Online Duplicate Document Detection: Signature Reliability in a Dynamic Retrieval Environment

Jack G. Conrad
Research & Development
Thomson Legal & Regulatory
St. Paul, MN 55123 USA
Jack.G.Conrad@Thomson.com

Xi S. Guo
Rapid Appl. Development
Thomson Legal & Regulatory
St. Paul, MN 55123 USA
Xi.Guo@Thomson.com

Cindy P. Schriber
Business & Information News
Thomson–West
St. Paul, MN 55123 USA
Cindy.Schriber@Thomson.com

ABSTRACT

As online document collections continue to expand, both on the Web and in proprietary environments, the need for duplicate detection becomes more critical. Few users wish to retrieve search results consisting of sets of duplicate documents, whether identical duplicates or close matches. Our goal in this work is to investigate the phenomenon and determine one or more approaches that minimize its impact on search results. Recent work has focused on using some form of signature to characterize a document in order to reduce the complexity of document comparisons. A representative technique constructs a 'fingerprint' of the rarest or richest features in a document using collection statistics as criteria for feature selection. One of the challenges of this approach, however, arises from the fact that in production environments, collections of documents are always changing, with new documents, or new versions of documents, arriving frequently, and other documents periodically removed. When an enterprise proceeds to freeze a training collection in order to stabilize the underlying repository of such features and its associated collection statistics, issues of coverage and completeness arise. We show that even with very large training collections possessing extremely high feature correlations before and after updates, underlying fingerprints remain sensitive to subtle changes. We explore alternative solutions that benefit from the development of massive meta-collections made up of sizable components from multiple domains. This technique appears to offer a practical foundation for fingerprint stability. We also consider mechanisms for updating training collections while mitigating signature instability.

Our research is divided into three parts. We begin with a study of the distribution of duplicate types in two broad-ranging news collections consisting of approximately 50 million documents. We then examine the utility of document signatures in addressing identical or nearly identical duplicate documents and their sensitivity to collection updates. Finally, we investigate a flexible method of characterizing and comparing documents in order to permit the identification of non-identical duplicates. This method has produced promising results following an extensive evaluation using a production-based test collection created by domain experts.

Keywords

data management, duplicate document detection, doc signatures

CIKM'03 November 3–8, 2003, New Orleans, Louisiana, USA

General Terms

Algorithms, Measurement, Performance, Design

Categories and Subject Descriptors

H.2.4 [**Information Systems**]: Database Management—*Systems–Textual Databases*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Selection Process*; H.3.2 [**Information Storage and Retrieval**]: Information Storage—*File Organization*; E.2 [**Data**]: Data Storage Representations

1. INTRODUCTION

Both on the World Wide Web and in proprietary data environments, it is currently possible to have tens of millions of documents indexed as part of the same collection.[1] News databases are particularly challenging in that, thanks to wire services that are released by different newspapers, they may contain dozens of copies of the same article. A number of other domains also produce collections of comparable size where the content of one document may be completely duplicated in another [11]. These domains include business & finance, science & technology, medicine & bio-informatics and intellectual property [17]. At Thomson Legal & Regulatory (TLR), massive data environments like Westlaw and Dialog possess on the order of 25 terabytes of data. In such an environment, the identification if not suppression of duplicate documents is critical to a practical and robust data delivery platform.

An important issue that forms the foundation for all deduplication work is the stability of the feature set selected to characterize a given document or to generate its "signature." We have found that many of the works that treat the subject have not adequately addressed the fact that contemporary online collections are extremely dynamic and frequently experience expansion (document additions via loads), contraction (document removals via deletions), and modifications (documents experiencing revisions and new metadata creation via reloads). Given this changing environment, reliance upon stable features is critical.

Our goal in this work is to determine the extent and the types of duplication existing in large textual collections. We also wish to devise one or more approaches that minimize its deleterious impact on search results in an operational environment. Recent work has focused on issues of computational efficiency and duplicate document detection (and, by extension, "deduping") effectiveness while relying on "collection statistics" to consistently recognize document replicas in full-text collections [16, 5]. Such research has tended to consist of academic studies that have worked with test collections constructed from the Web or by TREC [22].

[1] In this paper, we will use "collection" to refer to a database of textual documents.

They have tended to understate pivotal issues involving the constantly changing nature of the underlying textual collections. Some have suggested relying upon a relatively static training collection [8], but over time such dependence upon a collection that otherwise dynamically fluctuates can introduce uncertainties involving coverage and completeness. We show how critical the stability of an underlying "training" collection can be and the importance of a large comprehensive or multi-domain "meta-collection."

This work makes three distinct contributions. It is the first report to:

1. characterize the distribution duplicate documents in a large production environment consisting of tens of millions of documents representing over $\frac{1}{2}$ TB of data;
2. investigate the effect of collection expansion on collection statistics and representative document signatures;
3. create a deduping test collection by harnessing:
   (a) real user queries;
   (b) a massive collection from an operational setting;
   (c) professional assessors possessing substantial knowledge of the domain and its clients.

In addition, this work expands the discussion of online (real time) deduping addressed by Cooper, et al. [6]. Recent work has often been Web-based (focusing on issues such as URL instability), syntax-based rather than lexical-based, and offline-based (e.g., examining large numbers of permutations before constructing a feature set) and thus substantially different than our current efforts that target a production environment.

The remainder of this paper is organized as follows: Section 2 reviews related work in duplicate document detection. In Section 3, we present the methodology and results of our duplicate document study, conducted in the news domain. Section 4 outlines data from our production environment used to represent three distinct topical domains. Section 5 addresses collection statistics and how they can be exploited to select document features. In Section 6, we examine identical duplicate document detection along with the role of underlying idf-based features. Section 7 discusses the key deduping algorithm for non-identical duplicates and the preliminary trials to evaluate it. We describe presentation issues in Section 8 and share our conclusions in Section 9.

2. PREVIOUS WORK

2.1 Early Techniques

Manber reported on a technique and a system, *sif*, used to find "similar files" in a large file system. The goal of this work was to be able to identify files that come from the same source or contain parts that come from the same source [10]. The technique distinguishes itself from approaches relying on checksums and single fingerprints. Whereas these can be effective for exact equality testing, Manber wishes to detect similarities even if the mutual coverage is as low as 25%. The basic idea is to gather fingerprints from several parts of a file. Since a priori knowledge of the parts fingerprinted is unworkable, the challenge arises from the synchronizing of fingerprint sets between two separate documents. He computes fingerprints of nearly all possible substrings of a certain length, then chooses a subset of them based on their hash-like values. He claims that since two equal substrings will generate the same fingerprints, regardless of their position in the text, this method provides the desired synchronization, given that overlapping fingerprints are not allowed.

In a related work, Heintze focuses on effective document fingerprinting that scales to large environments and that can identify similarities between documents that have as little as 5% or less in common [9]. The approach is based on selecting a set of sub-sequences of *characters* from a document and generating a fingerprint based on the hash values of these sub-sequences. In full fingerprinting, similarity between two documents is measured by counting the number of common sub-sequences in the complete fingerprints. But the full process can become impractical because of the sizes of the fingerprints generated. To reduce the size of the fingerprints, a subset of the substrings is selected. He also compared the match percentage of selective fingerprints against full fingerprints to demonstrate that selective fingerprints normally perform within a factor of two of complete fingerprints.

Brin, Davis and García-Molina have proposed a system for registering documents and detecting either complete or partial copies [1]. They also address issues of performance, storage capacity, and accuracy. Although their initial focus is on monitoring the danger of illegal copying—in a prototype system called COPS—they also discuss implementation issues, experimental results, and useful parameter settings that may be instructive for other deduping applications. The authors propose reliance upon a *chunking strategy* and a set of *ordinary operational tests* (OOTs) that can be implemented efficiently, for instance, reflecting subset, overlap, and plagiarism measurements. Like the two previous works, hashing is used to assist in efficiently detecting matching chunks.

2.2 Web-based Approaches

Broder, et al. author a seminal work on clustering Web-based documents that are *syntactically* similar in order to address a number of issues involving document *resemblance* and *containment* (multiple hosts, versioning, different formats, dead links, slow access, subsumption, etc) [3]. The authors' technique has come to be known as *shingling* and is applied by representing a document as a series of simple numeric encodings representing an n-term window—or shingle—that is passed over a document to produce all possible shingles (e.g., for n=10). They then use filtering techniques to retain every mth shingle (e.g., for m=25), and, if necessary, select a subset of what remains by choosing the lowest s encoded shingles (e.g., for s=400). This process produces a document 'sketch.' To further reduce the computational complexity involved in processing large collections, the authors present a super-shingle technique that creates meta-sketches or sketches of sketches. Documents that have matching super-shingles thus have a sequence of sketches in common. Pairs of documents that have a high shingle match coefficient (resemblance) are asserted to be close duplicates while pairs that have lower match coefficients are similar. The authors used a resemblance threshold of 50% in their tests. As subsequent comparative tests have shown, the more distilled or abstracted the representations, the greater the chance for error [5, 6].

Shivakumar and García-Molina describe factors in identifying nearly identical documents on the Web for the benefit of Web crawlers and Web archivers [16]. They consequently concentrate on computing pairwise document over-

lap among pages commonly found on the Web. Their workshop draft specifies Web-based applications for the identification of near replicas: (1) *more efficient web-crawling*, focusing on speed and richer subsets rather than time-consumptive comprehensiveness; (2) *improved results ranking* (or reranking), inspecting the environments from which Web documents originate; and (3) *archiving Web documents*, enabling greater compression of shorter pages that replicate more complete doc sets. The authors reveal that there is a much greater incidence of (a) server aliasing; (b) URL aliasing; and (c) replication of popular documents such as FAQs and manuals than initially believed. Some of the resource-saving concepts they propose have been harnessed by a number of Web search engines including Google [2].

In one of the most comprehensive works to date, Chowdhury, et al. refine their collection statistic, idf-based deduping algorithm for efficiency and effectiveness while comparing its performance to other state-of-the-art techniques such as shingling and super-shingling [5]. They demonstrate that their approach, called I-Match, scales in terms of number of documents and works well for documents of diverse sizes. They evaluate I-Match on both Web-based and non-Web-based test collections and claim that in addition to improving accuracy over competing approaches like shingling, it executes in one-fifth the time. The authors briefly describe how the collection statistics for the algorithm can come from training collections in rapidly changing data environments. In Section 6, we further explore the importance of collection statistics and the role of underlying idf stability in pursuing a domain-optimized version of such idf-based features.

The recent Web-related research of Park, et al. relies heavily on the notion of lexical signatures, consisting of roughly five key identifying words in document, based either on their low *df* or high tf properties [13]. What distinguishes this work is that its eight signature variations are designed and evaluated for their ability either to retrieve the associated document in question in the top ranks of a search result (*unique identification*) or to retrieve alternative relevant documents should the document be "lost" (e.g., due to a broken link) (*relevance properties*). They determine that hybrid signatures consisting of only a couple of low *df* terms plus several high tf or high $tf \cdot idf$ terms produce the most effective unique and relevant properties for Web page signatures.

Cooper, et al., discuss methods for finding identical as well as similar documents returned from Web-based searches [6]. The techniques are based upon the creation of a digital signature composed of the sum of the hash codes of the "salient" terms found in a document. The document signatures are intended to provide a short-hand means of representing the top terms in documents to facilitate fast comparisons. Their tests generally rely upon a single query and may warrant more comprehensive evaluation. The authors describe their approach as the "logical extreme of supershingl[ing]," yet, characterizing a document by summing its Java hash codes for hundreds or more terms may raise questions about the principled, dependable nature of the technique.

In some environments, maintaining a vast registry of duplicate clusters is practically unsustainable due to the frequency of updates, computational expense, and the distribution of participating databases. As a result, researchers have begun to seriously address real-time deduplication, with Cooper, et al. performing $O(n^2)$ comparisons, albeit reduced by inter-document length restrictions [6]. An important related issue involves the fact that real-time deduping generally consists of two phases, the first to generate document signatures (offline), and the second to compare them (online). Approaches like shingling are more computationally expensive to perform [3], yet are outperformed by rich term-based techniques like I-Match [5] and, for short or similar documents, by hash encodings [6]. In Sections 6 and 7, we consider optimizations of a related idf-based feature set approach in order to "productionize" online deduping.

3. DUPLICATION STUDY

In order to investigate the types and distributions of duplicate documents in large collections, most notably in news collections, we performed an experiment in which real user queries were run against large news databases.

3.1 Methodology

We randomly selected 25 real user natural language queries submitted to the ALLNEWS collection and 25 to the ALLNEWSPLUS collection. The ALLNEWSPLUS collection contains all of the documents that ALLNEWS contains, plus additional up-to-date newswire articles. The latest versions of newswire articles are also normally maintained in their own WIRES database for up to 90 days. Simple collection statistics for these three data sets are shown in Table 1.[2] We ran the queries using the WIN natural language search engine, a cousin to UMass's InQuery [20, 18]. The top 20 results for each set of 25 queries were examined for their duplicate types. In all, 1,000 documents were examined in response to the 50 queries [$(25 + 25) \times 20$].

The categories used for duplicate types included the following:

1. exact duplicates (same title not required);
2. excerpt: one document takes first section (e.g., n-hundred words) from another (longer) article;
3. elaboration: one document adds one or more paragraphs to another (shorter) article;
4. insertion(s): one document is the same, but adds one or more sentences or phrases to the paragraphs of another article;
5. focus: one document is a rewrite, using visibly different vocabulary/descriptions/content than that of the other article, but about an identical or very similar topic;
6. revisions (etc): other, not covered in any of the categories above.

Database	Documents	Tokens
ALLNEWS	45,191,471	$O(10^7)$
ALLNEWSPLUS	55,167,244	$O(10^7)$
NEWSWIRES	781,695	$O(10^6)$

Table 1: Collection Statistics for ALLNEWS, ALLNEWSPLUS, and [NEWS]WIRES Databases

[2]ALLNEWSPLUS retains more historical newswire articles than the WIRES database does, which explains why WIRES does not equal the difference in size between the ALLNEWS and ALLNEWSPLUS databases.

Collection:	ALLNEWS		ALLNEWSPLUS	
Size (Doc Count)	45.2 Million		55.2 Million	
Category	Sets(%)	Documents (%)	Sets (%)	Documents (%)
1.	38.1%	36.1%	47.6%	54.9%
2.	7.3%	3.6%	11.1%	9.7%
3.	7.3%	5.9%	8.0%	6.9%
4.	7.3%	5.9%	0.0%	0.0%
5.	40.0%	48.5%	33.3%	28.5%
6.	0.0%	0.0%	0.0%	0.0%
Total:	100.0%	100.0%	100.0%	100.0%

Table 2: Duplicate Document Distribution

No.	Database	Domain	15 August 2002	15 September 2002 (% Change From Aug.)	15 October 2002 (% Change from Sept.)
1.	DJAP1	News/Finance	258,812	277,055 (7.1%)	291,178 (5.1%)
2.	DJAP2	News/Finance	216,250	241,735 (11.8%)	256,368 (6.1%)
3.	DJAPCOR	News/Finance	342,698	375,315 (9.5%)	388,609 (3.5%)
4.	DJMISC11	Science/Tech	712,028	833,589 (17.1%)	943,023 (13.1%)
5.	USFT58	Patents	108,169	108,170 (0.0%)	108,170 (0.0%)
6.	USFT60	Patents	124,569	124,569 (0.0%)	124,569 (0.0%)
7.	USFT61K	Patents	184,184	184,169 (-0.1%)	184,169 (0.0%)
8.	USFT62K	Patents	113,912	120,998 (16.0%)	132,177 (9.2%)
	Total:	Multiple Domains	2,060,622	2,265,600 (9.95%)	2,428,263 (7.18%)

Table 3: Document Distribution in Monthly Samples

3.2 Duplicate Findings

The total number of duplicate document sets returned for the 25 ALLNEWS queries was 53, representing 143 documents; the total number of duplicate document sets returned for the 25 ALLNEWSPLUS queries was 59, representing 145 documents. Note that a duplicate "set" can contain more than simply a pair of duplicate documents. The total number of duplicate documents accounts for 28.6% of all the documents returned on ALLNEWS and 29.0% of all the documents returned on ALLNEWSPLUS. Only two of the 25 ALLNEWS queries and three of the 25 ALLNEWSPLUS queries were free of duplicates.

The results, shown in Table 2, demonstrate that most of the duplicate documents are found in Categories 1 or 5 for both the ALLNEWS and ALLNEWSPLUS databases. This finding indicates that a large majority of the duplicates are either exact duplicates (category 1) or are a rewrite, that is, documents using a somewhat different vocabulary (category 5). The number of duplicate sets in Categories 1 and 5 comprise approximately 80% of the total number of duplicate sets for ALLNEWS queries and 79% of those for the ALLNEWSPLUS queries. We address these two important categories in Sections 6 and 7.

For the queries in Category 1, we note that only 30% of documents in the ALLNEWS duplicate sets have the same title whereas approximately 33% of the documents in the ALLNEWSPLUS duplicate sets have the same title.

4. DATA

To monitor the amount of growth and change that large collections in production environments experience over time, we assembled eight large physical databases with documents nearly evenly distributed over the domains of news, science, and intellectual property. Cumulatively these collections represent over 2 million documents. We tracked these collections and their growth over the course of the third quarter of 2002. The first three databases, as shown in Table 3, come from Dow Jones News. The fourth comes from a related set of Dow Jones Science & Technology documents. The remaining four databases come from the U.S. Patent Office. As can be seen from the table, the amount of growth shown in the months in question varies between roughly 7% and 10% (last line in Table 3). We revisit the impact such growth can have in the next section on Collection Statistics.

5. COLLECTION STATISTICS & FEATURE SETS

Collection statistics that are based on a term's inverse document frequency (idf—Section 5.1) are actually relying upon a presumed estimate of a word's rareness across a given collection and therefore its value to discriminate a document from others. A deduping technique might thus appear deficient if it did not incorporate relatively frequent updates to the underlying documents and the terms they contain. In so doing, however, the technique would permit its document signatures to shift, and that could have harmful consequences when they are used to compare two documents. By contrast, one could mitigate this situation by reducing the frequency of idf table updates while increasing the size and diversity of databases comprising this source of collection statistics. Terms that are compromised by the reduction in updates would tend to be concepts such as named entities that often rise and fall in prevalence over time (e.g., organizations, scientific procedures, chemical compounds, drugs, etc). Some terms normally satisfying inclusion thresholds may be temporarily omitted, for instance, certain named entities, yet the techniques we propose are not so sensitive that they require the absolute highest n idf words; rather, in order to be effective, they require only n high idf words. Moreover, in each of the experiments reported on here, a term is not considered for table membership unless it ap-

pears across the collections at least five times. Such a restriction effectively filters out many tokens attributable to misspellings, typographical errors, and others.

Chowdhury, et al. have alluded to the significance of this problem by describing how a "training set" could assist those tackling the problem [5]. They imply that by relying on such a training set, developers can potentially reduce the dimensionality of the problem. They further posit that idf values "change slightly" as collections grow; hence a training set represents an "acceptable solution" [8]. An alternative, they point out, would be a two pass approach, with the first pass determining the idf weights of the terms and the second pass applying the duplicate document detection algorithm.

In a production environment like ours, where the number of news articles now surpasses 50 million documents, a two pass approach would clearly be unsupportable. At the same time, these collections are constantly in a state of flux. One reason for this dynamic environment is that today's news cycles are no longer twice daily (for the traditional a.m. and p.m. editions), but rather, as frequently as several times an hour. Clearly a large, stable, domain-representative collection or meta-collection would be needed to provide a source from which to incorporate idf-related collection statistics.

Following the involvement of our production staff, we resolved to construct a large meta-collection in order to supply the needed collection statistics (idf values). In an environment that increasingly permits users to perform heterogeneous or federated searches across multiple databases (including multiple domains), a design that relied upon multiple sets of domain-dependent collection and term statistics would ultimately create problems. Take as an example when documents from multiple domains have to be compared—let alone merged into a single result set—following a multiple database search. Our proposed meta-collection would thus consist of O(10) large physical databases, databases that come from domains representative of our diverse non-legal content (news, business, finance, science, technology, biomedical, patents, trademark materials and other related filings). All totaled, the resultant meta-collection consists of over 3 million documents from the aforementioned domains. From each of the component data sets, we downloaded unstemmed term listings and corresponding document frequencies while filtering out numeric and alpha-numeric terms, along with terms with special characters, e.g., { . ' - & + / }. We then merged the term lists while paying close attention to their cross-collection document counts.

5.1 Normalized idf

Using the resultant meta-collection with alpha (non-numeric) terms of at least length 3, we sorted terms by their normalized idf values from (1).

$$idf_{norm} = \frac{log[\frac{(N+0.5)}{n_{docs}}]}{log(N+1.0)} \quad (1)$$

where n_{docs} = the number of documents containing the specific term and N = the total number of documents in the collection [where $N \approx 3.1$ million documents (from Dec. 2002)].

We rely on meta-collection normalized idf values, that in principle can range between 0 and 1 (but in reality range between 0.001 and 0.890), in order to permit us to monitor changes in term rank once updates to the idf table had been performed. In short, collection-normalized idfs allow us to make a reliable comparison between terms, their order, and how their idf or "rareness" in the collection(s) may fluctuate. In addition, we can track differences in term idf and relative rank in a collection when terms in a combined-domain (such as in our meta-collection) are compared with the same terms in specialized domain-specific tables (like news). Normalizing idf scores based on the number of documents in a collection permits one to make reasonably equitable comparisons between relative term rank in one collection and in another when these scores are available.

5.2 Spearman Rank Correlation Coefficient

The Spearman Rank Correlation Coefficient shown in (2) is an effective means by which to monitor the degree of variation in term rank in a list of tokens [15].

$$R = \frac{1 - \frac{6}{n^3-n}(\sum d_i^2 + \frac{1}{12}\sum(f_k^3 - f_k) + \frac{1}{12}\sum(g_m^3 - g_m))}{\sqrt{(1 - \frac{\sum(f_k^3-f_k)}{n^3-n})}\sqrt{(1 - \frac{\sum(g_m^3-g_m)}{n^3-n})}} \quad (2)$$

where d_i is the rank difference of common term i, n is the number of terms, f_k is the number of ties in the kth group of ties in list A's ordered term distribution, and g_m is the number of ties in the mth group ties in list B's ordered term distribution. When $R = 1.0$, the two orderings are identical; when $R = -1.0$, the two are in reverse order; and when $R = 0.0$, the two are uncorrelated.

Simpler versions of the Spearman coefficient have been used [12]. These versions tend to assume a complete ordering, which disregards terms that have the same rank or idf value (i.e., ties). Yet, collections contain many terms with the same document frequency. These terms would possess the same idf value and thus represent ties in collection ranking. As Callan has pointed out, coefficients that ignore the effects of ties can give misleading results (as was the case with their initial database sampling experiments) [4].

We use the Spearman rank correlation coefficient in Section 6.2 to examine the similarities between two collections' idf tables constructed from the same databases over a series of subsequent months. We compare the intersections of a collection's term list—i.e., its snapshot—over the course of three months. Since a collection generally experiences growth in the number of documents it possesses and therefore in its term lists, any similarities that are found may be overstated since the new terms would not be permitted to participate in the comparisons (since the intersection operation guarantees their exclusion); only those terms that were already present in the collection but experienced change could be compared.

6. IDENTICAL DUPLICATES

6.1 Algorithm Overview

In order to alleviate the exact duplicate document problem identified in our study, we wanted to determine the stability of document fingerprints constructed using terms found in a collection-based, idf-ranked table. The richest discriminating terms to be used in such fingerprints could be identified through the use of such an idf table that represents a snapshot of the underlying collection at a given point in time. We have determined empirically that using a fingerprint of as few as six top idf terms from a document, along

Time Period:	August-September		September-October		August-October	
Term Set	No. of Terms in Intersection	Spearman Coefficient	No. of Terms in Intersection	Spearman Coefficient	No. of Terms in Intersection	Spearman Coefficient
Alpha-only	780,736	0.9903	793,984	0.9881	769,535	0.9808
Alpha-numeric+	1,047,252	0.9888	1,064,878	0.9865	1,030,163	0.9783

Table 4: Comparison of Term Rank Correlations for Months of August/Sept/Oct (2002)

with their offsets, are sufficient to create a unique signature for the document. This approach is based on the observation that the probability of the same sequence of six rare terms coincidentally appearing in the same relative word positions with the same document offsets in two separate and non-duplicate documents—and in the same search result set—is an extremely remote prospect.[3]

The essence of the duplicate document detection algorithm for identical documents is as follows. It focuses on the core content of a document and ignores metadata and its associated tags.

1. During the load process, a complete document *signature* is produced and stored for each document, in the form of a metadata key [doc length (scalar) + fingerprint (vector)];
   (a) The document length [in tokens, excluding title(s), author(s), and other header or metadata information] is stored as part of the signature;[4]
   (b) The document fingerprint consists of the top six unique idf terms (excluding header or metadata tokens), along with their positions relative to each other, e.g., (1) prevarication[79], (2) hostage[0], (3) conspicuous[21], (4) intransigence[123], (5) brutality[163], (6) theater[13] (ranked by idf values).
      i. Note that the terms under consideration would exclude title and other headings (since these can clearly vary in different articles, editions, etc.);
      ii. Note also that terms with an unusually high idf, e.g., idf > 0.8, would not be considered among the top six candidates because these tend to be aberrant forms (i.e., typo and misspellings).
2. The construction of the fingerprint is completed when it is hashed into a key, for example, of length 20 bytes (160 bits) using a standard hashing algorithm such as NIST's SHA1 standard [21].

For relatively short news documents, averaging roughly 800 words in length, this concise fingerprint approach is a suitably economic means of recording a document's most discriminating and characteristic features. For longer docs, such a fingerprint may be more fragile for given headings and other presentation-related details that can vary by provider.

[3]We learned this through our work with variable length text string comparisons involving quotation verification for a citator service [7]. In a related work, Park, et al. rely on 5 terms to store content that functions as both uniqueness and similarity (to other documents) markers [13].

[4]Fingerprint comparisons can be avoided when the lengths of two documents differ by more than a designated (small) percentage.

Practical variations on the fingerprinting process outlined above have arisen because of what Phelps and Wilensky call the *uniqueness-robustness tradeoff* [14]. That is, when a signature is generated using high idf terms as a selection criteria, it will work effectively with *exact* duplicate documents, yet its robustness in the face of even small amendments is weakened. To help mitigate this situation, an alternative to step 1(b) above consists of "binning" the offsets of the selected terms into bins of size 10, 25, or n. So the actual offset of the term in question would be rounded up to the limit of the applicable bin. The motivation behind such binning is that if an article experienced, for instance, short substitutions or the insertion of a small number of stock tickers beside company names, the general functioning of the algorithm would remain intact. More radical modifications would obviously exceed the robustness of the binning. However, modifications would also arguably exceed the definitional boundaries of exact duplicate.

6.2 Evaluation of Signature Stability

After having previously examined the viability of document signatures as a means of representing documents and of reducing the complexity of textual comparisons when executing deduping strategies online [7], we wanted to determine the importance of collection stability over time in the light of frequent collection updates. As an initial step, we examine the collections' idf-based term rank correlations during a three month period, using the Spearman coefficient (Section 5.2). We compare both alpha-only and alpha-numeric tokens (the latter also including a few special characters, namely, { . ' - & + / }). Table 4 shows that the degree of correlation is consistent and extremely close for these paired token sets during the three month period. These comparisons do not consider the new terms added to the table as the collections are expanded. Given that the documents used in these comparisons were present in their respective collections at the beginning, previously unseen terms that were being added to the table later in the third-quarter have no bearing on the signature terms of these particular documents. Only the changing ranks of already present terms can be responsible for the signature mismatches. The Spearman rank correlation coefficient can thus serve as an effective means by which to measure stability. *The high degree of correlation shown in Table 4 is a preliminary indication that one can expect there to be a similarly high degree of stability among document signatures, even as the underlying collections experience growth through the updating process.*

To further test this degree of collection stability, we randomly selected 1,000 documents from each of our three domains of interest (news, science, intellectual property). We then produced a series of signatures—ranging from single highest idf term to top-30 idf terms (represented in the table)—for each document. These signatures are based on a table of idfs generated from the collections shown in Table 3. No document metadata was permitted to participate

in a signature. Thus, each document possesses three separate signature sets, one set resulting from each of the three idf tables (August, September, October). Shown below are figures depicting the extent to which collection growth can impact signature stability. The first figure illustrates signature stability for all 3,000 documents treated as one collection (Figure 1). If each of the 3,000 document signatures of length 12, for instance, that were created using an idf table from September matched those signatures of length 12 created using an idf table from October, then the corresponding data point (×) would be located at the top of figure 1, at (x=12, y=3000). The next three figures break down performance by domain—News (Figure 2), Science (Figure 3), and Intellectual Property (Figure 4).

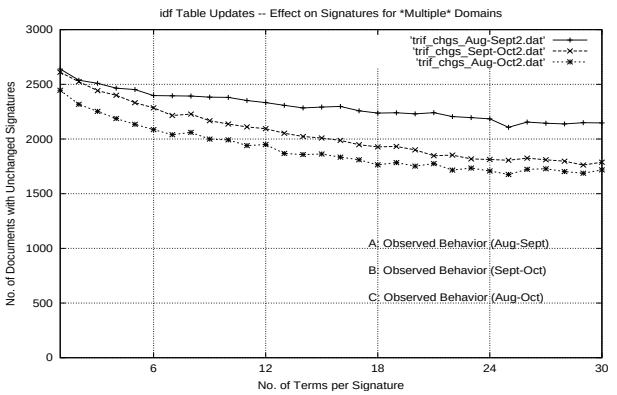

Figure 1. *Signature Changes from idf Table Update — Multiple Domains (News, Science, Patents)—*

Figures 1 through 4 reveal several surprising behavior patterns. Based on our preliminary Spearman results (and highly correlated underlying term distributions), we had anticipated very little difference in same-length signatures, especially for shorter length signatures. Yet as Figure 1 illustrates for the composite set of 3,000 documents, as many as one in six signatures changes for short signatures (e.g., with a length of 1 to 6 terms) and this fraction increases to approximately one in three (Aug-Sept) and five in twelve (for Sept-Oct and Aug-Oct) (e.g., for those with a length of 30 terms). As Figure 1 suggests, beyond 30 terms, the curves continue to level off.

Figures 2 through 4 confirm these results for their own domain-specific sets of 1,000 documents: even for signatures of length 1 token, roughly one signature in six changes with the changing idf table. This degree of instability increases with increased signature size, especially for the September-October and August-October pairwise comparisons for the shorter news and science documents.[5] Despite the reassuringly high initial Spearman rank correlation coefficients encountered, we find that high-idf term-based signatures appear to be quite sensitive to even modest levels of monthly updating in the 7%-10% range. The significant change in relative position of the Aug-Sept and Sept-Oct curves for the Intellectual Property material is likely attributable to the reduction of growth (and thereby change) in these collections during the Sept-Oct period.

[5]The only filtering that was performed on our data sets was the exclusion of terms with numerals or special characters, { . ' - & + / }, and tokens with less than 3 characters. Results from an earlier experiment that also allowed alpha-numerics and decimal points produced curves just a few percentage points lower than these.

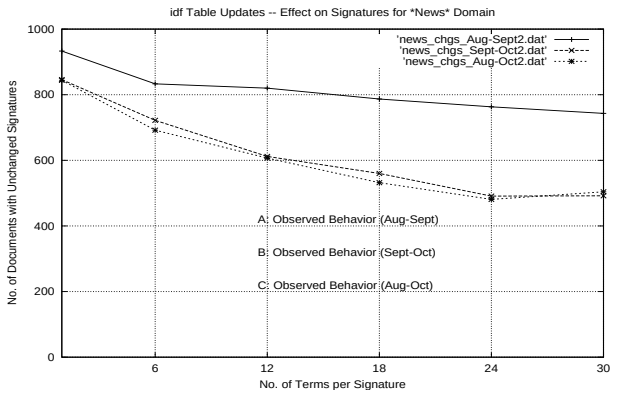

Figure 2. *Signature Changes from idf Table Update — News Domain —*

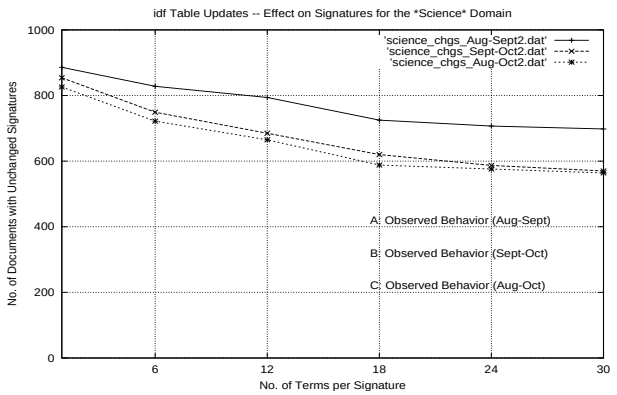

Figure 3. *Signature Changes from idf Table Update — Science Domain —*

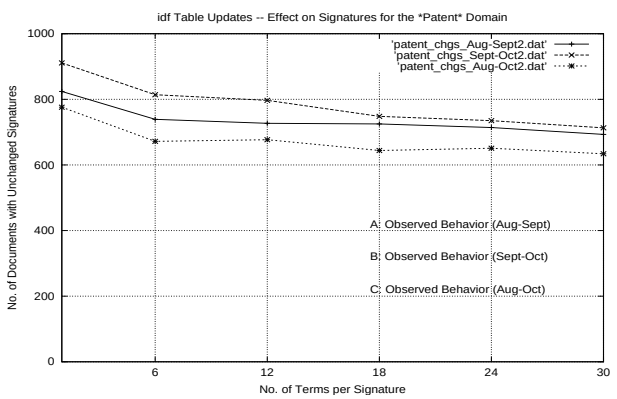

Figure 4. *Signature Changes from idf Table Update — Intellectual Property Domain —*

These findings suggest that frequent or regular updates to an underlying table representing collection statistics may introduce undesirable instability to a document management system that is based on signatures consisting of term fingerprints. Conversely, the findings support an alternative reliance upon a very large, less dynamic, multiple domain-based table or training collection.

7. NON-IDENTICAL DUPLICATES

Given our ability to identify and flag *identical* documents in an online context, we subsequently conjectured that there may exist an intuitively reasonable means by which to compare topically similar documents that are *not identical* in terms of their language, structure, or length.

7.1 Algorithm Overview

In order to determine our ability to identify and characterize such *non-identical* duplicate documents, we began investigating reliance upon an expanded multi-dimensional feature set or signature. These feature sets include:

- time component (pub_date);
- magnitude component (doc_length);
- core content component (term_vector).

The role of the first two is to reduce the need for more costly term comparisons. In addition to a publication date (e.g., days or weeks since Jan. 1, 1980) and document length (excluding metadata), a document's term_vector (or fingerprint) is represented by its *top n idf words*, where n falls somewhere between 30 and 60 words. We determined empirically that 60 words would serve as an optimal default vector size because (a) it offers substantially finer granularity to the process, and (b) it does not exceed the lower length limits of the vast majority of our shortest news documents. In a number of instances, there are not always many more than 60 terms to select for discrimination purposes. In rare instances when a document possesses less than 60 eligible terms a special procedure, discussed below, is applied that still permits the equitable comparison of vectors. Unlike in the previous approach that targets identical duplicates, no offset or rank information is retained in the vector—only the set of characteristic, discriminating words.

Aside from core document content, metadata indicating region, news sector, market sector, industry, product type, etc. is not used. We have determined that such categories tend to increase the number of false positives, since related but dissimilar documents may possess similar metadata classification terms.

7.2 Performance Evaluation

To test our hypothesis, we selected a total of 100 real user queries from logs in our production domain responsible for the largest percentage of duplicate documents: news, including financial. The queries were randomly selected with the exception that we required a results list of at least 20 documents. A sample of these queries is shown in Table 5. Each query was run using the Westlaw system which provides both Boolean and natural language search capability, depending on the preference of the user [19]. After running these queries against their corresponding ALLNEWSPLUS database, we assembled the top twenty documents returned. We had each set of twenty documents reviewed by two client research advisors (who also happen to have law degrees) in order to identify their duplicate sets in a manner similar to the examination reported in Section 3. This process produced standard training and test sets against which our computational approach would be compared.[6]

[6] "Training" is not used here in the Machine Learning sense; rather, it signifies an initial round in which we were permitted to establish the algorithm's optimal parameter settings.

As the queries below suggest, a sizeable majority of our News database subscribers prefer to use Boolean rather than

Type	**News & Finance** [DB: *ALLNEWSPLUS*] (55M docs)
Bool	"medical malpractice" & "public citizen" (1/25/03+)
Bool	"natural gas" & storage & "all time low"
Bool	"Eastern Europ*" & support & US & Iraq (02/01/03+)
Bool	John /3 Ashcroft (1/25/03+)
NL	pay reform for federal law enforcement officers
NL	"consumer fraud" "deceptive behavior" "unfair practice"

Table 5: Sample Queries for Non-Identical Dups

natural language queries, often due to the perceived control it offers users. The default results ranking for Boolean queries on Westnews is by date (i.e., reverse chronological order). This characteristic permits an initial binning-by-date that can be exploited later when avoiding costly term_vector comparisons. A similar economy can be established by binning-by-doc_length within each date-based bin.

Duplicate Document Detection	**Training Set**	**Test Set**
Total Queries	50	50
with Dup Sets	41	44
without Dup Sets	9	6

Table 6: Distribution of Duplicates Across Queries

In this trial, we applied a definition of non-exact duplicate that was generated by a customer work group consisting of 25 research librarians. These were individuals who typically service the information needs of a wide variety of users in their workplaces. The resulting definition states that two documents are duplicates if they retain much of the same language and are at least 80% similar.[7] To formally review the duplication status of the result sets, we assembled two teams of two client research advisors. The 100 queries were divided into two sets of 50, the first set to be used to train the system and the second set to test it. The process by which the query results were judged was scheduled over 4 weeks time.[8] During week 1, results from the training queries were assessed for their duplication status. Each team reviewed the results from 25 queries, 5 queries per team per day. Although members of the same team reviewed the same results, they did so independently. Week 2 served as an arbitration week. When members of the same team disagreed about a duplicate set, a member of the other team would serve as an arbitrator or tie-breaker. Weeks 3 and 4 were conducted in the same manner using the remaining 50 queries, thereby producing the test set. Table 6 presents the number of queries that yielded duplicate sets in the trial, while Table 7 shows the distribution of duplicate sets by size. The queries for the test set produced slightly fewer duplicate sets but also several larger duplicate sets consisting of 4, 5,

[7] (a) I.e., 80% of the words in one document are contained in the other (in terms of overall *terminology* rather than individual term *frequency*).
(b) For documents that do not meet a working threshold for similarity or *resemblance*, Broder, et al. monitors a second looser relationship described as *containment* [3].

[8] During the preceding week, we conducted a practice session for the subjects using a preliminary set of queries. Each participant was asked to assess the same results. Once this was completed, a feedback session was held to discuss nuances associated with the duplicate identification task and to propose and agree upon additional guidelines and heuristics as needed.

or 6 documents. In total, 2,000 documents were examined. The mean length of the news documents returned during the two rounds was 796 terms.

Duplicate Set Size	Training Set (Frequency)	Test Set (Frequency)
Pairs	68	64
Triplets	12	12
Quadruplets	8	2
Quintuplets	0	3
Sextuplets	0	1
Total	88	82

Table 7: Distribution of Total Resulting Dup Sets

Table 8 illustrates the performance of the algorithm relative to the standard established by the client research advisors, in terms of agreement (correct identification), false negatives (misses), and false positives (over-generation). The same idf table described in Section 5 is used here. A number of modifications were made to the algorithm during the training phase. Most notable is how it treats short documents (with fewer than 60 terms). A variety of options exist, including (i) comparing vectors of unequal length, (ii) comparing only the rarest n terms, where n is the size of the shortest doc's vector, and (iii) padding the short doc's term vector with entries not found in the table (in a manner that facilitates comparisons with similar docs). In the end, we found that amendments to the last approach yielded the best results. We nonetheless discovered that (atypical) documents of less than 20 terms yields a higher rate of false positives and thus are not recommended as reliable candidates for signature generation.

DDD Algorithm Performance	Training Set	Test Set
Agreement (Correct)	86 of 88	79 of 82
Misses (False Negatives)	2	3
Over-Identification (False Positives)	3	8

Table 8: DDD Algorithm-Assessor Correspondence

The algorithm recognized 98% of the dup sets identified by the assessors in the training round (86/88) with 3 false positives and 96% of the dup sets in the test round (79/82) with 8 false positives. Clearly the false positives in the test set are cause for some concern. Upon performing a failure analysis of these over-generated sets, we were able to make three key observations. First, by tightening our date window from 12 weeks to 6 weeks, three of the test round's false positives can be eliminated with no impact on the other results. Secondly, three of the false positives are sets resulting from the same financial query about "cotton futures" and are either practically all numerical or represent a boiler plate text of nearly identical content with only two numbers changing from one day's document to the next.[9] The take away value of these observations is that for those largely quantitative documents for which the algorithm was not designed, performance is spotty and some user education may be helpful. Thirdly, of the remaining false positives, the documents are often so close that the extent of their "erroneous" nature is debatable among the assessors (e.g, John Ashcroft's Justice Dept. budget presented before the House and the Senate—same material, different audience, different title).

7.3 Implementation Issues

Important questions need to be addressed regarding the production implementation of such an approach—signature storage requirements (Phase 1) and signature comparison cost requirements (Phase 2) are but two. Besides two bytes for each pub_date and doc_length value, each term_vector entry can be encoded into three bytes or less (based on an idf table with 1,000,000 or fewer entries). This indicates that the entire signature would require as much as 184 bytes [(60 $\times$ 3) + 2 + 2], if no form of additional compression were used. The worst case scenario for inter-vector comparisons is $O(n^2)$. Yet by invoking heuristics that leverage other significant features, these can be reduced to practically linear time. No vectors are compared when their pub_dates differ by more than 6 weeks, and no vectors are compared when the length of one doc varies by more than $\pm$20% with respect to another (thus, the application of the duplicates definition is not transitive across document pairs). In a large number of instances, no term comparisons need to be performed due to our binning procedures; when they do, on average no more than 15 need be compared before we know that a pair cannot score high enough to be considered duplicates. Tokens participating in a term vector are sorted from highest to lowest idf, so the (rarer) higher idf terms are always compared first. For docs that have greater similarity, generally well under 60 terms need be compared in order to know that a pair can score high enough for a duplicate designation (using a threshold of less than 50/60). Given the idf ordering of term vector participants and a numerical 3 byte encoding, term vector comparisons can cost no more than $O(c \cdot n)$, where n is 60 and c is roughly 0.25. The approach was found to be computationally effective for result sets averaging in the range of 500 documents.

8. DISCUSSION OF PRESENTATION ISSUES

In a professional field like law, where legal practitioners[10] need to be concerned about the recall as well as the precision of their search results, researchers cannot afford to leave any piece of evidence unexamined, whether supporting or contradicting their position. Such evidence could prove to be a liability were it strictly discarded from view because the associated documents were determined to be duplicates. A number of parameters exist to assist a system in deciding which duplicate document to retain—(1) the highest ranking in the results list, (2) either the first or the last most recently published (determined by date or time stamp), or (3) the longest document (considering added introductory material, subtitles, or local headings). Yet regardless of which selection criterion is invoked, there will always be situations where the researcher may have preferred one of the versions of the document that has not been displayed. It would be possible for a legal researcher investigating news stories, for example, to prefer a release of a story that was published in the *Dallas Observer* rather than in the *Los Angeles Times* as a material witness may have grown up in Texas and the

[9] In an IR context, the percentages presented correspond to recall. By contrast, 86/89 (96.5%) and 79/81 (97.5%) correspond to precision (excluding the 3 date and 3 numeric-resolved misses).

[10] Legal practitioners include judges, law clerks, attorneys, paralegals, and other professionals serving the legal domain.

article's subtitles may disclose special information regarding this detail. For this reason, we have found it prudent to avoid discarding any documents identified as duplicates; rather, we make them available in a separate grouping that is indented within the results list and follow the first occurrence of a fellow dup set member. In this manner, the user's frustration of encountering multiple duplicate documents in a results set is alleviated, yet the user can still retrieve any potentially "on point" documents in the "grouping of duplicates." Clearly, this is an implementation issue, but one that is important enough to discuss in sufficient detail to clarify how the algorithm can work for, rather than against, any efforts to improve researcher productivity.

9. CONCLUSIONS

The accelerated growth of massive electronic data environments, both Web-based and proprietary, has expanded the need for various forms of duplicate document detection. Depending on the nature of the domain and its customary search paradigms, this detection can take any of several forms, but may be largely characterized by either identical or non-identical deduping. Our own exploration addressed a real world replication problem occurring in a large production environment. In response to this investigation and its identification of frequently occurring categories of duplicates, we have pursued two distinct approaches to recognize and treat such similarities. They include the strict document fingerprint approach to recognize *identical* duplicate documents, and, a fuzzier feature set approach to identify highly similar but *non-identical* duplicate documents. One of the most significant observations of our research involves the instability of collection statistics (idfs) following updates. This finding serves to discourage the perceived benefits of frequent updates of collection statistics, and favors reliance upon a large, comprehensive, and more static multiple domain table or training collection.

For non-duplicate document detection, our dedicated test collection and trials suggest that a multi-dimensional feature set approach to characterizing and comparing documents can provide a strong indicator of the degree of duplication between two documents. The treatment of its multi-dimensional feature set frees it from reliance upon uni-dimensional features and the brittle syntactic structures that documents may possess.

10. ACKNOWLEDGMENTS

We recognize the role that Denis Hauptly played in this investigation; his determination helped get it started. We thank Shakila Xavier and Dan Dyke for their preliminary study into duplicate types. We are also grateful to Monem Meziou for his acquisition of our data sets from the production environment. We appreciate the duplicate assessment efforts of Brian Craig, Carol Jo Lechtenberg, Michael Paulos, and Tom Perusse. We acknowledge the support of Marilee Winiarski who invested in our non-identical duplicate research. And, lastly, we thank Bruce Getting and Jie Lin for their handling of computability and real-time processing issues in the production environment.

11. REFERENCES

[1] S. Brin, J. Davis, and H. García-Molina. Copy detection mechanisms for digital documents. In *Proceedings of the Special Interest Group on Management of Data (SIGMOD '95)*, pages 398–409. ACM Press, May 1995.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the Seventh Int'l World Wide Web Conference (WWW7 '98)*, pages 107–117. Elsevier Science, April 1998.

[3] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In *Proceedings of the Sixth Int'l World Wide Web Conference (WWW6 '97)*, pages 391–404. Elsevier Science, April 1997.

[4] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems (TOIS)*, 19(2):97–130, April 2001.

[5] A. Chowdhury, O. Frieder, D. Grossman, and M. C. McCabe. Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems (TOIS)*, 20(2):171–191, April 2002.

[6] J. W. Cooper, A. R. Coden, and E. W. Brown. Detecting similar documents using salient terms. In *Proceedings of the 11th Int'l Conference on Information and Knowledge Management (CIKM '02)*, pages 245–251. ACM Press, Nov. 2002.

[7] D. P. Dabney, H. R. Turtle, J. G. Conrad, et. al. *System and Method of Processing Formatted Text Documents in a Database*. U.S. Patent App. No. 09/120,170, 1999.

[8] O. Frieder, D. A. Grossman, A. Chowdhury, and G. Frieder. Efficiency considerations for scalable information retrieval servers. *Journal of Digital Information*, 1(5):26 pgs, Jan. 2000.

[9] N. Heintze. Scalable document fingerprinting. In *Proceedings of the Second USENIX Electronic Commerce Workshop*, pages 191–200, Nov. 1996.

[10] U. Manber. Finding similar files in a large file system. In *USENIX Winter 1994 Technical Conference Proceedings (USENIX '94)*, pages 1–10, Jan. 1994.

[11] C. Miller. Detecting duplicates: A researcher's dream come true. *Online*, 14(4):27–34, July 1990.

[12] M. J. Moroney. *Facts from Figures*, pages 334–370. Penguin Books, Middlesex, UK, 3rd edition, 1956.

[13] S.-T. Park, D. M. Pennock, C. L. Giles, and R. Krovetz. Analysis of lexical signatures for finding lost or related documents. In *Proceedings of the 25th Int'l Conference on Research and Development in Information Retrieval (SIGIR '02)*, pages 11–18. ACM Press, Aug. 2002.

[14] T. A. Phelps and R. Wilensky. Robust hyperlinks: Cheap, everywhere, now. In *Proceedings of the 8th Int'l Conference on Digital Documents and Electronic Publishing (DDEP '00)*. Springer-Verlag, Sept. 2000.

[15] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*, pages 507–510. Cambridge University Press, New York, NY, 2nd edition, 1992.

[16] N. Shrivakumar and H. García-Molina. Finding near-replicas of documents on the Web. In *Proceedings of Workshop on Web Databases (WebDB '98)*, pages 204–212, March 1998.

[17] C. Tenopir and P. Cahn. Target & Freestyle: DIALOG and Mead join the relevance ranks. *Online*, 18(3):31–47, 1994.

[18] P. Thompson, H. Turtle, B. Yang, and J. Flood. TREC-3 ad hoc experiments using the WIN system. In *Proc. of TREC-3*, pages 211–217. NIST, Nov. 1995.

[19] H. Turtle. Natural language vs. Boolean query evaluation: A comparison of retrieval performance. In *Proceedings of the 17th Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*, pages 212–221. Springer-Verlag, July 1994.

[20] H. R. Turtle. *Inference Networks for Document Retrieval*. Ph.D. Dissertation, Univ. of Massachusetts–Amherst, 1991.

[21] U.S. Department of Commerce/National Institute of Standards and Technology. *Secure Hash Std*, 1995.

[22] E. M. Voorhees and D. Harman. Overview of the Sixth Text REtrieval Conference (TREC-6). *Information Processing and Management*, 36(1):3–35, Jan. 2000.