Martin-Luther-Universität Halle-Wittenberg
Institute for Informatics
Degree Programme Informatik Bachelor (180LP)

Evaluation of Ad-Hoc Information Retrieval Systems on Web Crawls with Redundant Documents

Bachelor's Thesis

Jan Philipp Bittner
Born March 02, 1998 in Minden

Matriculation Number: 216243707

1. Referee: Prof. Dr. Matthias Hagen
2. Referee: Maik Fröbe

Submission date: July 13, 2020

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Halle (Saale), July 13, 2020

. . . . . .
Jan Philipp Bittner

Abstract

In this thesis, we study the occurrence of duplicates among corpora that consist of crawled web documents and corresponding TREC Ad-Hoc tracks. We study the impact of the novelty principle on the evaluation of TREC Ad-Hoc tracks. The motivation to assess the impact is the discrepancy between the information need of search engine users and the methods used to evaluate information retrieval systems. The information need of a search engine user stays unfulfilled if equivalent documents are shown, information retrieval systems are even rewarded for retrieving duplicates. We evaluate an approach to mitigate the impact of duplicates on the evaluation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the past years, searching for information on the web has become more critical than ever. The internet provides access to an unimaginable number of documents. Whenever we use a search engine, we formulate a query for an information retrieval system. "[T]he primary goal of an [information retrieval] system is to retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible" [BR11].

The development of information retrieval systems is driven by the evaluation of algorithms in terms of the ability to identify relevant documents [BR11]. The Cranfield paradigm was developed in the 1960s in the Cranfield experiments and laid the cornerstone for information retrieval system evaluations. The Cranfield paradigm dictates how an information retrieval system has to be evaluated so that the evaluation is reproducible and produces meaningful results. Test collections after the Cranfield paradigm contain a set of information needs (topics), a set of documents to be searched (corpus), and relevance judgments indicating which documents are relevant for which topics. Typically, an information retrieval system is tested with 50 different topics to ensure that the results are significant since only through repeated tests can meaningful information be derived. The judgments are created by hand from humans. The desired properties of the judgments are that they are relevant for the topic, independent of the time, independent of the user, and independent of retrieving other documents. The system that we want to measure stores it is results for the topics in a file. This allows us to measure the system's performance in a controlled environment. The performance of an information retrieval system is measured through the relevance judgments of the documents it retrieved and the corresponding position in the result list.

The Text REtrieval Conference (TREC) conducts "contests" to benchmark the performance of information retrieval systems, also known as tracks. The procedure is as follows. The track's organizers specify which corpus is used and

publish topics. The participants submit their system's results to the organizers. Then a subset of the retrieved documents out of all submissions is judged. Each document of the subset is individually judged on whether it is relevant for the corresponding topic. With these judgments, the performance of a system can be evaluated.

A document is considered novel if no equivalent document has ever been presented to the user. The novelty of a document is essential as in web search, a user's information need is not satisfied by retrieving an equivalent document. Bernstein and Zobel state that "[i]f a document is effectively identical to documents that have already been presented, then it is unnecessary for the user to see it". Furthermore, they showed in their experiments that widely used corpora consisting of crawled web document contain a significant number of content-equivalent documents. They showed that these redundant documents have a significant influence on the evaluation of information retrieval systems.

Bernstein and Zobel's experiments motivate that content-equivalent documents, which are expected in general web crawls, should be considered within the evaluation of information retrieval systems [BZ05]. Search engines should not be rewarded for retrieving a content-equivalent document of a document already shown to the user.

Inspired by Bernstein and Zobel's experiments, we conduct similar experiments on newer tracks and bigger corpora that consists of up to 1.2 billion documents. We study the occurrence of equivalent documents among the corpora. Specifically, we identify equivalent documents by their fingerprints and compare our results to other studies. Furthermore, we reproduce the experiments of Bernstein and Zobel for the .GOV corpora. Aside from fingerprinting, we identify near-duplicates in the tracks' judged documents with a more sophisticated approach. We reproduce and expand Bernstein and Zobel's experiments regarding the impact of duplicates on the tracks. We confirm that novelty-awareness has a significant impact on the performance. In the web track 2010, only 2 of the top 5 submissions remained in the top five, if duplicates are irrelevant. This reveals a problem since a user expects content that he has not seen yet, but the benchmarks, used to assess search engine performance, do not take novelty into account. Thus the evaluations do not reflect the user experience. An information retrieval system can score well in benchmarks but offers a poor user experience in real applications. The benchmark might devalue information retrieval systems that respect the novelty. This can invalidate research results based on these benchmarks. Parts of our results have been published at the European Conference on Information Retrieval (ECIR) in 2020 [Frö+20].

The negative effects of content-equivalent documents are larger on collections created after Bernstein and Zobel's findings were published. This means

either Bernstein's and Zobel's findings were not taken into account during the creation of the test collections, or the efforts to mitigate the impact failed. Addressing this gap, we discuss whether track organizers can reduce the overall impact of duplicates on the evaluation by removing the topics most affected by duplicates. Ideally, the organizers should be able to remove these topics before the judgments have been made so that no manual effort is wasted. We compare three scenarios with different levels of manual effort involved, starting with no judgments, and reaching up to conventional judgments.

Chapter 2

Related Work

Our work is closely related to near-duplicate-detection, and evaluation in information retrieval. Redundant documents can be detected by comparing documents directly. Allowing the use of all information in the documents, a direct comparison of documents can very accurately and gradually determine overlapping information.

Bernstein and Zobel showed that the .GOV 1 and .GOV 2 corpora contain a high number of redundant documents. Furthermore, they found "that 16.6% of all relevant documents in runs submitted to the TREC 2004 terabyte track were redundant."[BZ05]. They evaluated the impact of the novelty principle on the Terabyte 2004 Track.

Broder introduced the resemblance-score, which is designed to detect near-duplicates [Bro97]. The resemblance-score assembles the text chunks of fixed length of two documents in a two sets. The resemblance is the Jaccard index of the documents' sets.

Clarke et al. propose a similarity measure which uses binary document properties (nuggets) to asses the novelty of a document [Cla+08]. An ideal search engine within Clarkes similarity measure maximizes the number of distinct nuggets in the retrieved documents.

Zhang et al. propose a measure based on language models[ZCM02]. The measure calculates the amount of lost information if the language model of the old document approximates the language model of the new document using the Kullback–Leibler divergence. The language model is build by finding the language model with the highest likely-hood of producing the document together with a language model for the topic and for the (natural) language e.g. English.

The normalized discounted cumulative gain, also known as the nDCG-score, was first introduced by Järvelin and Kekäläinen [JK02], and is a standard measure for the evaluation of ad-hoc retrieval tracks. Compared to the MAP,

the nDCG-score has the advantage that it is receptive for fine grained judgments, as it directly uses the relevance. A document with relevance of 10 is much more relevant that a document with relevance 1. In contrast, the MAP only differentiates whether a documents is relevant or not. Since the relevance of a document depends on the topic, the systems obtain one nDCG-score per topic. Averaging the nDCG-scores of all topics provides the systems' overall nDCG-score. The nDCG derives from the cumulative gain (CG), which is the sum of the retrieved documents' relevance. Any document without a judgment has a relevance of 0. Let doc_i be the document at the position i in a systems result list. Let $rel(doc_i)$ be the relevance of the doc_i.

$$\mathrm{CG}(doc_1, ..., doc_n) := \sum_{i=1}^{n} rel(doc_i)$$

Discounting a document's relevance by the log of its position in the result list defines the discounted cumulative gain (DCG). The further down the documents appear in a system's submission, the higher is the discount. As in the official evaluation of the TREC conferences, we use the log base two.

$$\mathrm{DCG}(doc_1, ..., doc_n) := \sum_{i=1}^{n} \frac{rel(doc_i)}{log_2(i+1)}$$

The system with the highest DCG possible, retrieves all relevant documents ordered by their relevance and none of the irrelevant documents. To the score of such a system is referred to as DCG_I. Normalizing the DCG by the DCG_I defines the normalized discounted cumulative gain (nDCG).

$$\mathrm{nDCG}(doc_1, ..., doc_n) := \frac{\mathrm{DCG}(doc_1, ..., doc_n)}{DCG_I}$$

Chapter 3

Retrieval-Equivalence in Corpora

We adhere to Bernsteins and Zobels definition of retrieval-equivalence as "an easy-to-compute restricted form of content equivalence motivated by the operation of search engines" [BZ05]. In our experiment we consider documents retrieval-equivalent if their fingerprints have the same hash value. The fingerprint is the result of preprocessing the document's text and hashing the result. Through the use of hash values, we can identify equivalent documents across billions of documents. We analyze the extent of retrieval-equivalence in the .GOV Web Research Collections 1 and 2 (.GOV/.GOV2), Clueweb09, Clueweb12, NYT-AC and the Washinton Post corpora. Bernstein and Zobel analyzed the .GOV1 and .GOV2 corpora with the same methodology [BZ05]. We are able to reproduce their results and observe high numbers of duplicates, even on very recent corpora. This may pose a in the when comparing retrieval systems on these corpora.

As we only need the document to derive its hash value, the entire process is highly parallelizable. We use Apache Spark, an open-source program used to automate the organization of distributed computing and allows operations similar to the MapReduce programming model in many different programming languages. We compute the groups of equivalent documents on a cluster of 145 nodes and 1740 cores. The spark library splits the work into pieces and distributes the pieces across the cluster nodes for simultaneous processing. In the first phase, each hash value is computed and stored alongside the document id on the file system shared across the cluster. The document ids are grouped by the corresponding hash value, which is the second phase. The final result is a list of all groups of retrieval-equivalent documents and the corresponding hash values.

The hash value of every document in the corpus is computed in six steps. We arrange these six steps in the following pipeline:

(1) Extracting Text: As the documents are stored in the Hyper-Text-Markup-Language (HTML), they contain a manifold of information not part of the document's text. We remove the parts of the document that are invisible to the user using JSOUP, an open-source Java HTML-parser[1]. (2) Lowercase: As two texts with a difference in capitalization that are otherwise the same would be regarded as equivalent by a user, every capital letter is lowered. (3) Replace English Stop Words and Separators: In languages, there are some very frequently used words. A stop word is defined as a word with the same likelihood to occur in any given document. There are also domain-specific stop words, but as web documents, in general, do not share a common domain, we use a list of generic English stop words with prime examples like "the", "of", "to". As these words can appear in any context, they do not convey the text's information. Therefore, we replace the stop words with whitespace using the StopFilter and the English stop words of the open-source Lucene project. The same argument can be made for punctuation, hyphenation, and separators alike, which is why we replace all separators with whitespaces as well using Luecenes implementation of the Unicode Standard Annex #29 specification. (4) Collapse Whitespace Sequences: Under the assumption that no two reasonable texts significantly differ by a sequence of whitespaces, we collapse them into one whitespace character. Conveniently this also collapses any whitespace sequences caused by the replacement of stop words or punctuation. (5) Word Stemming: A stem of a word is shared by all variations the same word. To stem the words, we use an implementation of the Porter-Stemmer, which is based on a set of rules optimized for stemming words in English. The stem derived from the porter-stemmer of the words "interesting", "interest", "interested" is "interest", which is also the linguistic root of these words, though the stem of a word is not necessarily the linguistic root of the word. For example, the stem of "highly" is not "high", but "highli". With word-stemming texts that only differ by a variation of the same word become identical. (6) Hashing: A hash function maps a sequence of characters to a bit array of fixed size. To simplify the identification of identical documents, we hash the stemmed texts and build a key-value pair of the document id and the hash value. Depending on the hash function, it may be possible, but very unlikely, for two different texts to have the same hash value. We use the MD5 hash function, which provides hash values of the length of 128 bit. The likelihood of a hash collision between two texts is negligible. The collision probability on the largest corpus we study is

[1] https://jsoup.org/

approximately $2.11 * 10^{-21}$ [2]. The hash value of a document is much smaller than the original document and thus faster to compare. The properties of the MD5 hash function ensure that even slightly different text representations are reduced to very different hash values. The result of the first phase is a key-value-pair for each document in the corpus, where the key is the document id, and the value is the hash representation of the document's text.

The time complexity of the first phase, which maps documents to their hash representations, grows linearly with the number of documents. Nevertheless, it took the overwhelming majority of the computing time in our experiments. Both the effort of parsing the documents from the archive files and transforming them into hash values contributed to the computing time. Another factor might have been our infrastructure, which required us to read the documents from a network file system, thus limiting the throughput. In contrast, the second phase reads from the file system distributed across the cluster. Spark optimizes the schedule for data localization, which means that in the second phase, ideally, no remote readings take place, and the bottlenecks associated with the file system are reduced.

In the second phase, the key-value pairs of a document id and the associated hash value are grouped by their hash value. Grouping is an often occurring, cross-domain problem and can be parallelized to deal with large data sets like the Clueweb09 and Clueweb12 data sets. We use the parallel "reduceByKey" function of the spark library to solve our instances of the grouping problem faster using distributed computing. As each document is represented through a key-value pair of a hash value and a document id, finding equivalent documents in the corpus is reduced to sorting the key-value pairs by their key. First, the key-value pairs are sorted by the hash value. Then the sorted pairs can easily be grouped by iterating over the pairs. The process is parallelized by splitting the pairs into partitions based on their key. Splitting by the key ensures that the partitions can be independently sorted, grouped, and stored in the shared file system. The partitions are then distributed over the cluster nodes. Partitioning the data is linear in the number of documents, and grouping each partition has a theoretical time complexity in $O(p * log(p))$, where p is the partition size.

The result is a list of groups where every group consists of identical documents. Through the described procedure, we can not only replicate the original experiment of Bernstein and Zobel but also apply the same procedure to the newer and bigger data sets Clueweb09 and Clueweb12 to see whether the situation has changed in the meantime. Table 3.1 shows the sizes of the .GOV

[2] Assuming that the hash values are evenly distributed, we can calculate the probability of a hash collision using the Taylor expansion of the exponential function. The approximation for the largest corpus is $e^{-\frac{k*(k-1)}{2*n}}$, where k is the size of the corpus ($1.2 * 10^9$) and n the number of hash values (2^{128}).

Corpus	Size		Equivalent	
	Gzip	Docs	Duplicates	Groups
.GOV 1	4.6 GB	1.2m	6.54%	24,394
.GOV 2	81 GB	25.1m	23.39%	794.889
ClueWeb09	4.0 TB	1.04b	7.74%	49.2m
ClueWeb12	4.6 TB	0.73b	14.71%	39.1m
NYT-AC	8.3 GB	1.8m	2.11%	28,493
WaPo	1.6 GB	0.59m	12.52%	28,471

Table 3.1: The table lists the examined corpora and there associated size on the file system and documents-wise, and the number of duplicates as well as the number of groups of equivalent documents.

corpora contrasted against other corpora of our study, as well as the relative number of duplicates and the number of groups we found. Figure 3.1 shows the group size against the number of groups of any given size on log-axes. On all corpora, the observations fit a zipfy power-law distribution, which is consistent with other studies on the distribution of the group sizes [Hen06; FMN04]. Our .GOV2 results show strong similarities to the findings of Bernstein and Zobel [BZ05].

The oldest and smallest corpus we examine is the original .GOV corpus. It is a web crawl of over 1.2 million HTML documents and extracted texts of other document types from the .gov domain at the beginning of 2002 [CH02]. According to Craswell et al. the crawl itself had a size of 35 gigabyte, which was reduced to 18 gigabyte before its distribution by pruning each document after 100 kilobyte. The compressed corpus has a size of 4.6 gigabyte. Our results show that 6.54% of all documents are exact duplicates, which amounts to 78.48 thousand documents in over 24.39 thousand groups. For example, given 100 equal documents in a group, there are 99 duplicates in this group. There are many small groups with less than ten documents and only a small number of groups with a size greater or equal to 100.

We also examine its successor the .GOV2 corpus, which has 25 times more documents than its predecessor. It includes over 25 million HTML documents and extracted text of other document types from the .gov domain at the beginning of 2004 [CCS04]. The .GOV2 corpus has a size of 426 gigabyte and 81 gigabyte uncompressed, respectively. This corpus showed an exceptionally high number of exact duplicates in our experiments. Roughly 23.39% of all documents are duplicates. These 5.87 million documents are represented in over 794 thousand groups.

Comparing our number of groups in the original .GOV corpus and the .GOV2 corpus to Bernstein and Zobel's (22,870 and 865,362 classes, respectively), we

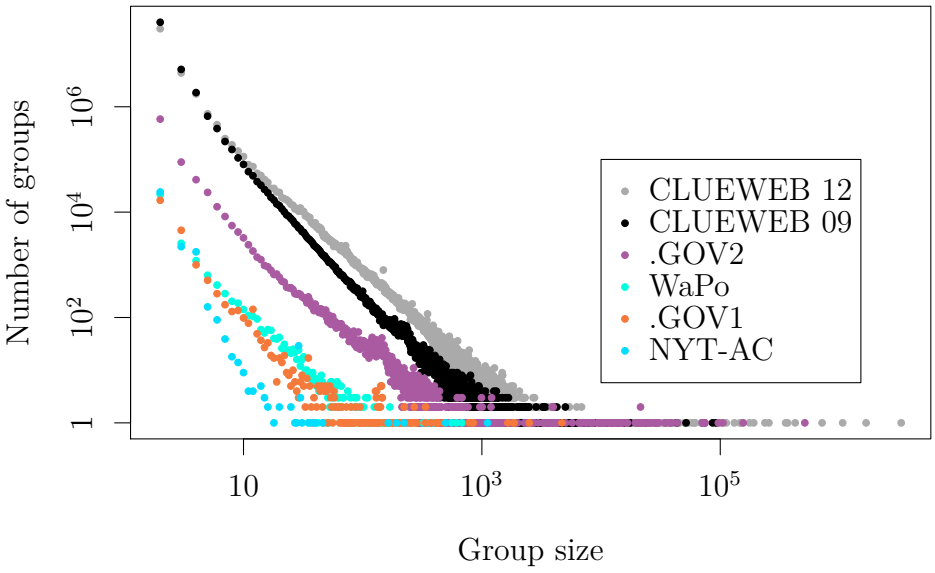


Figure 3.1: Log-scale plot of group size against the number group with the according size.

see that our results deviate from their results by only about 1%. Our findings for the percentage of duplicates on both corpora deviates from their results (6.36% and 24.31%) by less than 1 percentage point. Given the corpus sizes, this is sufficiently close to say that we can successfully reproduce their experiments.

The Clueweb09 corpus is by far the biggest corpus we examine in terms of documents. It consists of over 1 billion documents, has a size of 25 terabytes, and a compressed a size of 4 terabytes [3]. The documents were crawled from the general web using web pages from another corpus and top-ranked web pages from commercial search engines as starting points [4]. It also includes a full copy of the English Wikipedia at the time of creating the corpus. In the Clueweb09 corpus, we see that roughly 7.74% of all documents are exact duplicates or 80.49 million duplicates in over 49.2 million groups.

The Clueweb12 corpus is the successor of the Clueweb09 corpus. It consists of over 733 million documents and has a compressed size of 4.6 terabyte. The documents were crawled from the general web with a list of starting points. The first version of the corpus contained a huge amount of duplicates due to a

[3] http://lemurproject.org/clueweb09.php/
[4] http://boston.lti.cs.cmu.edu/Data/web08-bst/planning.html

software bug. We use a later version of the corpus which addressed this specific bug. [5] Nevertheless, we find that 14.71% of all documents are duplicates, which are 107 million duplicates in over 39.1 million groups.

For comparison, we also examined two corpora that do not consist of crawled web data. The New York Times Annotated Corpus consists of over 1.8 million new articles published between early 1987 and mid-2007 [San08]. In our experiments, we found about 2.11% of retrieval-equivalent duplicates, much less than in the other corpora. The other corpus we additionally feature is the Washington Post Corpus. It consists of over 600,000 Washington Post articles, columns, and blog posts from January 2012 through December 2019 [6]. There are duplicates in the corpus because "the Washington Post will republish an article, and the provenance history is not represented in the data" [SHH18]. We use version two of this corpus, on which some effort has been made to reduce duplicates. Nevertheless, we find over 12% retrieval-equivalent duplicates in over 28 thousand groups.

Our experiments on the retrieval-equivalence among corpora's documents show that there is a significant number of duplicates among all examined corpora. Furthermore, we can reproduce the results of Bernstein and Zobel. As Fetterly et al. and Henzinger, we observe a zipfy power distribution on the number of groups of a given size.

Of course, grouping documents by their hash value can only identify groups that consist of practically identical documents. Consequently, there are documents with different hash values a search engine user considers to be equivalent. For example, this is the case with documents that only differ by a timestamp. To find these near-duplicates a more sophisticated approach is employed in the next chapter.

[5] http://lemurproject.org/clueweb12.php/
[6] https://trec.nist.gov/data/wapost/

Chapter 4

Content-Equivalence in TREC Tracks

In the last chapter, we identified equivalent documents with identical fingerprints, which enabled us to find duplicates in large data sets comprising up to 1.2 billion documents. This approach fails to identify documents with the same content but minor differences, such as the documents shown in Figure 4.1. To find such near-duplicates, we use the Spex algorithm, which calculates the S3-score to identify so-called content-equivalent documents based on the amount of shared information between the documents. We study content-equivalent documents in the corpora presented in the last chapter, and reproduce the results of Bernstein and Zobel [BZ05] for the .GOV corpora.

The S3-Score measures the number of text chunks of fixed length shared by two documents. Bernstein and Zobel introduced it as a computationally less expensive version of the Broder resemblance score [Bro97]. The length of the text chunks determines the properties of the S3-score. On the one hand, short text chunks cause documents about the same topic to score higher because non-equivalent documents about the same topic share many of the same words. On the other hand, long text chunks cause near-duplicates to score lower because there are less exactly equivalent chunks. We use text chunks of length eight, also known as 8-grams, in which we find a reasonable balance between the specificity and the sensitivity of the S3-score. Figure 4.2 illustrates all the 8-grams of an example text. Let D_a, D_b be the set all 8-grams of document a and document b respectively. Then the S3-score is defined as follows:

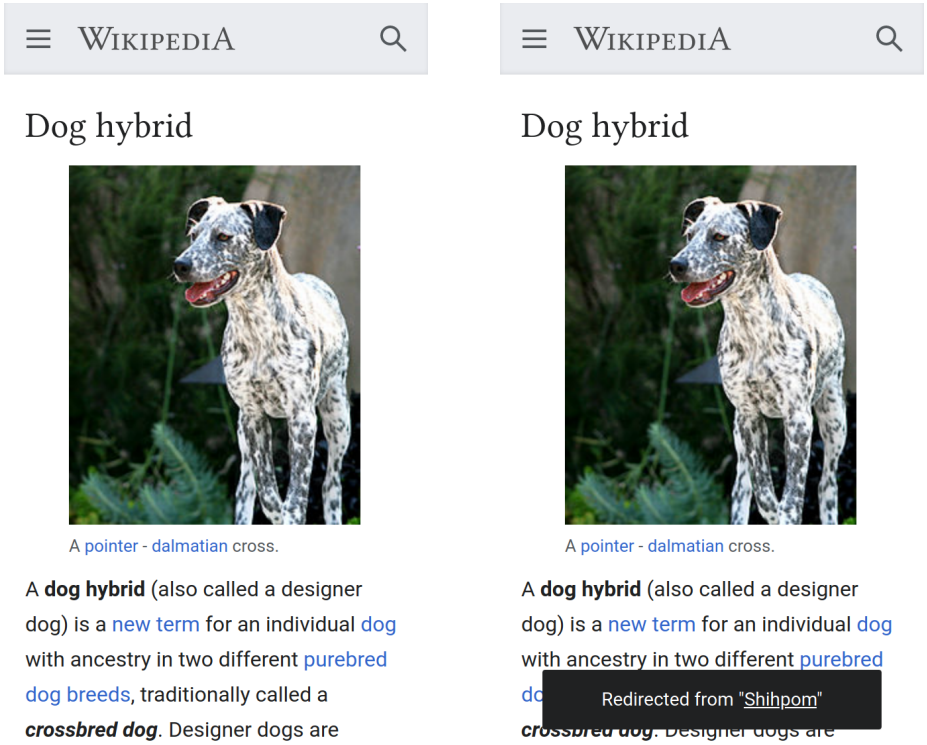
$$\text{S3-Score}(D_a, D_b) := \sum_{c \in D_a \wedge c \in D_b} \frac{1}{mean(|D_a|, |D_b|)} = \frac{|D_a \cap D_b|}{mean(|D_a|, |D_b|)}$$

Two documents are content-equivalent if the S3-score of these documents is above a certain threshold. We conducted a user study for the tracks on the

.GOV2 corpus, the Clueweb corpora, the NYT-AC corpus, and the Washington Post corpus. For each track, we sampled 100 document pairs, so that the documents are evenly distributed between an S3-score of 0.4 and 1. The participants were asked to rate a document pair on its equivalence from zero to three. Zero means that the documents are not equal under any circumstances. One means that if both documents are shown in the result of a reasonable search engine for a query, then the documents are nearly content-equivalent. Two means that if both documents are shown in the result of a reasonable search engine for a query, then the documents are content-equivalent. Three means that the documents are completely content-equivalent. From the user feedback, we derive the threshold so that the S3-score correctly classifies 95% of the document pair with a rating of one or higher. We found that a threshold of 0.68 for the GOV2 corpus, 0.84 for the Clueweb corpora, and 0.68 for NYT-AC and Washington Post corpora achieve an overall accuracy of 0.95.

In the following paragraphs, we explain our implementation of the Spex algorithm. We calculate the S3-Score of all document pairs in two phases on a cluster using the Apache Spark. In the first phase, we calculate lists of documents where the documents share a specific 8-gram and all the number of unique 8-grams in each document. This process is explained in the following four steps. (1) Text Extraction and Reduction: First, we extract the text from each document and remove everything that does not convey information about the text, as described in the previous Chapter in Section 3 phase 1, steps 1 to 5. (2) Sets of 8-grams: Each text is converted into an 8-gram set generated by sliding a window of size eight over the text. Every eight words inside the window are added to the text's 8-gram set. We store the cardinality of an 8-gram set alongside the document for later use. (3) Building 8-gram-Document-id Pairs: For every document's 8-gram set, we build key-value pairs consisting of an 8-gram in the set and the corresponding document-id. (4) Group by 8-gram: We group the key-value-pairs by their 8-gram using the same method as in chapter 3. The pairs are distributed across workers of the cluster based on their key, which is the 8-gram. Then each partition is sorted and grouped. Thus, if p is the partition size, this operation takes $O(n * log(n))$. We remove any 8-gram that is only present in one document. The result is a dictionary over the 8-grams. Each 8-ram entry in the dictionary holds a list of documents containing this 8-gram. This data structure is referred to as an 8-gram index.

In the second phase, we calculate the number of 8-grams any two documents have in common, thus obtain the numerator of the S3-scores, and finally calculate the S3-scores. (1) 8-gram co-occurrence: For each 8-gram in the 8-gram index, we compute every pair of documents sharing a particular 8-gram, by deriving all document combinations in the list of occurrences of an 8-gram. (2) Counting shared 8-grams: We count the co-occurrences of any



(a) clueweb09-enwp00-72-00842 **(b)** clueweb09-enwp01-57-22152

Figure 4.1: Two near-duplicates in the Clueweb09 corpus. (a) is the Wikipedia article for "Dog hybrid", (b) is the article for "Shihpom". The documents are identical except for the redirection message. The screenshots were taken from the Wikipedia archive. We had to modify the title to fit the documents from the crawl because the article was renamed years later.

Lorem ipsum dolor sit amet consectetur adipisici elit sed eiusmod

Figure 4.2: Example text with all possible 8-gram sequences. Each line represents one 8-gram. The set of all 8-grams represent the document.

two documents over all pairs. For any two documents, we gain the number of unique, shared 8-grams, which is equal to the numerator in the S3-score of these two documents. (3) S3-scores Calculation: The denominators of the S3-scores are the mean numbers of unique 8-grams in any two documents. Since we earlier stored the cardinality of the 8-gram sets to every document, we can look up these values and derive the mean, yielding the denominator. From there, we calculate the S3-Score for each document pair by dividing the numerator by the denominator.

With all S3-scores available, we can find all document pairs with an S3-sore above our user study's threshold. By interpreting the remaining document pairs as connected nodes in a graph, we find all groups of content-equivalent documents. The documents in each connected component in the constructed graph are content-equivalent documents. Note that we also identify content-equivalence in chained relations. For example, consider three documents A, B, C, all of which are different versions of a news article. A is connected to B, and B is connected to C. In other words, A and B have an S3-score above the threshold as well as B, and C. A and C are always in the same group because they are connected through B.

Though this approach allows us to identify content-equivalent documents, it is also computationally expensive, causing us to focus on a subset of documents in each corpus. We index the judged documents of the TREC tracks of each corpus. A judged document concerning a particular query is either relevant, represented by a positive integer, or irrelevant, represented by a negative integer or zero. Judged documents are an interesting subset of corpora because they are used to assess the retrieval systems' performance.

Table 4.1 gives an overview of the number of duplicates in the corpora. Among the web-crawled corpora .GOV2, Clueweb09, and Clueweb12, we find about 17% duplicates within the judged documents. The NYT-AC corpus has 1.22% duplicates. Among the relevant documents in Clueweb12 and NYT-AC, the percentage of duplicates is almost the same as that of the judged documents. Among the relevant documents in .GOV2 and Clueweb09, the percentage of duplicates is about 2 percentage points higher than the percentage among the judged documents. In the Washington Post corpus, we find an increase of about 5 percentage points in the relevant documents, possibly due to the small number of relevant documents. We find fewer duplicates than estimates suggest for general web crawls. For web pages collected during the PageTurner experiment, Fetterly et al. reported "that about 28% of all web pages are duplicates [...]" [FMN03]. Our lower numbers may be due to the corpus creators' efforts to minimize spam during crawling or to changes on the web.

Figure 4.3 shows box plots for the relative amount of duplicates within the relevant documents across all topics of the TREC tracks Terabyte 2004 to 2006 (TB04-TB06), Web 2009 to 2014 (Web09-Web14), as well as Core 2017 (Core17) and Core 2018 (Core18). The orange boxes mark the retrieval-equivalent duplicates as described in Chapter 3, while the purple boxes mark the content-equivalent duplicates as described in this chapter. The median percentage of redundant, content-equivalent documents and the number of outliers is higher among the tracks on a web-crawled corpus than the other tracks Core2017 and Core2018 on non-web corpora. In contrast to the tracks on web corpora, the tracks on non-web corpora show among the relevant documents

Table 4.1: The number of judged and relevant documents and the associated percentage of duplicates of the listed corpora. The judgments for the .GOV2 corpus are taken from the TREC Terabyte tracks 2004, 2005, and 2006. Analogously, the Clueweb09 corpus' judgments are taken from the TREC Web tracks 2009, 2010, 2011, and 2012. The judgments for the Clueweb12 corpus are taken from the TREC Web tracks 2013 and 2014. The judgments for the NYT-AC corpus are taken from the TREC 2017 Core track. The judgments for the Washington Post corpus are taken from the TREC Core 2018 track.

Corpus	Judged		Relevant	
	Size	Duplicates	Size	Duplicates
.GOV2	135352	16.47%	26917	18.38%
Clueweb09	73883	16.65%	19223	18.57%
Clueweb12	28906	17.25%	10605	17%
NYT-AC	30030	1.22%	9002	1.16%
WaPo	26233	8.76%	3948	13.42%

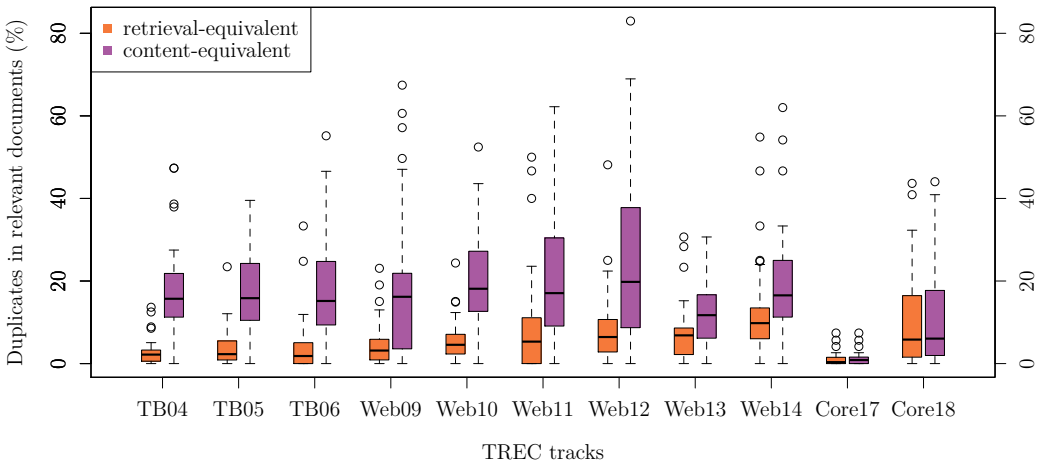


Figure 4.3: These box plots show the range of duplicates within the relevant documents across all topics of the respective TREC track. The whiskers have a maximal length of the 1.5-fold interquartile range (IQR).

similar amounts of content- and retrieval-equivalent duplicates. We found only a few documents that are exclusively content-equivalent in the NYT-AC corpus and the Washington Post corpus.

The interquartile range (IQR) ranges from the smallest percentage in the second quartile to the highest percentage in the third quartile. The Web12 track has an unusually broad IQR, which means that the percentage of duplicates is widely varying across the topics. For half of the Web 2012 track's topics, there are between 8% and 38% duplicates among the relevant documents, and a quarter of the topics consists of 20% and 38% duplicates. The outlying topic of Web12 (no. 194) has the highest percentage of duplicates with over 80%. The relevant documents of this topic include the example documents given in Figure 4.1 as well as 38 content-equivalent documents.

Besides the Web 2011 track, we observe an increase in duplicates from the Web 2009 track to the Web 2012 track. One possible explanation is that more duplicates are contained in a submission. Another reason might be that the judges classified more duplicates as relevant over time. Unfortunately, the investigation of these hypotheses goes beyond the scope of this thesis. In summary of this figure, there are some notable outliers among the topics regarding relevant duplicates. Generally, we identified many more equivalent documents content-equivalent documents than retrieval-equivalent documents

Bernstein and Zobel report the results for the specific topics of the Terabyte 2004 track [BZ05]. We did the same plots for all examined tracks, but limit our analyses on the Terabyte 2004 track. Figure 4.4 shows our results for the duplicates among relevant and irrelevant topics in the same format. Note that topic 704 was skipped since it does not have any judgments. On the bottom, the bars in light blue show the percentage of retrieval-equivalent duplicates, and the dark blue bars the percentage of content-equivalent duplicates, respectively. On the top, the bars in orange show the percentage of retrieval-equivalent duplicates, and the purple bars the percentage of content-equivalent duplicates, respectively. It seems like that the amount duplicates does insignificantly differ in relevant documents compared to irrelevant documents. We notice that we find less retrieval-equivalent, irrelevant, duplicates in the topics 714 and 715, and less retrieval-equivalent, relevant, comparing our results to Bernstein and Zobel's results duplicates in topic 708. Other than that are both results similar. We conclude that we successfully reproduced Bernstein and Zobel's results for the Terabyte 2004 track since the divergence is within the margin of error.

The amount of duplicates in the judgments leads to the question of whether there are content-equivalent documents with different judgments. We regard a document's judgment in a given topic as inconsistent if there exists a content-equivalent document with higher relevance for the same topic. We analyze the percentage of inconsistent judgments across the topics of the TREC tracks

from 2004 to 2014 and 2017 to 2018. Our results are shown in box plots in Figure 4.5. For most tracks, the median topic has less than 1% inconsistent judgments. Deviating from this observation, the tracks Terabyte 2004, Terabyte 2005, Web 2009, and Web 2010 have a median between 1.85% and 3.13%. The Core 2017 track mainly shows low inconsistency values below 1% and has a median of 0.1%. The outlier with the highest value of 17.83% is in the Core 2018. The top five outlying topics have inconsistencies lie within 10% and 18%. There are considerable discrepancies between the judgments of content-equivalent documents. These discrepancies are either due to an incorrectly claimed equivalence or du to real differences in the judgments of equivalent documents, which should not exist.

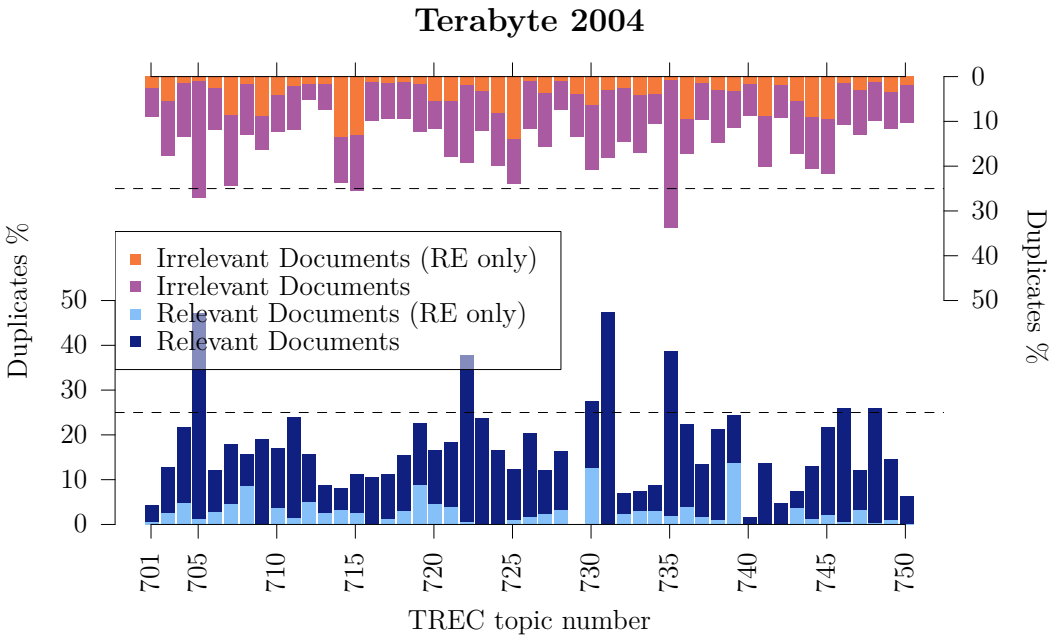


Figure 4.4: Percentage of duplicates among relevant and irrelevant documents across the Terabyte 2004's topics.

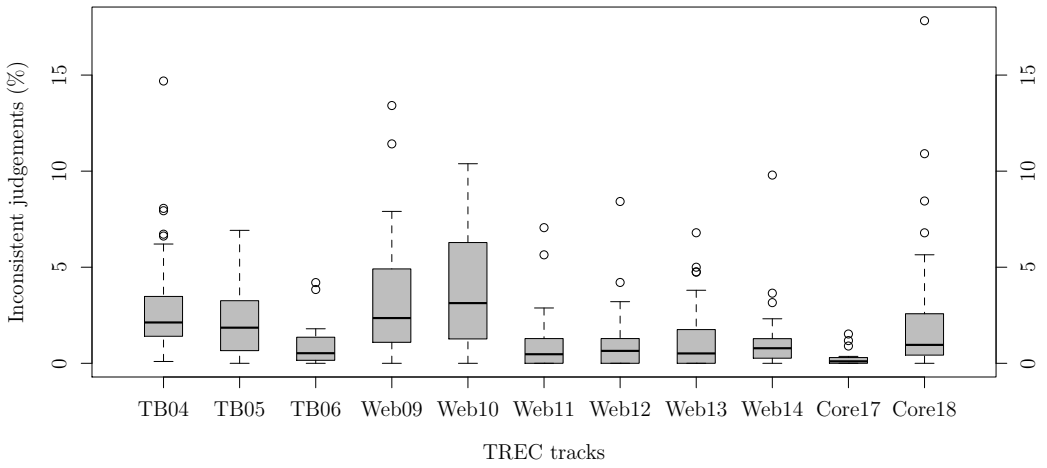


Figure 4.5: Percentage of inconsistent judgements among the topics. A judgement is inconsistent, if there exists a judgement with a higher relevance for a content-equivalent document.

Chapter 5

Impact of the Novelty Principle on Retrieval Evaluation

On the principle of novelty, a document is irrelevant if an equivalent document is already in the user's result list. This includes documents which, taken by themselves, would otherwise be relevant. In the last chapter, we saw topics with up to 83% duplicates within the relevant documents. A conventional evaluation approach favors systems that retrieve all relevant documents, including the duplicates. In this chapter, we study the effect of the novelty principle on the score and the ranking of competing retrieval systems.

To assess the impact of the novelty principle, we evaluate the systems' result lists under the novelty principle and compare the result to the conventional evaluation. Additionally, we simulate how the systems would have performed if they had removed all duplicates from their submissions. By combining the novelty-aware evaluation with the removal of duplicates from the submission, we can also simulate the systems' performance in a completely novelty-aware environment.

As our experiments showed in the last chapter, there are up to 18% percent of inconsistent judgments in the topics. In all our experiments, we make the judgments consistent by assigning the highest relevance within a group of content-equivalent documents to all group members in the respective topic. In detail, we discuss Bernstein and Zobel's interpretation of a novelty respecting evaluation algorithm [BZ05]. On this basis, we introduce an enhanced, alternative interpretation.

Bernstein's and Zobel's interpretation states that a novelty aware evaluation algorithm should treat a document as irrelevant if the information retrieval system already showed an equivalent document at a higher rank. We refer to Bernstein's and Zobel's interpretation as local judgments manipulation. Local judgment manipulation only treats duplicates as irrelevant if one document

doc-unique	relevant
doc-groupA-1	relevant
doc-groupA-2	relevant
doc-groupB-1	relevant
doc-groupB-2	relevant

(a) Original judgments

doc-groupA-1	doc-unique
doc-groupB-1	doc-groupA-1

(b) System 1 MAP: 0.4

(c) System 2 MAP: 0.4

Figure 5.1: Original judgements and the submission of two systems as well as the corresponding mean average precision (MAP). Both systems have the same MAP.

doc-unique	relevant
doc-groupA-1	relevant
doc-groupA-2	irrelevant
doc-groupB-1	relevant
doc-groupB-2	irrelevant

(a) System 1 MAP: $0.\overline{6}$

doc-unique	relevant
doc-groupA-1	relevant
doc-groupA-2	irrelevant
doc-groupB-1	relevant
doc-groupB-2	relevant

(b) System 2 MAP: 0.5

Figure 5.2: Judgments for the systems under local judgment manipulation and the corresponding mean average precision (MAP). Both systems retrieve exactly two relevant documents, but the MAP differs.

from the corresponding group has been retrieved by the system that is to be evaluated. As a result, submissions may have different scores, although they deliver equal quality from a novelty perspective. For a better understanding of the implications of local judgment manipulation, consider the following example. There is a single, relevant and unique document, as well as two groups A and B consisting of two equivalent and relevant documents each, as shown in Figure 5.1. We compare two hypothetical systems, both retrieve two relevant documents. In a conventional evaluation, both systems retrieve two out of five relevant documents, which corresponds to a mean average precision (MAP) of 0.4. Employing local judgment manipulation changes the judgments for each system individually, depending on which groups' documents they retrieved, as shown in Figure 5.2. System 1 retrieves one document from each of the two groups. Through the local judgment manipulation, the duplicates of both retrieved documents become irrelevant. Based on the resulting judgments, System 1 retrieves two out of three relevant documents, which corresponds to an MAP of $0.\overline{6}$ and matches our intuitive understanding of the novelty

principle. System 2 retrieves the unique document, and the first document of group A. As local judgment manipulation only manipulates the relevance of duplicates of *retrieved* documents, only the duplicate group A is marked irrelevant. In contrast, all documents in group B stay relevant. Thus System 2 retrieves two out of four relevant documents, which corresponds to an MAP of 0.5. Although both systems recall two relevant, novel documents, System 1 has a higher score because it retrieved one document from each group instead of the unique document. System 2 is penalized for failing to retrieve every duplicate in group B. Thus the strength of the penalty is proportionate to the number of documents in group B. In our opinion, a system should not be penalized for not retrieving a duplicate.

To address this issue, we expand the novelty principle to non-retrieved documents. This global judgment manipulation manipulates the relevance of *all* duplicates, even those that are not retrieved by a system. Each relevant group contains one document that remains relevant, while all other documents become irrelevant. For each system, the first retrieved document of each relevant group remains relevant. If a system does not retrieve any document of a group, a random document of this group stays relevant. Concerning the example from above, under global judgment manipulation System 2 has the judgments in Figure 5.2b, but additionally one of the documents in group B is irrelevant. System 1 has the same judgments in both judgment manipulations. We evaluate both systems with their respective judgments. As a result, both systems have a MAP of $0.\overline{6}$. Thus, the evaluations of different systems are comparable under global judgement manipulation. Furthermore, the systems are only penalized once for neglecting the retrieval of a novel document.

For the evaluation, we use the open-source program `trec_eval`, which is also used in the official evaluations of the TREC tracks. We configure `trec_eval` to process only the first 1000 retrieved documents of a submission.

We investigate how the application of the novelty principle affects the tracks ranking in different scenarios. Each participating system provides at least one submission. The ranking is the list of the systems' submissions ordered by their score. Since there is evidence that the Kendall correlation is preferable to other rank correlations measures [CD10], we use the Kendall rank correlation [Ken48] to compare the ranking of the original evaluation with the ranking emerging from our experiments.

We want to quantify how systems that put the novelty principle into practice competes against other systems in an original evaluation. In all reported numbers in this chapter, we only investigate the 75% of systems that performed best in the original evaluation. As a reference, we evaluate all systems conventionally and derive the systems' original ranking from their nDCG-score. For each system, we simulate novelty by removing duplicates from

its list of retrieved documents and evaluate this, under the prospects of novelty, ideal version of the system using the original judgments. We use the nDCG-score of the system's ideal version in combination with the nDCG-scores from the original evaluation of all other systems. Thus, the ideal version competes against the other original systems. By ranking this modified evaluation and the reference evaluation, we derive the change in the system's rank caused by holding on to the novelty principle. We report the median (med_I) and maximum (max_I) change in the rank across the submissions of a track.

Table 5.1 show our results. The column "Submissions" contains general information about the tracks' submissions. For comparison, we list the average nDCG of the original system evaluation alongside the number of submissions. This column also includes the median and maximal change in the system's rank caused by holding on to the novelty principle. The column "Duplicates irrelevant" features the evaluation using global judgment manipulation. In the column "+Duplicates removed", we also applied the global judgment manipulation, but *additionally* we removed all duplicates from the submissions. This simulates the novelty-aware evaluation of systems that only retrieve novel documents. We report deviations from the conventional evaluations average nDCG ($\Delta nDCG$) as well as the deviations in the ranking of all systems (τ) and the top five systems (τ@5) using the Kendall correlation.

In the following, we first give some general insight into the data and then discuss three tracks in more detail. The Web tracks have the lowest average nDCG values. There is no Terabyte or Core track with a more substantial difference in the average nDCG than in any Web track. If duplicates are treated as irrelevant during the evaluation, the average nDCG drops on all tracks. The Web 2012 track has the most significant difference of -17.3%, which is a drop from 0.295 to 0.244. If we additionally remove duplicates, the systems can gain back some of the lost performance. In the Terabyte 2005 track, the submission scored higher than than in the original evaluation.

There are striking differences between the rank correlation of the Terabyte and the Web tracks. None of the Terabyte tracks drops below a rank correlation of 0.94, whereas the rank correlation varies from 0.49 to 0.94 across the web tracks. The additional removal of duplicates increases the rank correlation on all tracks. The Core 2017 track even has rank correlation of 1, which means it the ranking in our experiments is the same as the original ranking.

More than half of the tracks record a rank correlation of or above 0.8 among the top five systems under global judgement manipulation. Especially the top systems in the newer tracks Web 2013 and Web 2014 appear to be less affected by an evaluation that considers novelty than the top five systems in the older Web tracks from 2010 to 2012. If the duplicates are removed from the submissions and marked as irrelevant, the Web 2011 and the Web 2012 track

Table 5.1: Impact assessment of the novelty principle for the given TREC tracks. The category Submissions lists the number of submissions (100%), the average nDCG in the original evaluation, as well as the median (med_I) and maximum (max_I) difference in the rank of the submissions if they only retrieve novel documents. The two other categories are the following scenarios are: (1) Duplicates irrelevant: All duplicates in the judgments are marked irrelevant, (2) +Duplicates removed: All duplicates in the judgments are marked irrelevant and removed from the systems' submissions. For the respective scenarios we report three values: (1) ΔnDCG is the difference in the original average nDCG (avg_{nDCG}) and the average nDCG of the corresponding scenario. (2) τ is the Kendall rank correlation coefficient of the systems' submissions in the original evaluation and the systems' submissions in the corresponding scenario. (3) τ@5 is the Kendall correlation coefficient of the top five submission in the original evaluation and the top five submissions in the corresponding scenario.

Track		**Submissions**				**Dupl. irrelevant**			**+Dupl. removed**		
		#	avg_{nDCG}	med_I	max_I	ΔnDCG	τ	τ@5	ΔnDCG	τ	τ@5
Terabyte	2004	70	0.425	-9.0	-19	-5.1%	0.96	0.80	+0.3%	0.98	1.00
	2005	58	0.586	-15.5	-27	-3.8%	0.95	0.20	+0.8%	0.98	0.80
	2006	80	0.654	-29.5	-53	-4.4%	0.94	1.00	-1.0%	0.94	0.86
Web	2009	71	0.323	-8.5	-24	-8.9%	0.89	0.80	-6.8%	0.91	0.80
	2010	56	0.302	-19.5	-39	-14.1%	0.49	0.42	-9.9%	0.57	0.33
	2011	37	0.341	-8.0	-13	-9.0%	0.85	0.40	-3.4%	0.92	0.80
	2012	28	0.295	-9.0	-16	-17.3%	0.72	0.61	-12.4%	0.81	0.73
	2013	34	0.324	-4.0	-8	-4.6%	0.86	0.80	-1.8%	0.90	0.80
	2014	30	0.380	-4.0	-11	-7.9%	0.87	1.00	-4.5%	0.94	1.00
Core	2017	75	0.560	-1.0	-9	-0.3%	0.99	1.00	+0.1%	1.00	1.00
	2018	72	0.541	-11.0	-26	-4.3%	0.92	1.00	-0.9%	0.93	0.73

improve their rank correlation among the top five systems. In contrast, the correlation among the top five systems on the Web 2010 track drops with the removal of duplicates to an even lower value of 0.33, indicating that duplicates heavily influence the ranking (of this track). In contrast to the Web 2010 track, the top five systems in the Terabyte 2005 track partly recover their from the low rank correlation of 0.2, as indicated by the rank correlation of 0.86 coming along with the removal of duplicates.

The system losing the most ranks under the principle of novelty is in the Terabyte 2006 track, which records a fall from the fourth down to the fourth-last rank. This corresponds to a loss of 53 ranks. Core 2017 is the track with the lowest median number of lost ranks at only 1 lost rank. Our results for the

Core 2017 track indicate only a minimal impact of the novelty principle since rank correlation is at or above 0.99 in all scenarios. The low impact is due to the track containing almost no duplicates (see Figures 4.3).

We take a deeper look at the Terabyte 2006, Web 2010 and the Core 2017 tracks. Figure 5.3 shows the score for each system. The systems are sorted by their score in the original evaluation. The Terabyte 2006 track reflects our expectations and is very similar to the results of the other Terabyte tracks. The systems drop in their performance under the global judgment manipulation. However, they get almost up to their original performance, if the duplicates are removed additionally. The tracks Web 2010 and Core 2017 are opposite extremes. Systems in the Core 2017 are very lightly affected by the application of the novelty principle and they perform almost constantly throughout the evaluation scenarios. Some systems in the Web 2010 have widely differing performances across the evaluation techniques.

In the Web 2010 track, there are submission that strongly profited from the retrieval of duplicates, for example, the submission 3 to 5. These submissions are on par with the top two submissions in the original evaluation but fall behind if duplicates become irrelevant. Also, they do not recover from their losses if the duplicates are removed from their result list. It might be worthwhile to compare the strongly affected submission to the more lightly affected submission within some Web tracks. However, this is, unfortunately, beyond the scope of this thesis.

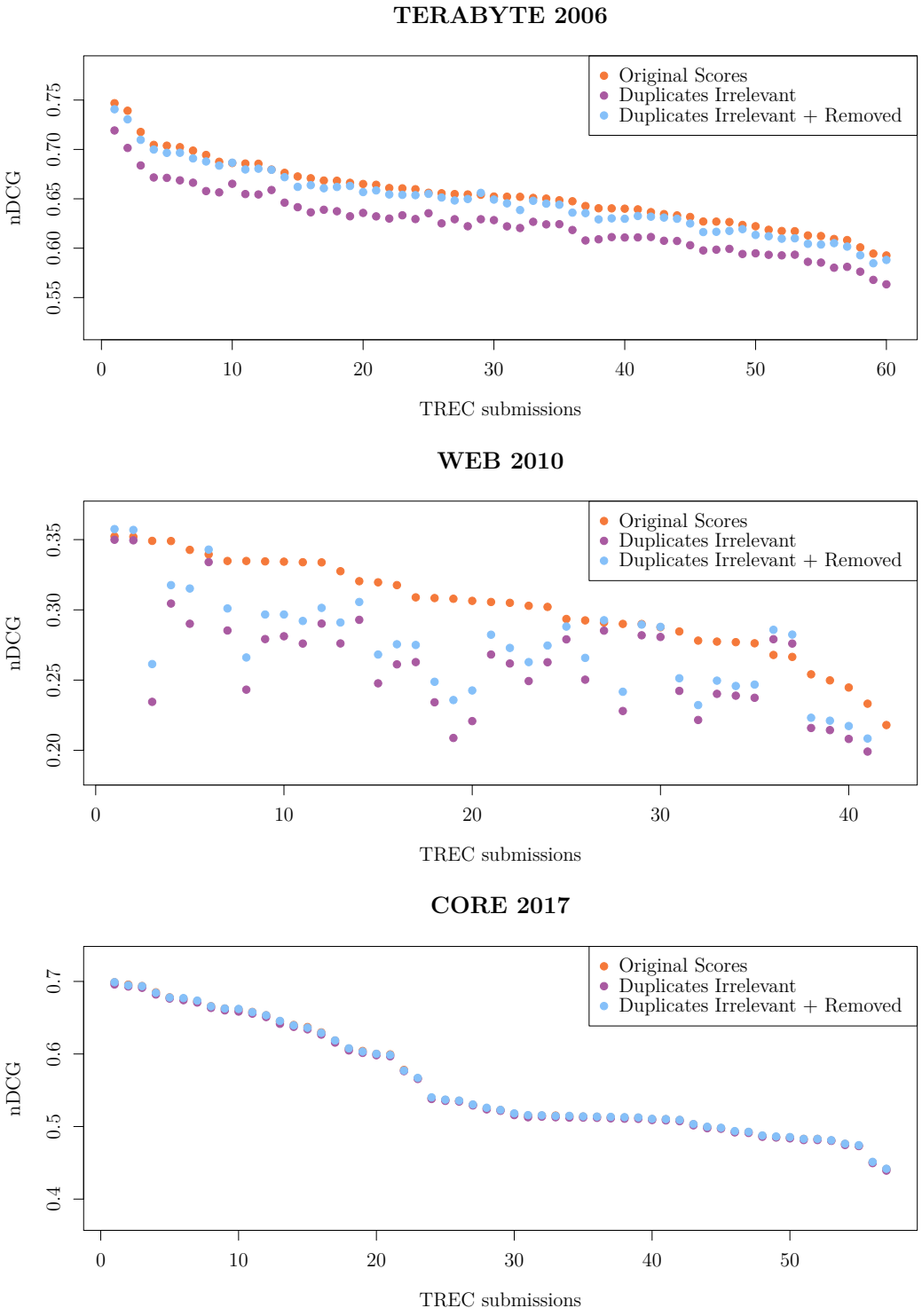
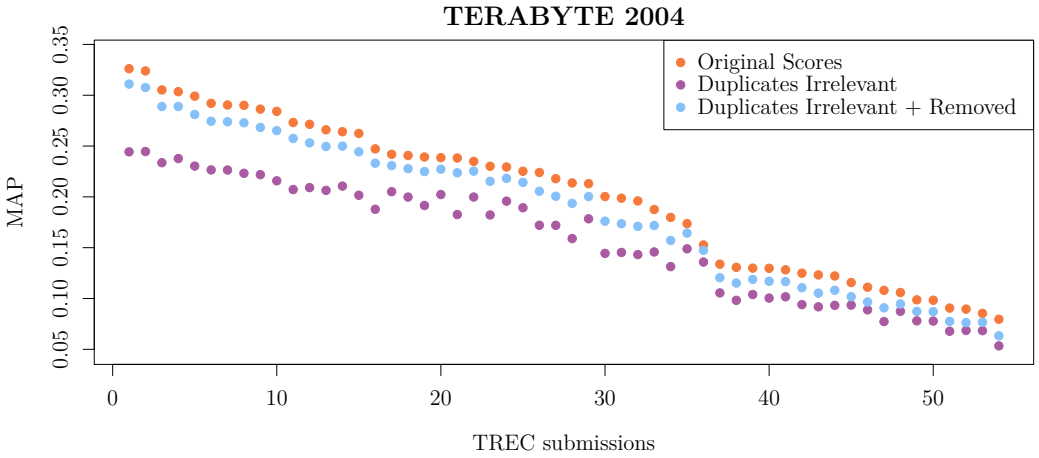
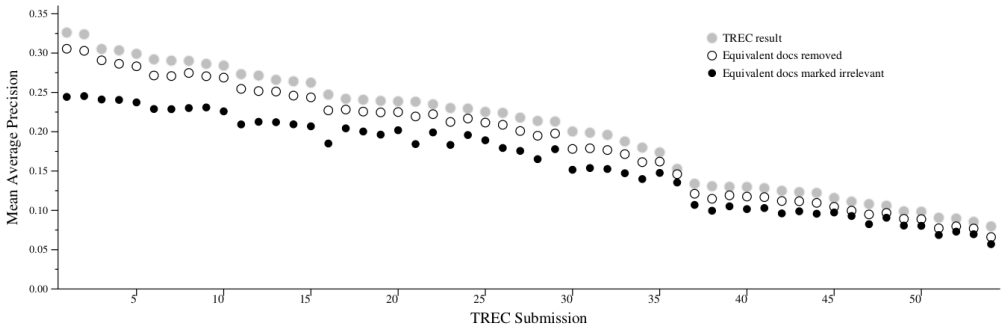


Figure 5.3: nDCG-scores in the different scenarios per submission using the global judgment manipulation, ordered by the original scores. Figures show the following tracks: Terabyte 2006, Web 2010, Core 2017.



(a) Results of our reimplementation of Bernstein and Zobels approach using local judgment manipulation and all retrieved documents.



(b) Results of Bernstein and Zobel taken from "Redundant Documents and Search Effectiveness"[BZ05]

Figure 5.4: Side by side the results of Bernstein and Zobel and our results for the mean average precision of the submission in the TREC 2004 Terabyte Track.

Bernstein and Zobel did similar experiments on the Terabyte 2004 track [BZ05]. They evaluated the submissions using the MAP and the local judgment manipulation. Also, they did not supply the -M1000 option to trec_eval, which means they used all retrieved documents in the evaluation. Figure 5.4a shows the results from our reimplementation of Bernstein and Zobel's approach. Comparing our results with Bernstein and Zobel's reported results, we are very confident that we successfully reproduced their experiments on the Terabyte 2004 track using local judgment manipulation.

In summary, our experiments show that duplicates can significantly impact the evaluation of the systems. Tracks on web corpora contain more duplicates than tracks on newspaper archives, making them naturally more susceptible to duplicates. Even across tracks on the same corpus, there are glaring differences

in duplicates' impact, for example, between Web 2009 and Web 2010. Our results suggest that some systems competing in tracks on the Clueweb corpora gained a considerably higher score by retrieving duplicates. Seeing the significant differences in the scores caused by duplicates, it is immanent that we need some way of reducing their impact. Ideally, the original scores and the scores from an evaluation that takes novelty into account converge, as observed in the Core 2017 track. In the next chapter, we evaluate different mitigation strategies to reduce the impact of duplicates.

Chapter 6

Mitigation

We examine whether organizers of shared tasks can detect and directly replace topics that cause significant changes in the ranking of retrieval systems by the novelty-principle in advance. In doing so, we directly address the main problem in the evaluation caused by duplicates. Considering novelty in the evaluation process can result in a significantly different ranking of the retrieval systems, as described in chapter 5. Our approach allows organizers of shared tasks to reduce these differences by removing topics that cause vast differences between evaluation techniques. Reducing the difference by removing topics is especially compelling if the minority of topics causes the majority of differences.

Figure 6.1 shows a box-plot of the topics and their difference between the nDCG-score in the conventional evaluation and the evaluation using global judgment manipulation. It clearly show that there are outliers, suggesting that removing topics may decrease the differences between the evaluations and is motivating our research. Our result show that removing carefully selected topics indeed decrease the difference between the evaluation techniques.

We simulate approaches with various levels of manual effort to estimate the amount of divergence a topic causes between the evaluation techniques' rankings. A topic is thereby risky if the estimated amount of divergence is high, thus we refer to these approaches as risk estimations. Our experiments suggest that it may be possible to reduce the difference in the rankings by removing topics selected by the considered risk estimations.

The examined risk estimations only require information associated with the topic at hand, which allows risk estimation on topics individually. None of the risk estimations require manual effort in other topics than the topic to estimate the risk on. All require the content-equivalent document groups for the documents in the submissions. We removed the inconsistencies in the official judgments before hand, as described in chapter 5. Throughout this chapter, we use all submissions of a track but exclude the Core tracks since

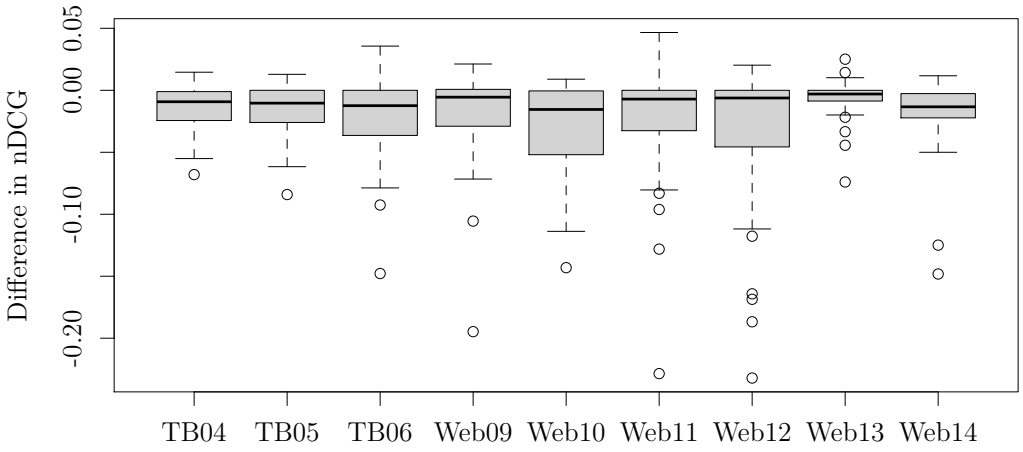


Figure 6.1: Box-plot of the topics' difference between the average score in the conventional evaluation and the evaluation using global judgment manipulation.

the duplicates' effect on them is low. We also enforce realistic similarity-score tie-breaking introduced by Cabanac et al. [Cab+10] on the retrieved documents of a system. The similarity scores of the documents in a submission determine which documents are first shown to the user. If two documents have the same similarity score, then we order them so that with these documents, the relevant documents always come last.

We test risk estimation in three different practical scenarios, two of which offer incomplete or no knowledge about the relevance of documents. The third scenario offers the relevance information about all judged documents.

(1) No-Judgments (dup-score): The submissions plus the duplicate groups are available. There is no requirement for judged documents, which means no manual effort, allowing fully automatic risk estimation immediately after a shared task ends. The risk is estimated by evaluating a topic using the submitted runs and treating all duplicate documents as relevant while treating any other document as unjudged. With artificial judgments generated from the duplicate groups, we can estimate the potential impact through the use of discounted measures like the normalized discounted cumulative gain (nDCG). The topic's risk is the average nDCG of this topic derived from the submissions and artificial judgments.

(2) Partial-Duplicate-Judgments (reldup-score): The submissions plus the duplicate groups are available, as well as judgments for duplicate groups. Since organizers of tracks are aware of the duplicate groups, they can obtain judgments for the entire group by judging only one representative, saving some judgments. Additionally, a pooling strategy can be employed to reduce manual effort even

further. The judgments are generated the same way as for the dup-score, with the difference being that only relevant duplicates generate a relevant judgment. As for the dup-score, the topic's risk is the average nDCG of the submissions using the generated judgments.

(3) Partial-Mixed-Judgments (impact-score): The submissions, the duplicate groups, and judgments for duplicates and additional judgments for non-duplicate documents are available. Likely almost every TREC ad-hoc shared task from the past can provide these judgments through the official judgments used for evaluation, allowing the calculation of the impact-score of the duplicates in a topic. A topic's risk is the average difference between the nDCG of the conventional evaluation and evaluation using global judgment manipulation described in chapter 5. In other words, for every topic, we subtract the nDCG-score, using only novel documents from the nDCG-score derived from the conventional evaluation.

We study whether the risk estimations with limited knowledge (dup-score and reldup-score) correlate to the risk estimation with full knowledge by comparing them in a correlation graph. We look at a well-chosen subset of tracks. Our study contains tracks that are strongly affected by duplicates and more lightly affected tracks. Strongly affected tracks are the TREC 2009 Web Track, the TREC 2010 Web Track, and the TREC 2012 Web Track. In contrast, the tracks TREC 2006 Terabyte Track and TREC 2014 Web Track, are more lightly affected.

We start by comparing the risk estimation without knowledge (dup-score) against the risk estimation with full knowledge (impact-score). The correlation graphs of the shared tasks are shown in Figure 6.2. The x-axes of the graphs denote the risk estimation with all information about the judgments (impact-score). The y-axes denote the risk estimation without any knowledge about the judgments (dup-score). The five topics with the highest dup-score are marked with + in the graphs, while the five topics with the highest impact-score are marked with $\bigtriangledown$. We mark any other topics with $\bigcirc$. A topic may be in both, one, or none of these top-fives. We analyze if the top 10% percent of topics in the impact-score overlap with the top 10% topics in the dup-score. In the examined tracks, 10% corresponds to 5 topics. Ideally, both risk estimations should agree on topics with a low risk near to zero. However, the dup-score has a positive offset compared to the measured impact-score. The offset is especially noticeable in the shared task web 2014. The offset is caused by the fact that all topics have duplicates, but not all duplicates are relevant. Thus, the dup-score might overestimate the risk. The majority of topics in all shared tasks has an impact-score near to zero, which supports our motivation that these topics can remain, while others should be removed. The tracks Web 2009 and Web 2010 show a promising correlation between the impact-score and the dup-score. The

risk estimations predict three out of five topics mutually. There are some topics with an impact-score near zero, but with a dup-score significantly above zero, which indicates a risk overestimation. On Terabyte 2006, the dup-score produces very different results compared to the impact-score, but still, two out of five topics were mutually identified. On web 12 and web 14, the dup-score is not very similar to the impact-score. Only one out of five topics are commonly identified. The dup-score performs poorly on the Web 2012 track because it assigns all topics a risk within a narrow range of values. Also, the dub-score performs poorly on Web 2014 because it assigns high-risk values to topics with an impact score near zero.

Extending the analysis, we compare the risk estimation with partial information about the judgments (reldup-score) against the risk estimation with full knowledge (impact-score). Figure 6.3 shows the correlation graphs of the tracks. Structurally these graphs do not differ from the graphs in Figure 6.2, but they show the reldup-score on the y-axes and mark the documents with the highest reldup-score with +. In these graphs, a correlation between both estimations is immediately noticeable on all tracks. On web 2009 and web 2012, the two risk estimations mutually identify four out of five topics. On web 2010, there are still three out of five commonly identified, but only two out of five on terabyte 2006 and web 2014, respectively. As before, there are two outliers in the web 2014, which have the highest impact-score, but a mediocre reldup-score.

The following paragraph concludes our study of the correlation between the risk estimations. The risk estimations share at least some of their top five topics and show an overall similarity, indicating that all of these risk estimations deliver results in the same ballpark. On some tracks like the Web 2009 track and Web 2010 track, the dup-score performs remotely similar to the impact-score. Nevertheless, the dup-score and impact-score do agree on some of the most risky topics. Unfortunately, the dup-score sometimes assigns a high risk to topics with a low impact, indicating an overestimation, especially on Web 2012 and Web 2014. Our experiments suggest that the reldup-score and the impact-score deliver similar results, despite the significantly lower number judgments required by the reldup-score. The relevant documents in the reldup-score are precisely the documents that become irrelevant through the global judgment manipulation, where the latter is part of the impact-score. This overlap explains the similarity between their results.

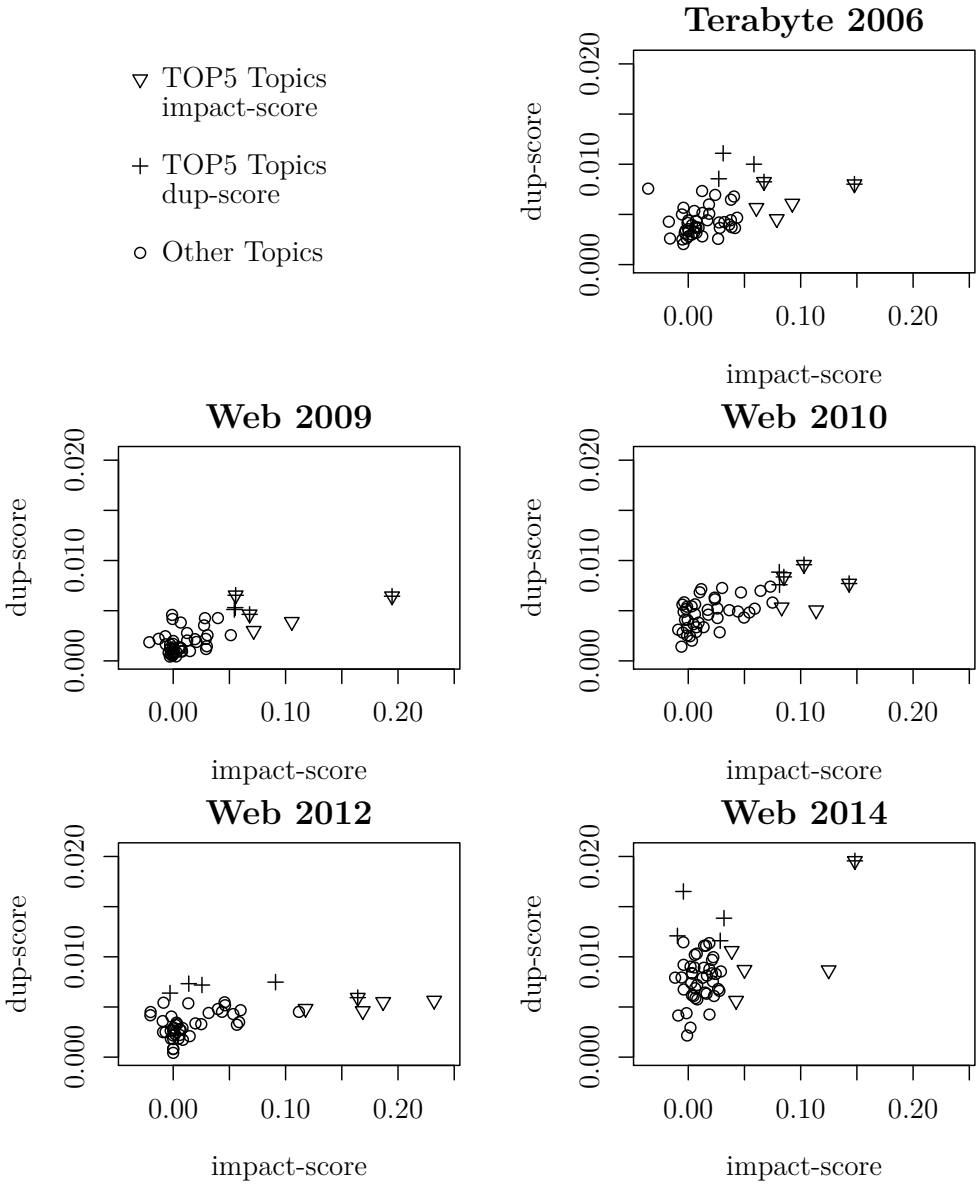


Figure 6.2: The risk estimation with full knowledge about the judgments (impact-score) plotted against the risk estimation without knowledge about the judgments (dup-score) for all topics of the TREC 2006 Terabyte Track, TREC 2009 Web Track, TREC 2010 Web Track, TREC 2012 Web Track, and TREC 2014 Web Track.

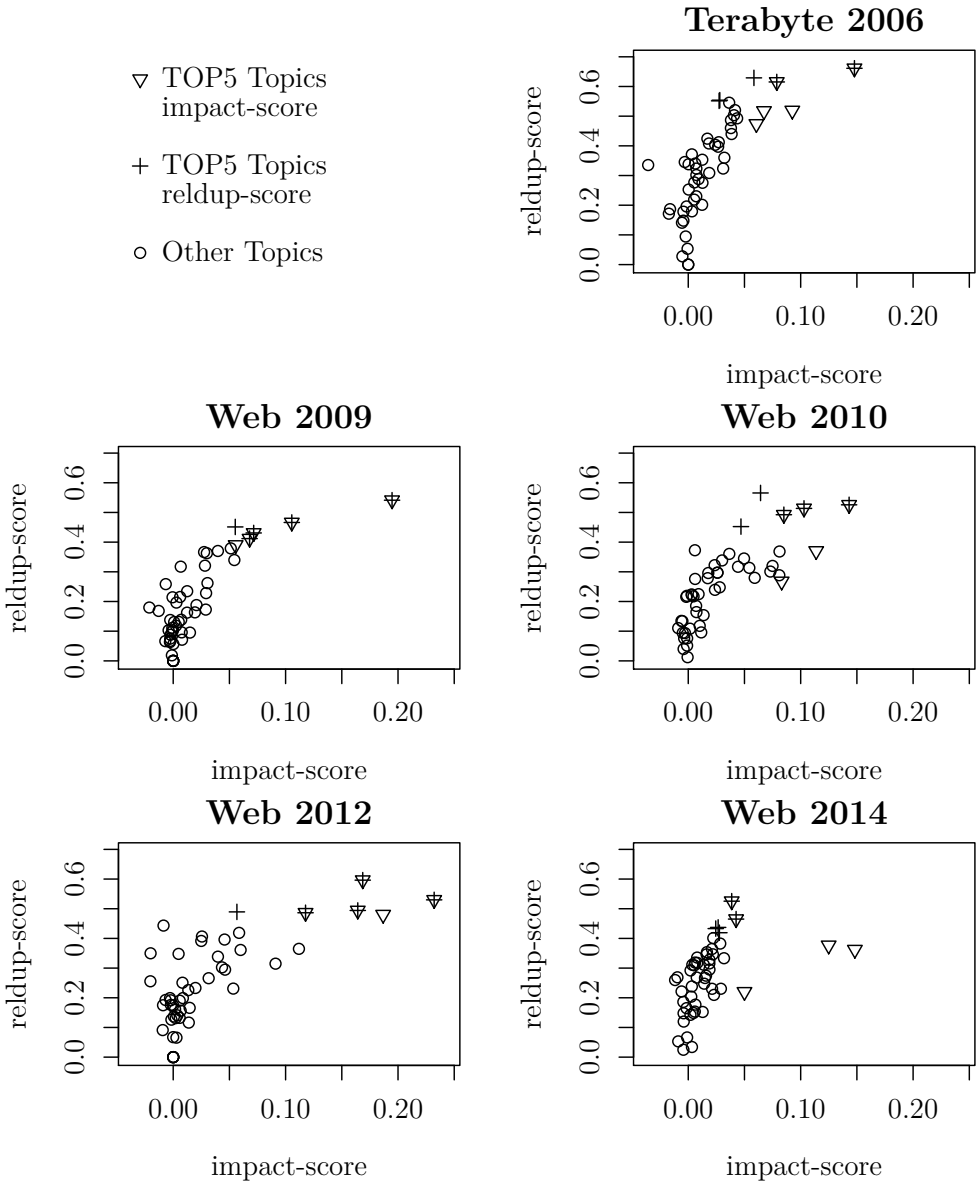


Figure 6.3: The risk estimation with full access to the judgments (impact-score) is plotted against the risk estimation with restricted access to the judgments (reldup-score) for all topics of the tracks TREC 2006 Terabyte Track, TREC 2009 Web Track, TREC 2010 Web Track, TREC 2012 Web Track and TREC 2014 Web Track.

Ideally, a risk estimation predicts the topics whose removal reduces the impact of duplicates on the overall evaluation, nearing an evaluation where only novel documents are relevant. To assess the quality of the risk estimations, we measure the systems' rank correlation between the conventional evaluation, from which the risky topics have been removed, and the evaluation that applies global judgment manipulation (gjm-evaluation). The rank correlation for the shared tasks TREC 2006 Terabyte Track, TREC 2009 Web Track, TREC 2010 Web Track, TREC 2012 Web Track and TREC 2014 Web Track is visualized in Figure 6.4. The x-axes denote the number of topics removed from the baseline evaluation. The y-axes denote the Kendall correlation between the gjm-evaluation and the conventional evaluation for the three risk estimations dup-score, reldup-score, and impact-score. A point with an x-value of five displays the rank correlation between gjm-evaluation and conventional evaluation, where the five documents with the highest risk are removed from the latter. The solid line shows the correlation's empirical maximum, which is the maximal correlation achievable if a certain number of topics is removed. Finding the set of documents resulting in the empirical maximum requires extensive computational effort, making it infeasible for us to calculate the empirical maximum for more than five documents. The horizontal, dotted line indicates the value of the correlation without the removal of topics.

On the Terabyte 2006 and Web 2014 tracks, removing topics selected by risk estimation could not raise the rank correlation. However, the low empirical maximum indicates that the potential for improvement is also low. Surprisingly, within the removal of five topics the impact-score performed the worst out of all risk estimations on Terabyte 2006 track and Web 2014 track. On these tracks, the reldup-score performed the best within the removal of five topics. It has to be noted that on Terabyte 2006, the dup-score performs nearly identical to the reldup-score.

On Web 2009, all risk estimations deliver marginal improvements and perform similar to each other. The empirical maximum indicates that half of the potential for improvement has been exhausted. On Web 2010, the impact-score and the dup-score perform nearly identical, while the reldup-score unexpectedly does not deliver any improvements. The risk estimations do not fully leverage the potential improvement. Nevertheless, the dup-score and the impact-score substantially improves the rank correlation.

On the Web 2012 track, the dup-score improves the correlation, if at all, only marginally. The reldup-score delivers the full potential of improvement when removing one to two documents but steeply falls behind the impact-score when more documents are removed. None of the risk estimations can reach the empirical maximum if three to five documents are removed.

In summary, the removal of topics can improve the rank correlation between the evaluation that applies global judgment manipulation and the conventional evaluation. On the one hand, the rank correlation of the lightly influenced tracks, like Terabyte 2006 and Web 2014, is marginally deteriorated by the risk estimations. On the other hand, the risk estimations improve the rank correlation on the tracks that are most strongly influenced by duplicates, like the tracks Web 2009, Web 2010, and Web 2012. The impact-score does overall deliver the results, but for the most part, one of the other risk estimations is on par with it. This might indicate that the reldup-score and the dup-score measure different sources of influence on the rank correlation. The reldup-score directly measures the influence of the relevant duplicates. The dup-score measure the impact of the duplicates on a topic in general. A combination of the dup-score and reldup-score might improve the rank correlation even more.

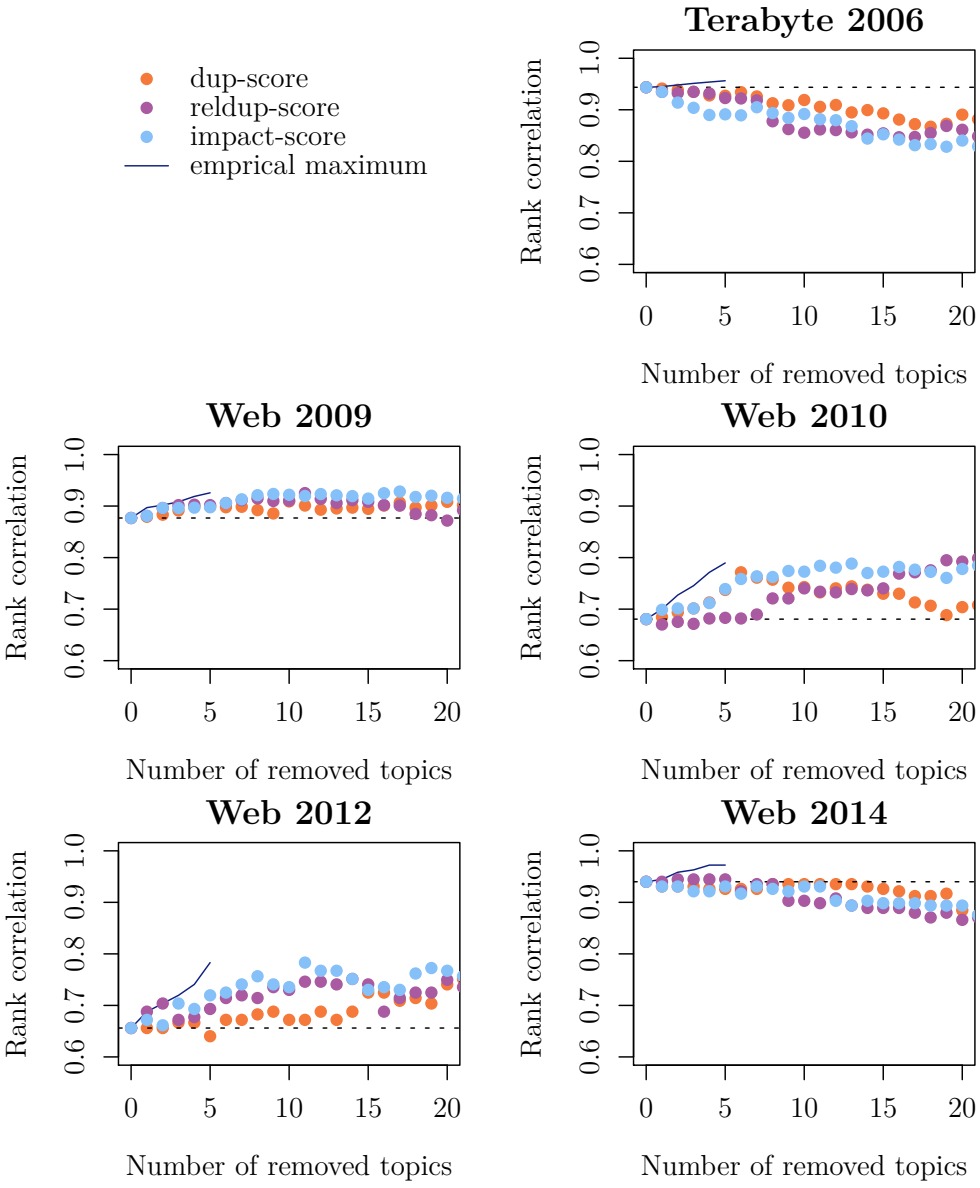


Figure 6.4: The graphs show the rank correlation of the system between the globalmax evaluation, where only novel documents can be relevant and the baseline evaluation less the most risky topics for the tracks TREC 2006 Terabyte Track, TREC 2009 Web Track, TREC 2010 Web Track, TREC 2012 Web Track and TREC 2014 Web Track. There are three differently colored sequence of correlation coefficients, where the risks are estimated by dup-score (orange), reldup-score (purple), and impact-score (lightblue).

Table 6.1: The Table lists the difference ($\Delta\tau$) rank correlation between the filtered conventional evaluation and the evaluation under global judgment manipulation, as well as the correlation without any applied filters (τ). The applied filters are the dup-score (dup), the reldup-score (reldup), impact-score (impact)

	τ	$\Delta\tau$		
Risk	-	dup	reldup	impact
TB04	0.975	−0.011	−0.004	−0.004
TB05	0.961	−0.025	+0.001	−0.008
TB06	0.943	−0.016	−0.020	−0.052
Web09	0.876	+0.021	+0.024	+0.020
Web10	0.680	+0.056	+0.002	+0.058
Web11	0.912	+0.012	+0.018	+0.015
Web12	0.656	−0.015	+0.037	+0.063
Web13	0.942	+0.007	+0.014	0.000
Web14	0.939	−0.013	+0.004	−0.009

Table 6.1 concludes our results. The removal of five topics selected by the reldup-score delivers improvements on most of the tracks. Despite the lower manual effort, the reldup-score is more reliable than the impact-score. On tracks strongly influenced by duplicates, both the reldup-score and the impact-score select topics that increase the rank correlation. Utilizing the dup-score increases the correlation on strongly affected tracks but fails on lightly affected tracks. Mainly the TREC 2010 Web Track sticks out, as removing the topics selected by the dup-score significantly increases the correlation. The improvement is almost as significant as with the impact-score, while the reldup-score only delivers marginal improvements.

Our results suggest that the reldup-score is a reliable risk estimation since it does not deteriorate the correlation. Compared to the impact, the reldup-score gives small to mediocre improvements. The dup-score and the impact-score seem to lack this reliability. Both deliver significant improvements on tracks, which are strongly influenced by duplicates, but also decrease the correlation on the other tracks. It may be possible to increase the performance by combining the dup-score and the reldup-score. Organizers of shared tasks may be able to select and discard risky topics based on the reldup-score, and reliably decrease the influence of duplicates on their shared tasks. Given the limited sample size, further research is needed to give a clear recommendation.

Chapter 7

Future Work and Conclusion

We studied the extend and impact of duplicates on web corpora as well as newspaper corpora.

We implemented a highly parallel program to identify retrieval-equivalence across billions of documents and successfully deployed it on a cluster. All studied web corpora contain high amounts of retrieval-equivalent documents. On all corpora, the duplicates distribution follows Zipf's law. We reproduced Bernstein and Zobel's results on the .GOV corpora. The percentage of duplicates in the Clueweb corpora is lower than in the .GOV 2 corpus. Nevertheless, both Clueweb corpora still contain over 7.74% retrieval-equivalent duplicates. Neither of the newspaper corpora contains a significant amount of retrieval-equivalent documents.

We reimplemented Bernstein and Zobel's approach to also find documents that contain the same content but slightly deviate from each other, for example, due to a timestamp. The web corpora's judged documents contain roughly 18% content-equivalent documents. Relevant documents do not contain significantly more content-equivalent documents than all judged documents. The median number of relevant duplicates per topic is reasonable close to the tracks' average number of duplicates. However, there are topics with extreme amounts of relevant duplicates of up to 80%. Inconsistencies among the judgments are particularly high in the Terabyte 2004/2005 and Web 2009/2010 tracks. Our results support the findings of Bernstein and Zobel for the Terabyte 2004 track. We successfully reproduced their results within margins of error.

We also replicated their experiments on the impact of duplicates on the Terabyte 2004 track and reproduced their results. Furthermore, we enhanced their approach by introducing global judgment manipulation and examined more and newer tracks. Duplicates significantly impact the evaluations of all the Terabyte and Web tracks, but mainly affected tracks on the Clueweb09 corpus. Our results suggest, that some systems on the Web 2010 track actively

avoided the retrieval of duplicates, while others did not, causing substantial performance drops under a novelty aware evaluation. These performance drops even occur among the top three submissions of the Web 2010 track. Further research and a detailed analysis of the submission may cement this conjecture. The impact on the Core 2017/2018 tracks is insignificant due to the low number of overall duplicates in their respective corpora. Parts of our results from the experiments described in chapter 5 and chapter 6 have been published at the European Conference on Information Retrieval in 2020, which shows the relevance of our research [Frö+20].

Duplicate free corpora eliminate the need for novelty aware measures. However, a duplicate free corpus may not always achievable nor desirable, depending on the tracks target domain. A novelty aware evaluation technique can solve these issues, but might not appeal to everyone. For that reason we tested a strategy to minimize the impact of duplicates, by removing strongly by duplicates affected topics before the evaluation. To asses which topics to remove, we tested three different risk estimation. Even though we found combinations of topics that deliver great improvement by brute force, none of these proofed reliable and effective enough. Further research may discover more reliable, and more effective risk estimations.

In conclusion duplicates still significantly impact the research into information retrieval system and there are no simple solutions to this problem.

Bibliography

[BR11] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. 2nd ed. Harlow, England: Pearson Addison Wesley, 2011. ISBN: 978-0-321-41691-9.

[Bro97] Andrei Z Broder. "On the resemblance and containment of documents". In: *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*. IEEE. 1997, pp. 21–29.

[BZ05] Yaniv Bernstein and Justin Zobel. "Redundant documents and search effectiveness". In: *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*. 2005, pp. 736–743. DOI: 10.1145/1099554.1099733. URL: https://doi.org/10.1145/1099554.1099733.

[Cab+10] Guillaume Cabanac et al. "Tie-breaking bias: effect of an uncontrolled parameter on information retrieval evaluation". In: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer. 2010, pp. 112–123.

[CCS04] Charles LA Clarke, Nick Craswell, and Ian Soboroff. "Overview of the TREC 2004 Terabyte Track." In: *TREC*. Vol. 4. 2004, p. 74.

[CD10] Christophe Croux and Catherine Dehon. "Influence functions of the Spearman and Kendall correlation measures". In: *Statistical methods & applications* 19.4 (2010), pp. 497–515.

[CH02] Nick Craswell and David Hawking. "Overview of the TREC-2002 web track". In: (2002).

[Cla+08] Charles L. A. Clarke et al. "Novelty and diversity in information retrieval evaluation". In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*. 2008, pp. 659–666. DOI: 10.1145/1390334.1390446. URL: https://doi.org/10.1145/1390334.1390446.

[FMN03] Dennis Fetterly, Mark Manasse, and Marc Najork. "On the evolution of clusters of near-duplicate web pages". In: *Proceedings of the IEEE/LEOS 3rd International Conference on Numerical Simulation of Semiconductor Optoelectronic Devices (IEEE Cat. No. 03EX726)*. IEEE. 2003, pp. 37–45.

[FMN04] Dennis Fetterly, Mark Manasse, and Marc Najork. "Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages". In: *Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS 2004*. 2004, pp. 1–6.

[Frö+20] Maik Fröbe et al. "The Effect of Content-Equivalent Near-Duplicates on the Evaluation of Search Engines". In: *European Conference on Information Retrieval*. Springer. 2020, pp. 12–19.

[Hen06] Monika Henzinger. "Finding near-duplicate web pages: a large-scale evaluation of algorithms". In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 2006, pp. 284–291.

[JK02] Kalervo Järvelin and Jaana Kekäläinen. "Cumulated gain-based evaluation of IR techniques". In: *ACM Transactions on Information Systems (TOIS)* 20.4 (2002), pp. 422–446.

[Ken48] Maurice George Kendall. "Rank correlation methods." In: (1948).

[San08] Evan Sandhaus. "New york times corpus: Corpus overview". In: *LDC catalogue entry LDC2008T19* (2008).

[SHH18] Ian Soboroff, Shudong Huang, and Donna Harman. "TREC 2018 News Track Overview." In: *TREC*. 2018.

[ZCM02] Yi Zhang, James P. Callan, and Thomas P. Minka. "Novelty and redundancy detection in adaptive filtering". In: *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland*. 2002, pp. 81–88. DOI: 10.1145/564376.564393. URL: https://doi.org/10.1145/564376.564393.