An accurate copy detection server for digital documents

Mohamed A. Ismail, Amani A. Saad and Ghada H. Badr
Computer Science and Automatic Control Dept., Faculty of Eng., Alexandria University
Alexandria 21544, Egypt

The number of networked users has increased rapidly with the widespread proliferation of computers and networks. If the Internet is any indication, the number of people who have started using online services has increased dramatically in recent years. Two types of users can be found on the Internet: searchers for information and content publishers. Two related problems arise for both users, which are: the Duplicate Detection in Information Retrieval or Information Dissemination Systems problem, and the Copy Guarantees for Digital Publishers problem. The work in this paper is motivated by the need for a new Copy Detection Approach that can be used to solve both of the previous problems and can give accurate results. A new approach is proposed Overlap Sentence Copy Detection Server (OS-CDS), implemented and tested with respect to accuracy and performance issues. Several experiments are made and the results are compared to the results of previous work in this field.

يقترح هذا البحث نظام جديد ودقيق للكشف عن النسخ بين الوثائق الرقمية. ويخدم هذا البحث نوعين من مستخدمي شبكة الإنترنت وهما الباحثين عن المعلومات والناشرين. ويتم ذلك عن طريق حل مشكلتان وهما: مشكلة الكشف عن تكرار النسخ في أنظمة استرجاع أو بث المعلومات، ومشكلة ضمان عدم نسخ الوثائق الرقمية. وقد تم تطبيق هذا النظام واختباره من حيث الدقة والأداء. كما أجريت العديد من التجارب والمقارنات بنتائج الأعمال السابقة فى هذا الحقل التي أثبتت دقة هذا النظام وتفوقة على النظم السابقة.

Keywords: Copy detection server, Duplicate removal, Information dissemenation, Information retrieval

1. Introduction

Two related problems arise for both searchers and publishers on the Internet. These are: *Duplicate Detection* in information Retrieval or information dissemination systems, and *Copy Guarantees* for digital publishers.

1.1. Duplicate document detection problem

A suite of tools has emerged for network information finding and discovery; e.g., wide Area Information Service (WAIS) and World Wide Web (WWW). However, these new tools have one important missing element. They provide a means to search for existing information, but lack a mechanism for continuously informing the user of new information. The exploding volume of digital information makes it difficult for the user to keep up with the fast pace of information generation. Instead of making the user go after the information, it is desirable to have information selectively flow to the user [1, 2, 3, 4].

An information dissemination service helps to maintain an ongoing awareness of what material is available on a given set of topics. Because new documents are rarely indexed in a single location, someone wishing to be familiar with a given topic will have difficulty finding new documents of interest as they appear. Solutions attempt to use sample documents that users classify as relevant or not relevant to their interests and to generate a profile for use in finding new documents of interest[2, 5]. The user subscribes to an information dissemination server by submitting profiles that describe his interests. He then passively receives new, filtered information. A profile is typically made up of a number of keyword queries. The system collects new documents from underlying sources, matches them against user profiles, and routes relevant information to users. Complementary to traditional search mechanisms, information dissemination helps users cope with information overload.

In such information systems, one of the many challenging problems is the proliferation of redundant information or duplicate documents [5].

1.2. Copy guarantees problem for digital publishers

The WWW creates new problems for digital content publishers [6]. Once a customer has bought some goods and paid for them, the merchant has to deliver the content. In case of digital content, the merchant may use some information exchange protocol such as HTTP or e-mail to deliver the goods to the customer. However, once the provider delivers the digital content, the customer may offer this content on his web site, post it to a UseNet newsgroup or mail it to friends for a reduced price or perhaps even for free. The publisher will lose revenues due to reduced sales, if other potential customers start accessing the content from this *cyber-pirate*. Cyber-piracy [6] is a major problem even if the publisher does not make the content available digitally, but if the content can be digitized.

Content publishers currently face a dilemma. On one hand, they would like to use the web to (1) tap into the "impulse buyer" market that relishes immediate access to content, (2) make higher profits due to lower distribution and delivery costs, and (3) increase sales by attracting a larger customer base. On the other hand, they will lose revenues due to cyber-piracy and the proliferation of web sites with illegal copies.

Commercial publishers may be reluctant to offer documents in a *Digital Library* if their documents can be easily retransmitted into *UseNet* newsgroups, or offered on alternate ftp or web sites for free [7].

Therefore, a critical problem that needs to be addressed before Digital Libraries are widely used, is the prevention or detection of illegal copies (full or partial).

The work in this paper is motivated by the need for a new Copy Detection Approach that can be used to solve both previously illustrated problems.

The rest of the paper is organized as follows: Section 2 presents some definitions. Section 3 discusses different recent approaches to solve each problem. Section 4 presents the details of a Copy Detection Server CDS and different design issues. Section 5 presents the Proposed CDS (OS-CDS). The proposed architecture is illustrated and the proposed Data Structures and Algorithms are explained. Section 6 presents some *experiments*, explains the data set used and shows how the accuracy of the proposed system is calculated. Section 7 discusses the *results* of the experiments, and finally, Section 8 gives the conclusions and suggests some future Work.

2. Definitions

In this section, some definitions [8] are given for some terms that are relevant to *Information Retrieval* IR in general:

Document: A piece of information the user may want to retrieve. This could be a text file, a WWW page, a newsgroup posting, a picture, or a sentence from a book.

Collection: A group of documents that a user wishes to get information from.

Information Filtering: Given a large amount of data, return the data that the user wants to see.

Information Retrieval: The study of systems for indexing, searching, and recalling data, particularly text or other unstructured forms.

Information Need: What a user really wants to know. A query is an approximation to the information need.

Relevance: An abstract measure of how well a document satisfies the user's information need. Ideally, a system should retrieve all of the relevant documents for you. Unfortunately, this is a subjective notion and difficult to quantify.

Precision: A standard measure of IR accuracy. It is defined as the number of relevant documents retrieved divided by the total number of documents retrieved.

Recall: A standard measure of IR accuracy. It is defined as the number of relevant documents retrieved divided by the total number of relevant documents in the collection.

Term: A single word or concept that occurs in a model for a document or query. It can also refer to words in the original text.

Inverted File: A representation for a collection that is essentially an index. For each word or term that appears in the collection, an inverted file lists each document where it appears.

Stemming: The process of removing prefixes and suffixes from words in a document or query in the formation of terms in the system's internal model. This is done to group words that have the same conceptual meaning, such as walk, walked, walker, and walking. Hence, the user does not have to be specific in a query. The Porter Stemmer [9] is a well-known algorithm for this task.

Stopword: A word such as preposition or article that has a little semantic content. It also refers to words that have high frequency across a collection.

Vector Space Model: A representation of documents and queries where they are converted into vectors. The features of these vectors are usually words in the document or query, after stemming and removing stopwords. The vectors are weighted to give emphasis to terms that exemplify meaning, and are useful in retrieval. In retrieval, the query vector is compared to each document vector. Those that are the closest to the query are considered to be similar, and are returned. There are many systems that use the vector space model [10, 11].

Signature File: A representation of a collection where documents are hashed to a bit string. This is essentially a compression technique to permit faster searching.

3. Previous work

This Section presents different recent approaches to solve each of the problems discussed in Section 1.

3.1. Duplicate detection in information dissemination problem

In this age of information overload, an important value-added service that should be performed by search engines and databases is to remove duplicates of articles before presenting the results of search to users. An information dissemination system that automatically removes duplicate news articles is the SIFT server [1]. Duplicates or near duplicates of documents may exist due to multiple formats or because of replication of articles by cross-posting for newsgroups, forwarding of articles etc. It is shown in [5] how a Copy Detection Blackbox CDB may be used to automatically remove multiple copies of the same article, and how a user may dynamically discard certain classes of articles that have sufficient overlap.

A promising approach to building a CDB is to use document-clustering techniques. In the Scatter/Gather clustering approach [12], users can dynamically recluster documents based on topics they wish to pursue and those they wish to discard.

Another approach to build the CDB is to use the Registration based Copy Detection Server CDS. The first approach is session-specific while the latter is more user- specific.

3.2. Copy guarantees for the digital publishers problem

There are two main philosophies for addressing this problem: prevention and detection.

Copy prevention [13] schemes include physical isolation of the information (e.g., by placing the documents on stand-alone CD-ROM systems), using special purpose hardware, or active documents which are documents encapsulated by programs. It is found that prevention techniques may be cumbersome, may get in the way of honest users, and may make it difficult to share information. Furthermore, prevention schemes are not always safe since documents may be recorded by using software emulators [14].

Copy detection schemes do not place restrictions on the distribution of documents, but detect illegal copies. Detection schemes fall into two categories, signature based and registration based. In signature based schemes, a signature is added to the document, and this signature can be used to trace the origins of the document. For example, one popular approach is to incorporate watermarks such as word spacings and checksum into documents [7].

Signature based schemes have two major weaknesses [13]: (a) the signatures often can

be removed automatically, leading to untraceable documents, and (b) they are not useful for detecting partial overlap. For these reasons we advocate registration based copy detection schemes.

With registration based [13] copy detection schemes users (such as authors, publishers or users of information dissemination systems) register their valuable digital documents at the server. These documents are "chunked" (broken) into primitive units such as words, sentences or paragraphs, and these chunks are stored in a repository. Query documents such as NetNews articles, and documents available at ftp sites and web pages are chunked into the same primitive units as registered documents. These chunks are compared with the set of registered chunks for overlap, and in case of sufficient overlap, in case of digital libraries, the owner of the registered copy is notified of the location of possible illegal copies. In case of information dissemination systems, the document is not sent to the user again.

Registration servers can be implemented in a variety of ways. The major design issues are [7]:

1. The chunking strategy (e.g., small vs. large chunks, overlapping vs. non-overlapping chunks).
2. The data structures used to store chunks and to find matches between registered and queried chunks.
3. The decision function used to determine when a queried document has substantial overlap with the registered one.

4. Copy detection server CDS

In this section we begin to discuss the details of the CDS and discuss some issues that need to be considered while building a CDS. These are the server architecture, storage data structures and the textual units used for comparison. Finally, the overlap measures used to calculate the similarity between two documents are illustrated.

4.1. The CDS architecture

Content publishers register their digital content into the CDS registry [6]. The CDS then accesses publicly accessible popular web sites, identifies potential copies and notifies the corresponding content publishers. Documents that are to be registered are chunked and inserted into the repository. We define chunking of a document to be the process of breaking up a document into more primitive units such as sentences, words, or overlapping sentences. New documents that arrive are chunked into the same units and are compared against the pre-registered documents for overlap.The two main components of such a CDS architecture are [6]:

1. Crawler: The CDS employs a variety of crawlers to retrieve digital documents from the web.
2. Comparator: Assume we have a boolean predicate Similar(p, q) that checks if two digital objects p and q are similar. The comparator module supports the following operation: find operation: Given a query object p, the find operation finds each registered object r such that Similar(p, r)=True. For example, a Professor can check if a Student report is plagiarized from some registered document, using this operation.

4.2. Inverted index storage

An inverted index structure for storing chunks of registered documents is used. It is an index of chunks in the vocabulary, i.e., the set of occurrences of all distinct chunks in the registered documents. It is constructed and maintained at registration time. Each entry for the chunk points to a set of postings that indicate the documents where the chunk occurs. Every posting for a given chunk has docnum, which is a unique identifier of a registered document in which the chunk exists.

4.3. Units of chunking

The unit of chunking chosen for copy detection is critical since it shapes the subsequent overlap search cost and storage cost.

The bigger the chunking unit, the lower the probability of matching unrelated documents. On the other hand, the bigger the chunking

unit, the higher the probability of missing actual overlaps. The larger the chunk, the higher the potential number of distinct chunks that will be stored. Hence, we see that the potential size of the chunk index is higher when the chunking unit chosen is larger. Of course, the number of postings per chunk is larger when the chunking unit is small (as in words). However, a small chunking unit increases locality. That is most documents will have a relatively small working set of words rather than sentences.

4.4. Overlap measures

In the past few years, a few research CDS prototypes, for example: SIFT [1], COPS [14], SCAM [6], have been developed. The underlying approaches of these systems [6] can be classified into (1) chunking based and (2) Information Retrieval IR based techniques.

In chunking based technique, each text document is broken into a set of chunks, i.e., smaller text units. These chunks are usually then compressed (e.g., by hashing) and sampled for performance reasons (e.g., lower storage requirements). Two documents are then compared by means of the number of compressed chunks they share. For instance, if two documents share more than some minimum threshold of chunks, they are defined to be similar. The possible chunking strategies discussed in [6] are: Sentence, n-gram, Hashed break points measure, and word chunking strategies.

As for IR-based techniques, when queries arrive from users, the query set of keywords is compared against the documents, and some measure of relevance between the document and the query is obtained. Similarly, a metric was needed to measure the overlap between an incoming document and a pre-registered document. The most important models that were established are: the Vector Space Model VSM [13] and the Relative Frequency Model RFM [6].

5. The proposed copy detection server: OS-CDS

The objective of this work is to try to find a more Accurate Copy Detection Approach that combines the advantages of both of the Sentence chunking strategy and the Word chunking strategy. That is a new chunking based technique is proposed that tries to achieve more accurate results. The Proposed idea is:

- Try to get the smallest value for the false positive error of the sentence chunking schemes by using sentences as a chunking strategy.
- Try to get the smallest value for the false negative error of the word chunking schemes by detecting overlaps in words between sentences.

The Data Structures of the SCAM [6] is modified in order to allow us to detect sentence overlaps.

The system has two phases: The probing phase and the registration phase. In the probing phase the new document is tested against the registered documents for overlap. In the registration phase, if the new document is accepted (i.e., not a copy of any document in the system), the new document is registered in the system.

In this Section, the Data Structures used are explained. Then, the algorithms that use these Data Structures to calculate the overlap percentages between two documents are presented. Finally, the overlap metric and the decision function used are presented.

5.1. Data structures

In Fig. 1 we present the main data structures used in the proposed system. The Data structures used can be classified into Persistent data structures and Transient data structures. The Persistent data structures compose the repository of our registration based CDS that stores the chunks and all the information needed about a registered document. These data structures are: The Mapping structures, the Sentence Inverted Index SII, and the Word Inverted Index WII. The Transient data structures are used for the calculation purpose and are not stored in the repository. These structures are: The Merging structures, the Relevance Array RA, and the Overlap Array OA [16, 17, 18, 19, 20].

5.1.1. Mapping structures

There are three mapping tables:

1. Map Document MD Table: This table stores the mapping from the actual file name or URL of the web page to a unique document identifier used in postings.
2. Map Document Sentence Count MDSC Table: This table stores the mapping from the unique document identifier to the actual document file name or URL of the document web page. Also, the number of sentences in the document is stored. Other data for the document is stored in the table such as the author name, date of publishing and any other information needed about the document.
3. Map Sentence MS Table: This table stores the mapping from the actual sentence to a unique Sentence Identifier SId used in postings.

5.1.2. Sentence inverted index SII

In Fig 2, the SII of the SCAM [13], in which the chunks (sentences) of registered documents are stored, is modified in order to allow us to detect the common sentences between the query document and registered documents, as well as, the sentences which overlap. In this structure, the posting indicating the nonzero occurrence frequency of sentences in documents is maintained.

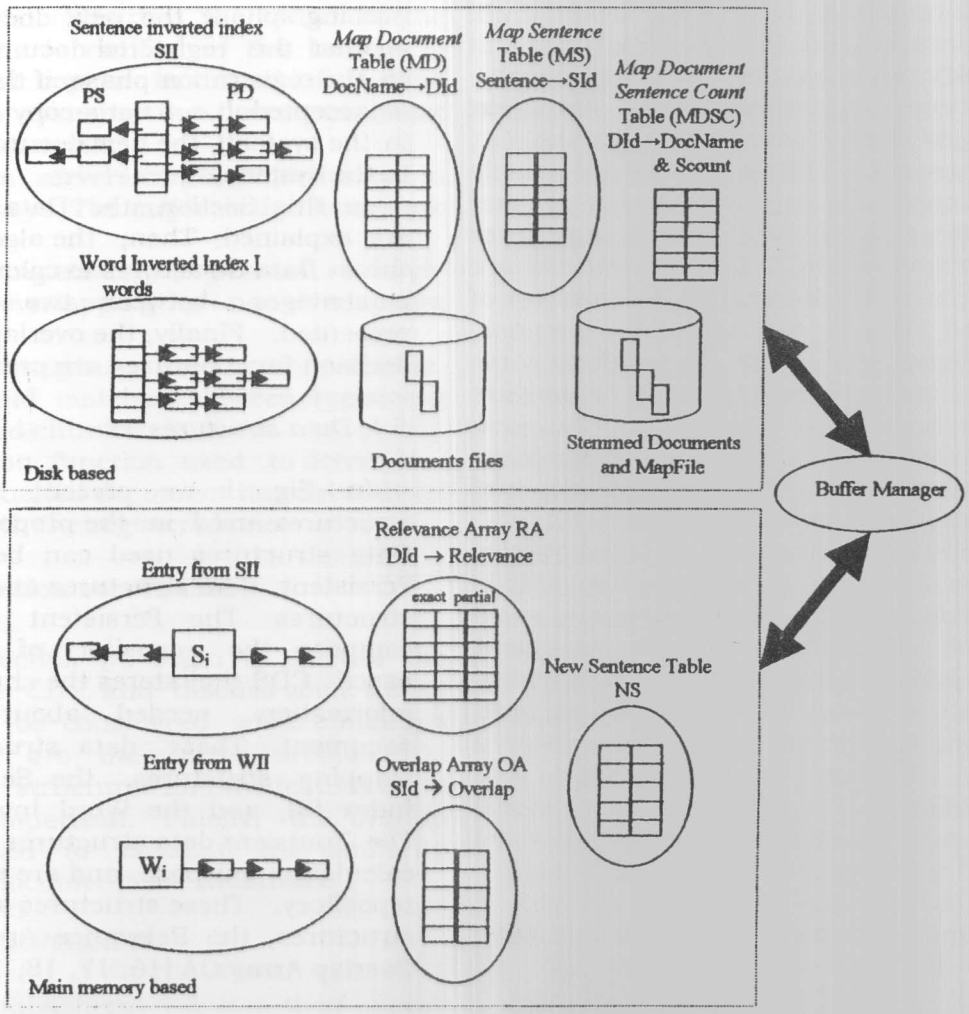

Fig. 1. General Data Structures used in the System.

Another posting is added indicating the nonzero overlap of the sentence in that entry with other sentences of the registered documents, i.e. only sentences in the index. Let the posting for documents to be PD, and the posting for overlap sentences to be PS. For instance, a PD may be of the form ("ABCD", 11) to indicate that the sentence "ABCD" found in a document with a unique identifier 11. A PS may be of the form ("ABCD", "ACD", 75) to indicate that the sentence "ABCD" overlaps with sentence "ACD" with 75%. The sentences identifiers are used in the postings instead of the sentence itself in order to decrease the storage requirements.

The number of words in the sentence is stored at the beginning of the PS to be used in the calculation of the overlap percentage of the sentence with other new sentences.

5.1.3. Relevance array RA

In SCAM [13] overlaps between a query document and a set of registered documents is usually kept track of by incrementing some score whenever a common sentence is found which denotes the Exact matches. For each document that overlaps with the query document with at least one sentence, an entry is made in the Relevance array RA for that document. For instance, say a query document Q has 5 sentences in common with R1 and 7 sentences in common with R2, where R1 and R2 are some registered documents. Two entries will be found in the Relevance array for R1 and R2 with scores 5 and 7 respectively. When a new sentence is found in common with R2, the score of R2 is incremented to 8.

The modification in that structure is as follows: A new entry is added that keeps track of some score indicating the accumulated overlap percentages of sentences in the registered document corresponding to that entry in the RA which denotes the Partial match. These sentences overlap with sentences in the query document with a percentage (a fraction between 0 and 1) over a certain percentage threshold. This threshold, which is set by the user, indicates the overlap percentage over which the sentence is considered an overlapping sentence. For instance, say Q has an overlapping sentence S1 with R1 with percentage 60% and another overlapping sentence S2 with R1 with percentage 90% and the overlap threshold is set to 65%. The overlap entry for R1 is only incremented by 0.9.

5.1.4. Word inverted index WII

The nonzero overlap percentage of any sentence is calculated and stored in the SII within the PS's entries. But, how are these overlap percentages calculated? The WII is used for this reason.

If a new sentence is found in the new document, the overlap percentage of this new sentence with each registered sentence in the system has to be calculated. The trivial way is to calculate this overlap percentage with each registered sentence one by one. But, this way will take a lot of time and will not be efficient. That is why an inverted index is used in the same way as the overlap between documents is calculated. The Index is based on words found in the registered sentences of the system.

In this structure, the postings indicating the (nonzero) occurrences of words in registered sentences are maintained, These are the only sentences stored in the persistent sentence inverted index. For instance, a posting may be of the form ('A', 17,50) to indicate that the word 'A' is found in a sentence with ID equals 17 and in a sentence with ID equals 50. The words stored in this index are only those words that appear in the registered sentences. Fig. 3 shows an example.

5.1.5. Overlap array OA

Overlaps between a new sentence and a set of registered sentences are usually kept track of, by incrementing some score whenever a common word is found. For each registered sentence that overlaps with the new sentence with at least one word, an entry is made in the OA for that sentence. For instance, say a new sentence S has two words in common with S1 and four words in common with S2, where S1 and S2 are some registered sentences. Two entries will be found in the OA for S1 and S2 with scores two and four respectively.

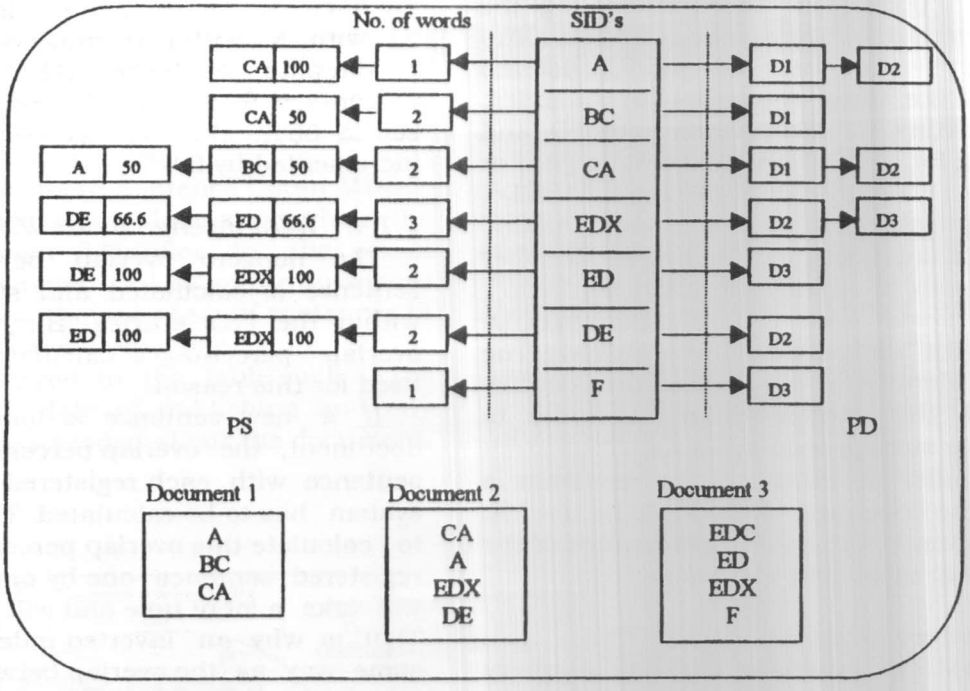

Fig. 2. An example of the sentence inverted index SII.

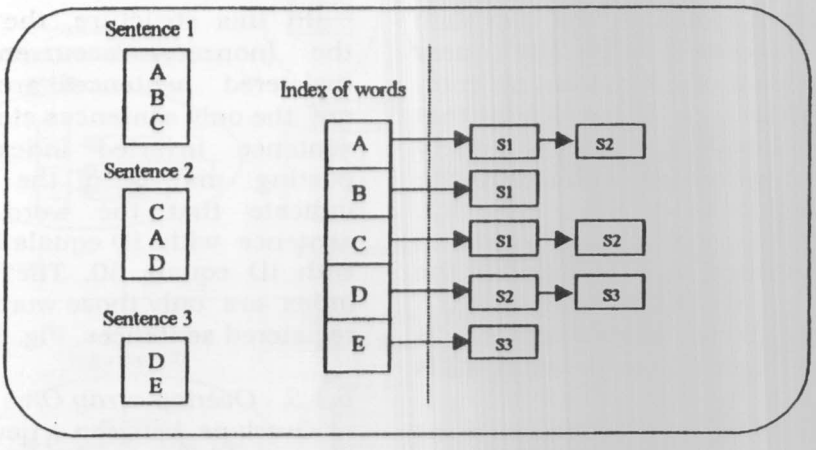

Fig. 3. An Example of word inverted index WII. (A,B, C, D, and E are words in sentences 1,2, and 3,).

When a new word is found in common with S2, the score of S2 is incremented to five. These scores will be used after that to calculate the overlap percentages of S with other sentences in the OA.

5.1.6. Merging structures

These structures are used as intermediate storage for data that will be needed in the registration phase as a result of the probing phase to update the Persistent data structures. The merging structures are:

- The New Sentence Table NS
- The MapFile.

The New Sentence Table NS is used to store the data needed for new sentences found in a new document during the probing phase. This data is used to update the SII and the MS Table during the registration phase. It is stored in the table without order because it

will be processed one by one. NS is implemented as an array in memory.

The MapFile is a file that is used to store sequentially the SId's of the sentences found in the new document and already registered in the system. The SId's stored in this file are used to update the SII during the registration phase if the new document is found not to be a copy. In this way, the MS Table does not need to be accessed during the registration phase so the system time is saved.

5.2. Algorithms

Fig. 4 describes the main algorithm for any registration based copy detection scheme. In such schemes, users (such as authors, publishers or users of information dissemination systems) register their valuable digital documents at the server. These documents are "chunked" (broken) into primitive units, which are sentences in our system, and these sentences are stored in a repository. Query documents such as Netnews articles, and documents available at ftp sites and web pages are chunked into the same primitive units as registered documents. These sentences are compared with the set of registered sentences for overlap, and in case of "sufficient" overlap the document is not sent to the user again.

The proposed system have four main algorithms, which are: The Document Chunking, the Probing Phase, the Registration phase, and the Sentence Overlap algorithms.

5.2.1. The document chunking algorithm

In this part, procedures are implemented to automatically extract sentences from a text document. These procedures are applied to each new document in the same order as follows:

1. Change all letters to lowercase.
2. Remove non-letters except:
   Space character: word terminator
   { . , ? , ! , : , ; }: sentence terminators→{ . }
3. Remove stopping list [21].
4. Remove Excess spaces.
5. Apply Porter Stemming Algorithm: algorithm for suffix stripping [9].
6. Remove repeated words in the sentences.
7. Remove repeated sentences in the document.
8. Remove sentences with one word.

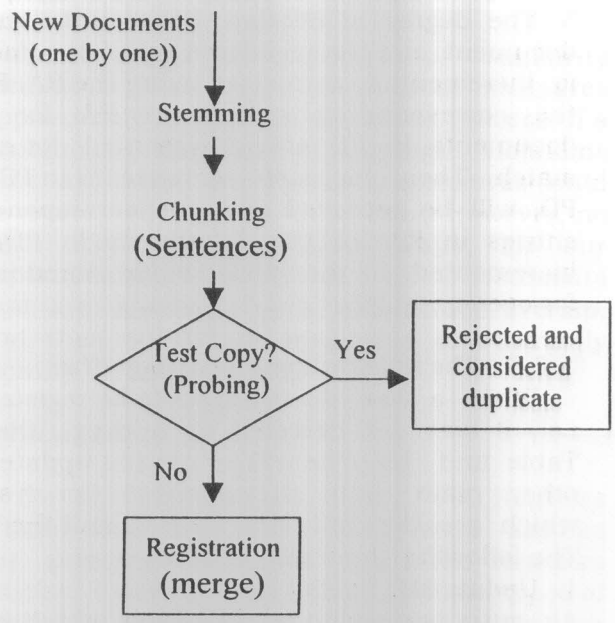

Fig. 4. Registration based copy detection algorithm.

5.2.2. The probing phase algorithm

When a query document is to be checked against a repository, it will be divided into sentences. Each sentence S in the query document will be processed one by one as follows:

1. The MS Table is checked to see if the sentence S is registered in the system or not.
2. If S was registered before, it will be converted to an integer value representing the Sentence Identifier SId. This SId is stored in the MapFile to be used after that in the registration phase to update the SII.
3. If S is a new one, i.e., not registered in the system before, a new SId will be given to it. S and its SId are both stored in the NS Table to be processed after that to calculate the overlap percentage of S with all other registered sentences in the system and get its corresponding PS list using the Sentence Overlap algorithm.
4. If S was registered before, the corresponding entry for its SId in the SII will be retrieved in memory. Both postings, PS and

PD, will be retrieved to find the set of documents containing that sentence (exact match) and to find the set of sentences that overlap with that sentence (overlap match).

5. The degree of overlap between the query document and the registered set of documents is incrementally computed using the RA. First the corresponding entries in RA for the documents in PD are incremented for *exact* match. Then, for each sentence S_i in PS, its PD_i will be retrieved and the corresponding entries in RA for the documents in PD_i are incremented by the overlap value stored in PS for *overlap* match.

5.2.3. The registration phase algorithm

When a new document is to be registered, i.e., it was not detected as a copy, the NS Table and MapFile will be used to update the other main data structures in the system which are the MS Table, the SII and the WII. The following steps will be done:

1. Update MD Table:

An entry is made in the MD Table with the file name of the document or the URL. A unique integer, Document Identifier DId, is generated and stored which maps to the file name of the document or the URL.

2. Update MS Table:

For each new sentence in the NS Table, an entry is made in the MS Table to store the sentence and its SId.

3. Update SII:

- For each new sentence in the NS table, an entry is made in SII. A new file is created. The file name will be "PS+SId". In this file the nodes of the PS list of the new sentence will be stored sequentially. These nodes contain the SId's of the registered sentences and the overlap percentages with which the new sentence overlaps with the registered sentences. The number of words will be stored at the beginning of the file.
- Then, for each registered sentence S in the PS list of the new sentence, a new node is added in the corresponding PS of S which contains the new sentence SId and the overlap percentage with which S overlaps with the new sentence. Thus, when two sentences overlap their SId's and overlap percentage will be added to the PS lists of each other. This overlap percentage of S with the new sentence will be calculated from the overlap percentage of the new sentence and S as follows:

Let WS be the number of words in S retrieved from the corresponding PS of S, WNS be the number of words in the new sentence stored in NS Table, and OP be the overlap percentage with which the new sentence overlaps with S. Then the overlap percentage with which S overlaps with the new sentence will be:

OP*WNS / WS.

- For each new sentence in NS Table, a new PD file is created which contains the new DId. The file name will be "PD+SId".
- For each SId in MapFile, a new node is added to the corresponding PD list by appending the DId value to the end of the corresponding PD file with name "PD+SId". In this way the new document is added to the registered sentences PD lists that are previously stored in the system.

4. Update WII:

For each new sentence in the NS Table, and for each word w in this sentence, the SId of the new sentence will be added to the posting of the word w. This will be implemented by appending SId to the end of file with name the same as w. If w is a new word in the system, a new file will be created in which the new SId will be stored.

5.2.4. The sentence overlap algorithm

When a new sentence is to be checked against other registered sentences for overlap, the postings in the persistent WII are retrieved for each word in the new sentence one by one to find the set of registered sentences that overlap with that sentence. The number of words in common between the new sentence and the set of registered sentences with which it overlaps is incrementally computed using the Overlap Array OA.

The number of words stored in OA will be used to calculate the overlap percentage as follows: Let WNS be the number of distinct words in the new sentence and WC be the number of words in common between the new sentences NS and a registered sentence S. Then the overlap percentage OP between NS and S will be: **OP=WC/WNS**.

5.3. Overlap metric and decision function

When a new document d arrives to the system, it is tested to see if it is a copy of one of the registered documents. The result of the system for the processing of d will be a set of registered documents R with which it may have overlap.

The overlap of d with each document r in the set R will be represented by:

- the number of Exact Match EM sentences between d and r , and
- the Overlap Match OM, which are fractions between 0 and 1 denoting the percentage of overlap between a sentence in d and another in r. These fractions are considered when their values are above a certain Sentence Threshold Sth that is set by the user.
- the Sum of Overlap Match SOM, which represents the accumulated value for OM with respect to all sentences in d that overlap with sentences in r.

To test whether a document r in set R is a copy of d or not, we need an overlap metric and a decision function. The overlap metric uses the values, EM and OM, for document r to give us the overlap percentage OP with which document d overlaps with r. Then the decision function tells us if OP indicates that d is a copy of r or not.

Let Nd be the number of sentences in d, and Nr be the number of sentences in r. Then, the metric that measures the overlap between an incoming new document d and a registered document r in set R will be:

$$\text{For each r in R, } OP_r = \frac{EM + SOM}{\min(Nd, Nr)}$$

Where a minimum function is used to allow us to detect both subset and superset documents.

The decision function $f(r)$ which indicates if document d is a copy of document r or not will be:

$$f(r)=(OP \geq Dth) = \begin{cases} \text{False , if d is not a copy of r} \\ \text{True , if d is a copy of r} \end{cases}$$

Where Dth is the document threshold set by the user according to his requirements. This threshold depends on what type of overlap is targeted. For example, to identify exact copies we can look for pairs of documents with 100% Dth. If we are looking for documents with high overlap, we can look for documents with 60% Dth [7].

6. Experiments

Evaluating the quality of similarity measures is tricky, since these measures approximate a human's decision process. If a benchmark database of copyright violations were available, the similarity measures could be evaluated against this data. However, no such benchmark exists. Thus, for our experiments, we start with a set of documents that are known to have "substantial" overlap, and then see if our measures can correctly identify these cases [23].

6.1. Data Set

For the experiments, about 300 NetNews articles from September 2000 were used as our primary document set. The NetNews articles were chosen [22, 7] as these types of articles "stress-test" copy detection mechanisms for several reasons. NetNews articles are relatively small, so substantial overlap between articles may be a few sentences, making it harder to detect. Also, the articles are informal, so when people copy text into an article, they may copy incomplete sentences. So partial sentences overlap cases will exist so that our system could be tested and evaluated.

6.2. Human classification

The notion of correctness of a copy detection system such as COPS and SCAM depends on what our ultimate goal is. This goal can only be specified as a manual test where a human decides if a pair of documents actually involve plagiarism or substantial overlap. This is not a trivial problem. Indeed, two humans may not agree if two documents are similar. These possible manual tests can be called Document Target Tests DTTs [13]. The goal of a copy detection system is to predict the outcome of these DTTs. There can be four different DTTs: Plagiarized: if a document includes some parts of another article, Subset: if a document is almost

completely included in another document, Copies: if two documents appear to be exact copies, and Related: if two documents appear to have a common thread relating them.

In general, different humans may have different responses to the DTT's since they are subjective. Hence the results described below should be considered in an illustrative rather than in an "absolute" sense.

6.3. Approximating the human classification

The ultimate goal of a copy detection mechanism is to identify pairs of documents that the human end user would consider to have "substantial" overlap[7]. Since the term "substantial" depends on the goals or interests of the human, the DTT's can be approximated as follows [7]:

- Exact Copies: These were articles that were identical or that include all of the text of the original article.
- High Overlap: These were articles that include most of the text of the original article.
- Some Overlap: This is similar to the previous category, except that only small portions of the original article has been included.

This classification was very subjective, but was our best attempt to identify documents that most humans would agree were closely related.

6.4. Empirical accuracy results

To compute the accuracy of the system, we proceeded as follows [7, 23]. We started with three sets of documents: Exact, High and Some. To obtain these sets, we examined the 300 documents in our data set in great detail and chose about 80 documents that had many partial and complete duplicates. The rest of the articles (about 220 documents) were placed into the three sets according to the "degree" of overlap with each of the 80 articles as judged by human from the manual test. These sets are: H_E, H_H and H_S respectively. The objective of a copy detection scheme is then to identify the articles that were selected in the manual classification. In particular, we compare our scheme Overlap Sentences Copy Detection Server OS-CDS with the sentence chunk-based CDS used in SCAM [7] showing the advantages of the proposed system.

Each of these strategies requires a Document threshold Dth to determine when there is substantial overlap [7]. This Dth depends on what type of overlap is targeted. In this paper, the results were examined with the following Dth sets, i.e., the boundaries between Exact, High, and Some were set to be:

- Exact: 100%
- High: 66%, 60%, or 50%
- Some: 33%, 10% or 5%

Then for each of these Dth sets, the three sets of documents S_E, S_H and S_S were calculated for each of the two schemes, for each document in the 80 document set. In the SCAM sentence chunk-based scheme [7], the boundaries are set to be: 100% Exact, 50% High, and 5% Some. In the experiments, all the boundaries will be tested for the best accuracy results.

The Sentence threshold Sth which is used to determine when a certain sentence overlaps with another sentence previously registered in the OS-CDS system is tested to see which value will give more accurate results. The values tested for Sth are: 50%, 60%, 70%, 75%, 80%, 85%, 90% and 100%.

The accuracy of the two schemes is measured by the **false negative** and **false positive** errors [7] computed by comparing S_E, S_H and S_S and H_E, H_H and H_S.

The False negative error denotes the percentage of the documents that the system missed by either not reporting them as overlaps, or by placing them into a lower category, for instance, placing some document in H_E into S_H rather than S_E. They are computed as a percentage of missed documents to the number of overlapping documents expected by human. For instance, if a document has 10 overlapping documents in H_E and the system reports only 8 of those overlapping documents in S_E, the false negative errors will be 2/10 = 20%.

The False positive error for a given article denotes the percentage of documents in some S_X for that article, but not in H_X for the same article, where X = {E, H, S}. These errors are computed as a percentage of documents that are false alerts to the total number of registered documents.

The false positive and the false negative errors are computed for each of the 80 chosen articles, then the average value is calculated.

7. Results

As for evaluating accuracy, 81 experiments are conducted in order to compute the accuracy of OS-CDS for different values of Dth and Sth, and to compare it with the SCAM sentence chunk-based approach. Due to space limitations in the paper, only the best results are presented.

Similar to the SCAM system results, the best Dths values were found to be: 100% for Exact, 50% for High, and 5% for Some.

On the other hand, from the experiments for Sth, the best value for Sth was found to be 80%.

From table 1, the following results were reached:

Table 1
Accuracy for: 100% exact, 50% high, and 5% some

		False negative	False positive
SCAM	Exact	0.268750	0.000000
	High	0.113333	0.001512
	Some	0.028542	0.012965
80% OS-CDS	Exact	0.000000	0.000233
	High	0.037500	0.000581
	Some	0.026875	0.016628

- As for the Exact overlap case, we have a high decrease in False Negative error and the False Positive error is around the same value as SCAM.
- As for the High overlap case, we have a high decrease in False Negative error and a high decrease in False Positive error

As for the Some overlap case, both systems are approximately the same.

As for the Some Overlap case, the manual test was very difficult to obtain. Some documents that we decided not to be Some Overlap cases, may be actually decided by other users to be Some Overlap cases. This difficulty is due to the small percentage (5%) taken for Dth. If we increase Dth to be 10% or even 33%, both error values of the OS-CDS will be much better than SCAM, but, the False Negative error will be higher in value for both systems. Table 2 illustrates these results.

Table 2
Accuracy for some overlap case for other Dths

	Dth	False Negative	False Positive
80% OS-CDS	10 %	0.033125	0.010640
SCAM	10 %	0.034792	0.009128
80% OS-CDS	33 %	0.172083	0.000465
SCAM	33 %	0.194583	0.001105

8. Conclusions and future work

In this paper, a Copy Detection Server CDS that can identify partial or complete overlap between documents is proposed. A prototype implementation of this server, OS-CDS is proposed.

The accuracy of the OS-CDS is tested with respect to the false negative and the false positive errors in nine sets of experiments. The results demonstrate the following three points:

1. The OS-CDS can accurately catch many copies of documents that can deceive the SCAM scheme. In other words, the OS-CDS decreases the false negative error in detecting document copies. This is due to the fact that, it is easy to deceive the SCAM sentence chunk-based system by:
   - changing the order of words in the sentences of a document, or
   - changing even one word in some sentences of a document, or
   - copying only parts of the sentences of a document.
2. The OS-CDS reports good values for the false positive errors with respect to the SCAM sentence chunk-based scheme. This means that, the OS-CDS does not falsely detect many copies as the SCAM word chunk-based system. This is because the OS-CDS uses sentences as a chunk unit.
3. A value of 80% for Sth is the best to be used by the OS-CDS.

Thus, from the first two results, it is shown that the main objective of this work is reached. The objective was designing a new system that makes use of the small false positive error in the sentence chunk-based systems, as the COPS system, and to make use of the small false negative error in a word chunk-based system as the SCAM system by detecting overlaps between sentences themselves.

In the future, several extensions of this work can be done such as; Studying how to use the proposed technique to automatically identify mirrored collections on the web, so that a variety of tasks can be performed more effectively. These tasks include: Crawling, Ranking, Archiving, and Caching and Building a content delivery system that uses copy protection, watermarking, and copy detection to protect digital contents. A content delivery system should be designed and implemented based on the level of security that is required.

References

[1] T. W. Yan and H. Garcia-Molina. "SIFT - A Tool for Wide-Area Information Dissemination". Proceeding of the 1995 USENIX Technical Conference, 177-86, February, (1995).

[2] N. J. Davies, R. Weeks and M. C. Revett. "Information Agents for the Word Wide Web". Lecture notes in Artificial Intelligence 1198, Springer Verlag Publishing Company (1997).

[3] H. S. Nwana. "Software Agents: An Overview". Knowledge Engineering Review. Vol.11, (3), pp. 205-244 (1996).

[4] H. S. Nwana and M. Wooldridge. "Software Agent Technologies". Lecture notes in Artificial Intelligence 1198, Springer Verlag Publishing Company (1997).

[5] T. W. Yan and H. Garcia-Molina. "Duplicate Removal in Information Dissemination". Proceedings of Very Large Database (VLDB'95) Conference, Zurich, Switzerland, September (1995).

[6] N. Shivakumar. "Detecting Digital Copyright Violations on the Internet". Ph.D. Thesis, Department of Computer Science, Stanford University (1999).

[7] N. Shivakumar and H. Garcia-Molina. "Building a Scalable and Accurate Copy Detection Mechanism". In Proceedings of 1st ACM Conference on Digital Libraries (DL'96), Bethesda, Maryland, March (1996).

[8] C. Faloutsos and D. Oard. "A Survey of Information Retrieval and Filtering Methods". Salton and McGill, Introduction to Modern Information Retrieval, McGraw-Hill (1995).

[9] M. F. Porter, "An Algorithm for Suffix Stripping", Program, Vol.14 (3), pp. 130-137 (1980).

[10] Smart system: http://www.cs.jhu.edu/~weiss/projects.html

[11] T. W. Yan and H. Garcia-Molina. "Index Structures for Selective Dissemination of Information Under the Vector Space Model". In Proc. International Conference on Data Engineering, pp. 337-347 (1994).

[12] N. Shivakumar and H. Garcia-Molina. "The SCAM Approach to Copy Detection in Digital Libraries". Department of Computer Science, Stanford University. http://www-db.standford.edu/pub/papers/dlmag.htm

[13] N. Shivakumar and H. Garcia-Molina. "SCAM: A Copy Detection Mechanism for Digital Documents". Proceedings of the 2nd International Conference on Theory and Practice of Digital Libraries, Austin, Texas (1995).

[14] S. Brin, J. Davis, and H. Garcia-Molina. "Copy Detection Mechanisms for Digital Documents". Proceedings of the ACM SIGMOD International Conference on Management of Data, San Francisco, CA, May, pp. 398-409 (1995).

[15] N. Declaris, D. Harman, C. Faloustsos, S. Dumais and D. Oart. "Information Filtering and Retrieval: Overview, Issues and Directions", Basis for Panel Discussion, 26th Annual Conference of the IEEE Engineering in Medicine and Biology Society, Baltimore, MD, November 3-6 (1994).

[16] R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems", the Benjamin/Cummings Publishing Company, Inc. (1994).

[17] E. Horowitz, S. Sahni, and S. Aderson-Freed. "Fundamentals of Data Structures in C". Computer Science Press, W. H. Freeman and Company (1993).

[18] A. L. Tharp. "File Organization and Processing". John wiley and Sons Publishing Company (1988).

[19] S. Holzner. "Visual C++ Programming". Brady Publishing (1993).

[20] T. W. Yan, H. Garcia-Molina. "Index Structures for Selective Dissemination of Information Under the Boolean Model". IEEE Transactions on Database Systems, June (1994).

[21] Stopping-list: (http://mystic.biomed.mcgill.ca/MedinfHome/stem/q2/colorlect10Apr8.html)

[22] M. Horton. "Stanford for interchange of *UseNet* Messages", RFC-850 *UseNet* Project, June (1983).

[23] Gh. H. Badr. "An Accurate Copy Detection Server for Digital Documents". M.Sc. Thesis, Department of Computer Science, Alexandria University (2001).

Received June 7,2001
Accepted July 31,2001