Essential Deduplication Functions for Transactional Databases in Law Firms

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

Edward L. Raymond, Jr.
Content Operations
Thomson–West
Rochester, New York 14694 USA
Ed.Raymond@Thomson.com

ABSTRACT

As massive document repositories and knowledge management systems continue to expand, in proprietary environments as well as on the Web, the need for duplicate detection becomes increasingly important. In business enterprises such as law firms, effective retrieval applications depend upon such functionality. Today's Internet-savvy users are not interested in search results containing numerous sets of duplicate documents, whether exact duplicates or near variants.

This report addresses our work in the domain of legal information retrieval, working with a large, *transactional* knowledge management system. We specifically explore the occurrence and treatment of identical, near-identical, and fuzzy duplicate sub-documents ('clauses') in a contracts database. To our knowledge, we are the first to use principled methods to construct a test collection of transactional documents for such research purposes, one which identifies a variety of duplicate types and is deployed to establish baseline algorithmic approaches to deduplication.

We subsequently investigate the application of digital signature techniques to characterize and compare similar clauses in order to identify duplicates and near duplicates. This approach establishes a baseline using methods and algorithms first developed in a parallel domain. It produces a set of promising results following an extensive assessment phase involving direct comparisons with gold training and test data created by expert attorneys working in the transactional domain.

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.m [**Information Storage and Retrieval**]: Miscellaneous—*Test Collections*

General Terms

Experimentation, Measurement, Design, Algorithms

Keywords

data management, duplicate detection, document signatures

ICAIL '07, June 4-8, 2007, Palo Alto, California USA

1. INTRODUCTION

Both on the World Wide Web and in proprietary data environments, it is currently possible to have tens of millions of textual objects indexed as part of the same collection.[1] Transactional databases are particularly challenging in that the contracts they contain consist of a highly hierarchical structure where the same text object may appear at several levels across documents. In large knowledge management environments like law firms, there may be terabytes of information stored. In such environments, the identification of duplicate documents is an important factor for a practical and robust data delivery platform.

One goal of this work is to leverage domain expertise in order to characterize the duplication existing in such large textual collections. We subsequently try to validate the completeness and reliability of this effort with analyses of assessor agreement, error rates, and significance.

This work makes two significant contributions. First, it creates and deploys a deduping test collection by harnessing:

(a) real user queries;
(b) a significant 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 in Cooper, et al. [10]. Other recent work has often been syntax rather than lexical-based, Web-based (focusing on issues such as URL replication and instability), and conducted offline (e.g., examining large numbers of permutations before constructing a feature set). Previous research is substantially different than our current efforts which target a dynamic law firm environment. The novelty of this work thus derives from its being the first to focus on duplicate document detection for *transactional* documents in the legal domain.

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 used to assemble our duplicate document detection collection. Section 4 describes a baseline deduping algorithm for non-identical duplicates and the preliminary trials to assess it. Section 5 delves into performance and evaluation issues associated with the algorithm. We present our conclusions in Section 6 and discuss Future Work in Section 7.

[1] In this paper, we will use "collection" to refer to a database of textual documents, and "deduping" to refer to duplicate document detection and subsequent removal or suppression.

2. PREVIOUS WORK

2.1 Knowledge Management Applications

The problem of duplicate entries in large transactional document repositories is well known. Solutions to this problem are not widely publicized. Little if anything has been written on the subject of transactional deduplication in research proceedings or patent databases. Occasionally one comes across references to the problem and its treatment in trade journals or white papers [28]. In one system designed to facilitate transactional knowledge management and acceleration of a firm's document creation process, the focus is on providing transactional resources for a limited number of practice areas (e.g., real estate, employment, licensing), while tying the work environment closely to the Office 2003 suite (and the MS Word drafting environment), with little or no mention of how to address the duplicate document or clause issue [21].

2.2 Earlier Studies

Some of the first duplicate document detection studies addressed problems such as plagiarism, intellectual property violations, and partial replications within file systems [1, 16, 18]. In many of these instances, researchers either owned or constructed their own data sets for the purposes of testing.

Concerning publicly available collections, in a published technical report Sanderson [24] described a set of tests he developed for the identification and potential removal of duplicate documents present in the Reuters test collection of over 22,000 news articles [24].[2] He performed a series of three tests to determine (i) documents that are highly similar, but reported as separate events; (ii) documents that are very similar, where one is a longer version of the other; and (iii) documents that are exact duplicates of each other. Candidate documents were found by submitting a document as a distinct query and examining the results. Documents were considered exact duplicates if the first retrieved the second and vice-versa. To avoid retrieving too many similar documents about related but different events (such as financial transactions), a condition was established requiring candidate pairs to be published within a 48-hour window of each other.

For test (i), of 33 candidates for similar article, different topics, 29 (88%) were not about different events. For test (ii), of 283 candidates, 139 (49%) turned out not to be longer versions of the other. And lastly, for test (iii), of 322 candidates, 320 (99%) passed the exact duplicates test. By presenting these findings, Sanderson helped characterize the nature and scope of the duplication problem in collections of news documents. Note that a more comprehensive review of pre-Web duplicate document detection research can be found in Conrad, Guo, and Schriber [8].

2.3 Recent Web-based Approaches

Much of the dedicated duplicate document research performed in the last decade has focused on TREC data or ad hoc corpora constructed from informal collections of Web pages, e.g., in [7].

Broder, Glassman, Manasse and Zweig [3] 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.). They conduct tests on virtually all of the Web at the time (1996). 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 m-th 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 like the Web, 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 simply 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 [7, 10].

This work was expanded upon by one of the co-authors, Manasse, et al., in a subsequent set of Web-based experiments [11]. They identified clusters of near-duplicate documents and tracked their stability over time. They relied upon "mega-shingles" to compute clusters of near duplicate documents, where near-duplicate documents were defined as documents having at least two super-shingles in common (i.e., a common mega-shingle). The authors found that two documents that are 95% similar have an almost 90% chance of having a mega-shingle in common; yet, two documents that are 80% similar have only a 2.6% chance of having a mega-shingle in common. In contrast to Broder, et al., Fetterly, et al. determined that their mega-shingling near-duplicate identification approach (using a union-find data structure) had a run time that was almost linear in the number of documents.[3]

In on-going work by Fetterly, et al., the authors study the evolution of the Web [13] and come to harness experimental collections of 151 million and 96 million Web pages and identify the distribution of duplicates therein, i.e., via short five-token syntactic (non-linguistic) phrases, in order to reveal Web sites replete with "spam" [12].

Another approach used by Schleimer, Wilkerson and Aiken is known as "Winnowing" [25]. Like shingling, it can be adapted to a subset of local document fingerprints created by hashing; unlike shingling, it is based on strings of characters rather than strings of tokens. As such, winnowing ignores knowledge of its particular application domain (news & finance) as well as standard English text (tokens and their rarity). In some respects, winnowing operates at a logical extreme of the fingerprint by hashing. It applies an appreciable amount of math to the digital signature problem, but without harnessing domain expertise, semantic knowledge,

[2]The results Sanderson reported apply to both the original Reuters collection of 22,173 documents and the newer Reuters collection of 21,578 documents:
www.daviddlewis.com/resources/testcollections/reuters21578/

[3]Broder, et al.'s multi-step process took 10 CPU days to treat 30 million documents, while Fetterly, et al. processed 150 million documents in a fraction of that time.

or even term distribution information. It may be effective for general Web-based information about which we may know little, but for directed domains for which we do know quite a bit, it may work at a disadvantage.

Both of the above approaches rely on hash values for each document sub-section, and both prune these hash values to reduce the number of comparisons that the algorithms must perform. The computational complexity and thus resultant efficiency of the schemes are therefore quite dependent on the manner and extent to which the pruning is performed. The more aggressive the pruning, the more efficient are the algorithms, at the cost of increasing the prospects for identifying false positive duplicates.

Chaudhuri, Ganti and Motwani recently approached duplicate detection from a record merging perspective and focused on eliminating the problem based on two fundamental properties of duplicate tuples: compact set and sparse neighborhood [6].

Shivakumar and Garcia-Molina describe factors in identifying nearly identical documents on the Web for the benefit of Web crawlers and Web archivers [26]. They consequently concentrate on computing pairwise document overlap 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 re-ranking), inspecting the environments from which Web documents originate; and (3) archiving Web documents, enabling greater compression of shorter pages that replicate more complete document 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, Frieder, Grossman and McCabe [7] refine their collection statistic, idf-based deduping algorithm for efficiency and effectiveness on both Web-based and non-Web-based test collections. They also compare its performance to other state-of-the-art techniques such as shingling and super-shingling. The authors demonstrate that their approach, called I-Match, scales in terms of number of documents and works well for documents of diverse sizes. They 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 more recent work, Kołcz, Chowdhury and Alspector offer an alternative to I-Match that relies upon a set of digital signatures for a document created from randomized subsets of the global lexicon [17]. The motivation for this approach is to compensate for the case where the fraction of terms participating in the I-Match signature (hash) relative to the terms in the lexicon used is small. The significance of the approach stems from the fact that I-Match may result in false positive matches if a large document has a small term intersection with the lexicon used. The authors show that this approach outperforms traditional I-Match with an improvement in overall recall of 40% to 60%. An advantage of the scheme is its increased insensitivity to word permutations and its document length independence. The authors do not quantify, however, the additional cost associated with generating the multiple lexicons, creating the multiple (K + 1) signatures, and comparing one (K +1) tuple with another.[4] The computational cost of this improved performance appears to be implementation dependent. For non-critical applications such as that mentioned by the authors—reducing spam by a significant percentage in a large ISP provider e-mail system—the benefits of the technique may outweigh its costs and justify its deployment.

The Web-related research of Park, Pennock, Giles and Krovetz 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 [23]. 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 · idf terms produce the most effective unique and relevant properties for Web page signatures.

Cooper, Coden and Brown discuss methods for finding identical as well as similar documents returned from Web-based and internal IBM enterprise searches [10]. 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 super-shingl[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.[5]

The significance of this overview is that there has not yet been established a standard information retrieval (IR) test collection for duplicate document detection. As we approached the problem, this was our first essential step, since without a validated test collection, we could not have confidence in the approaches and performance measures that followed.

3. METHODOLOGY

3.1 Background

Initially the Thomson business unit responsible for law firm knowledge management (West km) asked us for technologies to identify and treat duplicate documents in transactional databases. In response, we began characterizing the distribution of duplicate types across a representative contracts collection that we constructed from documents obtained through an acquisition. The collection statistics for the resulting contracts database are shown in Table 1.

[4](K + 1): 1 represents the original and complete I-Match signature and K represents the number of permutations of the original lexicon. Kołcz, et al. experimented with K ranging from 1 to 10.

[5]The test to determine whether a technique is principled, in this case, depends upon whether it avoids leaving anything to chance or probabilistic uncertainty. In short, is the approach highly reliable?

Duplicate Type	Definition
Identical	Clauses which contain no variation in the substantive words used, but which may have variations in non-substantive words, such as articles (e.g., we would view clauses which used the same words except that one used "Company" and another "the Company" as identical).
Near Identical	Identical but for defined terms or keywords (e.g., company name, address, dollar amount, jurisdiction, etc.). Clauses which are the same but for the use of synonyms (e.g., Executor vs. Personal Representative) would fall into this category.
Fuzzy	Essentially the same language and legal meaning. Clause lengths must not vary by more than ± 20% of each other and clauses must contain at least an 80% terminology match.

Table 2: Definitions for Duplicate Types

Duplicate Type	Sample Pairs
Near Identical	Section 14.06 **Successors and Assignors.** All covenants and agreements in this *Agreement by the Sponsor* shall bind its successors and assigns, whether so expressed or not.
	Section 11.09. **Successors and Assignors.** All covenants and agreements in this *Indenture by the Issuer* shall bind its successors and assigns, whether so expressed or not.
Fuzzy	3.1 **Successors and Assigns.** This agreement enures to the benefit of and is binding on the parties hereto and their respective successors and assigns.
	32. **Successors and Assigns.** This agreement shall be binding upon, and enure to the benefit of, the parties, their legal representatives, successors and assigns.

Table 3: Illustrations of Non-Identical Duplicate Types

Component ('Doc' Type)	Size (Total No.)
Files	2,410
Documents	4,694
Clauses	*82,485*
Defined Terms	75,834

Table 1: Transactional Test Collection Statistics

We then initially proceeded to address two of the three largest and most significant categories of duplicates associated with this type of database. These duplicate-types include: (1) identical duplicates, (2) near identical duplicates, and (3) 'fuzzy' duplicates. The definitions of these duplicate-types, referring to clause-level granularity, are presented in Table 2, while illustrations of the two **non-**identical duplicate categories are presented in Table 3. These were some of the tools we provided to our attorney assessors when asking them to evaluate sample result sets.

Much effort has addressed issues surrounding relevance assessments in various contexts of Information Retrieval over the years [4, 14, 29]. At a certain level of abstraction, the task we eventually asked our assessors to perform is similar in function to that of a standard relevance judgment. Given an initial target document (that may be viewed conceptually as a query), our assessors are asked to identify other documents in the same result set that satisfy the similarity metrics (i.e., are "highly relevant" to it).

3.2 Problem Definition and Client Feedback

In an earlier and related project, we conducted a feedback session with 25 members of our Library Advisory Board, who represented a variety of our clients' enterprises and firms [9]. Most of the group's formal training comes from the field of Library Science. In all, 17 of the 25 participants provided non-trivial replies to our suite of questions. The role of the members of this Board is typically to field information needs from their enterprise's legal practitioners and engage in a variety of related research projects. As such, they are uniquely positioned to provide domain expertise in their focus areas and an excellent group to consult.

The objective of the session we conducted was to describe, both qualitatively and quantitatively, the nature of the most annoying duplicate documents or textual segments such as clauses and to receive feedback from participants on these types. This exercise resulted in the following description: *a non-identical duplicate pair consists of two documents that possess a terminology overlap of at least 80% and where one document does not vary in length from that of the other by more than ± 20%.* It was generally believed that to call documents with less than an 80% terminology overlap duplicates would be problematic. Although such documents might adequately satisfy Broder's definition of *containment*, they could not reasonably satisfy a definition for *resemblance* [3], which is our principal objective.

In subsequent discussions with transactional domain experts within West km who were very familiar with the needs and expectation of their customers, the required degree of overlap was raised to 90%. A priority was also placed upon *clause-level* rather than document-level deduplication. Note, however, that a 90% overlap condition does not imply that 90% of the shared text must be identical. In some examples, even though at the paragraph, sentence, phrase, and word levels, documents may differ substantially (not to mention at the title level), they may still satisfy the similarity conditions of this definition and would thus be judged as valid non-identical duplicates.

These guidelines produced a working definition of "near duplicate" pairs with which we proceeded. Note that implicit in this definition is the fact that these relations are *not* transitive. That is to say, if texts A & B are duplicates and B is 80% the length of text A, and texts B & C are duplicates and C is 80% the length of text B, it does not follow that C is also a duplicate of A. In this instance, that is clearly not the case.

3.3 Collection Generation and Domain Expert Assessments

To test our approach, we selected a total of 50 real user information requests from a query log that also included a human-assigned transactional query *category*. These logs originated from a production environment that was incorporated into West km. The queries were randomly selected with the exception that a results list of at least 20 documents was required. A sample of these categories is shown in Table 4 while a sample of the queries is shown in Table 5. The average query contained roughly three terms. Each query was run using the West km Transactional system which provides natural language search capability, depending on the preference of the user. After running these queries against our test collection, which consisted of approximately 82,500 clauses, we assembled the top twenty clauses returned from each query. While the cumulative result set consisted of nearly 1,000 clauses, each set of twenty clauses was reviewed by two attorney-editors,[6] in order to identify their duplicate subsets.[7] This process helped us produce standard training and test sets against which computational approaches would be compared.[8] Collection statistics for the resultant training and test sets are shown in Table 6.

No.	Category of Contract (Selected)
A.	Standard Provisions (All Contracts)
B.	Acquisition Agreements
C.	Employment Agreements
D.	Escrow Agreements
E.	Investor Rights Agreements
F.	Joint Ventures
G.	License Agreements
H.	Limited Partnership Agreements
I.	Loan Agreements
J.	Merger Agreements
K.	Real Estate (REIT/Partnerships)
L.	Reorganization Agreements
M.	Security Agreements
N.	Underwriting Agreements

Table 4: Transactional Law–Sub-Categories

Query Type	Transactional Law Queries
Acquisitions	"excluded liabilities"
Employment	"put option"
Joint Venture	"event of default"
Limited Partnership	"initial capital contribution"
Loan Agreement	"fixed charge coverage ratio"
Security Agreement	"sale of collateral"

Table 5: Sample Qrys–Duplicate Set Construction

3.3.1 *Details of the Document Inspections*

In this trial, we applied definitions of non-identical duplicates that were drafted by customer and business unit work groups. The resulting definition states that two texts are duplicates if they retain much of the same language and are at least 90% similar.[9] To formally review the duplication status of our result sets, we assembled twelve attorney-editors. The 50 sample queries were divided into two sets of 25, the first set to be used to train a prototype system and the second set to test it. The process by which the query results were judged was scheduled over four weeks time (as indicated in Table 7). During week 1, results from the training queries were assessed for their duplication status. Each team reviewed the results from 25 queries, approximately 5 queries per team per day. Although members of the same team reviewed the same results, they did so independently.

Assessor Pair	Team A	Team B
Week 1	25 Qrys	25 Qrys
Week 2	*Arbitration*	*Arbitration*
Total	25 Qrys	25 Qrys
Combined	50 Qrys	

Table 7: Scheduling of Assessments

The assessors also had access to the term counts available in the core documents (which excluded only a limited amount of metadata, such as name of source file, as shown in Table 8). Week 2 served as an arbitration week. When members of the same team disagreed about a duplicate set, a senior attorney-editor not on that team would serve as an arbitrator or tie-breaker. In this way, a virtual voting system was established. Every result set would thus be reviewed by a minimum of two assessors, and sometimes by three. This approach was intended to produce dependable judgments from the process.

Type	Sample Instantiations
ClauseTitle	Section 7.9 WAIVER OF JURY TRIAL
DocTitle	Pledged Bonds Custody & Security Agreement
DMSFile	PledgeCustody.1March00.doc
ClauseTitle	(m) LEVERAGE RATIO
DocTitle	Letter of Credit & Reimbursement Agreement
DMSFile	NNZX18!.doc
ClauseTitle	6. Compensation of Escrow Agent
DocTitle	EXHIBIT C ESCROW AGREEMENT
DMSFile	0587980.doc

Table 8: Metadata Classifications for Clauses

To further help ensure judgment reliability and consistency, a training document was prepared for the assessors that included illustrations and detailed instructions. In addition, a preliminary training exercise was developed for each team that included real user query result sets and the opportunity for the particpants to discuss their judgments as well as the granularity of their inspection. All of the assessors participated in the same initial training session and were asked to apply their knowledge to the same pair of sample result sets. Training guidelines were amended as a result of these sessions in order to clarify the level of granularity of analysis necessary for the task. In general, the assessors found these training cases quite instructive. As beneficial

[6] Our attorney-editors, who are required to have law degrees, spend a significant portion of their day working closely with essential analytical legal texts.

[7] Inter-assessor agreement is discussed in Section 3.4.

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

[9] (a) I.e., 90% of the words in one text are contained in the other (in terms of overall *terminology* rather than individual term *frequency*).

(b) For texts that do not meet a working threshold for similarity or *resemblance*, Broder, et al. monitor a second looser relationship described as *containment* [3].

Query Set	Query Count	Total Clauses	Clauses / Query	Mean Length (in tokens)	Standard Deviation	Median Length (in tokens)
Training	25	478	19.1	532.9	1120	154.0
Test	25	464	18.6	233.5	247	143.5
Combined	50	942	18.8	385.3	829	152.0

Table 6: Collection Statistics for Query-generated Clause-level Training and Test Sets

as this training round was, the assessors did not produce completely uniform judgments. Information and statistics about inter-assessor agreement can be found in Section 3.4.

Table 9 presents the number of queries that yielded duplicate sets in the trial. Some queries produced no duplicate sets—two in the training set and none in the test set. These were retained for two main reasons. First, they were produced by our random sampling and are therefore presumably representative and, second, they can still be instructive in terms of false positive sensitivity experiments, since these queries should produce no duplicate sets.

By contrast, Table 10 shows the distribution of duplicate sets by size. The queries for the test set produced slightly fewer duplicate sets; but both sets also produced several larger duplicate sets consisting of 4, 5, 6, or more clauses. The assessors identified an average of 3.8 duplicate sets per query-result set (4.2 in the training set and 3.4 in the test set).

Duplicate Document Detection	Training Set	Test Set
Total Queries	25	25
with Duplicate Sets	23	25
without Duplicate Sets	2	0

Table 9: Distribution of Duplicates Across Queries

Duplicate Set Size	Training Set (Frequency)	Test Set (Frequency)
Pairs	69	63
Triplets	19	14
Quadruplets	13	5
Quintuplets	1	2
Sextuplets	1	1
More than 6	1	1
Total	104	86

Table 10: Distribution of Total Resulting Duplicate Sets

3.4 Inter-assessor Agreement

When asked to verbally characterize the nature of the duplicate sets identified in relation to exact duplicates, the assessors were in agreement that the sets they found spanned the identical–non-identical duplicates spectrum.[10]

Of the 50 queries reviewed by a pair of assessors, 23 resulted in complete agreement between the assessors. Furthermore, Team A agreed on over 3 in 4 of its duplicate sets, while Team B agreed on over 4 in 5 of its sets. Disagreements between asessors were resolved by means of a voting process, whereby a senior attorney-editor not on the team served as an arbitrator and cast a third and tie-breaking judgment.

We used the Kappa statistic for nominally scaled data in order to compare our inter-assessor concordances over the 50 result sets [27]. The Kappa coefficient of agreement is the ratio of the proportion of times that the assessors agree (corrected for chance agreement) to the maximum proportion of times that the assessors could agree (corrected for chance agreement):

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where *P(A)* is the proportion of times that the k assessors agree and *P(E)* is the proportion of times that we would expect the k assessors to agree by chance. If there is complete agreement among the assessors, then $\kappa = 1$; whereas if there is no agreement (other than the agreement that would be expected by chance) among the assessors, then $\kappa = 0$. We used as our baseline set of candidate duplicates the set of all textual (clausal) pairs identified by at least one of our assessors. The results are presented in Table 11.[11]

Computational linguists have taken $\kappa = 0.8$ as the norm for significantly good agreement, although some argue that there is insufficient evidence to choose 0.8 over, for instance, other values between 0.6 and 0.9 [19].

Given a result set of $n = 20$ clauses, there are $n(n-1)/2$ or 190 total comparisons required. We had two assessors make categorical judgments with respect to each of these candidate pairs: exact duplicate, fuzzy duplicate, or non-duplicate. We computed the Kappa statistic over the comparison space described using a more tractable binary comparison (duplicate versus non-duplicate).

Because the majority of clausal pairs are not duplicates, the possibility for chance agreement is high. But this marshals the strength of the Kappa statistic—it corrects observed agreement with respect to chance agreement. Given the size of the space (190 pair-wise comparisons), the resulting Kappa values we obtain are likely to be slightly inflated (given that the vast majority of the 190 comparisons are non-duplicates), but not significantly [20].

After determining the value of the Kappa statistic, κ, it is customary to determine whether the observed value is greater than the value which would be expected by chance. This can be done by calculating the value of the statistic z, where,

$$z = \frac{\kappa}{\sqrt{var(\kappa)}}$$

in order to test the hypothesis $H_o : \kappa = 0$ against the hypothesis $H_1 : \kappa > 0$ [5, 27].

[10] In another work, we categorize and quantify into six classes the distribution of duplicates found in our test collection [8].

[11] For the macro-averaged scores, the Kappa statistic is calculated using a single table for all the comparisons involved in the entire query set. The micro-averaged scores would be calculated using a separate table for the comparisons from each query; these scores in turn would be averaged together to derive the composite Kappa score.

Assessor Pair	Editor–Editor	System–Editor-Arb.
Training *(First 25 Queries)*	$\kappa = 0.87^{*}$	$\kappa = 0.95$
Test *(Second 25 Queries)*	$\kappa = 0.92^{+}$	$\kappa = 0.94$
Combined (50 Qrys) *(Train & Test)*	$\kappa = 0.898$	$\kappa = 0.943$

Table 11: Kappa Statistics for Inter-assessor Agreements for Duplicate Set Identification (macro-averaged scores)

The above value of κ for the combined query set yields z = 3.925 (Team A, Queries 1-25)* and z = 5.191 (Team B, Queries 26-50).$^{+}$ These values exceed the $\alpha = 0.001$ significance level (where z = 3.090). Therefore, we may conclude that the assessors exhibit significant agreement on this categorization task. It is important to note that these results were produced *before* we introduced the arbitration round, wherein another attorney-edtor not on the team resolved differences in judgments between the two original assessors. Given a third expert casting a "vote" on these differences, the final duplication judgments are arguably more reliable than those examined during the Kappa analysis.

4. OVERVIEW OF INITIAL ALGORITHM

Sections 4 and 5 are included to examine the utility of the resulting transactional duplicate detection collection when designing, developing, and testing algorithmic approaches to deduplication of fuzzy duplicates.

Note that there have been efforts to completely automatically detect "redundancy" in result sets [31], but these appear to eliminate the role of the client and focus exclusively on mathematical models of content, even in highly dynamic retrieval environments. In order to determine our ability to identify and characterize such non-identical duplicate documents using the contributions from our client base, we began investigating reliance upon an expanded multi-dimensional feature set or "digital signature." This feature set includes:

- magnitude component (doc_length);
- hash of the top N rarest terms and their locations (hash_key);
- core content component (term_vector).

The role of the first two is to provide heuristics to reduce the need for more costly term comparisons. They do not reduce the number of candidate pairs as much as reduce the search space for valid duplicate candidates. In addition to document length (excluding metadata) and top-term hash, a document's term_vector 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 for *documents* of moderate length, because (a) it offers substantially finer granularity to the process, and (b) it does not exceed the short length limits of the vast majority of such documents. For clause-level deduplication, however, where the texts can and are often considerably shorter, the lower bound of 30 terms was found to be more practical, but also invites an algorithmic means of smoothly accommodating still shorter length texts.

The percent overlap between two documents' term_vectors served as our de facto similarity measure. In practice, once the heuristics completed their reduction of eligible candidate pairs, the algorithm then used as its matching criterion a 90% vector overlap.

Aside from core content from contract clauses, metadata indicating law firm, key indexing terms, source file, etc. (some shown in Table 8) is not used. We have determined that such supplemental content tends to increase the number of false positives, since related but dissimilar documents may possess similar metadata and classification terms.

It is worth noting that even though these metadata classification indexes are not considered part of the core document, they were not suppressed from our assessors (though the assessors were generally discouraged from using them in their determination of duplication status, because of the false positive risk discussed above). Nonetheless, in the comprehensive collection that resulted, these fields are still viewed as intrinsic to the corpus and are therefore retained.

5. COLLECTION DEPLOYMENT AND PERFORMANCE EVALUATION

5.1 Test Corpus and Algorithm Assessment

Figures 1 (a) and (b) and Table 12 show the performance of the algorithm outlined above relative to the gold data standard established by the attorney assessors, in terms of agreement (correct identification), false negatives (misses), and false positives (over-generation). An idf table constructed from a separate training collection of over 2 million documents is used to identify the rarest terms. A number of modifications were made to the algorithm during the training phase. Most notable is how it treats short texts (with fewer than 30 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 text's vector, and (iii) padding the short text'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.

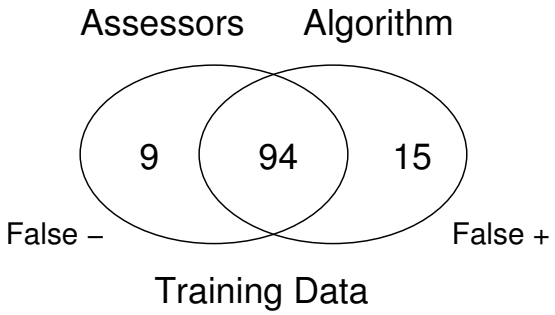

Figure 1 (a). *Dup Sets Identified in Training Round*

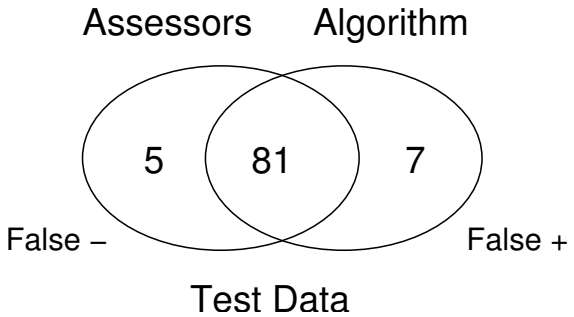

Figure 1 (b). *Dup Sets Identified in Test Round*

Duplicate Type	Training Set				Test Set			
	Capture	Miss	Total	Percentage	Capture	Miss	Total	Percentage
Identical	65	2	67	97.0%	60	0	60	100.0%
Near Identical	12	1	13	92.3%	15	1	16	93.8%
Fuzzy	17	6	23	73.9%	6	4	10	60.0%
Total	94	11	103	91.3%	81	7	86	94.0%

Table 12: Distribution of Results for Clause-level Duplicate Set Identification

We nonetheless discovered that (atypical) texts of less than 10 terms yield a higher rate of false positives and thus are not reliable candidates for signature generation.

The algorithm recognized 91% of the duplicate sets identified by the assessors in the training round (94/103) with 15 false positives and 94% of the duplicate sets in the test round (81/86) with 7 false positives.[12] Upon performing a failure analysis of our false positives, we were able to make three key observations. First, by tightening parameters such as length of term vector, from 60 to 30 terms, several of the training round's false positives can be eliminated with no impact on the other results. Secondly, several of the false positives are sets resulting from algebraic clauses with equations or rates of exchange and are either practically all numerical or represent a boiler plate text of nearly identical content with only two numbers changing from one document's clause to the next. The significance of this observation is that for those largely quantitative clauses 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., with slight adjectival differences distinguishing one clause from another).

If we define *precision* as the percentage of duplicate documents identified by the algorithm that agree with the assessors and *recall* as the percentage of the total number of duplicate documents identified by the assessors also identified by the algorithm, then our results can be found in Table 13.

DDD Algorithm Performance	Training Set	Test Set
Precision (%)	94 of 109 86.2%	81 of 88 92.0%
Recall (%)	94 of 103 91.2%	81 of 86 94.2%

Table 13: DDD Algorithm-Assessor Correspondence

Given this preliminary effort to investigate effective deduplication for contract clauses, the initial performance of the algorithm, in terms of both precision and recall, is encouraging. Although the majority of its recall misses currently occur under the targeted "fuzzy" category (Table 12), it nonetheless provides a useful baseline upon which additional enhancements can build.

5.2 Comparative Evaluation

An analysis like the one presented above invites a discussion of comparative evaluation. The most meaningful comparison to examine is that of idf-based deduping techniques (addressed in the previous section) and well-known alternatives such as shingling [3], in terms of both timing and effectiveness. It is significant to mention that when we incorporate features such as doc_length and hash_key into the digital signature, they are selected to minimize the impact on overall performance. That is, we select a range (of length) and a hash (of rarest terms) such that no duplicates would be lost by their introduction; they serve strictly to reduce the computational cost of comparisons. For this reason, the comprehensive trials conducted by Chowdhury, et al. [7]. provide relevant comparative insights. They examined how idf-based signature approaches to deduping perform relative to selective windowing techniques such as shingling. They determined that given identical data, an optimized idf fingerprint approach is nine times faster than shingling (six times faster than super shingling) when run against the 2 GB NIST Web collection (on a Sun ES-450) [15].

In terms of actual deduplication effectiveness, because shingling does not cover every portion of a textual document and is not sensitive to the rareness of participating terms, it consistently under identified duplicates in a diverse duplicate set constructed from TREC's *Los Angeles Times* sub-collection (which consists of 10 duplicate sets of 11 documents each) [30]. This result occurred as shingling produced more than the optimal number of duplicate sets when processing the automatically generated test collection. Although both approaches use principled techniques, a key distinction between them is that shingling relies upon undiscriminated strings of tokens (shingles) as its representatve content (discussed in Section 2.3). By contrast, the idf-based algorithms distinguish between richer, rarer content-bearing terms and those which are not. This characteristic appears to be one of the chief shortcomings of shingling and a strength of idf-based approaches like ours.

It is worth noting, however, that shingling was devised for very large heterogeneous content sets like those found on the Web, whereas algorithms like that reported on here, along with their associated heuristics for computational efficiency, were designed with a dedicated and more circumscribed domain in mind, while also placing a premium on precision. Measures to extend and improve our algorithm, in terms of leveraging the named entities found in transactional documents, are addressed in Section 7.

5.3 Accuracy and Confidence Levels

It is important to note that our evaluation of the algorithm's results on our test set provides only an approximation of its true accuracy. After all, we applied our algorithm to a combined sample set of nearly 1,000 clauses out of a collection of over 80,000. A reasonable question is thus–how good of an approximation is this? Stated differently, what is our confidence level that the performance measures on this

[12] In an IR context, the percentages presented correspond to recall. By contrast, 94/109 (86.2%) and 81/88 (92.0%) correspond to precision (cf: Table 13).

set reflect true accuracy on the complete set? Mitchell has addressed this problem in the context of Machine Learning [22]. For a collection C, $error_C$ can be defined as the ratio of false positives and false negatives in the algorithm's results on C. Our evaluation test set of sample S produces $error_S$. Mitchell assumes that the probability of having a specific ratio of errors (r) is approximated by a normally distributed random variable with a mean $error_S$ and standard deviation:

$$\sigma_{error_S} = \frac{\sigma_r}{|S|} \approx \sqrt{\frac{error_S(1 - error_S)}{|S|}}$$

where $|S|$ is the size of the sample. The true error can be viewed as drawing a bell curve that is centered on the observed error. So with probability $N\%$, $error_C$ is within z_N standard deviations of $error_S$, where z_N is the z-value. In our case, there is a 95% chance of $error_C$ is within 1.96 standard deviations of $error_S$. For instance, for an observed error ratio of 2.3% (22 errors among 942 document-clauses), there is a 95% chance that the error on the full collection is within the range 2.34% ± 0.49%. For 44 errors among 1,884 document-clauses, the interval would be 2.34% ± 0.35%. This analysis likely warrants further investigation since as one moves beyond consideration of a result set consisting of 20 documents, the number of pair-wise comparisons required per query increases exponentially. It would be instructive to determine whether this fanout has any appreciable impact on error rate. In subsequent tests on result sets consisting of approximately 1,000 documents coming from the business domain, we found no deviation in performance.

6. 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 deduplication. Our own exploration addresses a real world replication problem occurring in the transactional law domain. We designed a methodology that invited our clients, both internal and external, to define the scope of the problem, and then commissioned pairs of professional legal assessors to use our working definition together with additional principled methods to construct a test collection in which non-identical duplicates are identified. We have also attempted to validate the decisions of our assessors using a follow-up Kappa analysis. For non-identical duplicate text detection, our applied test collection proved beneficial and the subsequent dedicated trials suggest that a multi-dimensional feature set approach to characterizing and comparing clause-level texts can provide a solid indication of the degree of duplication between two texts. The treatment of its multi-dimensional feature set frees it from reliance upon singular features and permits heuristics to save on more costly comparisons.

7. FUTURE WORK

In subsequent work, we plan to add a layer of Entity Recognition (ER) in order to address the "near identical" duplicates category. Such ER research would include the categorization of entity (e.g., party, organization, location, financial amount, etc.) as well as whether two entities would warrant resolution to a single canonical form. Once such a follow-up process were added, we would be better able to improve the granularity of our existing evaluation measures.

8. ACKNOWLEDGMENTS

We appreciate the duplicate assessment efforts of Rodney Brown, Scott Ratcliffe, Cara Cardinale, Frank Wozniak, Yasmin Alexander, Joanne Rhoton, Lora Thody, Bill Bremer, Elizabeth Randisi, Stephanie Harth, Kevin Duerinck and Lisa Kless. We thank Ely Razin and Kingsley Martin for their invaluable contribution of domain expertise. We are also grateful to George May, Anudeep Parhar and Matt Canavan who supported our non-identical duplicate research. And lastly, we acknowledge the assistance of Bart Matzek and Doug Heger in handling computability and real-time processing issues in the production environment.

9. REFERENCES

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

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

[3] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the Web. In *Proceedings of the Sixth Int'l World Wide Web Conference (WWW6 '97) (Santa Clara, California)*, pages 391–404. Elsevier Science, April 1997.

[4] Robert Burgin. Variations in relevance judgments and the evaluation of retrieval performance. *Information Processing and Management*, 28(5):619–627, 1992.

[5] Jean Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[6] Surajit Chaudhuri, Venkatesh Ganti, and Rajeev Motwani. Robust identification of fuzzy duplicates. In *Proceedings of the 21st International Conference on Data Engineering (ICDE05)*, page 12, Tokyo, Japan, April 2005. IEEE Computer Society.

[7] Abdur Chowdhury, Ophir Frieder, David Grossman, and Mary Catherine McCabe. Collection statistics for fast duplicate document detection. *ACM Transactions on Information Systems (TOIS)*, 20(2):171–191, April 2002.

[8] Jack G. Conrad, Xu S. Guo, and Cindy P. Schriber. Online duplicate document detection: Signature reliability in a dynamic retrieval environment. In *Proceedings of the 12th Int'l Conference on Information and Knowledge Management (CIKM'03) (New Orleans, LA)*, pages 443–452. ACM Press, Nov. 2003.

[9] Jack G. Conrad and Cindy P. Schriber. Managing déjà vu: Collection building for identifying non-identical duplicate documents. In *Journal of the American*

Society of Information Science and Technology (JASIST), volume 57(7), pages 919–930, Hoboken, NJ, May 2006. John Wiley & Sons.

[10] James W. Cooper, Anni R. Coden, and Eric W. Brown. Detecting similar documents using salient terms. In *Proceedings of the 11th Int'l Conference on Information and Knowledge Management (McLean, Virginia)*, pages 245–251. ACM Press, Nov. 2002.

[11] Dennis Fetterly, Mark Manasse, and Marc Najork. On the evolution of clusters of near-duplicate Web pages. In *Proceedings of the First Latin American Web Congress (Santiago, Chile)*, pages 37–45. IEEE, Nov. 2003.

[12] Dennis Fetterly, Mark Manasse, and Marc Najork. Detecting phrase-level duplication on the World Wide Web. In *Proceedings of the 28th Annual Int'l Conference on Research and Development in Information Retrieval (SIGIR '05) (Salvador, Brazil)*, pages 170–177, Aug. 2005.

[13] Dennis Fetterly, Mark Manasse, Marc Najork, and Janet Wiener. A large-scale study of the evolution of Web pages. In *Proceedings of the Twelfth Int'l World Wide Web Conference (WWW '03) (Budapest, Hungary)*, pages 669–678, May 2003.

[14] Stephen P. Harter. Variations in relevance assessments and the measurement of retrieval effectiveness. *Journal of the American Society for Information Science*, 47(1):37–49, 1996.

[15] David Hawking, Ellen Voorhees, Nick Craswell, and Peter Bailey. Overview of TREC-8 Web track. In *The 8th Text REtrieval Conference*, pages 131–148. NIST, 2000. Retrieved from http://trec.nist.gov/pubs/trec8/papers/web_overview.pdf.

[16] Nevin Heintze. Scalable document fingerprinting. In *Proceedings of the Second USENIX Electronic Commerce Workshop (Oakland, CA)*, pages 191–200, Nov. 1996.

[17] Aleksander Kołcz, Abdur Chowdhury, and Joshua Alspector. Improved robustness of signature-based near-replica detection via lexicon randomization. In *Proceedings of the Tenth Int'l Conference on Knowledge Discovery and Data Mining (SIGKDD '04) (Seattle, WA)*, pages 605–610. ACM Press, Aug. 2004.

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

[19] Daniel Marcu. *Personal Communication*. Information Sciences Institute (ISI), University of Southern California, Los Angeles, CA, April 24, 2002.

[20] Daniel Marcu. *Personal Communication*. Information Sciences Institute (ISI), University of Southern California, Los Angeles, CA, Dec. 16, 2003.

[21] Microsystems. D3 – Dynamic Document Drafting. www.microsystems.com.

[22] Tom M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[23] Seung-Taek Park, David M. Pennock, C. Lee Giles, and Robert Krovetz. Analysis of lexical signatures for finding lost or related documents. In *Proceedings of the 25th Annual Int'l Conference on Research and Development in Information Retrieval (SIGIR '02) (Tampere, Finland)*, pages 11–18. ACM Press, Aug. 2002.

[24] Mark Sanderson. Duplicate detection in the Reuters collection. *Technical Report (TR-1997-5)*, 1997.

[25] Saul Schleimer, Daniel S. Wilkerson, and Alex Aiken. Winnowing: Local algorithms for document fingerprinting. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD '03) (San Diego, CA)*, pages 76–85. ACM Press, June 2003.

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

[27] Sidney Siegel and N. John Castellan, Jr. *Nonparametric Statistics for the Behavioral Sciences*, chapter 9: Measures of Association and Their Tests of Significance, pages 284–289. McGraw Hill, Boston, 1988.

[28] Teradata. a division of NCR. www.teradata.com.

[29] Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, 36:697–716, 2000.

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

[31] Yi Zhang, Jamie Callan, and Thomas Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02) (Tampere, Finland)*, pages 81–88. ACM Press, Aug. 2002.