

Detecting Duplicate Web Documents using Clickthrough Data

Filip Radlinski
Microsoft
840 Cambie Street
Vancouver, BC, Canada
filiprad@microsoft.com

Paul N. Bennett
Microsoft Research
1 Microsoft Way
Redmond, WA, USA
pauben@microsoft.com

Emine Yilmaz
Microsoft
7 J J Thomson Ave
Cambridge, UK
eminey@microsoft.com

ABSTRACT

The web contains many duplicate and near-duplicate documents. Given that user satisfaction is negatively affected by redundant information in search results, a significant amount of research has been devoted to developing duplicate detection algorithms. However, most such algorithms rely solely on document content to detect duplication, ignoring the fact that a primary goal of duplicate detection is to identify documents that contain redundant *information* with respect to a particular user *query*. Similarly, although query-*dependent* result diversification algorithms compute a query-dependent ranking, they tend to do so on the basis of a query-independent content similarity score.

In this paper, we bridge the gap between query-dependent redundancy and query-independent duplication by showing how user click behavior following a query provides evidence about the relative novelty of web documents. While most previous work on interpreting user clicks on search results has assumed that they reflect just result relevance, we show that clicks also provide information about duplication between web documents since users consider search results in the context of previously seen documents. Moreover, we find that duplication explains a substantial amount of presentation bias observed in clicking behavior. We identify three distinct types of redundancy that commonly occur on the web and show how click data can be used to detect these different types.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms

Experimentation, Measurement

Keywords

Clickthrough, Duplication, Redundancy, Utility, Web Search

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSDM'11, February 9–12, 2011, Hong Kong, China.

Copyright 2011 ACM 978-1-4503-0493-1/11/02 ...\$10.00.

1. INTRODUCTION

It is well known that presenting redundant search results to search engine users is often suboptimal. This has resulted in a large amount of research on producing diverse rankings of documents that avoid showing multiple documents with redundant content to users (e.g., [1, 8, 13, 28, 34, 36]). In particular, this research is often motivated by ambiguity often seen in queries issued to web search engines, with a variety of techniques being developed to assess how well a particular ranking or set of documents satisfies the possible intents for a given query (e.g., [15, 20, 35]).

Separately, information retrieval practitioners have long noted that many documents are duplicated on the web, leading to research in efficiently identifying and removing such duplicates (e.g., [5, 22]). This form of duplication is an extreme form of document redundancy in that it is just based on lexical similarity between documents and is query *independent*. It has thus previously been studied separately from novelty with respect to a query.

However, non-identical documents can be redundant with respect to some queries but not others. Furthermore, documents that are different than each other in terms of full content can still duplicate the information they contain with respect to a *particular query*. Similarly, documents that are very similar to each other in terms of overall content may still contain key different information for a particular query, and hence may not in fact be duplicates of each other.

In this paper, we are concerned with better characterizing redundancy and duplication, studying how query *dependent* duplication relates to user *utility*, and measuring whether duplication in terms of query-specific information content can be detected by observing search user behavior. Our key contributions are a unified taxonomy of redundancy of web documents and a straightforward technique for identifying whether a pair of documents falls into one of the duplication classes based on usage behavior. Importantly, we show that search engine usage can provide information about document redundancy which is separate from information about the relevance of the web results to the user's query. While it has been noted that presentation bias (i.e. search engine users' strong preference to look at and click on higher ranked results independently of document relevance) makes inferring relevance from usage data difficult [21, 25, 16], our results show that different types of presentation bias occur when different types of duplicates occur in search results.

We propose that there are three types of redundancy. The first and simplest readily identifiable type of redundancy is *exact duplication* of web pages. This occurs when two web

pages consist of identical content. The second, which we term *content duplication*, involves pairs of web pages that present essentially redundant information with respect to a user query, although provided by different sources or in different formats. Importantly, users may consider two results duplicate even if the textual content of the documents differs substantially – as in the case of competing web sites allowing users to play the same game online. Finally, usage data shows that when users click on the wrong page of the right web site given their query, they often navigate to the correct page directly rather than by returning to the search results. This leads us to define *navigational duplication* as pairs of pages where it is immediately obvious to a user how to navigate from the wrong one to the right one.

In the remainder of this paper, we first present related work and then a detailed description of how we constructed a collection for measuring duplication based on search engine usage data. We next describe how we learned a model for predicting duplication based on this usage, and we conclude with an evaluation of the model and its implications.

2. RELATED WORK

Most previous work in identifying duplicates is based on using similarities between document contents. Since discovering all possible duplicate documents in a document set of size N requires $O(N^2)$ comparisons, efficiency as well as accuracy are two main concerns of existing algorithms in the literature.

The simplest approach used for detecting *exact* duplicates is based on a fingerprint that is a succinct digest of the characters in a document. When the fingerprints of two documents are identical, the documents are further compared, and identical documents are identified as duplicates [26].

The above approach does not solve the problem of identifying *near* duplicates: web pages that are not identical but still very similar in content. Almost all previous algorithms for identifying near duplicates are based on generating n-gram vectors from documents and computing a similarity score between these vectors based on a particular similarity metric [22, 31, 12, 14]. If the similarity between two documents are above a particular threshold, the two documents are considered to be near duplicates of each other. One of the commonly used methods for detecting near duplicates is based on *shingling* [5, 7]. Given a sequence of terms in a document d , shingling is based on encoding each n-gram by a 64-bit Rabin fingerprint [27], which is referred to as a *shingle*. The similarity between two documents is then measured using the Jaccard coefficient between the shingle vectors [22]. A similar technique that combines the accuracy of similarity based methods and the efficiency of *shingling* was developed by Charikar [12]. Note that these are all query independent techniques.

In a different line of work, a number of algorithms have been proposed to enhance the novelty of the documents returned given a user query. Most such algorithms are based on the idea of maximal marginal relevance (MMR) [8]. The general idea of MMR is to re-rank an initial set of documents retrieved for a given query by iteratively selecting documents with the highest relevance to the query and highest dissimilarity to those already selected. Thus, MMR essentially re-ranks documents based on their conditional utility to the user. A number of approaches that aim at maximizing MMR have been devised based on different methods to

compute the similarity among documents: Carbonell and Goldstein [8] use a content-based similarity function; Zhai et al. [35] model relevance and novelty within the language modeling framework; and Wang & Zhu [33] employ a portfolio theory using the correlation between documents as a measure of their similarity.

All aforementioned algorithms for duplicate detection are based on using document contents. Methods that solely depend on similarities in terms of document contents have an important drawback: the main purpose of duplicate detection is to identify documents that contain similar *information* with respect to a user need. That is, in most cases, duplicate detection is aimed at identifying documents where either has zero utility given the other. However, when only document contents are used for duplicate detection, utility is ignored. Two documents can be of the same utility (containing duplicate information) even if the contents are different. For example, two newspaper articles describing exactly the same event but with different words may be duplicates of each other. Furthermore, two documents can be of different utility to an end user even if their contents are very similar. For example, two different documents containing a biography of Britney Spears, identically written except that one contains the birthday of Britney Spears while the other does not, are not duplicates of each other when the goal of the user is to find out Britney Spears’ age.

User behavior and click data, on the other hand, contain much information about the utility of documents. Consider two documents $d1$ and $d2$. If these documents are duplicates of each other, we should be able to observe from user behavior that document $d1$ is not clicked very often when it is retrieved below document $d2$ and vice versa. Previous work has mainly assumed that user behavior is mostly affected by the relevance of documents shown to the user. Consequently, models of user click behavior [3, 16, 11, 18] have been mainly used to infer relevance of documents [24, 30, 2, 10]. In this work, we show how click behavior can be interpreted as a signal for redundancy in a duplicate detection model, specifically, to identify documents containing the same information with respect to an information need.

Our work differs from previous work in several key aspects. First, while previous work has focused primarily on determining document redundancy from document content, we hypothesize and demonstrate a link between click behavior and the novelty of results. Second, this link enables us to broaden the scope of document redundancy into the realm of query-*dependent* document redundancy in a scalable (driven by search engine logs) fashion where most previous work has been limited to query independent redundancy. Finally, we not only demonstrate that click behavior can be used to predict these duplicates through learned models, the demonstrated click-behavior link enables us to mine query-dependent duplicates at a higher rate than random sampling – thus allowing efficient building of training sets for richer prediction models.

3. DUPLICATION IN SEARCH RESULTS

3.1 Usage Cues of Redundancy

Among the first methods to infer document relevance from usage was Joachims’ observation that if a user clicks on a document after skipping another higher ranked document, this likely indicates that the clicked document is more rele-

vant than the skipped document [24]. Noticing that such “skip-click” relevance judgments always oppose the ranking order, thus making it impossible to conclude that the higher ranked document is more relevant (i.e. if the original ranking was correct), Radlinski and Joachims extended the model by suggesting that if a pair of documents is shown in both possible orders, the document that is clicked more often when shown at the lower position is more relevant [29]. Further, they proposed the FairPairs algorithm to actively collect such relevance data.

We take their model a step further, considering all possible outcomes for both orders in which two documents can be presented. Consider a pair of web results, u and v being shown at adjacent positions in a web search ranking, at positions i and $i + 1$. These URLs can be shown in the order u, v , with u immediately above v , as well as in the order v, u .

Suppose u and v are observed at adjacent positions in a ranking for a fixed query q . Let $c^{\hat{u}v}(q)$ (subsequently $c^{\hat{u}v}$ for brevity) be the number of times the results were shown with u immediately above v , where u was clicked¹ and v was not. Similarly, let $c^{u\hat{v}}$ be the number of times v was clicked and u was not, and $c^{\hat{u}\hat{v}}$ be the number of times both results were clicked.

We hypothesize that the minimum fraction of clicks that occur only on the top result in both presentation orders, for a fixed pair of documents, is strongly related to the redundancy of the documents with respect to that particular query:

$$r(u, v) = \min \left(\frac{c^{\hat{u}v}}{c^{\hat{u}v} + c^{u\hat{v}} + c^{\hat{u}\hat{v}}}, \frac{c^{\hat{v}u}}{c^{\hat{v}u} + c^{v\hat{u}} + c^{\hat{v}\hat{u}}} \right) \quad (1)$$

While extreme presentation bias would also cause this redundancy score to be high, we hypothesize that duplicate pairs of results have a higher score (also interpretable as stronger presentation bias) than non-duplicate pairs. One of the goals of this work is to validate whether this simple metric correctly identifies duplicate pairs of documents, and if it does, we further desire to find stronger indicators of duplication.

To verify that this score does in fact differ across different pairs of search results, Figure 1 shows the distribution of redundancy score for a commercial search engine for all pairs of results shown in both orders at least ten times with a click on either of the results. We see that there is a wide distribution of redundancy scores, and thus we can assess whether the value of this score correlates with redundancy. Moreover, we postulate that the distribution of redundancy score for a particular web search system is a usage-based measure of how often duplicate results occur in the search result rankings².

We also observe that our redundancy score is very related to the metric used by Radlinski and Joachims [29], where

¹As suggested by Fox et al. [19], to obtain cleaner data we consider a result to be clicked on if a user clicked on this result, and did not click on any other result or issue another search query within 30 seconds.

²However, note that the distribution seen in Figure 1 comes from search engine logs that did not involve controlled randomization of results, thus the pairs of results that were frequently shown in both orders are not necessarily a representative sample, and in fact are likely to be pairs of results that are scored similarly by the search engine ranker, swapping in the ranking due to small day-to-day changes in ranking feature values.

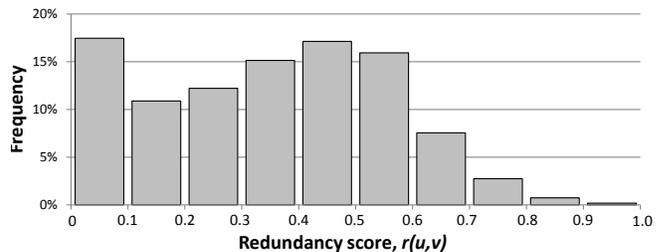


Figure 1: Redundancy score distribution for all pairs of results observed and clicked in both orders by at least ten users.

the authors considered one document more relevant if

$$\frac{c^{u\hat{v}} + c^{\hat{u}\hat{v}}}{c^{\hat{u}v} + c^{u\hat{v}} + c^{\hat{u}\hat{v}} + c^{uv}} \neq \frac{c^{v\hat{u}} + c^{\hat{v}\hat{u}}}{c^{\hat{v}u} + c^{v\hat{u}} + c^{\hat{v}\hat{u}} + c^{vu}}$$

with statistical significance. The difference is that whereas they considered clicks on the lower result as a *relevance* signal, we postulate that a click on the top result is a *redundancy* signal. We will further discuss the relationship between relevance and redundancy in the evaluation.

3.2 Classes of Duplication

Observing pairs of documents with high redundancy score $r(u, v)$ in search engine logs, we found three types of redundant pairs of results:

- **Exact duplicates**, where both pages appear identical, perhaps with the exception of advertisements.
- **Content duplicates**, where both pages essentially provide the same information with respect to the query, but from different sources.
- **Navigational duplicates**, where navigating from one page to the other is very easy. Note that one of the pages may be more relevant than the other.

While exact duplicates do not require examples, we illustrate the others with specific examples.

Examples of content duplicates include two different web sites with lyrics for the same song, two different documents with similar recipes for oatmeal cookies, or two different sites for converting centimeters into inches. While these alternatives may differ in relevance (for example, due to the clarity of presentation), we expect that most users would find either redundant if they have already observed the other. This study will determine whether users in fact behave in a way consistent with this hypothesis.

Note the difference between our definition of content duplicates and the concept of near duplicates studied in the literature: content duplication here refers to duplication in terms of *information* content, whereas near duplication pertains to duplication in terms of lexical content. As an example, two different documents with similar recipes for oatmeal cookies can be content duplicates but not near duplicates if the documents are describing similar recipes with different words.

Navigational duplicates are different: here one page is very often more relevant than the other, but it is conceivably easier (or less risky) for a user to get to the more relevant result from the less relevant result by browsing than by returning

to the web search results. This is often the case when a user clicks on one search result without having considered the next result; if the user expects that the cost of backing out from an almost correct result to find the right result in a search engine ranking is higher than the expected cost of navigating, then the user is likely to choose to navigate instead. Examples of navigational duplicates are the homepage and the sports page of a newspaper, or the online banking login page of a bank and the “contact us” page of the same bank. Occasionally, we also observed pairs of results where neither was quite what the user was looking for, but navigating to the correct page from either was equally trivial.

3.3 Acting on Duplication

Assuming that it is possible to detect various types of duplication in web search results, this information can be used in many different ways. It is already recognized that presenting exact duplicates of search results hurts user experience, as it makes it more difficult for users to find alternative information if the duplicated document is not satisfying. Identifying such duplicates not detected by content-based techniques will clearly improve user satisfaction.

For the same reasons that exact duplication hurts search experience, we expect navigational duplication to often be suboptimal. Reducing a pair of navigationally duplicate results to one is clearly more difficult, especially as one is often more relevant. However, we expect that in such cases a different user interface, for example combining navigationally duplicate results into one larger search result that users can more clearly recognize as different entries into the same website, may be the optimal treatment. We note that in some cases, navigational duplicates are not obviously detectable by observing the hostname of the URLs, as certain web sites use a number of different domains.

Content duplicates may well motivate an entirely different search result treatment. For instance, if different sources of related information are available to search results, it would likely be beneficial to highlight to users the *differences* between the alternatives rather than simply focusing summaries on why the results are relevant to the query. This would assist users in selecting the best result for their query without needing to visit all such results.

Beyond users, when building search engines relevance judgments are needed both for optimization and evaluation purposes. Relevance judgments are usually human-generated, and are therefore expensive and slow to produce. Duplicate detection can also be used to reduce the human judgment effort needed to generate the relevance judgments. As an example, duplicate detection can be used to automatically label the training data as being redundant or belonging to a category of duplication or redundancy. This automates one stage of the labeling process, thereby reducing the time to hand-annotate the relevance judgments. Duplicate detection can also be used to verify the labeled data. For example, two documents that are identified as duplicates of each other should have similar relevance labels.

Another application of the duplicate detection technique is to infer improved relevance values given clicks. Most previous work on click modeling assumes that clicks are a direct function of relevance, ignoring the fact that clicks are highly affected by duplication. If a highly relevant document d_1 is always presented below its duplicate d_2 , it is unlikely that

d_1 will be clicked. Previous work infers from this that d_1 is not relevant with respect to the given query. Duplicate detection can be used to enhance the previous click prediction models to incorporate the effect of duplication, resulting in more accurate inferences.

4. EXPERIMENT DESIGN

To validate our proposed taxonomy, we constructed a corpus of queries and associated pairs of URLs that exhibit a range of redundancy scores. In this section, we describe how this corpus was constructed.

4.1 Sampling URL Pairs

The goal of our corpus was to study redundancy in web search results, hence we obtained a sample of pairs of results with varying levels of redundancy with respect to a query. Using one month of logs from a commercial search engine (from May 2010), we first extracted all tuples `<query, result-1, result-2>` with the results shown adjacently in both orders for the query, where either of the results was clicked¹ at least 10 times in each presentation order. From these tuples, we performed stratified sampling according to the result redundancy score as defined in Equation 1. Specifically, we randomly sampled 120 tuples with $r(u, v) < 0.1$, 120 tuples with $0.1 \leq r(u, v) < 0.2$, 120 tuples with $0.2 \leq r(u, v) \leq 0.3$ and so forth.

We also extracted all tuples where the results were shown in both orders at least 100 times, and at least one result on the web page was clicked, irrespective of whether one of the results in the pair was clicked. From this set, we randomly sampled 126 tuples.

Combining this stratified sample into one set, we removed all pairs of results that obviously contain adult content, ending up with 1350 distinct tuples consisting of a query and a pair of URLs.

4.2 Document Pair Judgments

For each query and result pair, a judge was then shown the query, the two result URLs, and the content of the pages side by side (but not the snippet shown by the search engine or any information about user behavior on the results). The tuples were judged in random order by the authors of this paper according to the following three questions, motivated by the three hypothesized classes of duplicates:

1. Which page is most relevant to the query?

The judge could select:

- *Relevant to Different Intent* if the two results appear to be possibly relevant to the query, but are relevant to different meanings of the query.
- *Left* or *Right* if one of the results is more relevant.
- *Neither relevant* if neither result is relevant to any intent of the query.
- *Both equally relevant* if both results are approximately equally relevant to the same intent of the query.

2. How similar is the utility of these two pages for the query?

The judge could select:

- *Identical pages* if the pages are exactly identical (with minor exceptions such as advertisements).

- *Very similar* if the pages are different but appear to be essentially of equal utility for the query.
- *Related* if the pages convey some of the same information, for example with one page providing more details than the other.
- *Different* if the pages provide different information, with one of the pages potentially much more useful to the user.

3. Is it easy to navigate from either page to the other?

The judge could select

- *Yes (within site)* if it was obvious how to navigate from one of the pages to the other, and the two pages appeared to be on the same website (in terms of style, but not necessarily at the same hostname).
- *Yes (across sites)* if it was obvious how to navigate from one of the pages to the other, but the two pages are hosted on different websites.
- *No* otherwise.

The reason we used a separate question for comparing the relevance of the two documents is mainly to identify the effect of more authoritative pages: two documents may be of very similar utility yet one may still be more relevant than the other.

To limit the judging effort, we simplified the judging in two cases: (1) if a judge selected that two results are relevant to different intents for the first question, the second question was disabled since it would be difficult for a judge to compare the utility of one page to one intent with the utility of the other page to the other intent, and (2) if a judge selected that two pages are identical for the second question, the third question was disabled as it is irrelevant.

As an alternative, we considered simply asking the judges to label each pair as falling into one of the duplication classes, but found that using a descriptive question rather than a class name improved the clarity of the task required of the judges. We also considered judging pairs of URLs without considering the query, however found that utility of two pages was often query dependent, with for example two song lyrics sites perhaps equally useful if the query is “*song name* lyrics”, but not equally useful if the query indicates the need for additional information such as “*song name* lyrics meaning” which one site may provide while the other does not.

Of the 1,350 sampled (query, result, result) tuples, 90 were judged by all three judges, with the remaining tuples split equally between all judges (all tuples judged by a judge were randomly sorted, so the judges did not know which tuples were to be used for inter-judge agreement). Any tuples where the results are in a foreign language, where either of the results contains adult content, or where either result did not permit itself to be showed in a html frame element (used in the judging interface) were not judged.

In this way, we obtained 1,102 judged tuples, mostly due to many of the tuples containing at least one result that does not show in a html frame element, or due to one of the URLs in the tuple no longer being valid.

Table 1: Judgment frequency on the corpus of pairs of results for the three questions in Section 4.2.

	More Relevant?	Utility?	Navigation?	Freq.
E	Both equally	Identical	-	6%
	Both equally	<i>any</i>	Yes within	5%
N	Left or Right	<i>any</i>	Yes within	13%
	Both equally	Very similar	No	16%
C	Left or Right	Very similar	No	8%
	Both equally	Related	No	4%
C_w	Left or Right	Related	No	15%
	Different intents	-	No	12%
	Left or Right	Different	No	12%
	Left/Right/Both	<i>any</i>	Yes across	2%
	Neither Relevant	<i>any</i>	<i>any</i>	4%
	<i>other</i>	<i>other</i>	<i>other</i>	3%

5. EVALUATION

We now evaluate our duplication taxonomy from two perspectives. First, in terms of judges, we assess both whether duplication can be reliably judged and whether the judges found many duplicates among the tuples judged. Second, we evaluate in terms of prediction: given usage behavior, is it possible to train a model to determine if a pair of URLs is a duplicate, and is it possible to distinguish the different classes of duplicates?

5.1 Judgment Analysis

The patterns of judgments observed for the tuples sampled are shown in Table 1, with the answer to each question in a separate column, and the last column showing the frequency with which this pattern of answers was observed. Further, we labeled the tuples with duplication classes as shown on the left of the table, where E indicates *exact duplicates*, C indicates *content duplicates*, and N indicates *navigational duplicates*.

The small fraction of tuples where the judge indicated that both results were not relevant to the query were removed (as they could be reasonably considered as either not duplicate or duplicate), and all other judgment patterns were taken to indicate *not duplicate*. In particular, we chose to only consider within-site navigation as indicative of navigational duplicates, as obvious cross-site navigation appears to be rare. Also, as we will see later, assessing content duplication proved to be hardest for judges, so we annotate two patterns as *weak* content duplicates, with C_w, although we will consider tuples labeled with these patterns as non-duplicates in the remainder of this paper.

Given the frequencies in the rightmost column, about 48% of the tuples were judged as duplicate in some way. Note that this frequency is so high due to the sampling technique used, with tuples with low redundancy scores much more frequent in natural search results (as shown in Figure 1) than in the stratified sample judged. Note also that our approach is such that the same (query, result, result) tuple cannot be considered both a navigational duplicate and a content duplicate, with navigational duplication effectively given priority in such cases.

Inter-Judge Agreement

Of the 90 tuples judged by all three judges, 79 were considered judgeable by at least two of the three judges. On

Table 2: Frequency with which each possible pair of judgments was made for tuples judged by two or more judges.

Judgment 2	Judgment 1				
	Ex.	Nav.	Cont.	C_w	Not Dup.
Exact (E)	0	0	0	0	0
Navigational (N)		19	1	3	2
Content (C)			31	27	12
Weak Cont. (C_w)				15	32
Not Duplicate					78

these judgments, 87% of individual judgment labels agreed with the majority label (where such a majority existed, otherwise the tuple majority was taken to be *not duplicate*). Additionally, each of the judges produced each class label approximately equally often.

Despite this, when considering all pairs of judgments obtained for the same tuple, we found relatively poor inter-judge agreement, particularly in assessing whether a pair of URLs is a content duplicate. Table 2 summarizes how many times each combination of judgments was made by every pair of judges for every tuple, including weak content duplicates as an outcome.

We see that there were no exact duplicates in the shared judged set (which is not too surprising given the small size of the set judged by all judges). The inter-judge agreement on navigational duplicates is high, with the disagreements being caused by the judges disagreeing if navigating from one result to the other is obvious.

However, consider the content and weak content judgments: We see that many tuples judged as content duplicates by one judge (i.e. with the two URLs having “very similar” utility to the query) were judged as weak content duplicates by another judge (i.e. with “related” utility). Similarly, often a tuple judged as a weak content duplicate was judged as not duplicate by another judge. This shows that it was difficult for the judges to determine the threshold between the possible judgments for the second question. Despite this, note that if one judge considered two documents as having “very similar” utility, the other judge(s) would usually rate the documents at least as “related”. We chose to be conservative in determining the labels for tuples, only taking pairs judged “very similar” in utility as content duplicates, effectively erring on the side of precision rather than recall when the documents are presented to an end user. This is motivated by the assumption that treating two results as duplicate when they are not would have higher cost to a user than treating two results as non-duplicate when they in fact are.

On Relevance versus Duplication

As shown in Table 1, for 8% of tuples the two documents were considered to be of very similar utility, but with one of them considered more relevant than the other. This difference often happened where one document was better presented or more authoritative, while both documents provided essentially the same information (for example, a Wikipedia document versus a more poorly presented non-Wikipedia reference article, or two online stores where one appears more trustworthy). This suggests that even when

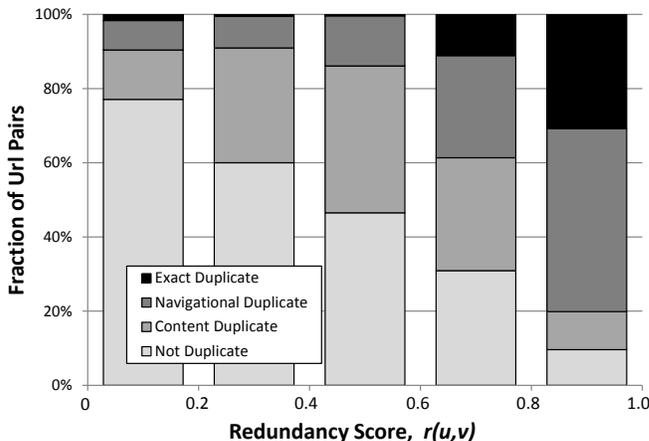


Figure 2: Judgment distribution for each range of redundancy score as computed by Equation 1.

two documents may be redundant to users, picking the correct representative is still likely to be important.

A related navigational case was also found to occur, where neither of the documents judged was exactly relevant to the user’s query, yet both allowed the user to trivially navigate to a third, correct, URL. Depending on whether it was possible to navigate from one result to the *other result*, these may have been labeled as navigational duplicates, but this effect suggests an alternative definition of navigational duplicates that may better reflect utility to end users.

5.2 Duplicate Prediction

We now use the duplication judgments to assess whether our initial hypothesis about click behavior is correct. We will see that usage behavior allows duplicates to be identified, and that users respond differently to the different classes of duplicates when they are present in search results.

Correlation of Redundancy Score with Duplication

Figure 2 shows the fraction of tuples judged as exact, content, navigational and non-duplicate as a function of redundancy score. It confirms our hypothesis from Section 3.1: Tuples with low redundancy scores were usually judged as non-duplicate, with tuples with high redundancy scores usually being judged as exact or navigational duplicates. Around the middle of the range, content duplicates are more common. Seen another way, presentation bias effects are strongest for navigational and exact duplicates and much weaker for non-duplicates.

While this result shows us that for a given redundancy score we can estimate the probability that a pair of URLs is redundant for a particular query, Figure 3 shows the distribution of redundancy scores for each class of duplicates. Interestingly, we see that while exact duplicates usually have high redundancy scores, sometimes the redundancy score is small. This demonstrates an implicit assumption in all click modeling: that search result snippets reliably reflect the content of web pages, so that clicks can reflect user’s perceptions of the target page. Upon further investigation, it appears that in some cases exact duplicate results were shown by the search engine with different short descriptions (or snippets), with one strongly preferred by users.

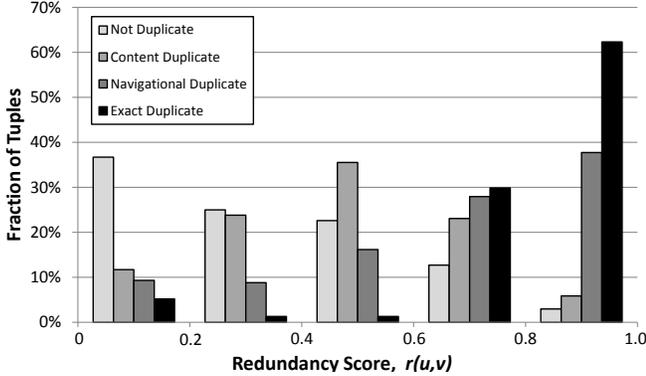


Figure 3: Distribution of redundancy score for each class of duplicate, as well as overall.

Moreover, these results indicate that the classes of duplicate do promote different usage behavior, with users more likely to click the higher presented result in the case of exact or navigational duplicates than in the case of content duplicates. In fact, few pairs of content duplicates just have the higher position result clicked reliably. Rather, the less-often clicked result is clicked less than 50% of the time when shown higher for half of all content duplicate pairs. However, content duplicates do still result in different user behavior than non-duplicate pairs, where presentation order matters even less in determining which result users will click on.

This last result from Figure 3 suggests that if we wish to infer *relevance* information from clicking behavior, controlling for redundancy may be key. Specifically, the redundancy score is a measure of the minimum rate, given both presentation orders, at which the top result is clicked. When it is small, there exists a presentation order in which the bottom result is clicked often. Thus, if we know that a pair of results is not duplicate, the effect of presentation order on which document is clicked is much smaller than on average, and thus clicking is more indicative of relevance.

Key Indicators of Duplicate Class

To better understand the different click signal indications of duplication, we trained a decision tree classifier [4] using only basic click and URL features to identify the different duplicate classes, using our complete judged corpus. The resulting decision tree is shown in Figure 4, giving an intuitive and easy to read classifier, although not one with optimal prediction performance.

In the figure, the *top-click rate* is the fraction of clicks on the top result of a pair, for a particular presentation order. The *minimum top-click rate* is the minimum fraction of clicks on the top result across both possible orders, i.e. exactly the redundancy score as described in Equation 1. Similarly, the *bottom-click rate* is the fraction of clicks on the bottom result of a pair for a particular presentation order, and the *both-click rate* is the fraction of impressions for which both results were clicked.

We first see that if the two result URLs have the same hostname, they are classified as navigational duplicates. This is largely due to the limited size of our collection, as adjacent pairs of results from the same host are relatively rare

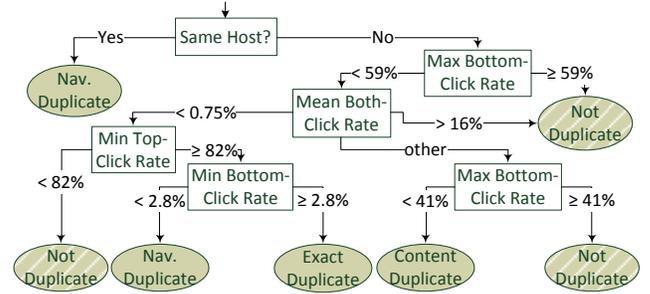


Figure 4: Decision tree duplicate classifier.

in search results. However, this is also indicative of most navigational duplicates being on the same hostname. In our case, about 65% of the examples on the same host are in fact labeled as navigational duplicates, with most of the remainder (24%) being labeled as exact duplicates. Only about 7.5% of document pairs on the same host in our corpus are labeled as not duplicate.

Considering pairs of URLs not on the same hostname, we see that if there is an ordering of the results where the lower ranked result is clicked more than 59% of the time, or if both results are clicked often, then the pair of results is likely not duplicate for this particular query. This makes intuitive sense, as it indicates that users often consider one result much less relevant (i.e. skip over it), or that users often click on one result, are not sufficiently satisfied, and hence also click on the other one. It is particularly interesting to observe how content and navigational duplicates are distinguished: If both results are almost never clicked, then a pair of results is more likely to be a navigational duplicate. In contrast, users more often click on both results of a content duplicate pair. Finally, the node distinguishing between navigational and exact duplicates likely indicates that exact duplicates more often include one result with particularly poor snippet quality than navigational duplicates.

It is also informative to observe the label confusion rate of this very simple classification model, which is shown below.

		Predicted			
		Content	Nav	Exact	Not
Actual	Content	107	10	2	154
	Navigational	13	131	7	53
	Exact	4	49	17	7
	Not Duplicate	57	20	1	426

This shows that using this simple model about 81% of tuples classified as one of the three classes of duplicates are in fact duplicates of some form, with a recall of about 61%. We see that the model confuses navigational and exact duplicates most often. The model predicts many examples as content duplicates when they are not labeled as such (with a precision about about 60%), however we recall that our definition of content duplicate is conservative, as discussed in Section 5.1. The confusion between navigational duplicates and non-duplicates, and exact duplicates and non-duplicates is lower. As such, this model appears to be a reasonable weak indicator of result duplication that is stronger than simply relying on redundancy score, which in particular does not model how often users click on both results versus skipping over one of them.

Table 3: Distribution of duplicate types for navigational and non-navigational queries.

Duplicate Type	Query Type	
	Non-Navigational	Navigational
Content	35%	8%
Navigational	11%	35%
Exact	5%	13%
Not Duplicate	49%	44%

Effect of the Query on Duplication

We classified the queries for which judgments were collected as either navigational or non-navigational. A navigational query is one where the user intends to reach a specific URL [6]. To classify the queries, we observed if users who issue the query have a strong preference for exactly one URL, i.e. clicking on this one URL at least 60% of the time. Approximately one third (354) of the queries for which at least one pair of URLs had been judged were thus classified as navigational, with the remaining two thirds (704) classified as non-navigational.

Table 3 shows the distribution of duplicate labels observed in these two classes of queries. We see that the types of duplicates observed for navigational queries differ substantially from those observed for non-navigational queries, although in both cases the frequency with which a pair of URLs was judged as some form of duplicate did not differ substantially. Specifically, if the query is navigational, navigational and exact duplicates are observed more often. Conversely, if the query is not navigational, most duplicates are content duplicates.

While this is to be expected, as navigational queries are more likely to produce search results with multiple pages on the same site between which a user may wish to browse, this shows that all types of duplicates commonly occur for both navigation and non-navigational queries. Moreover, if the goal of duplicate detection is to identify redundant results for specific queries, this suggests that the best model to identify a duplicate may well depend on the type of the query issued by the user.

Predicting Duplicate Class

Using the basic click and URL features mentioned above in addition to the query type (navigation or non-navigational), we learned a variety of models to predict whether or not a pair was a duplicate and the type of duplicate that it was. For all methods, we used 10-fold cross-validation over the data. As above, we learn a decision tree model using CART. In addition, we also learn a model using logistic regression (LogReg below). We assume that a high precision classifier (with respect to predicting duplicate) is likely of interest for automatic use within a system and focus our attention on this performance measure by reporting $F_{0.5}$ [32] which weights precision twice as heavily as recall.

In addition, we use MetaCost [17] to learn a cost-sensitive classifier to emphasize the desired direction of high precision performance. MetaCost is an algorithm that can turn any other classification algorithm into a cost-sensitive algorithm treating the inner algorithm as a black box. For the cost settings, we set the cost between duplicate types to be 0.2, the cost of misclassifying a duplicate as a non-duplicate as 0.5, and the cost of misclassifying a non-duplicate as a dupli-

Table 4: Basic Performance Summary.

Method	Dupe Prec.	Dupe Rec.	Dupe $F_{0.5}$
Baseline	0.52	1.00	0.58
CART	0.75	0.65	0.73
MetaCost CART	0.82	0.58	0.76
LogReg	0.77	0.62	0.74
MetaCost LogReg	0.87	0.46	0.74

Table 5: LogReg Confusion Matrix

		Predicted			
		Content	Nav	Exact	Not
Actual	Content	104	10	3	153
	Navigational	9	124	20	53
	Exact	9	55	8	5
	Not Duplicate	75	24	2	408

Table 6: MetaCost CART Confusion Matrix

		Predicted			
		Content	Nav	Exact	Not
Actual	Content	77	12	7	174
	Navigational	11	132	15	48
	Exact	3	57	9	8
	Not Duplicate	47	22	2	438

cate as 1.0. These were simply chosen to illustrate our point that a variety of cost/benefit trade-off points are possible. For particular applications more precise utility costs can be estimated from data based on perceived impact on the user. All models were learned using the WEKA [23] toolkit with default parameter settings.

Table 4 presents a basic summary for the baseline of always predicting duplicate versus the learned models. Here for simplification in result analysis, precision and recall have been collapsed to “duplicate” versus “not duplicate” where predicting the wrong type of duplicate is not penalized. As can be seen in the table, all of the learned methods achieve similar $F_{0.5}$ scores and all are significantly higher than the baseline. Since $F_{0.5}$ only represents one point in the precision-recall trade-off, Figure 5 presents a full precision-recall curve.

Examining the curve, one notices that the simple logistic regression model produces nearly the best precision at every level of recall. The exception to this is around a recall of 0.6 where the MetaCost CART model is best – this is exactly the region where the optimal $F_{0.5}$ lies. As we go to the left of the curve, although both MetaCost CART and LogReg models perform well, the logistic regression model is able to achieve slightly higher precision. For example the logistic regression model achieves a recall of 43% (identifying nearly half of all of the duplicates) while maintaining a precision of 90%. Note that recall above this starts dropping off more rapidly with a recall of 19% at a precision of 95% indicating the challenge in capturing the remaining portion of precision.

Since the LogReg and MetaCost CART models are the best performers, we present the full confusion matrices for only these two models in Tables 5 and 6. We note the general trends are similar, with the primary challenge for both models being the separation of content duplicates from those that are not duplicates.

We note that extending these classifiers with content features based on the documents, snippets, anchor text, and so

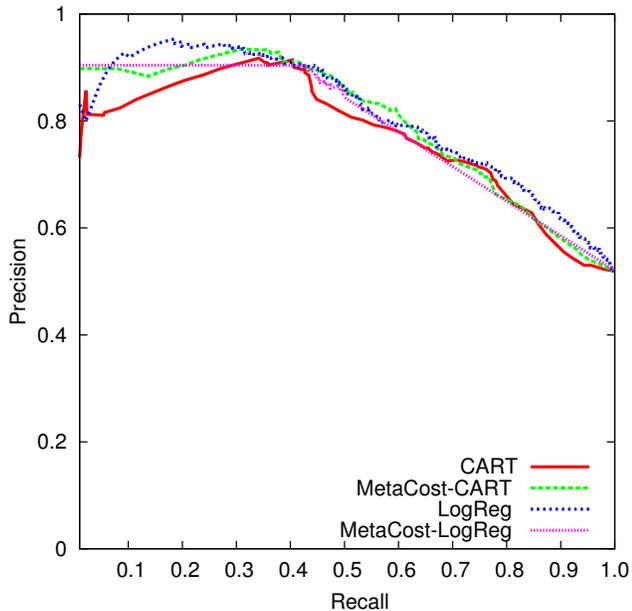


Figure 5: Precision-Recall curve for predicting “Duplicate” (all types).

forth, is very likely to increase performance. The goal here was simply to demonstrate that a range of reasonable prediction performance settings were possible using simple non-content features primarily determined by behavioral data.

5.3 Discussion

User Utility of Duplicates

Given that a pair of results is classified as some form of duplicate, we now discuss the question of how this affects the utility of a set of search results.

As seen in our results, users respond differently to different types of duplicates, so they each have a different impact on user satisfaction. Exact duplicates are by definition identical, hence different ways of getting to the same goal for users: If a user arrives at either result, the user is equally satisfied. Navigational duplicates are similar, in that the user can easily reach one result from the other, but one result may be more relevant. However, if a user navigates from the wrong result to the right result, we can only infer that the user did this because he or she considered it easier to reach their goal by navigation than by returning to the search results and scanning further. This provides a low upper-bound on the utility of presenting a navigational duplicate.

Content duplication is different, as we have seen that some fraction of users do visit both results of pairs that were judged as content duplicate. This suggests that the utility of having both results is sometimes larger than the utility of just showing one. We hypothesize that the utility of the second result in some cases may even be higher once we know that the user considers the first result relevant, and that if one result is non-relevant then the other is more likely to be non-relevant as well.

Obtaining Gold Standard Data for Duplication

It is well recognized that inter-judge agreement when judging documents on multiple levels of relevance is often poor (e.g. [9]). Our results suggest that obtaining high inter-judge agreement for duplication can be similarly difficult. As we found high levels of inter-judge agreement for navigational duplicates, we focus on the correct procedure for obtaining content duplication judgments.

The key question for assessing content duplication in this work was *How similar is the utility of these two pages for the query?* Providing the judges with a calibration as to when two pages are sufficiently duplicate is clearly necessary. However, it is also difficult for a judge to trade off different aspects: For example, much as in the case of relevance judgments, it would be difficult for a judge to decide how much better does the design of one page need to be to offset a slightly lower level of detail in responding to a user’s information need. We see how to achieve such a calibration as an open problem.

Other Duplication Signals

As we have observed, click signals are clearly strong indicators of duplication within search results. However, our duplication model ignores other important signals of duplication, in particular the content of the two pages being compared. As noted earlier, this means that the behavioral information is particularly poor at identifying even exactly duplicated web results if the snippets are of very different quality. Further, our model does not detect duplication among results that answer the user’s query directly in the search result snippet, as may be the case if the user is looking for an address that is shown in multiple snippets.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a taxonomy of duplication that shows how web results can be redundant in terms of *information* content (or *utility*) to users in different ways – as exact duplicates, as navigational alternatives, and as equally useful content. We have seen that users behave differently when presented with the different types of duplicates, suggesting that the utility of the classes of duplicates is different, and that they should be treated differently when evaluating the quality of search results. Moreover, we saw that if two results are not duplicate, the effect of presentation bias is much smaller than on average, hence suggesting that if one could control for duplication, clicks would become a much stronger relevance signal.

To improve the prediction performance of duplicate classification, non-usage based features, observations of user behavior on results other than those being considered, as well as larger samples of labeled data are all likely to be useful. However, we have also seen that it can be difficult to judge the level to which two results are duplicate, arguing for refined judgment guidelines.

7. REFERENCES

- [1] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying Search Results. In *Proceedings of WSDM ‘09*, pages 5–14, 2009.
- [2] R. Agrawal, A. Halverson, K. Kenthapadi, N. Mishra, and P. Tsaparas. Generating labels from clicks. In *Proceedings of WSDM ‘09*, pages 172–181, 2009.

- [3] R. Baeza-Yates, C. Hurtado, M. Mendoza, and G. Dupret. Modeling user search behavior. In *The Third Latin American Web Conference*, pages 242–251, 2005.
- [4] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.
- [5] A. Z. Broder. Identifying and filtering near-duplicate documents. In *COM '00: Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*, pages 1–10. Springer-Verlag, 2000.
- [6] A. Z. Broder. A taxonomy of web search. *SIGIR Forum*, 26(2):3–10, 2002.
- [7] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. *Comput. Netw. ISDN Syst.*, 29(8-13):1157–1166, 1997.
- [8] J. Carbonell and J. Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of SIGIR '98*, pages 335–336, 1998.
- [9] B. Carterette, P. N. Bennett, D. M. Chickering, and S. T. Dumais. Here or there: Preference judgments for relevance. In *Proceedings of ECIR '08*, 2008.
- [10] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and clicks. In *Proceedings of NIPS '07*, 2007.
- [11] O. Chapelle, D. Metzler, Y. Zhang, and P. Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of CIKM '09*, pages 621–630, 2009.
- [12] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC '02: Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.
- [13] H. Chen and D. R. Karger. Less is More: Probabilistic Models for Retrieving Fewer Relevant Documents. In *Proceedings of SIGIR '06*, 2006.
- [14] A. Chowdhury, O. Frieder, D. Grossman, and M. C. McCabe. Collection statistics for fast duplicate document detection. *ACM Trans. Inf. Syst.*, 20(2):171–191, 2002.
- [15] C. L. Clarke, M. Kolla, and O. Vechtomova. An Effectiveness Measure for Ambiguous and Underspecified Queries. In *Proceedings of ICTIR '09*, pages 188–199, 2009.
- [16] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An Experimental Comparison of Click Position-Bias Models. In *Proceedings of WSDM '08*, 2008.
- [17] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of KDD '99*, 1999.
- [18] G. Dupret and B. Piwowarski. A user behavior model for average precision and its generalization to graded judgments. In *Proceedings of SIGIR '10*, pages 531–538, 2010.
- [19] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating Implicit Measures to Improve Web Search. *TOIS*, 23(2):147–168, April 2005.
- [20] S. Gollapudi and A. Sharma. An axiomatic approach to result diversification. In *Proceedings of WWW '09*, pages 381–390, 2009.
- [21] L. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *Proceedings of SIGIR '04*, pages 478–479, 2004.
- [22] H. Hajishirzi, W. tau Yih, and A. Kolcz. Adaptive Near-Duplicate Detection via Similarity Learning. In *Proceedings of SIGIR '10*, pages 10–17, 2010.
- [23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.
- [24] T. Joachims. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of KDD '02*, pages 133–142, 2002.
- [25] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems (TOIS)*, 25(2), April 2007.
- [26] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [27] M. Rabin. Fingerprinting by random polynomials. Report TR-1581. Technical report, Harvard University, 1981.
- [28] F. Radlinski, P. N. Bennett, B. Carterette, and T. Joachims. SIGIR Workshop Report: Redundancy, Diversity and Interdependent Document Relevance. *SIGIR Forum*, 43(2):46–52, 2009.
- [29] F. Radlinski and T. Joachims. Minimally Invasive Randomization for Collecting Unbiased Preferences from Clickthrough Logs. In *Proceedings of AAAI '06*, 2006.
- [30] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *Proceedings of KDD '07*, pages 570–579, 2007.
- [31] M. Theobald, J. Siddharth, and A. Paepcke. Spotsigs: robust and efficient near duplicate detection in large web collections. In *Proceedings of SIGIR '08*, pages 563–570, 2008.
- [32] C. van Rijsbergen. *Information Retrieval*. Butterworth, 2nd edition, 1979.
- [33] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *SIGIR*, pages 115–122, 2009.
- [34] Y. Yue and T. Joachims. Predicting Diverse Subsets using Structural SVMs. In *Proceedings of ICML '08*, pages 1224–1231, 2008.
- [35] C. Zhai, W. Cohen, and J. Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of SIGIR '03*, pages 10–17, 2003.
- [36] X. Zhu, A. B. Goldberg, J. Van Gael, and D. Andrzejewski. Improving Diversity in Ranking using Absorbing Random Walks. In *Proceedings of HLT/NAACL '07*, 2007.