributed to the assessment of the given topic. Thus the opinion of each worker across the groups and topics they assessed contributed equally to the preference rates shown in columns 2 to 4 in Table 5.9.

As Table 5.9 describes, workers liked methods of preference (QUESTION1) and absolute relevance (QUESTION2) better than assigning relevance ratio (QUESTION3). It is probably because QUESTION3 is harder and requires longer time for thinking and typing (rather

than clicking only in QUESTION1 and QUESTION2) to complete. Worker's preference between QUESTION1 and QUESTION2 significantly depended on the number of groups that the worker assessed ($p = 0.013$ in a GLMM). Workers who completed more groups of assessments preferred QUESTION2 to QUESTION1. Moreover, QUESTION1 tended to be preferred in easier topics ($p = 0.002$ in GLMM), where the difficulty of topics is defined by the mean AP score of TREC-8 systems [26]. The higher the mean of system AP scores is, the easier the topic is.

Numbers of Documents and Workers To ensure that we had enough distinct assessors contributing to the crowd-sourced judgments, we computed the number of workers and their average workload. In total, there were 806 different workers contributed to the assessments of nine topics, and 112 of them completed the assessment of only one group. The most prolific worker provided judgments for 32 groups.

In Figure 5.12, each sub-graph (for each topic) illustrates the number of workers who judged each document (in horizontal axis) and the number of such documents (in vertical axis) for the topic. As the Figure 5.12 shows, most documents received judgments (in different pairs) from multiple assessors, thus the generated relevance judgments are unlikely to be worker-biased.

As stated in Section 5.1.1, the number of pooled documents for each topic is different and so the number of partitions X shown in each sub-graph is varied. Topics with greater pool sizes have more document pairs to be judged (so we can ensure that each document is compared with about 15% other pooled documents of the topic), thus they were generally assessed by more workers. As expected, for each of nine topics, the number of assessors for single documents shows similar normal distribution. Most documents received judgments from many different workers.

Figure 5.13 illustrates the number of documents judged by each worker. In each sub-graph (for one topic), each bar shows the number of documents assessed by the worker and the color indicates how many different times documents were judged by this worker. Workers are sorted by the total number of assessments. As nine sub-graphs in Figure 5.13 show, workers who judged more documents in total have higher probabilities to see previously judged documents again. Topics with greater number of pooled documents (and so larger X) involve more workers in total to complete the assessment. Most workers completed the assessment for more than one group (bar heights above eight) in our experiment.

Time to Judge To compare the costs (payment rate) of judgments collected using different scales, we computed the average time that worker spent on each document in our experiments.

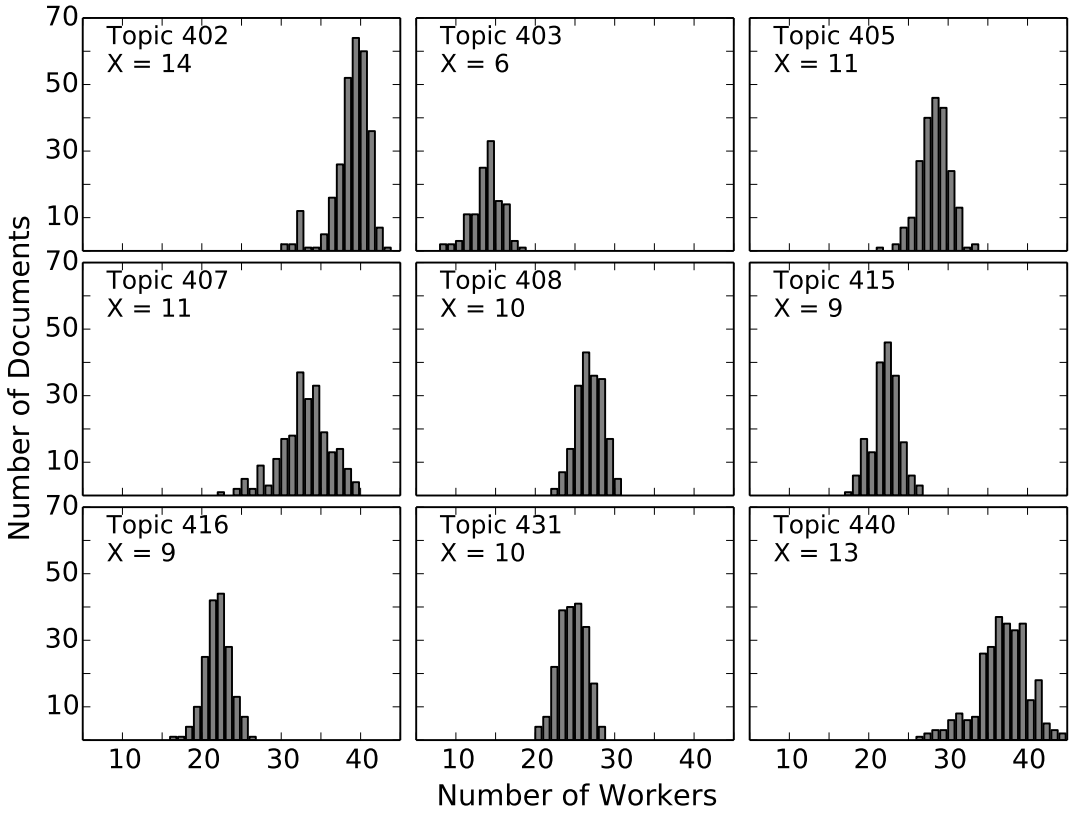


Figure 5.12: The number of documents judged by different workers. Each sub-graph is for one topic (the topic number and the number of partitions are shown in the upper left corner of each sub-graph). The x-axis shows the number of distinct workers who judged the same document and the y-axis shows the number of such documents. The nine sub-graphs share both x-axis and y-axis scales. Note that the number of pooled documents for each topic is different. To ensure the fidelity of the final normalized judgments, topics with more documents require greater number of partitions, and so viewed by more workers.

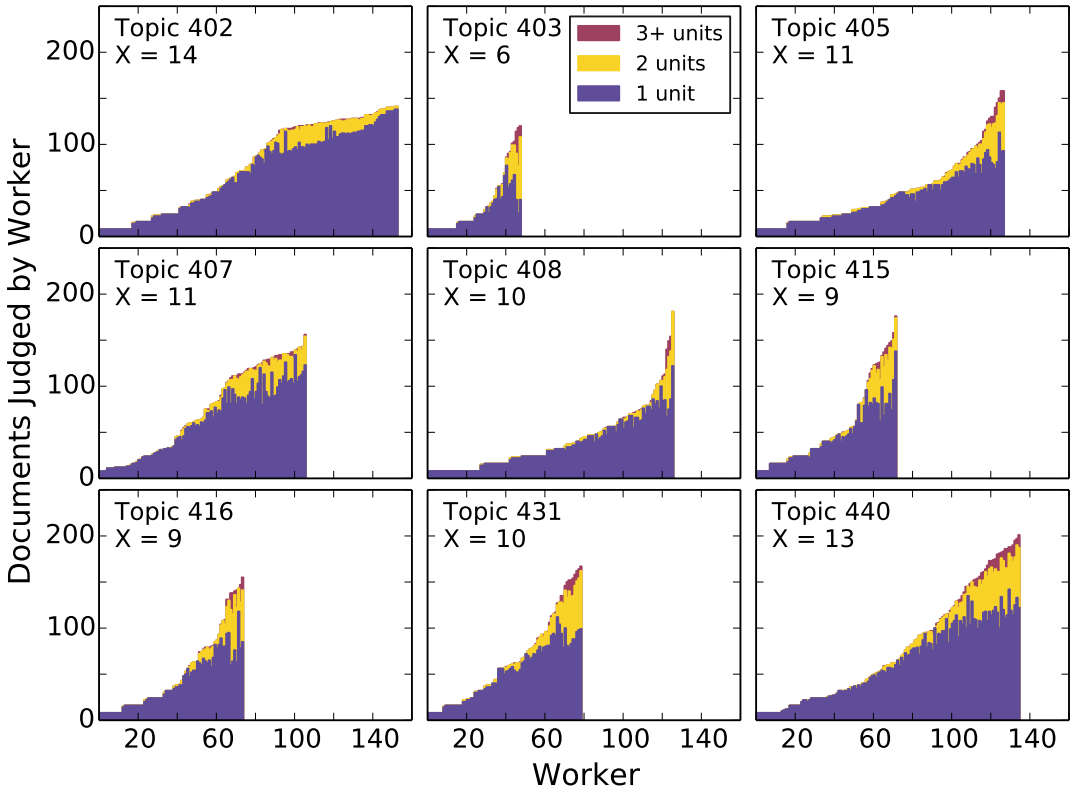


Figure 5.13: The number of documents judged by each worker. In each sub-figure (topic), each bar represents one worker and bars are sorted in increasing order. The purple part in the bar shows the number of documents viewed and judged by the worker only in one group; the yellow and red illustrate documents that were judged by the worker in two or more than two groups, respectively.

We firstly measured the time that workers spent on the assessment of a pair, shown in Figure 5.14, via separate experimentation, assuming (perhaps bravely) that while working within the `Figure8` interface they were not engaged in any other concurrent activities. Each of the sub-graphs (topics) show a similar pattern of average time that workers spent on reading and answering three questions for one pair. The overall average is about 60 seconds.

Then we compared the time that `Figure8` workers spent on judging document pairs using different methods. Table 5.10 summarizes the average time that workers spent on judging one pair of documents for different topics using the combined method (answering all three questions) in the second column. We additionally used the same pairs but asked QUESTION1 and QUESTION2 alone in two separate `Figure8` tasks. The average time that workers spent on these single-question tasks is shown in the third and forth columns of Table 5.10. For each topic, the tested pairs and the payment rate for each pair were same for all of these variant experiments.

As expected, answering all the three questions for each pair requires more time than

Topic ID	Q1+Q2+Q3	Q1 only	Q2 only	Q2 – Q1
402	63.00	33.87	38.00	4.13
403	60.17	40.55	52.87	12.32
405	54.43	41.82	40.33	–1.49
407	61.30	38.76	37.83	–0.93
408	52.43	46.88	47.77	0.89
415	62.64	44.96	46.36	1.40
416	57.95	39.44	44.42	4.98
431	64.68	37.15	35.66	–1.49
440	60.89	32.28	37.89	5.61

Table 5.10: Average judging time (seconds per document pair) for nine tested topics when all three questions (second column), only QUESTION1 (third column), and only QUESTION2 (forth column) are asked for one pair. The last column shows the time difference between QUESTION2 and QUESTION1. The p-value of the paired t-test taking the average assessing time of QUESTION1 and QUESTION2 (the third and forth column) is 0.095. QUESTION1's advantage of assessing speed is obvious in most topics.

answering a single question. The last column of Table 5.10 reports that QUESTION2 requires longer time than QUESTION1 in six out of nine topics and the time difference is obvious in four (402, 403, 416 and 440) of them. In topic 405, 407 and 431, workers' assessing time of QUESTION2 is faster than time of QUESTION1, even the advantages are generally very small (under 1.5s per pair). Referring back to Table 5.9, assessors like QUESTION2 rather than other two questions for these three topics.

However, we need to note that, within one group, workers only need to read and give absolute relevance (QUESTION2) for eight documents, that is answering QUESTION2 for all pairs in the group only requires their eight decisions. But workers need to make twelve preference decisions to answer QUESTION1 for all pairs in the group. In summary, if the a worker's judgments of a group are all consistent, there will be twelve preference judgments and eight binary judgments.

Thus we conclude that for some topics, QUESTION2 may be a preferred (easier for making decisions) method. But for most other topics, making preference choices (QUESTION1) is a generally more efficient method for collecting relevance judgments.

We then explored whether factors such as the average document length, the total number and the relevance level of assessed documents affected the assessing speed of workers. For each topic, the average length, the total number and the average ME relevance score of documents assessed by each worker who contributed to the judgments of this topic were counted and plotted in Figure 5.15

The top graph of Figure 5.15 illustrates the correlation between the average length of documents assessed by each worker and the average judging time that the worker spent

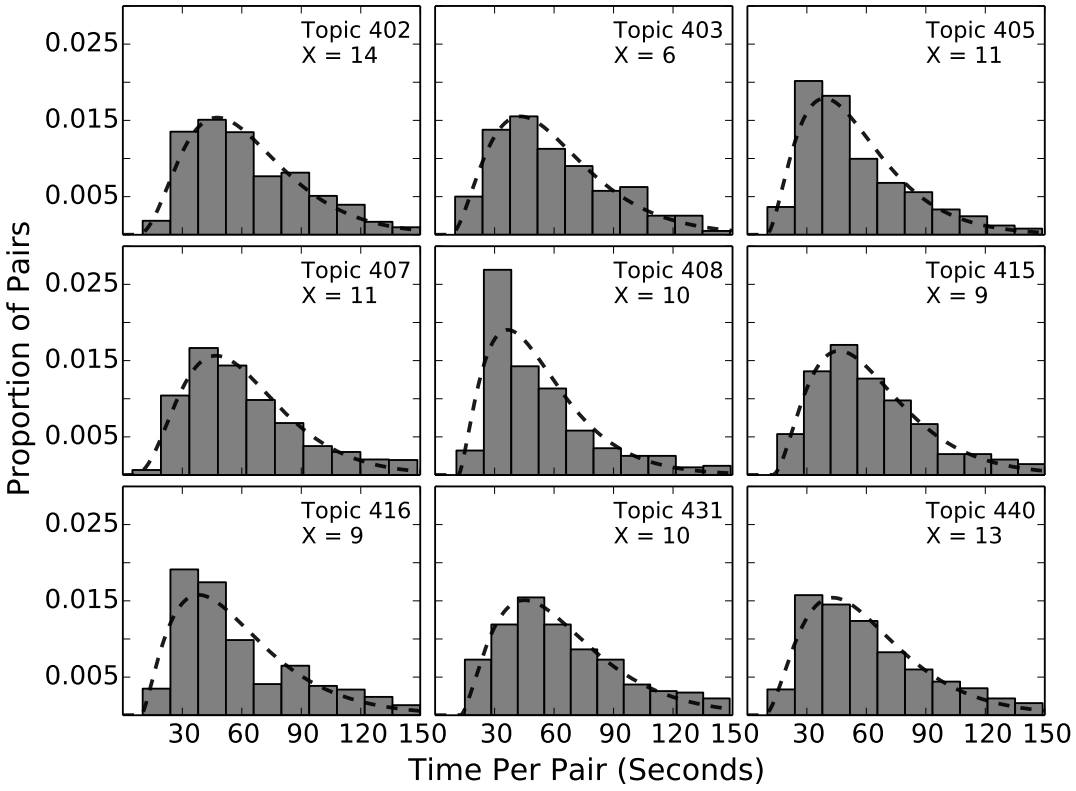


Figure 5.14: The distribution of judging time per document pair for each topic, with three questions to be answered per pair. In each sub-graph, the sum of bar heights is 1.0. The dashed lines in each sub-graph are fitted using the gamma distribution. The average pair judging time of each topic (sub-graph) is shown in the the second column of Table 5.10.

on each document to answer all of QUESTION1, QUESTION2 and QUESTION3. Note that, the average assessing time per document was computed by the total time that the worker spent on assessing a group divided by eight (***DocsPerGroup***=8). For each worker, the average assessed document length is plotted as the vertical axis. The average time that the worker spent per document is plotted on the horizontal axis.

Although the average document length and task difficulty for each topic are varied, there is no clear relationship between the document length and the judging time within each single topic. The Kendall's τ of all topic-worker combinations (dots) shown in the top graph of Figure 5.15 is -0.047. Thus we conclude that judging time of document might not related to the document length only. Some other factors such as topic and document difficulty may have greater effect to the assessing time.

The total number and the average relevance level of documents assessed by the worker are similarly illustrated as functions of worker's assessing speed in the lower two graphs of Figure 5.15. The Kendall's τ of the points in these two graphs are -0.069 and 0.0191 which conclude that neither number of documents nor relevance level of documents

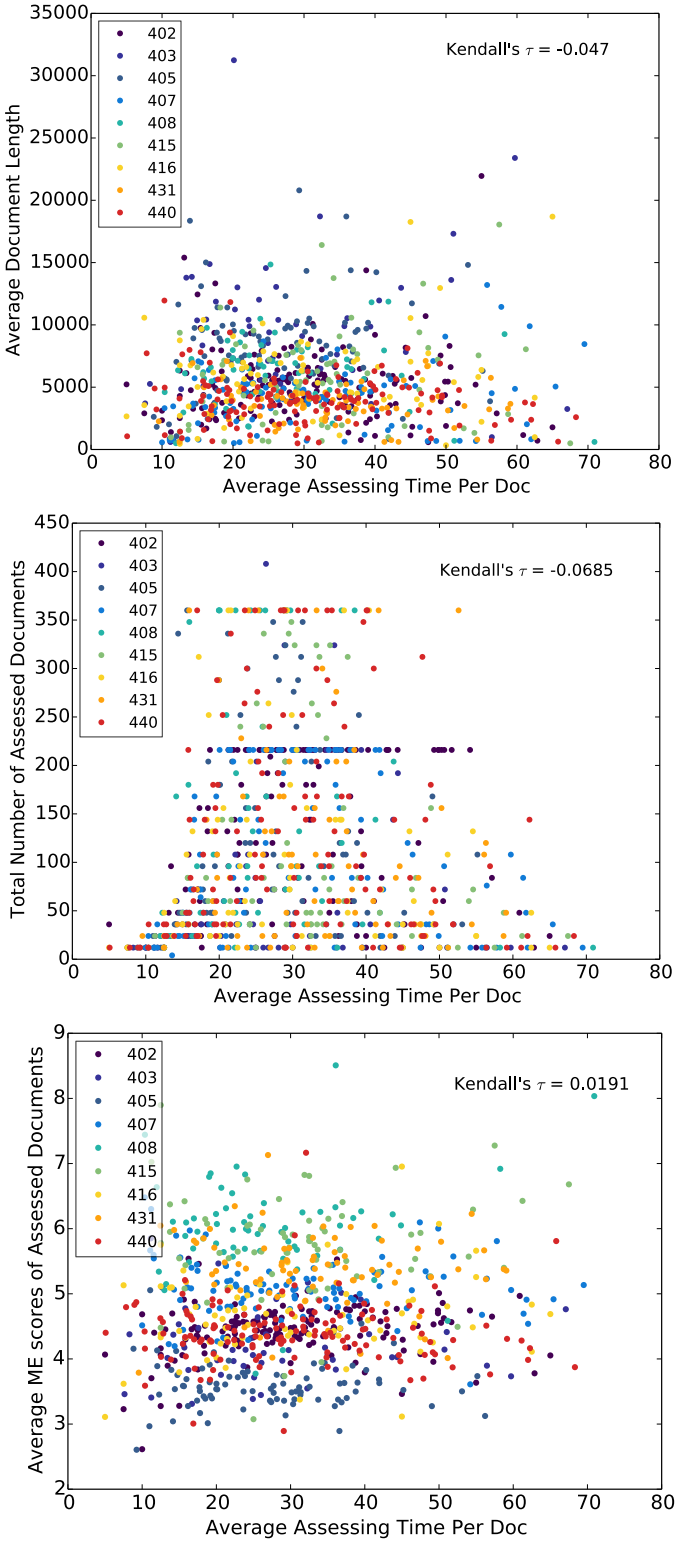


Figure 5.15: Average length of documents assessed by the worker (top), number of documents assessed by the worker (middle), and document relevance score in ME judgments (bottom) as a function of average judging time per document of the worker respectively. Each dot in the graph represents a worker, and colors indicate topics.

Topic	[0, 0] [1, 1]	left	right
402	10380	0.515	0.485
403	1383	0.524	0.476
405	6261	0.529	0.471
407	5604	0.506	0.494
408	3705	0.523	0.477
415	3198	0.514	0.486
416	3087	0.521	0.479
431	4365	0.516	0.484
440	10224	0.529	0.471
Average		0.520	0.480

Table 5.11: The total number of [0, 0] and [1, 1] pairs in the generated pair lists, and the proportions that assessors choose the left documents, and the right documents in the pairs in QUESTION1 for nine topics respectively.

viewed by the worker affects the assessing speed of the worker.

Position Bias As described in Section 5.1.3, the paired documents were horizontally presented to assessors side by side. Even though the positions of documents were randomly chosen when generating document pair lists, as one of the paired documents was stayed in the next pair, and whose position did not change, we then explored if the left-right position of documents caused bias to the answers of QUESTION1.

When assessors considered that the paired documents were equally relevant to the given topic, we assumed that assessor would randomly choose one of them, that is, the document in the left-hand side and the document located in the right-hand side should have equal possibility (50%) to be selected in QUESTION1. For each of the nine tested topics, we counted the number of document pairs in which document relevance scores are both 0, or both 1 in the NIST Binary, shown in the second column of Table 5.11. According to the answers of QUESTION1 of each topic, the proportion of the generated pairs that left documents (shown in the third column) are selected in pairs is generally higher than the proportion that right documents are chosen (shown in the fourth column). When the relevance of the paired documents are similar, the document located on the left-hand side has an average a 4% higher chance of being chosen as the more relevant document.

To further explore the bias, we picked four topics 405, 407, 408, and 415, randomly generated pair lists with same the parameters as before, and launched crowd jobs for the new pair lists on Figure8. Table 5.12 shows the sum of generated pairs that are [0, 0] and [1, 1] in NIST Binary in the second column, and proportions of choosing left and right documents in pairs for each of four topics in the third and fourth columns respectively. In the new experiments completed by different workers, for the same four

Topic	[0,0] [1,1]	left	right	[1,0]	[1,0] agree	[0,1]	[0,1] agree
405	6261	0.529	0.471	1182	0.914	1068	0.904
	6273	0.509	0.491	1134	0.848	1134	0.870
407	5604	0.506	0.494	1773	0.795	1797	0.762
	5583	0.526	0.474	1788	0.697	1800	0.661
408	3705	0.523	0.477	1875	0.871	1650	0.854
	3774	0.511	0.489	1749	0.768	1761	0.755
415	3198	0.514	0.486	1374	0.910	1254	0.886
	3195	0.529	0.471	1290	0.793	1302	0.723

Table 5.12: The number of pairs, left-right choosing proportions for pairs of documents with equal relevance levels, and agreements of document ordering for pairs in which documents have different relevance scores in NIST Binary. For each topic, the upper row shows the results of the original experiment, and the lower row shows the new experiment with the same parameters but different pair lists.

topics but different document pairs, assessors still tended to choose documents on the left 4% more than documents on the right. Did this bias affect the results of preference judgments? This question could be answered by comparing the judgment agreements of document orderings for pairs [1, 0] and [0, 1], shown in the sixth and the last columns in Table 5.12.

As the last four columns shown in Table 5.12, preference judgments generally have higher agreement with Binary for [1, 0] pairs, no matter if [1, 0] pairs are fewer or more than [0, 1] pairs. The p-value of the t-test taking the agreements between preference judgments and NIST Binary for [1, 0] and [0, 1] pairs respectively, is 0.07. Although it is above the significance level $\alpha = 0.05$, the confidence that the left-right bias does affect the preference judgments (when compared with NIST Binary) is very high. We recommend to involve bias adjustment process when aggregating the preference judgments. One possible strategy could be assigning a confidence weight to each preference answer. For example, if the left-right bias is 0.04, for each preference judgment of a pair, the preferred document receive a score of 0.96, instead of 1.00 when there is no bias, and the other document receive 0.04, instead of 0.00. The method for adjusting results when there is a bias could be further explored in future works.

5.3.6 Agreement and Consistency of Judgments

For each topic, we chose the number of partitions ($\boldsymbol{X}$) so that each document would be paired with 15% of other documents. We now explore that decision. In particular, if we

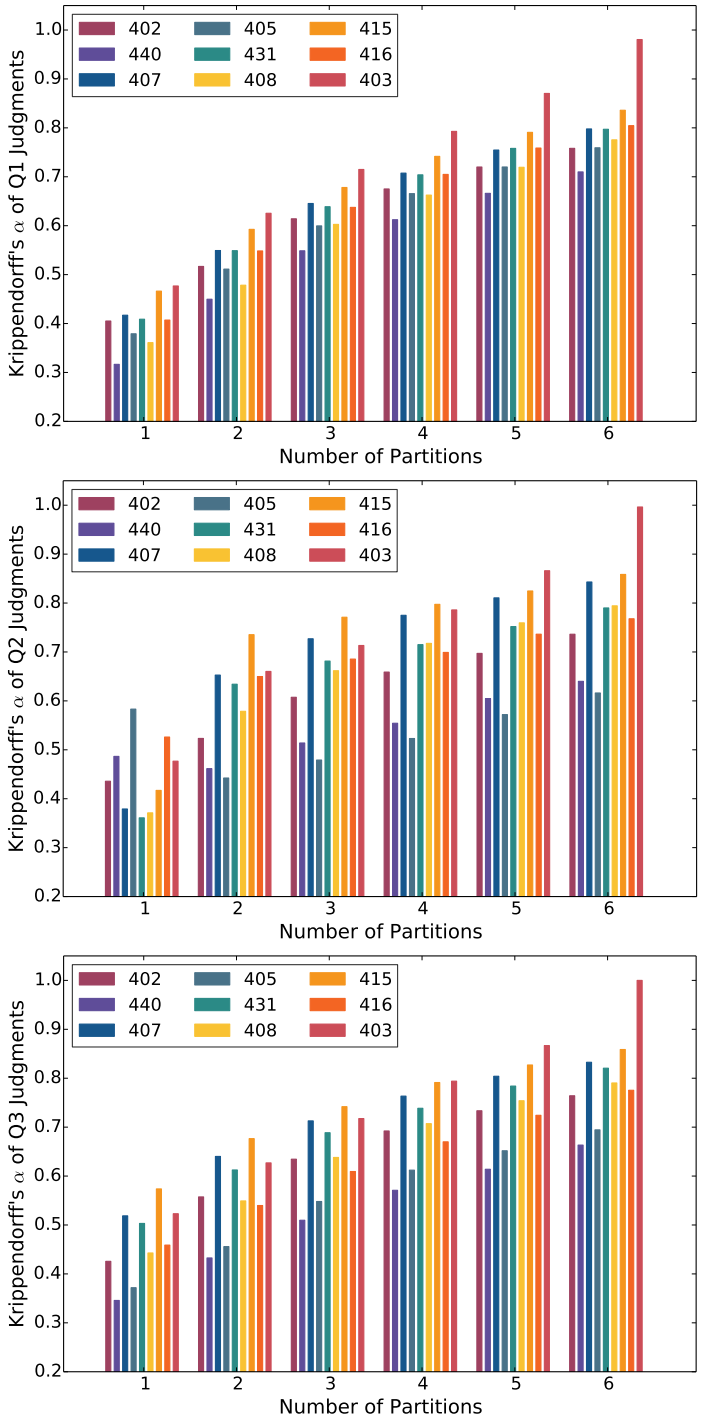


Figure 5.16: Krippendorff's α of all document pairs' relevance orders given by 50 sets judgments generated using $x' \in \{1, 2, 3, 4, 5, 6\}$ randomly selected partitions. Graphs are illustrated for three judgment collecting methods. In each graph and for each partition number, the grouped bars of α for nine tested topics which are sorted in decreasing order of the number of pooled documents and shown in distinct colors.

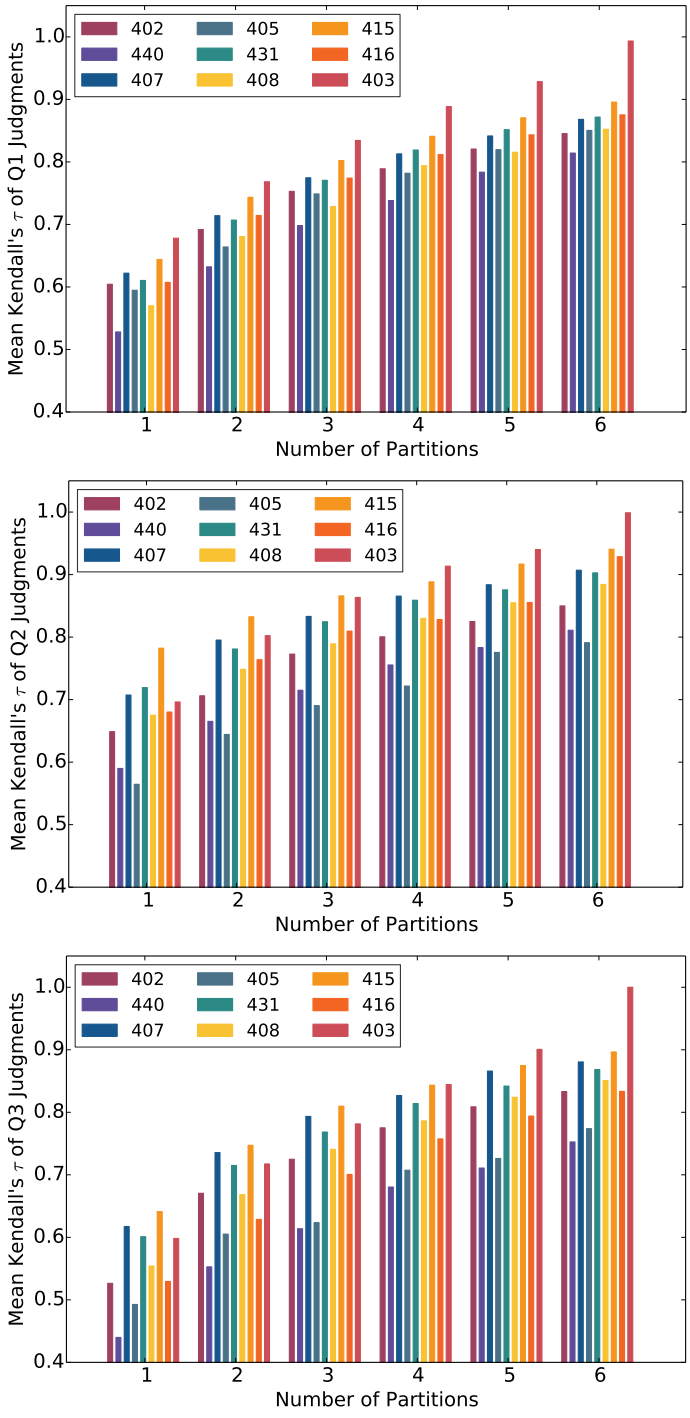


Figure 5.17: Mean Kendall's τ of document orderings given by judgments using different number of randomly selected partitions and the full judgments built using all partitions (as the reference), for three collection methods. In each graph and for each count of partitions used, the bars for the nine tested topics are sorted in decreasing pool size order, and shown as distinct colors. For each topic, Kendall's τ is measured between the document relevance ordering given by the full judgments, and an incrementally growing subset of the topic's partitions. Each bar represents the average of ten τ values computed over ten randomly generated combinations of the available partitions for that topic.

use answers of the same number of pairs but from different partitions to build the judgments, do the judgments agree with each other? And, what is the relationship between the quality of judgments and the number of pairs involved in collecting those judgments?

In the first experiment, for each topic, we randomly selected x' partitions to generate a judgments set, repeating the selection process 50 times. The agreement of these 50 sets of judgments built using x' partitions was then measured by Krippendorff's α. As described in Section 2.4.3, the relevance scores of documents are treated as categories rather than numeric values by Krippendorff's α, so we generated all pairs of documents for each topic, and measured the agreement of judgments by computing the Krippendorff's α, taking the paired documents' relevance orderings given by different judgments as inputs. Figure 5.16 shows the Krippendorff's α of 50 sets of judgments generated using $x' \in \{1, 2, 3, 4, 5, 6\}$ partitions for each topic (in different colors) and each of the three judgment methods (in different sub-graphs).

As the pattern shows in all three sub-graphs, for each topic, when the number of partitions used to generate judgments increases, the agreement of the built judgments grows ($p < 0.05$ in a GLMM). That is, if we pair each document with more other documents, the judgments that are generated will be more predictable and reliable.

Within each group of nine bars, the topics are sorted by the number of documents that were pooled from largest to smallest; the α value does not visibly increase when the topic pool size decreases. However, a Linear Mixed Model concludes that the pool size has negative effect ($p < 0.05$) on the α values associated with the three questions.

There are no significant differences between QUESTION1, QUESTION2, and QUESTION3, according to the t-test.

As for each topic we only had a limited number of partitions to select, meaning that some partitions used to generate different judgments might be the same, thus the disagreement of partitions (and so generated judgments) might not be fully detected and measured when a larger number of partitions were used. That is why the agreement α increases as $\boldsymbol{X}$ grows.

Kendall's τ can be employed if we want to compare the document orderings given by two judgment sets. Figure 5.17 shows the Kendall's τ of document score orderings given by the full judgment (the reference, when all partitions were used) when compared with the qrels generated by randomly selected partitions. Each group of nine bars again represents partial partitions being used, and the Kendall's τ shown by the heights of bars is the average correlation between document orderings induced by the qrels of judgments generated by all, and different subsets of the partitions.

As more and more judging information is added to the qrels, the τ score increases. The increment of bars is larger when the number partitions is smaller and tend to be more stable when the judgments are closer to the full qrels. The patterns shown in the graphs of the three different questions are similar, but the τ values for QUESTION3 (relevance ratio) generally a little lower than for the other two questions.

In summary, document relevance scores in judgments become more stable if more partitions (documents pairs) are used. The agreements of judgments built by QUESTION3 answers are generally the lowest. For some topics, with the same number of partitions, judgments using answers of QUESTION2 are closer to judgments using full partitions than judgments of QUESTION1. But for some other topics, judgments of QUESTION1 become stable faster than judgments of QUESTION2.

5.3.7 Consistent Discrimination

The key goal of collecting relevance judgment in IR is to measure whether one system performs better than another. To compare the judgments' ability in terms of system discrimination, we paired 123 TREC-8 systems (in total, 7503 pairs) and computed their mean RBP scores with $\phi = 0.9$ over nine topics using a variety of qrels. For each system pair, the student t-test was applied to the scores of the paired two systems over the nine topics. If the computed p-value is less than the significance level 0.05, the pair was *distinguishable*, otherwise it was *tied*.

Using NIST Binary judgments as the reference, system discriminations of different relevance judgments are summarized in Table 5.13. For each qrel Q, system pairs can be categorized into:

- TP – both Q and NIST Binary judgments distinguished the pair;
- FP – Q distinguished the pair but NIST Binary did not;
- FN – NIST Binary distinguished the pair but Q did not; and
- FN – neither Q nor NIST Binary distinguished the pair.

The qrels shown in Table 5.13 also include combined relevance scores for documents by computing the geometric means of scores arising from different questions (QUESTION1, QUESTION2 and QUESTION3), to see if involving more information from different aspects would improve the system discrimination of relevance judgments. Denote $\mathsf{Rel}_1(\boldsymbol{d})$, $\mathsf{Rel}_2(\boldsymbol{d})$ and $\mathsf{Rel}_3(\boldsymbol{d})$ as relevance scores of document $\boldsymbol{d}$ in the qrels files generated by answers of QUESTION1, QUESTION2 and QUESTION3 alone. The score of $\boldsymbol{d}$ in the combined qrels of the three judgments is computed as a three-element adjusted geometric mean:

$$\sqrt[3]{(\mathsf{Rel}_1(\boldsymbol{d}) + \epsilon) \cdot (\mathsf{Rel}_2(\boldsymbol{d}) + \epsilon) \cdot (\mathsf{Rel}_3(\boldsymbol{d}) + \epsilon)} - \epsilon,$$

where we chose $\epsilon = 1$. The calculations for geometrically combining two judgments are similar.

In the first row of Table 5.13, there are 2734 pairs of systems indicated as significantly different (distinguishable) by both the Sormunen and the Binary judgments, and 3787 pairs of systems that could not be distinguished by either of the two sets of qrels. Another 490 system pairs were tied according to the Binary judgments, but distinguished

qrels	System Pair			
	TP	FP	FN	TN
Sormunen	2734	490	492	3787
Q1, Q2 and Q3	2770	842	456	3435
Q1 and Q3	2782	845	444	3432
Q2 and Q3	2764	833	462	3444
Q1 and Q2	2752	862	474	3415
Q1	2744	906	482	3371
Q2	2735	826	491	3451
Q3	2763	824	463	3453
Random	206	245	3020	4032

Table 5.13: Applying a two-tailed paired t-test (significance level 0.05) to compare 123 TREC-8 systems in pairs (7503 pairs in total) using the NIST Binary judgments as the reference, and counting the number of significant differences when comparing systems using Sormunen judgments and the judgments generated via QUESTION1, QUESTION2 and QUESTION3, in various combinations.

by the Sormunen judgments, according to FP. However, the Sormunen judgments failed to separate another 492 pairs of systems which the Binary judgments did distinguish.

As Table 5.13 shows, using the preference judgments (QUESTION1) alone distinguishes the greatest number of systems in pairs (TP plus FP, in total of 3650), while the binary judgments distinguish 3226 pairs of systems and the Sormunen qrels distinguish 3224 (significantly different with $p < 0.05$ using a χ^2 test). Using any one of QUESTION1, QUESTION2, QUESTION3 and all possible combinations of judgments derived from them can distinguish more systems than binary and Sormunen. The judgments of Sormunen have the most similar system comparing results (TP plus TN of 6521, or 86.9%) with judgments of binary. The combined judgments of QUESTION1 and QUESTION3 have the highest agreement of identifying different systems with binary (TP equal to 2782).

Table 5.14 shows the system discrimination of judgments collected using QUESTION1, QUESTION2 and QUESTION3 when partitions are incrementally added. The numbers shown in the table are the mean over five random selections. For all the three methods, when the number of partitions used grows, the number of TP pairs increases and the number of FN pairs reduces. In general, discriminations of judgments collected using QUESTION1, QUESTION2 and QUESTION3 all grow with the number of partitions used to build the judgments.

Figure 5.18 and Figure 5.19 illustrate the paired t-test p-values of judgments used to distinguish systems in pairs. Each system pair is represented as a dot in each sub-graph and is distinguishable by the judgments on x-axis if its horizontal value is less than 0.05, and similarly for the other set of judgments on the vertical axis.

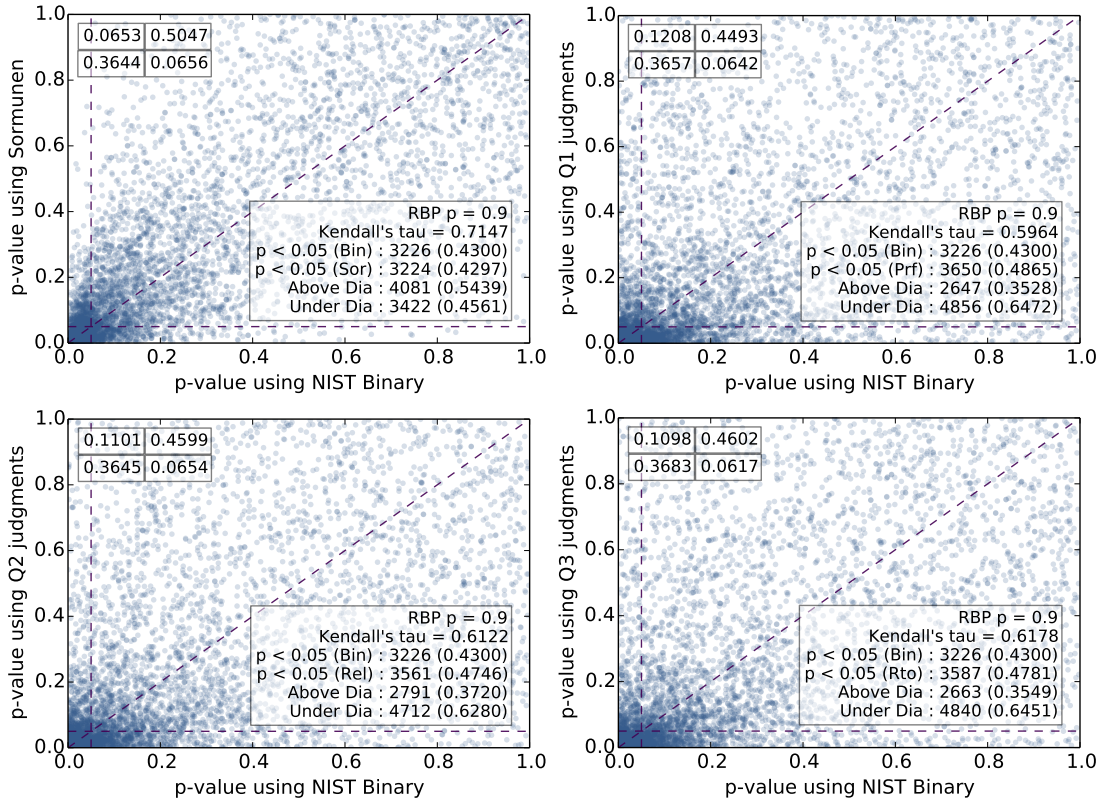


Figure 5.18: The p-value scores of the paired t-test taking the RBP ($\phi = 0.9$) scores of TREC-8 systems evaluated by judgments shown on axes. The judgments of Sormunen, QUESTION1, QUESTION2, and QUESTION3 (shown on y-axis) are compared with judgments of NIST Binary (shown on x-axis). There are 7503 system pairs and each of them is shown as a dot in the graph. If the p-value of the paired systems is below the significance level, 0.05, the pair is deemed as distinguishable using the given judgments. The Kendall's τ of all points as well as the number of pairs whose p-value < 0.05 according to each of the two compared judgments are shown in the bottom right corner. The horizontal and vertical dash lines split the graph into four parts, TP, FP, FN and TN. The proportion of pairs in each of the four quadrants is shown in the top left corner.

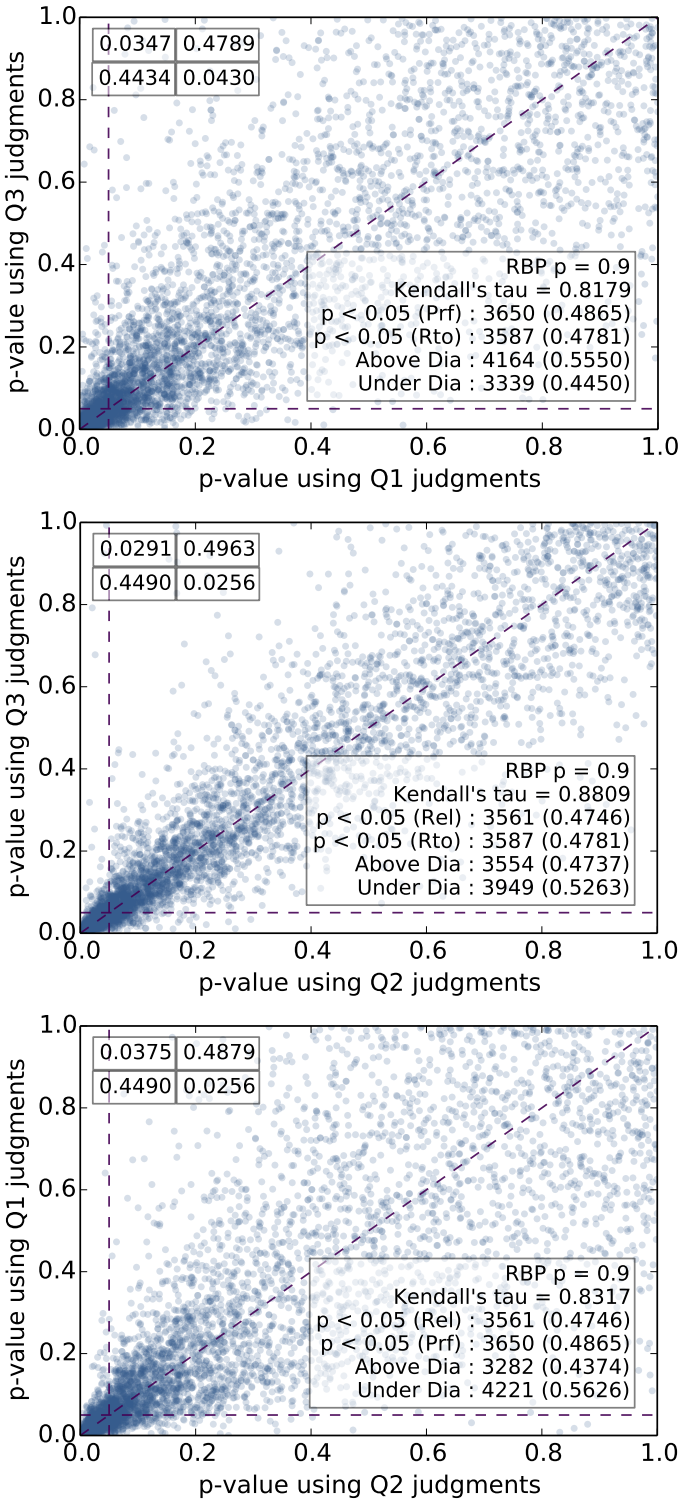


Figure 5.19: The p-value scores of the paired t-test taking the RBP ($\phi = 0.9$) scores of TREC-8 systems evaluated by pairwise judgments.

NumParts	Q1				Q2				Q3			
	TP	FP	FN	TN	TP	FP	FN	TN	TP	FP	FN	TN
1	2457	635	769	3642	2561	697	665	3580	2484	566	742	3711
2	2596	700	630	3577	2610	658	616	3619	2602	679	624	3598
3	2706	893	520	3384	2670	847	556	3431	2688	778	538	3499
4	2714	895	512	3382	2714	895	512	3382	2717	836	509	3441
5	2728	918	498	3359	2684	887	541	3391	2704	767	522	3510
6	2732	915	494	3362	2706	864	519	3414	2746	837	480	3440
All	2744	906	482	3371	2735	826	491	3451	2763	824	463	3453

Table 5.14: System discrimination of judgments generated by ***NumParts*** partitions. Applying a two-tailed paired t-test (significance level 0.05) to compare 123 TREC-8 systems in pairs (7503 pairs in total) using the NIST Binary judgments as the reference, counting the differences when using judgments generated by some number of partitions via QUESTION1, QUESTION2 and QUESTION3. For each question and each row (***NumParts*** $\in \{1, 2, 3, 4, 5, 6, All\}$), partitions were randomly selected and repeated five times for each row used to build the judgments. The count of system pairs in each category and each row is the average of results obtained from the five random partition selections.

The Kendall's τ of all points in each sub-graph (shown in the bottom right corner) indicates the correlation of the two compared judgments in terms of system discrimination. The judgments of QUESTION1, QUESTION2 and QUESTION3 have high agreements with each other on system discrimination, and all have lower correlation with NIST Binary than with the Sormunen judgments.

When compared with NIST Binary, the judgments of QUESTION1 have the highest TP (number of system pairs that can be distinguished by both). Sormunen has the greatest Kendall's τ correlation with NIST Binary because its TP (number of system pairs that cannot be distinguished by either) is high, and so the TP+TN are the largest.

By looking at the number of system pairs whose p-value < 0.05, deemed as system discrimination, the number of system pairs which can be distinguished is the highest when using judgments of QUESTION1, followed by QUESTION3, QUESTION2, NIST Binary and Sormunen in decreasing order.

In the three graphs in Figure 5.19, the system discriminations of judgments generated by answers to be three questions using all partitions are compared. If we reduce the number of partitions used, will the correlations (Kendall's τ) of their discriminations decrease? And which two of these three methods are the most similar regarding to distinguishing systems in pairs?

For every two methods, the τ of p-values obtained from two-tail paired t-tests of comparing systems in pairs using judgments built by ***NumParts*** $\in \{1, 2, 3, 4, 5, 6, All\}$

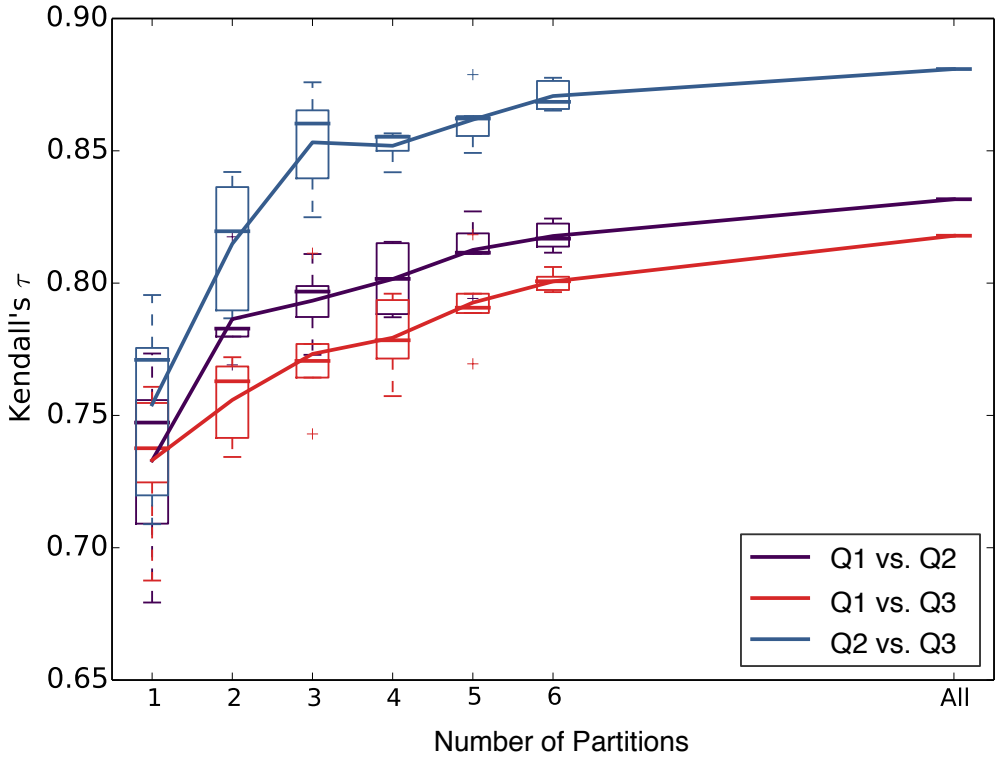


Figure 5.20: The Kendall's τ of p-values obtained from two-tail paired t-tests comparing systems in pairs using two sets of judgments collected by: QUESTION1 (Pref), QUESTION2 (Rele) and QUESTION3 (Ratio). The x-axis shows the number of partitions (randomly selected from all partitions) used to build the final judgments for the t-test. In each column and for each pair of comparing methods, the random selection of partitions and t-test were repeated ten times, ten then the Kendall's τ scores are plotted as a whisker box. The solid line in each color represents the mean of the ten Kendall's τ scores in each box.

randomly selected partitions and collected by these two methods respectively, was computed and shown in Figure 5.20. As the selection of partitions was random, the process was repeated ten times for each ***NumParts***. That is, for each number of partitions (column) in Figure 5.20, the whisker box was generated by ten τ values of ten random selections.

As is shown by the pattern in Figure 5.20, the correlations τ between judgments increase with ***NumParts*** (that is, when relevance scores of documents become more stable). The mean Kendall's τ sharply grows until the ***NumParts*** increases to three. The variance (whisker box's length) of ten τ scores also becomes smaller when more partitions are added. The judgments generated by answers to QUESTION2 and QUESTION3 are the most correlated, while judgments of QUESTION1 and QUESTION3 are the most different.

The similarity of judgments collected using these three methods can also be measured

NumPart	Q1	Q2	Q3
1	0.8236	0.8012	0.7750
2	0.8616	0.8340	0.8294
3	0.8984	0.8719	0.8421
4	0.9179	0.8805	0.8411
5	0.9346	0.8914	0.8717
All	1.0000	0.9002	0.8846

Table 5.15: RBO ($\phi = 0.98$) scores of TREC-8 system rankings evaluated by RBP ($\phi = 0.9$) using full QUESTION1 judgments (as reference) and judgments generated using answers of three questions in ***NumParts*** of partitions.

by computing the agreement (overlap) of their system orderings. Figure 5.21 shows the overall system orderings given by RBP ($\phi = 0.9$) using judgments generated by ***NumParts*** partitions. Whisker boxes represent systems, and are sorted by decreasing RBP scores using the full QUESTION1 judgments. Similar patterns emerge in all subgraphs, for judgments of all three methodologies, top systems (according to full preference judgments) receive high RBP scores even when only a small number of partitions are used to build the qrels. The systems' RBP scores tend to be more stable (the length of boxes decreases) if more partitions are involved when generating judgments.

To further explore the overlaps (agreements) of system orderings between judgments generated by QUESTION1, QUESTION2 and QUESTION3, the RBO ($\phi = 0.98$) scores of TREC-8 system rankings evaluated by RBP with $\phi = 0.9$ using different judgments, using the QUESTION1 judgments generated using all partitions as the reference point, are shown in Table 5.15.

As more partitions are used, QUESTION1 judgments self-converge with the reference (using QUESTION1 answers in all partitions). The agreements of system orderings between the reference and judgments of QUESTION2 and QUESTION3 also increase. But with more partitions involved, system orderings given by judgments of three questions becomes more similar. The RBO scores of QUESTION2 judgments are always higher than scores of QUESTION3. Referring back to Figure 5.20, it also appears that the QUESTION2 judgments agree more with QUESTION1 than the QUESTION3 judgments do, even through for each pair the ratio score assigned by workers in QUESTION3 has to agree with the preference choice in QUESTION1, guaranteed by the embedded validation in the experiment design.

5.4 Summary

Instead of sequentially presenting documents and collecting their relevance judgments using an ordinal relevance scale, we randomly and equally divided documents into groups

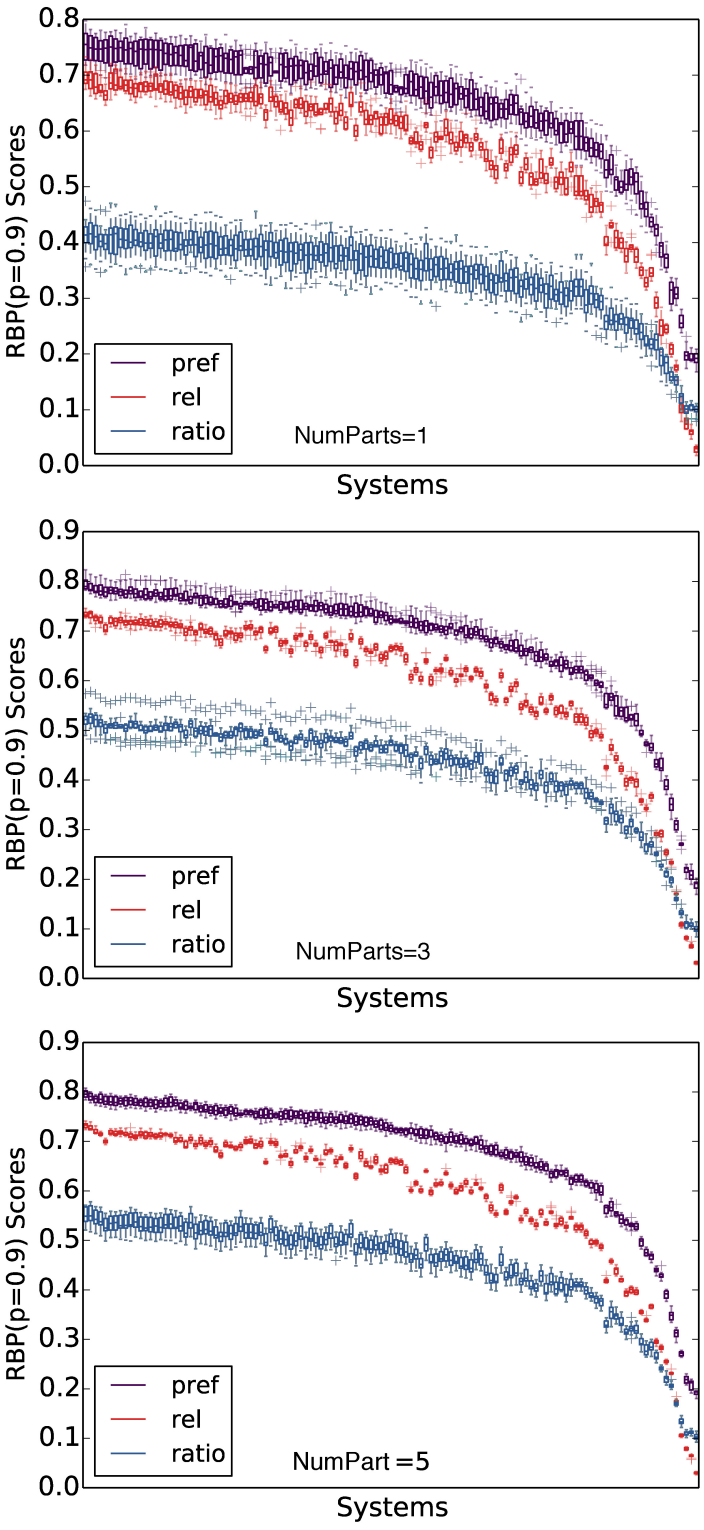


Figure 5.21: RBP ($\phi = 0.9$) scores of TREC-8 systems using judgments of three questions (shown in different colors) generated by ***NumParts*** of randomly picked partition(s) for ten times. Each whisker box is built by ten RBP ($\phi = 0.9$) scores of a system across nine topics. Systems (whiskers) are sorted based on the RBP score evaluated using full preference judgments (QUESTION1).

and paired each document the same number of other documents in the same group to collect crowd-sourced pairwise judgments by asking their (1) preference, (2) absolute relevance and (3) relevance ratio within each document pair. For each question (method), we aggregated and normalized the answers and calculated a numeric relevance score for each topic-document combination.

To address **RQ5**, which asks if our methods can be used to collect relevance judgments similar to those obtained by the previous approaches, we generated qrels files using answers of each questions and measured their agreements with judgment sets provided by NIST Binary, Sormunen and ME in aspects of score, ordering of documents in pairs and system orderings, described in Section 5.3.2 and 5.3.4. We concluded that judgments collected using our methods were not dissimilar to those generated by previous schemes.

As we did not have costs for NIST Binary and Sormunen judgments, **RQ6** is harder to resolve. But according to a reasonable estimate that an expert would be paid at least USD$\$60$ an hour and might judge one document per minute, without considering training time, collecting judgments for $1,876$ topic-document combinations that we constructed would cost USD$\$1,876$. In our experiments, tasks of each topic could usually be completed in one day and we only spent USD$\$1,303$ to collect three sets of judgments on `Figure8`. Even we have multiple validation processes for each document and transaction fees paid to `Figure8`, the overall cost was almost certainly lower.

We were interested in factors that might affect the relevance assessment quality (**RQ7**). In Section 5.3.5, we computed the consistency and accuracy of workers and analyzed the survey they completed about which methods were preferred for assessing the relevance of documents for the given topic. Factors of topic difficulty, workload (number of assessments the worker completed) and method complexity all influenced worker's preferences of relevance scales. The worker's experience, and document length, might also have an effect, but they were not significant factors according to statistical tests over the crowd-sourced data. In Section 5.3.6 and 5.3.7, the volume of opinions was the most obvious factor that affected the consistency and discrimination of judgments collected by our methods. The more documents paired and more workers involved in the relevance examination of a document, the more consistent the overall assessment of that document is likely to be.

In regard to **RQ8**, we tested if our crowd-sourced judgments could give similar system comparison outcomes to the conventional relevance judgments. The results described in Section 5.3.2 and 5.3.7 concluded that they did. We also demonstrated that judgments collected using our methods allow finer system distinctions to be identified.

Overall, in this chapter, we proposed a new design to collect pairwise judgments using a "three-in-one" crowd-sourced approach in order to generate relevance judgments with higher fidelity for finer system evaluation. The results indicate that using our design is cheap, effective, and acceptable to assessors. The generated judgments are reliable

and appear to have greater system discriminations than judgments collected using conventional ordinal scales.

Chapter 6

Conclusion and Future Work

Evaluation is one of the key elements of developing better Information Retrieval systems. In the long history of discovering effective and efficient methodologies to evaluate IR systems, researchers have explored different aspects of IR evaluation, such as user's information needs, metrics and models of user behaviour, collecting relevance judgments, and so on. In Chapter 2, the commonly-used evaluation strategies of comparing the effectiveness of systems, such as batch evaluation techniques, were introduced and compared. Chapter 2 also described the challenges faced in IR evaluations including (1) how to evaluate systems generating similarity score ties using reliable methods; (2) how to measure the uncertainty of system scores caused by unjudged relevance judgments; and (3) how to economically collect reliable relevance judgments with high fidelity.

6.1 Contributions

For a given query and a set of documents, the effectiveness of a system depends on the algorithm employed to compute similarity scores and generate rankings of documents sorted in decreasing relevance order, and can be measured by a variety of established evaluation metrics. In Chapter 3, we found that a number of TREC systems assigned the same similarity scores to different documents when scoring them. That is, many documents are tied with others on similarity scores. As the ranked lists of documents must be linearized for sequential presentation, systems need to employ a mechanism for ordering tied documents, such as sorting tied documents by their document identifiers. We discovered that the employed mechanisms for sequentially ranking tied documents in the runs seemed to have almost no effect to TREC system effectiveness evaluations when NIST Binary judgments were used.

The similarity score ties might be caused by score rounding for computing efficiency when systems scoring documents. As the similarity score ties only have minor effect to the TREC system evaluation when the fidelity of the used relevance judgments (such as NIST Binary) was low, in further experiments, we deliberately introduced ties to the TREC runs to different extents, and aimed to answer the first general research question in this thesis:

RQ1, 2: how similarity score ties affect IR evaluations, and to what extent can similarity score rounding be tolerated without alerting the system discrimination of IR evaluation metrics?

We proposed a strategy that grouped documents in TREC runs into bands, and assigned the same score to the documents in each band. The score assigned to each band was decreasing, and the length of bands was a geometric progression, controlled by a parameter ρ. This grouping process allowed similarity score ties to be deliberately generated to different extents. We applied this method to the TREC runs with a range of ρ values, and used a statistical test to compare each system pair using the original run scores and the banded run scores respectively. The calculated p-value for each system pair would indicate whether the grouping process affected the system comparisons. Using the Binary judgments, we showed that even when $\rho = 2$, the generated ties had no significant effect on system evaluations. The generated ties did not alter the conclusion of system comparisons on average for 98% of pairs. That is, in TREC evaluations using Binary judgments, systems do not need to keep high-precision of similarity scores, and therefore they can potentially boost their computing speeds. Allowing ties caused by the decrease of similarity score precision, to the level which is equal or less than the extent of $\rho = 2$ is acceptable for system evaluations.

In particular, we considered the impact made by similarity score ties to the system effectiveness scores evaluated by metrics of RR, RBP($\phi = 0.5$), RBP($\phi = 0.85$) and AP respectively. We demonstrated that ties affected the shallow metrics (whose expected evaluation depths are shallow, such as RR) less than deep metrics (such as AP) in regards to run score differences when $\rho \leq 3$. When ρ increased larger than 3, as more ties appeared in top ranks, the metrics that treated top ranks more importantly were affected greater by the generated ties.

The grouping strategy that deliberately introduces ties in similarity score computations without altering the system comparison results could be further explored for improving the search speed, and reducing the space used of IR systems (see Section 6.2).

However, the extent that similarity score ties could be allowed without affecting system comparisons could relate to the precision (number of relevance levels) of the employed relevance judgments. When the fidelity of judgments is high, that is there are more relevance categories in the relevance scale, the metric scores would be more sensitive to change with the ordering of the tied documents (described in Section 3.4). As the number of documents and topics has grown "big", collecting relevance judgments for every topic-document combination with high relevance fidelity may be expensive. There are also usually trade-offs between the size and fidelity of relevance judgments because of the budget. Therefore, to address these problems, we explored (1) how to measure the uncertainty of metric scores caused by unjudged documents (in Chapter 4); and (2) how to collect more precise relevance judgments with cheaper cost (in Chapter 5).

In test collections, such as TREC, built for IR evaluations (typically, at a scale smaller than commercial Web search), tracks for different search purposes are added by years, meanwhile the number of documents also grows. Pooling has become a necessary process when collecting relevance judgments for test collections. We reviewed and compared some proposed pooling strategies in Chapter 2, but none of them judge every document returned by the systems, the number of topic-document judgments still depends on the limitation of total costs. For the unjudged documents in the retrieved run, the uncertainty caused by them in the metric scores is established by residual scores of utility-based metrics such as RBP. But recall-based metrics such as AP do not have residuals, and usually consider unjudged documents as irrelevant. The uncertainty of recall-based metric scores is therefore hard to measure. These limitations led us to investigate:

RQ3, 4: how to estimate recall-based metrics using RBP, and compute the uncertainty of recall-based metrics, which is caused by the unjudged documents in the run, via RBP residuals;

as the second general research question in Chapter 4. We chose different values for RBP's continuing probability parameter ϕ, until the system ordering of the adapted RBP is most similar to the system ordering given by the estimated recall-based metric. Using this method, we can "link" the recall-based metric with utility-based metrics. And then the uncertainty of the estimated recall-based metric score could be approximated by the residual score calculated by the adapted RBP. The residual score indicates the upper limit of metric score changes caused by the unjudged documents. If the size of the used relevance judgments is small, and so there are a large number of unjudged documents in the retrieved runs, the confidence we have for the system evaluation results would be small because of the great uncertainty of system effectiveness scores. Our proposed strategy is for measuring the uncertainty of system scores for those metrics that do not have methods for computing residuals.

Although large numbers of unjudged documents did not bring great changes to TREC system scores, we still suggested researchers report the uncertainty of effectiveness metric scores brought by the incomplete relevance judgments. If there is no method to compute the residual of the employed metric, for example when AP is used, we recommend to consider using a weighted-precision metric such as RBP to estimate the used metric, and then compute the corresponding residuals.

Because of the increasing scale of data, efficiently and economically collecting precise relevance judgments for IR evaluation has also become challenging. Although relevant information is desired in any field [92], "relevance" is subjective to individuals, and can be hard to define and measure. When we want to collect judgments with high fidelity, if conventional ordinal relevance scales are used, the different expectations of assessors to the distinctions between relevance categories might lead to disagreements in assessments, especially when the number of relevance categories (that is, judgment fidelity) is high. Thus in Chapter 5, we proposed collecting pairwise judgments on a crowd-sourcing

platform via three techniques: pairwise preference, absolute relevance, and relevance ratio. For each given topic, we asked three questions for each document pair: (1) which document is more relevant; (2) whether the documents are relevant to the given topic; and (3) what is the relevance ratio between the paired documents?

We proposed a methodology to randomly generate document pairs. We ensured that each document is randomly paired with the same number of other documents, pairs in each generated pair list were constructed by a same group of documents, and displayed in a "chain". Our design saves the reading time for assessor, and help to reduce the assessment complexity. Moreover, we employed several quality control techniques to ensure the high quality of the obtained answers. We enabled the test mode on `Figure8` and designed test questions to select crowd workers with high accuracy to assess the real document pairs in the work mode. Overall, our proposed methodology helps to improve the assessment efficiency, meanwhile guarantee the high quality of the collected judgments.

The obtained answers of each of the three questions were then normalized into numeric relevance scores for documents. We aimed to use this new methodology to collect relevance judgments with fewer relevance score ties (higher fidelity) on the crowdsourcing platform, which provides a large number of human workers who could complete human intelligent tasks without requiring high payments. Hence,

RQ5, 6, 7, 8: whether the combination of pairwise judging techniques can be employed to collect relevance judgments with high fidelity and low cost on crowdsourcing platform; and which factors might affect the assessment quality,

is the third general research question in this thesis.

We analyzed the answers collected using our new method in experiments launched on `Figure8`, and compared the normalized results with the judgments of NIST Binary, Sormunen, and Magnitude Estimation. We demonstrated that judgments collected using each of the three techniques were similar to the previous schemes in terms of document ordering, system ordering, and system discrimination. The relevance information included in our judgments were richer than ordinal judgments, that is, the precisions of relevance scores in our judgments are higher, and hence our judgments could distinguish more documents on relevance than Sormunen and NIST Binary judgments. We also showed that the overall cost was lower than the compared approaches. Moreover, as the fidelity of our judgments was higher than conventional ordinal relevance judgments, we showed that the collected pairwise judgments could identify finer system distinctions than NIST Binary and Sormunen judgments.

The judgments generated by the normalized answers of three questions are also compared with each other. We demonstrated that the preference, absolute relevance and relevance ratio had high agreement with each other in terms of document ordering, system ordering, and system discrimination. The relevance score distributions of three sets of judgments showed that ratio judgments could separate documents over all the ranks, but

Method	Judgments Fidelity		Difficulty	Assessor's Preference		Consistency
	relevant	irrelevant		Easy Topics	Hard Topis	
Pref	Medium	Medium	Medium	High	Low	High
Rele	High	Low	Low	Low	High	Low
Ratio	High	High	High	Low	Low	High

Table 6.1: Comparisons of three pairwise methods: pairwise preference (Pref), absolute relevance (Rele), relevance ratio (Ratio).

the relevance scores in absolute relevance judgments were mostly tied in bottom ranks (for irrelevant documents). The discrimination of preference judgments was almost equal across all the ranks, but its relevance score tie rate was similar to the absolute relevance judgments. The consistencies of workers' answers to preference and ratio questions were higher than answers to the question asking the absolute relevance of the paired documents. The absolute relevance and ratio judgments were illustrated as the most similar with respect to distinguishing systems in pairs. Assigning relevance ratio was the most disliked method for crowd workers according to the survey answers, probably because it required more cognitive effort and typing. The preference between pairwise preference and absolute relevance scales significantly depended on the topic difficulty, and the workload of the assessor.

We also compared the system discriminations of different combinations of our judgments with conventional judgments, and tested the reliability of judgments generated by some subsets of the collected answers.

In general, choosing which pairwise scale(s) to use, or which of their combinations to collect judgments for, depends on budget, and the desired relevance fidelity and distribution. Table 6.1 summarizes the pros and cons for the three pairwise approaches. The relevance ratio can collect the most precise judgments across all the relevance levels, and it is not affected by the topic difficulty, but it is the most costly. The absolute relevance judgments are not good at separating documents at low relevance levels, but can discriminate highly relevant documents. The pairwise preference can separate documents by relevance though all the relevance levels, and are relatively easy (cheap) in terms of workload. However, pairwise preference is only liked by assessors for easy topics. The absolute relevance scale is usually preferred for difficult topics.

By investigating the raw answers of each assessment question, the accuracy of answers to the gold standard pair was found to be at least 80% on average for each topic. The assessment consistency of workers was also shown to be high (98% for pairwise preference and relevance ratio, 89% for absolute relevance, on average).

According to the results of the survey, we found that factors of topic difficulty, workload, and complexity of judgment method might not affect worker's consistency of assessments, but they do relate to worker's preference of relevance scales. The most obvious factor that affected the consistency and discrimination of the judgments collected using the pairwise methods is the volume of opinions, that is, the number of pairs and the number of workers involved.

6.2 Future Work

Similarity Score Rounding In Chapter 3, we proposed a method that deliberately introduced similarity score ties by grouping documents in the run into bands. The length of bands was a geometric sequence controlled by ρ in our experiment. In practice, if the precision of similarity scores in the runs is reduced, the length of tied document groups may not be a geometric progression, but probably depends on the scoring regimes employed the system. Thus the algorithm for extending the grouping for practical runs need to be discovered, and so help to improve the efficiency of systems by reducing the precision of similarity scores without affecting the effectiveness of systems.

We only employed metrics of RR, RBP, and AP to evaluate the banded runs, and compared systems in our experiments. The generated ties might not affect the discrimination of these tested effectiveness metrics, but the results of evaluations using other metrics such as NDCG and INST might not be similar. It worth further exploring the grouping method with other datasets and metrics.

As described above, when the NIST Binary relevance judgments were used, the ties could be tolerated without greatly affecting the system comparisons. However, when the fidelity of the relevance judgments used is increased, the changes to metric scores caused by different orderings of tied documents might be larger. In Chapter 5, we used a new methodology to collect relevance judgments with high fidelity. Therefore, another possible extension to our work in Chapter 3 is to use the new collected pairwise judgments to evaluate the banded runs, and explore whether the introduced ties still have no effect on the metric scores and system comparisons.

Pooling Strategy and Residuals In Chapter 4 we proposed a method to measure the score uncertainty for recall-based metrics using the residual score computed by the utility-based metrics. We also explored whether the pooling depth k of Depth@k, the pooling strategy employed by TREC-8, affected the score uncertainty of effectiveness metrics. Using this method, the relationship between the metric score uncertainty and the pool size of other pooling strategy such as Take@N [60], BordaTake@N [3], CombTake@N [61] and so on, can also be explored. Therefore, the pooling strategies can be compared via metric score uncertainties caused by documents outside the pool, when pool sizes of the compared pooling strategies are the same.

Selecting Relevance Scale In Chapter 5, we compared the pairwise judgments collected by three methods: preference, absolute relevance, and relevance ratio. We showed the advantages and disadvantages of each method in regard to assessing time, assessor's preference, complexity, judgment consistency, similarity to other judgments, system discriminations and so on. But the question "which relevance scale is the best for collecting relevance judgments for the given topic" does not have unique answer that is suitable for every topics.

The relevance judgments with high fidelity (such as ME and the normalized pairwise judgments) can separate more documents on relevance, but they may also lead to more expensive computations when scoring systems via IR evaluation metrics due to the high precision of scores. Exploring the best number of relevance levels for collecting judgments that balances the discrimination and costs could be considered as a future work direction. Moreover, the precision of document relevance could be varied across different rank ranges. For example, the relevance score precision for top relevant documents could be high, and differences between irrelevant documents could be ignored to some extent.

Another direction for future work could be dynamically combining multiple relevance scales. If judgments are collected on the Web, methods that dynamically decide the "best" relevance scale used for collecting judgments for rest document pairs based on the collected information could be explored. In addition, document pairs could be dynamically generated according to the obtained results [20]. Combining these strategies may bring notable improvement for collecting relevance judgments with regard to both effectiveness and efficiency.

Individual Judgments Relevance is subjective to users [92], and thus there is no real ground-truth relevance judgments for every user. There are many individual factors (some of them not yet considered) that might affect user perceptions of relevance. Pairwise judgments can be obtained (for example, by analyzing search log [116]) in practical search. Therefore, it may be possible to extend our method to collect individual relevance judgments for a single user, or a group of similar users. This technology could be adapted for recommender system, or to improve personal search.

Position Bias In Chapter 5, we discovered that a slight left-right bias exists when crowd workers assess documents in pairs. Assessors tend to choose the left document 4% more than choosing the right document. This bias was found to have effect to the relevance assessment. An extension of this work could be to seek strategies to adjust the obtained judgments, and so reduce the data noise introduced by the slight left-right bias.

6.3 Summary

In this thesis, we considered similarity score ties in TREC runs, and explored possible strategies to handle ties when evaluating the effectiveness of systems. We proposed a grouping method to deliberately introduce tied similarity scores into runs, and hence explored the extent to which the ties could be allowed without affecting system comparisons. We concluded that similarity score ties could be greatly tolerated when the number of relevance categories in the employed judgments was low. That assessment needs to be further tested with high-fidelity relevance judgments.

However, as the dataset has been growing in recent years, it is prohibitive to collect judgments for all of the topic-document combinations. Moreover, efficiently collecting high-fidelity and good quality relevance judgments for pooled documents has also become a challenge. We proposed to use three pairwise relevance scales for collecting high fidelity relevance judgments for the pooled documents with low costs on the crowdsourcing platform. For documents outside the pool, we discovered a method to measure the uncertainty of metric scores caused by the unjudged documents, the confidence of the effectiveness evaluation results could therefore be quantified.

Overall, our findings contribute new methods to the field of IR evaluations, by balancing the effectiveness and costs during score computations and relevance judgment collections. We proposed a range of methods to measure the uncertainty and lose caused by reduced score precision and pool sizes, to ensure that when systems are being compared for effectiveness, we know what is being measured, and can have increased confidence in the outcomes of the measurement process.

Bibliography

[1] E. Agichtein, E. Brill, S. T. Dumais, and R. Ragno. "Learning user interaction models for predicting web search result preferences". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2006, pp. 3–10.

[2] O. Alonso, D. E. Rose, and B. Stewart. "Crowdsourcing for relevance evaluation". In: *SIGIR Forum* 42.2 (2008), pp. 9–15.

[3] J. A. Aslam and M. H. Montague. "Models for metasearch". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2001, pp. 275–284.

[4] A. H. Awadallah and R. W. White. "Personalized models of search satisfaction". In: *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*. 2013, pp. 2009–2018.

[5] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. "Relevance assessment: Are judges exchangeable and does it matter". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2008, pp. 667–674.

[6] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. "User variability and IR system evaluation". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2015, pp. 625–634.

[7] E. G. Bard, D. Robertson, and A. Sorace. "Magnitude estimation of linguistic acceptability". In: *Journal of Language* 72.1 (1996), pp. 32–68.

[8] S. Bozóki, J. Fülöp, and L. Rónyai. "On optimal completion of incomplete pairwise comparison matrices". In: *Mathematical and Computer Modelling* 52.1 (2010), pp. 318–333.

[9] N. E. Breslow and D. G. Clayton. "Approximate inference in generalized linear mixed models". In: *Journal of the American Statistical Association* 88.421 (1993), pp. 9–25.

[10] A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, and J. Zien. "Efficient query evaluation using a two-level retrieval process". In: *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*. 2003, pp. 426–434.

[11] C. Buckley and E. M. Voorhees. "Retrieval evaluation with incomplete information". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2004, pp. 25–32.

[12] C. Buckley, D. Dimmick, I. Soboroff, and E. M. Voorhees. "Bias and the limits of pooling for large collections". In: *Information Retrieval* 10.6 (2007), pp. 491–508.

[13] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender. "Learning to rank using gradient descent". In: *Proc. International Conf. on Machine Learning (ICML)*. 2005, pp. 89–96.

[14] R. Burgin. "Variations in relevance judgments and the evaluation of retrieval performance". In: *Information Processing & Management* 28.5 (1992), pp. 619–628.

[15] S. Büttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. "Reliable information retrieval evaluation with incomplete and biased judgements". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2007, pp. 63–70.

[16] B. Carterette and D. Petkova. "Learning a ranking from pairwise preferences". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2006, pp. 629–630.

[17] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais. "Here or there: Preference judgments for relevance". In: *Proc. European Conf. on Information Retrieval (ECIR)*. 2008, pp. 16–27.

[18] P. Chandar and B. Carterette. "Using preference judgments for novel document retrieval". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2012, pp. 861–870.

[19] O. Chapelle, D. Metlzer, Y. Zhang, and P. Grinspan. "Expected reciprocal rank for graded relevance". In: *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*. 2009, pp. 621–630.

[20] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. "Pairwise ranking aggregation in a crowdsourced setting". In: *Proc. ACM Int. Conf. on Web Search and Data Mining (WSDM)*. 2013, pp. 193–202.

[21] Y. Chen, K. Zhou, Y. Liu, M. Zhang, and S. Ma. "Meta-evaluation of online and offline Web search evaluation metrics". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2017, pp. 15–24.

[22] C. L. A. Clarke, F. Scholer, and I. Soboroff. "The TREC 2005 terabyte track". In: *Proc. Text Retrieval Conf. (TREC)*. 2005.

[23] C. Cleverdon. "The Cranfield tests on index language devices". In: *Aslib Proceedings*. Vol. 19. 6. 1967, pp. 173–194.

[24] J. Condorcet, F. Sommerlad, and I. McLean. *The political theory of Condorcet*. University of Oxford, Faculty of Social Studies, 1989.

[25] W. B Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.

[26] T. Damessie, F. Scholer, and J. S. Culpepper. "The influence of topic difficulty, relevance level, and document ordering on relevance judging". In: *Proc. Australasian Document Computing Symp. (ADCS)*. 2016, pp. 41–48.

[27] J. Dawes. "Do data characteristics change according to the number of scale points used? An experiment using 5-point, 7-point and 10-point scales". In: *International Journal of Market Research* 50.1 (2008), pp. 61–104.

[28] J. S. Dumas. "User-based Evaluations". In: *The human-computer interaction handbook*. Ed. by J. A. Jacko and A. Sears. L. Erlbaum Associates Inc., 2003, pp. 1093–1117.

[29] M. Eisenberg and C. L. Barry. "Order effects: A study of the possible influence of presentation order on user judgments of document relevance". In: *Journal of the American Society for Information Science (JASIS)* 39.5 (1988), pp. 293–300.

[30] R. A. Fairthorne. "Implications of test procedures". In: (1963), pp. 109–113.

[31] *Figure Eight and EMOS*. [Online; accessed 25-March-2019]. URL: `https://www.figure-eight.com/success-stories/emos/`.

[32] R. A. Fisher. "Statistical methods for research workers". In: *Breakthroughs in Statistics*. 1992, pp. 66–70.

[33] S. Fox, K. Karnawat, M. Mydland, S. T. Dumais, and T. White. "Evaluating implicit measures to improve web search". In: *ACM Trans. on Information Systems* 23.2 (2005), pp. 147–168.

[34] J. Giles. "Internet encyclopaedias go head to head". In: *Nature* 438 (2005), pp. 900–901.

[35] H. Greisdorf and A. Spink. "A new way to evaluate IR systems performance: median measure". In: *Proc. National Online Meeting (NOM)*. 2000, pp. 137–144.

[36] L. Han, K. Roitero, E. Maddalena, S. Mizzaro, and G. Demartini. "On transforming relevance scales". In: *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*. 2019, pp. 39–48.

[37] D. K. Harman. "The TREC Test Collections". In: *TREC: Experiment and evaluation in information retrieval*. Ed. by E. M. Voorhees and D. K. Harman. MIT Press, 2005. Chap. 2, pp. 21–52.

[38] S. P. Harter. "Psychological relevance and information science". In: *Journal of the American Society for information Science (JASIST)* 43.9 (1992), pp. 602–615.

[39] K. Hofmann. "Online experimentation for information retrieval". In: *Information Retrieval - 8th Russian Summer School (RuSSIR)*. 2014, pp. 21–41.

[40] J. Howe. "Pure, unadulterated (and scalable) crowdsourcing". In: *Crowdsourcing: Tracking the rise of the amateur* 15 (2006).

[41] J. Howe. "The rise of crowdsourcing". In: *Wired Magazine* 14.6 (2006), pp. 1–4.

[42] M. Huang and H. Wang. "The influence of document presentation order and number of documents judged on users' judgments of relevance". In: *Journal of the American Society for Information Science and Technology (JASIST)* 55.11 (2004), pp. 970–979.

[43] InnoCentive. *About Innocentive*. [Online; accessed 15-March-2019]. URL: `http://www.innocentive.com/about/index.html`.

[44] K. Järvelin and J. Kekäläinen. "Cumulated gain-based evaluation of IR techniques". In: *ACM Trans. on Information Systems* 20.4 (2002), pp. 422–446.

[45] J. Jiang and J. Allan. "Adaptive persistence for search effectiveness measures". In: *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*. 2017, pp. 747–756.

[46] T. Joachims, L. A. Granka, B. Pan, H. Hembrooke, and G. Gay. "Accurately interpreting clickthrough data as implicit feedback". In: *SIGIR Forum* 51.1 (2017), pp. 4–11.

[47] R. V. Katter. "The influence of scale form on relevance judgments". In: *Information Storage and Retrieval* 4.1 (1968), pp. 1–11.

[48] D. Kelly. "Methods for evaluating interactive information retrieval systems with users". In: *Foundations & Trends in Information Retrieval* 3.1 (2009), pp. 1–224.

[49] D. Kelly and C. R. Sugimoto. "A systematic review of interactive information retrieval evaluation studies, 1967–2006". In: *Journal of the American Society for Information Science and Technology (JASIST)* 64.4 (2013), pp. 745–770.

[50] M.G. Kendall. "Rank correlation methods". In: *Charles Griffin Book Series* (1975).

[51] Y. Kim, A. H. Awadallah, R. W. White, and I. Zitouni. "Modeling dwell time to predict click-level satisfaction". In: *Proc. ACM Int. Conf. on Web Search and Data Mining (WSDM)*. 2014, pp. 193–202.

[52] B. Koopman and G. Zuccon. "Why assessing relevance in medical IR is demanding". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2014, pp. 16–19.

[53] K. Krippendorff. "Content analysis: An introduction to its methodology". In: *Personnel Psychology* 57.4 (2004), p. 1110.

[54] K. R Lakhani, L. B. Jeppesen, P. A. Lohse, and J. A Panetta. *The value of openness in scientific problem solving*. Division of Research, Harvard Business School Boston, MA, 2007.

[55] J. Langford, A. L. Strehl, and J. Wortman. "Exploration scavenging". In: *Proc. Int. Conf. on Machine Learning (ICML)*. 2008, pp. 528–535.

[56] L. Li, W. Chu, J. Langford, and R. E. Schapire. "A contextual-bandit approach to personalized news article recommendation". In: *Proc. Conf. on the World Wide Web (WWW)*. 2010, pp. 661–670.

[57] R. Likert. "A technique for the measurement of attitudes." In: *Archives of Psychology* 22.140 (1932), pp. 1–55.

[58] A. Lipani. "Fairness in information retrieval". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2016, p. 1171.

[59] A. Lipani, M. Lupu, and A. Hanbury. "Splitting water: Precision and anti-precision to reduce pool bias". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2015, pp. 103–112.

[60] A. Lipani, M. Lupu, and A. Hanbury. "The curious incidence of bias corrections in the pool". In: *Proc. European Conf. on Information Retrieval (ECIR)*. 2016, pp. 267–279.

[61] A. Lipani, M. Lupu, J. R. M. Palotti, G. Zuccon, and A. Hanbury. "Fixed budget pooling strategies based on fusion methods". In: *Proc. ACM Symp. Applied Computing (SAC)*. 2017, pp. 919–924.

[62] D. E. Losada, J. Parapar, and A. Barreiro. "When to stop making relevance judgments? A study of stopping methods for building information retrieval test collections". In: *Journal of the Association for Information Science and Technology (JASIST)* 70.1 (2019), pp. 49–60.

[63] X. Lu, A. Moffat, and J. S. Culpepper. "Can deep effectiveness metrics be evaluated using shallow judgment pools?" In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2017, pp. 35–44.

[64] J. Mackenzie, F. Scholer, and J. S. Culpepper. "Early termination heuristics for score-at-a-time index traversal". In: *Proc. Australasian Document Computing Symp. (ADCS)*. 2017, 8:1–8:8.

[65] E. Maddalena, S. Mizzaro, F. Scholer, and A. Turpin. "On crowdsourcing relevance magnitudes for information retrieval evaluation". In: *ACM Trans. on Information Systems* 35.3 (2017), 19:1–19:32.

[66] J. Mao, Y. Liu, K. Zhou, J. Nie, J. Song, M. Zhang, S. Ma, J. Sun, and H. Luo. "When does relevance mean usefulness and user satisfaction in Web search?" In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2016, pp. 463–472.

[67] M. S. Matell and J. Jacoby. "Is there an optimal number of alternatives for Likert-scale items? Effects of testing time and scale properties." In: *Journal of Applied Psychology* (1972).

[68] M. S. Matell and J Jacoby. "Is there an optimal number of Likert scale items? Study I: Reliability and validity". In: *Educational and Psychological Measurement* 31.3 (1971), pp. 657–674.

[69] F. McSherry and M. Najork. "Computing Information Retrieval performance measures efficiently in the presence of tied scores". In: *Proc. European Conf. on Information Retrieval (ECIR)*. 2008, pp. 414–421.

[70] A. Moffat, F. Scholer, and P. Thomas. "Models and metrics: IR evaluation as a user process". In: *Proc. Australasian Document Computing Symp. (ADCS)*. 2012, pp. 47–54.

[71] A. Moffat, P. Thomas, and F. Scholer. "Users versus models: What observation tells us about effectiveness metrics". In: *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*. 2013, pp. 659–668.

[72] A. Moffat, W. Webber, and J. Zobel. "Strategic system comparisons via targeted relevance judgments". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2007, pp. 375–382.

[73] A. Moffat and J. Zobel. "Rank-biased precision for measurement of retrieval effectiveness". In: *ACM Trans. on Information Systems* 27.1 (2008), pp. 2.1–2.27.

[74] A. Moffat, P. Bailey, F. Scholer, and P. Thomas. "Incorporating user expectations and behavior into the measurement of search effectiveness". In: *ACM Trans. on Information Systems* 35.3 (2017), 24:1–24:38.

[75] A. Moffat, P. Bailey, F. Scholer, and P. Thomas. "INST: An adaptive metric for information retrieval evaluation". In: *Proc. Australasian Document Computing Symp. (ADCS)*. 2015, 5:1–5:4.

[76] S. D. Ravana and A. Moffat. "Score aggregation techniques in retrieval experimentation". In: *Database Technologies 2009, Twentieth Australasian Database Conference (ADC)*. 2009, pp. 59–67.

[77] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979.

[78] S. Robertson. "A new interpretation of average precision". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2008, pp. 689–690.

[79] S. Robertson. "On GMAP: And other transformations". In: *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*. 2006, pp. 78–83.

[80] K. Roitero, E. Maddalena, G. Demartini, and S. Mizzaro. "On fine-grained relevance scales". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2018, pp. 675–684.

[81] T. Sakai. "Alternatives to BPref". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2007, pp. 71–78.

[82] T. Sakai. "Comparing metrics across TREC and NTCIR: The robustness to pool depth bias". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2008, pp. 691–692.

[83] T. Sakai. "Comparing metrics across TREC and NTCIR: The robustness to system bias". In: *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*. 2008, pp. 581–590.

[84] T. Sakai. "Conducting laboratory experiments properly with statistical tools: An easy hands-on tutorial". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2018, pp. 1369–1370.

[85] T. Sakai. "Metrics, statistics, tests". In: *Bridging Between Information Retrieval and Databases - PROMISE Winter School*. 2013, pp. 116–163.

[86] T. Sakai. "Statistical reform in information retrieval?" In: *SIGIR Forum* 48.1 (2014), pp. 3–12.

[87] T. Sakai. "Statistical significance, power, and sample sizes: A systematic review of SIGIR and TOIS, 2006-2015". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2016, pp. 5–14.

[88] T. Sakai and N. Kando. "On information retrieval metrics designed for evaluation with incomplete relevance assessments". In: *Information Retrieval* 11.5 (2008), pp. 447–470.

[89] G. Salton and M. Lesk. "Computer evaluation of indexing and text processing". In: *Journal of the ACM* 15.1 (1968), pp. 8–36.

[90] M. Sanderson. "Test collection based evaluation of information retrieval systems". In: *Foundations and Trends in Information Retrieval* 4.4 (2010), pp. 247–375.

[91] M. Sanderson and J. Zobel. "Information retrieval system evaluation: effort, sensitivity, and reliability". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2005, pp. 162–169.

[92] T. Saracevic. *The notion of relevance in information science: Everybody knows what relevance is. But, what is it really?* Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2016.

[93] L. Schamber. "Relevance and information behavior." In: *Annual Review of Information Science and Technology (ARIST)* 29 (1994), pp. 3–48.

[94] F. Scholer and A. Turpin. "Metric and relevance mismatch in retrieval evaluation". In: *Proc. Asia Information Retrieval Societies Conf. (AIRS)*. 2009, pp. 50–62.

[95] F. Scholer, A. Turpin, and M. Sanderson. "Quantifying test collection quality based on the consistency of relevance judgements". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2011, pp. 1063–1072.

[96] F. Scholer, A. Turpin, and M. Wu. "Measuring user relevance criteria". In: *Proc. Evaluating Information Access (EVIA)*. 2008, pp. 50–62.

[97] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press, 2003.

[98] Mark D. Smucker, James Allan, and Ben Carterette. "A comparison of statistical significance tests for information retrieval evaluation". In: *Proc. ACM Int. Conf. on Information and Knowledge Management (CIKM)*. 2007, pp. 623–632.

[99] E. Sormunen. "Liberal relevance criteria of TREC: Counting on negligible documents?" In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2002, pp. 324–330.

[100] K. Spärck Jones and C. Van Rijsbergen. *Report on the Need for and Provision of an "Ideal" Information Retrieval Test Collection. British Library Research and Development Report 5266*. Tech. rep. Computer Laboratory, University of Cambridge, 1975.

[101] K. Spärck Jones and P. Willett. "Evaluation". In: *Readings in Information Retrieval*. Ed. by K. Spärck Jones and P. Willett. Morgan Kaufmann Publishers Inc., 1997, pp. 167–174.

[102] C. Spearman. "The proof and measurement of association between two things". In: *The American Journal of Psychology* 15.1 (1904), pp. 72–101.

[103] A. Tonon, G. Demartini, and P. Cudré-Mauroux. "Pooling-based continuous evaluation of information retrieval systems". In: *Information Retrieval* 18.5 (2015), pp. 445–472.

[104] A. Turpin and F. Scholer. "Modelling disagreement between judges for information retrieval system evaluation". In: *Proc. Australasian Document Computing Symp. (ADCS)*. 2009, pp. 51–58.

[105] A. Turpin, F. Scholer, S. Mizzaro, and E. Maddalena. "The benefits of magnitude estimation relevance assessments for information retrieval evaluation". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2015, pp. 565–574.

[106] H. R. Turtle and J. Flood. "Query evaluation: Strategies and optimizations". In: *Inf. Process. Manage.* 31.6 (1995), pp. 831–850.

[107] E. M. Voorhees. "Overview of the TREC 2004 robust retrieval track". In: *Proc. Text Retrieval Conf. (TREC)*. 2004.

[108] E. M. Voorhees. "Overview of the TREC 2005 robust retrieval track". In: *Proc. Text Retrieval Conf. (TREC)*. 2005.

[109] E. M. Voorhees. "Overview of TREC 2001". In: *Proc. Text Retrieval Conf. (TREC)*. 2001.

[110] E. M. Voorhees. "Overview of TREC 2004". In: *Proc. Text Retrieval Conf. (TREC)*. 2004.

[111] E. M. Voorhees. "The philosophy of information retrieval evaluation". In: *Workshop of the Cross-Language Evaluation Forum (CLEF) for European Languages*. 2001, pp. 355–370.

[112] E. M. Voorhees and D. K. Harman. "Overview of the eighth Text REtrieval Conference". In: *Proc. Text Retrieval Conf. (TREC)*. NIST Special Publication 500-246. 1999.

[113] E. M. Voorhees and D. K. Harman. "Overview of the seventh Text REtrieval Conference". In: *Proc. Text Retrieval Conf. (TREC)*. 1998.

[114] Y. Wang, H. Wu, and H. Fang. "An exploration of tie-breaking for microblog retrieval". In: *Proc. European Conf. on Information Retrieval (ECIR)*. 2014, pp. 713–719.

[115] W. Webber, A. Moffat, and J. Zobel. "A similarity measure for indefinite rankings". In: *ACM Trans. on Information Systems* 28.4 (2010), 20:1–20:38.

[116] A. F. Wicaksono and A. Moffat. "Exploring interaction patterns in job search". In: *Proc. Australasian Document Computing Symp. (ADCS)*. 2018, 2:1–2:8.

[117] A. F. Wicaksono, A. Moffat, and J. Zobel. "Modeling user actions in job search". In: *Proc. European Conf. on Information Retrieval (ECIR)*. 2019, pp. 652–664.

[118] Y. Xu and D. Wang. "Order effect in relevance judgment". In: *Journal of the Association for Information Science and Technology (JASIST)* 59.8 (2008), pp. 1264–1275.

[119] F. Zhang, Y. Liu, X. Li, M. Zhang, Y. Xu, and S. Ma. "Evaluating Web search with a Bejeweled Player Model". In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 2017, pp. 425–434.

[120] J. Zobel. "How reliable are the results of large-scale information retrieval experiments?" In: *Proc. ACM Int. Conf. on Research and Development in Information Retrieval (SIGIR)*. 1998, pp. 307–314.