DETECTING SIMILAR HTML DOCUMENTS USING A SENTENCE-BASED COPY DETECTION APPROACH

by

Rajiv Yerra

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree

Master of Science

Department of Computer Science
Brigham Young University
July 2005

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Rajiv Yerra

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date
Yiu-Kai Dennis Ng, Chair

Date
Phillip Windley

Date
Eric Mercer

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Rajiv Yerra in its final form and have found that (1) its format, citations and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Yiu-Kai Dennis Ng
Chair, Graduate Committee

Accepted for the Department

Parris Egbert
Graduate Coordinator

Accepted for the College

G. Rex Bryce
Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

DETECTING SIMILAR HTML DOCUMENTS USING A SENTENCE-BASED COPY DETECTION APPROACH

Rajiv Yerra
Department of Computer Science
Master of Science

Web documents that are either partially or completely duplicated in content are easily found on the Internet these days. Not only these documents create redundant information on the Web, which take longer to filter unique information and cause additional storage space, but also they degrade the efficiency of Web information retrieval. In this thesis, we present a new approach for detecting similar (HTML) Web documents and evaluate its performance. To detect similar documents, we first apply our sentence-based copy detection approach to determine whether sentences in any two documents should be treated as the same or different according to the degrees of similarity of the sentences, which is computed by using either the three least-frequent 4-gram approach or the fuzzy set information retrieval (IR) approach. These copy detection approaches, which achieve a high success rate in detection similar

(not necessary the same) sentences, (i) handles wide range of documents in different subject areas (such as sports, news, and science, etc.) and (ii) does not require static word lists, which means that there is no need to look up for words in a predefined dictionary/thesaurus to determine the similarity among words. Not only we can detect similar sentences in two documents, we can graphically display the relative locations of similar (not necessary the same) sentences detected in the documents using the dotplot views, which is a graphical tool. Experimental results show that the fuzzy set IR approach outperforms the three least-frequent 4-gram approach in copy detection. For this reason we adopt the fuzzy set IR copy detection approach for detecting similar Web documents, especially HTML documents, by computing the degree of resemblance between any two HTML documents, which represents to what extent the two documents under consideration are similar. Hereafter, we match the corresponding hierarchical content of the two documents using a simple tree matching algorithm.

Our copy detection approach is unique since it is sentence-based, instead of word-based on which most of the existing copy detection approaches are developed, and can specify the relative positions of same (or similar) sentences in their corresponding HTML documents graphically, as well as hierarchically, according to the document structures. The targeted documents to which our copy detection approach applies is different from others, since it (i) performs copy detection on HTML documents, instead of any plain text documents, (ii) detects HTML documents with similar sentences apart from exact matches, and (iii) is simple, as it uses the fuzzy set IR model for determining related words in documents and filtering redundant Web documents, and is supported by well-known and yet simple mathematical models.

Experimental results on detection of similar documents have been performed to check for accuracy using false positives, false negatives, precision, recall, and F-measure values. With over 90% F-measure, which indicates that the percentage of error is relatively small, our approach to detect similar documents performs reasonably well. The time complexity for our copy detection approach is $\mathcal{O}(n^2)$, where, n is the total number of sentences in a HTML document, whereas the time complexity for detecting similar HTML documents using our copy detection approach is $\mathcal{O}(n\ log\ n)$. The overall time complexity of our copy detection and similar HTML documents detection approach is $\mathcal{O}(n\ log\ n + n^2) \cong \mathcal{O}(n^2)$.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Besides piracy one of the problems on the Internet these days is redundant information, which exist due to replicated pages archived at different locations like mirror sites. As a result, the burden is on Web users to sort through retrieved Web documents to identify non-redundant data, which is a tedious and tiring process. Such documents can be found in different forms, such as documents in different versions; small documents combined with others to form a larger document; large documents co-existing with documents that are split from them. One classic example of such documents is news articles where new information is added to an original article and is republished as a new (updated) article. Since the amount of information available on the Internet increases on a daily basis, filtering redundant and similar documents becomes a more difficult task to the user. Fortunately, copy detection can be used as a filter, which identifies and excludes documents that overlap in content with other documents.

Redundant Web documents include HTML documents, and there are significant number of HTML documents on the Internet. HTML is a markup language, which means that the author of an HTML document must follow the HTML syntax to specify formatting commands intermixing with the text content of the document.

Unlike images, it is not obvious to the naked eye if two text documents are similar, needless to say distinguishing text content from tags in HTML documents. Since tags are intermixed with the actual content of an HTML document, they further complicate the problem of visualizing which portions of HTML documents match as matching content in HTML documents requires careful examination of the documents. Document analysis techniques are required in order to automate the entire process and provide a metric to show the extent of similarity of two HTML documents. We propose a unique method in detecting similar HTML documents, which (i) performs copy detection on HTML documents, (ii) detects HTML documents with similar[1] sentences, and (iii) specifies the overlapping locations of any two HTML documents graphically.

In our copy detection approach, each document is first passed through a stopword-removal [BYRN99] and stemming [Por80] process, which removes all the stopwords and reduces every word to its stem. Stopwords in a sentence should first be removed since stopwords, such as articles, conjunctions, prepositions, punctuation marks, numbers, non-alphabetic characters, etc., often do not play a significant role in representing the sentence. This process reduces the size of a document for comparison and subsequently the complexity on copy detection. Since our copy detection approach is a sentence-based approach, i.e., documents are compared sentence-by-sentence, non-stop, stemmed words in each sentence in one document are compared with non-stop, stemmed words in a sentence in another document to determine whether they are the same using either the *three least-frequent 4-gram* (4-gram for short) or the *fuzzy-set information retrieval* (*IR*) approach. Similar sentences in different documents

[1] From now on, whenever we use the term *similar* sentences (documents), we mean sentences (documents) that are semantically the same but may be different in terms of words used in the sentences (documents).

are detected, and the relative locations of matched sentences in the documents are graphically depicted by a dotplot view [Hel96].

We adopt our fuzzy set IR copy detection approach for detecting similar HTML documents. We have chosen the fuzzy set IR approach over 4-grams approach, since the experimental results show that the fuzzy-set IR approach outperforms the 4-grams approach in detecting similar, but not necessary the same, sentences in different documents, including HTML documents.

We proceed to present our results as follows. In Chapter 2, we discuss existing approaches for copy detection and similar document detection. In Chapter 3, we introduce our copy detection approach, present dotplot views that display similar sentences in two documents, and include the experimental results of our copy detection approach. In Chapter 4, we propose our method in quantifying the similarity of different HTML documents, first by extracting data content of an HTML document in a hierarchical structure and then performing tree matching on the hierarchical structures. In the chapter, we include the experimental results of our similar document detection approach. In Chapter 5, we give a concluding remark.

Chapter 2

Related Work

In this chapter, we discuss related work in copy detection and existing similar document detection approaches that appear in the literature.

2.1 Related Work for Copy Detection

Many efforts have been made in the past [HO82, BDGM95, Dam95, SGM95, Nev96, CCS00] in finding similarities among documents. Well-known copy detection methods include (i) Diff [Uni] (Unix/Linux man pages), which displays the differences in two files by printing the lines in the files that are different, (ii) SCAM [SGM95] (Stanford Copy Analysis Mechanism), which performs word-based copy detection, (iii) SIF [Man94], which detects similar files, (iv) COPS [BDGM95] and KOALA [Nev96], which are designed for plagiarism detection, and (v) the copy detection system for digital documents [CCS00].

Diff, which is designed for source code, text, and other line-oriented files, shows differences between two textual documents, even spaces. It captures the differences between two text documents one line at a time and checks lines in the same order. SCAM, which does not specify the location of overlap between documents, is geared towards small documents. Differed from SCAM, SIF finds similar files in a file system by using the fingerprinting[1] scheme to characterize documents. SIF, however, cannot gauge the extent of document overlap nor display the location of overlap, and the notion of similarity as defined in SIF is completely syntactic, e.g., files containing the same information but using different sentence structures will not be considered similar. COPS, which is developed specifically to detect plagiarism among documents, uses hash-based scheme for copy detection. It compares hash values of given documents with that in the database for copy detection. The basic scheme of SIF and COPS is similar; however, COPS generates syntactic hash units as opposed to fixed-length strings adopted by SIF and does have its limitations: (i) it uses a hash function that produces large number of collisions, (ii) documents to be compared by COPS must have at least 10 sentences or more, and (iii) it has problems selecting correct sentence boundaries. KOALA, like COPS, which is specifically designed for plagiarism and is a compromise between random fingerprinting of SIF and exhaustive fingerprinting of COPS, selects substrings of a document based on their usage. This results in lower usage of memory and increase accuracy. Neither KOALA nor COPS, however, can report the location of overlap of two documents and handle documents with varying size. [LPF02] present an approach to identify duplicated HTML documents; however, their identification approach is significantly differed from ours, since they consider HTML documents only, whereas our copy detection approach applies to any Web documents. Furthermore, two HTML documents are treated as the same by [LPF02] if the number of occurrences of their HTML tags are the same, which are not considered by us because tags are relatively insignificant in terms of representing document content. The copy detection approach of [CCS00] is closer to ours than

[1] Fingerprints of a document yield the set of all possible document substrings of a certain length, called *fingerprints*, and fingerprinting is the process of generating fingerprints.

others, since it is also sentence-based; however, their copy detection is restricted to copy detection of same sentences. In addition, our copy detection approach does not require static word lists and hence is applicable to Web documents in different subject areas.

Our copy detection approach overcomes most of the limitations of existing approaches. Since our approach is not line-oriented, it overcomes the problems posed by Diff. Unlike KOALA and COPS, our approach is more general, since our approach is not particularly targeted for plagiarism, even though we can handle the problem. Compared with SIF and SCAM, which cannot measure the extent of overlap, our copy-detection approach can specify the relative positions of same (or similar) sentences in their corresponding documents.

2.2 Related Work for Detecting Similar Documents

In the past, numerous methods for detecting similar documents [YLW$^+$99, CP00, RSF$^+$01, CCB02, PZ04] have been proposed. [PZ04] use the fingerprint approach to represent a given document, which then plays the role of a query to a search engine to retrieve documents for further comparisons using the shingles and Patricia tree methods. As discussed in Section 2.1, the fingerprint approach is either completely syntactic or suffers from collisions. [YLW$^+$99] introduce a statistical method to identify multiple text databases for retrieving similar documents, which along with queries are presented as vectors in the vector-space model (VSM) for comparisons. VSM is a well-known and widely used IR model [BYRN99], however, its reliance on term frequency without considering thesaurus of index terms in documents could be a drawback in detecting similar documents. [CP00, CCB02] characterize documents using multi-word terms, in which each term is reduced to its canonical form, i.e., stemming, and is assigned a measure (called IQ) based on term frequency for ranking, which is essentially the VSM approach. Besides using VSM, [RSF$^+$01] also consider user's profiles [BYRN99], which describe users' preferences, in retrieving documents, an approach that has not been widely used and proved. In contrary, we adopt a copy detection approach, which consider similar, in addition to exact matching of, sentences in HTML documents to determine the degrees of similarity among different documents. We detect HTML documents that are similar/dissimilar in terms of their data content captured in a hierarchical manner and exclude tags, which are unimportant in representing data content in an HTML document.

Chapter 3

Our Copy Detection Approach

In our copy detection approach, a stopword-removal and stemming process is first performed on sentences in each Web document to yield what we call "refined" sentences, i.e., sentences without stopwords and words in the sentences are stemmed. Since our copy detection approach is a sentence-based approach,[1] refined sentences in one document are compared with refined sentences in another document to determine whether they are similar or the same using either the three least-frequent 4-gram or the fuzzy-set information retrieval (IR) approach.

3.1 Eliminating Non-Essential Data

Words in sentences of a Web document involved in our copy detection approach are made to undergo stopword list removal and stemming to represent each sentence independent of its grammatical (dis)similarities so that sentences which have different structures (such as active and passive voice) but convey the same meaning are recognized as same sentences, i.e., similar sentences. During the process of stopwords removal from a Web document, each individual sentence is parsed against the

[1] We use the boundary disambiguation algorithm in [Cam97], which has accuracy in the 90% range but operates in linear time, to determine sentence boundary.

list of pre-determined stopwords found in English language. (Our copy detection problem is restricted to English Web documents, which can easily be extended to other languages.) This step would reduce the unwanted complexity of processing insignificant/non-representative information and hence can speed up the copy detection process. For example, consider the following sentence S in a Web document, $Sistani_1$.html:

> *Iraq's top Shi'ite Muslim cleric, Grand Ayatollah Ali al-Sistani, travelled to, Britain where he is expected to receive treatment for a heart condition, a spokesman said.*

Removing all the stopwords from S yields

> *iraqs top shiite muslim cleric grand ayatollah ali alsistani travelled britain expected receive treatment heart condition spokesman said.*

denoted S_r. Next, the stemming algorithm [Por80], which is a suffix removal algorithm, is applied. During the stemming process, an explicit list of suffixes is used, and with each suffix, the criterion under which it may be removed from a word to leave a valid stem is considered based on pre-defined rules to find the root of the word. Applying the stemming algorithm to S_r yields

> *iraqs top shiite muslim cleric grand ayatollah ali alsistani travel britain expect receive treat heart condi spoke said.*

3.2 Detecting Similar Sentences in Web Documents

After performing stopword removal and stemming on two Web documents, we use either (i) the three least-frequent 4-gram or (ii) the fuzzy-set IR approach to determine which sentences in the documents are similar or different.

3.2.1 Sentence Representations by Three Least-Frequent 4-Grams

The 4-gram approach, which is an exhaustive fingerprinting approach, allows an intuitive, well-defined notion of similarity between documents to be defined [CCS00]. A 4-gram of a string S is a 4-character substring of S. For example, the 4-grams of the string "novel creations" are *nove, ovel, velc, elcr, lcre, crea, reat, eati, atio, tion*, and *ions*. For comparing two sentences using substrings of sentences, two substring selection strategies are usually considered: (i) selection based on N*-grams* and (ii) selection based on *words*. We adopt the N-gram approach over word frequencies because (i) N-grams frequencies are largely independent of the type of documents from which they come from [Dam95], (ii) N-grams handle novel words robustly and provide an elegant solution to the zero-frequency problem addressed by [WB91], which estimates the likelihood of the occurrence of a novel event, and (iii) N-grams do not require static-word sets, and thus, are applicable to various domain sizes. The only disadvantage of N-grams over words is an increase in computation time. However, extracting N-grams is simple and the memory storage of *selected* N-grams that represent a sentence is far less than storing a whole sentence in the memory. For these reasons, we consider the 4-gram approach as an elegant choice for copy detection.

In using the N-gram approach, it is critical to choose the right value of N to provide good discrimination among sentences. Although any value of N can be considered, $N = 4$ is an ideal choice. This is because as the value of N increases, the better it becomes to distinguish words in one sentence from words in another. Since each N-gram requires N bytes of storage, the memory requirements become too large for $N = 5$. This is because for any given N, each N-gram requires N bytes of storage. For smaller values of N (i.e., $N = 1$, 2, or 3), however, it has been observed [CCS00] that they do not provide good discrimination between sentences. To illustrate the 4-gram construction process, let's consider the following sentence S:

> *iraqs top shiite muslim cleric grand ayatollah ali alsistani travel britain expect receive treat heart condi spoke said*

which is obtained from its original sentence after applying the stopword-removal and stemming steps (as computed in Section 3.1). The individual 4-grams for the sentence are *iraq*, *raqs*, *aqst*, *qsto*, *stop*, *tops*, etc., which are shown in the 4^{th} entry in the list of concatenated 4-grams in Table 3.1. As part of the pre-processing step, which is not included in the real-time copy detection process, each document extracted from the large TREC archive data set (http://trec.nist.gov/data/t5_confusion.html) [TRE], the *Gutenberg* collection, and numerous Web sites (as shown in Table 3.2) is registered into the database, and each sentence from the document is converted into a set of four grams and stored in the form of a tree. These 4-grams are then used to determine the three *least-frequent* 4-grams in a sentence from a document, which is the best option to represent the sentence uniquely [Dam95]. The three least-frequent 4-grams in a sentence are concatenated to *represent* the sentence from a document to be compared with the three least-frequent 4-gram representations of sentences in another document. For example, the three least-frequent 4-grams of S given above are *raqs, aqst*, and *qsto*, and S is represented as *raqsaqstqsto*. Two sentences are treated the *same* if their corresponding three least-frequent 4-gram representations are the same. Converting sentences into their three least-frequent 4-gram representations speeds up the processing time of comparing sentences. All the *refined* sentences in documents involved in the copy detection process are converted into their three least-frequent 4-grams for comparison.

A total number of 200 documents, which were randomly sampled from the set of

$Sistani_1$.html	
Concatenated 4-grams	Three Least Frequent 4-grams
(1) tops opsh pshi shii hiit iite itec tecl ecle cler leri eric ricl iclo clon lond ondo ndon donh onhe nhea hear eart artt rttr ttre trea reat . . .	(1) opsh, pshi, hiit
(2) lond ondo ndon done onen neng engl ngla glan land anda ndap dapi apir pira iraq raqs aqst qsto stop tops opsh pshi shii hiit iite item temu emus musl usli slim limc imcl mcle . . .	(2) raqs, aqst, qsto
(3) sist ista stan tani ania niar iarr arri rriv rivy ivye vyes yest este ster terd erda rday daya ayaf yaft afte fter tern erno rnoo . . .	(3) anis, smai, ainl
(4) iraq raqs aqst qsto stop tops opsh pshi shii hiit iite item temu emus musl usli slim limc imcl mcle cler leri eric ricg icgr cgra . . .	(4) raqs, aqst, qsto
⋮	⋮
(8) durm urmi rmin minu inut nute utes test esto stop topo opov . . .	(8) durm, urmi, rmin
⋮	⋮
(12) sist ista stan tani ania niac iacc acco ccom comp ompa mpan pani anit nith ithr thre here reea eeai . . .	(12) tani, niac, aicc
⋮	⋮

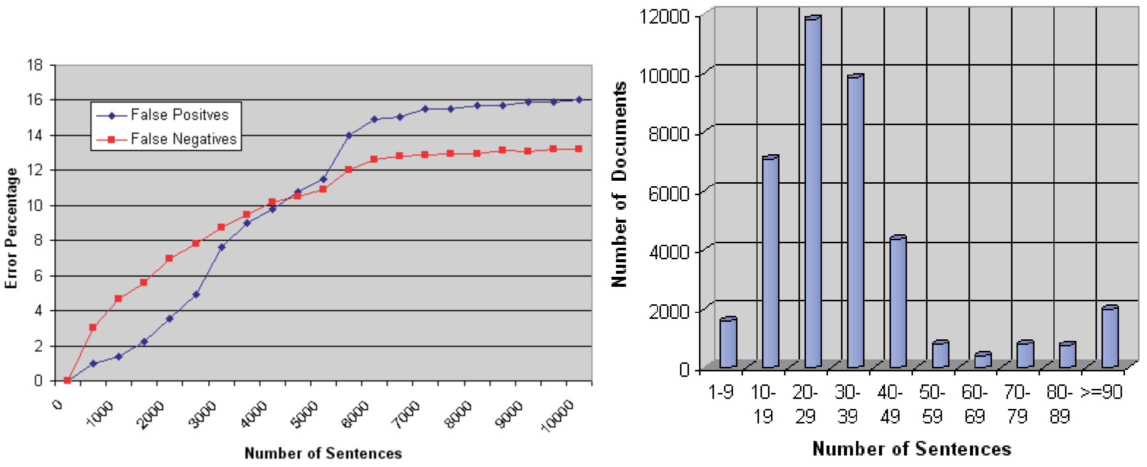
Table 3.1: List of 4-grams and the three least-frequent 4-grams of sentences in $Sistani_1$.html

Sources	Size	Number of Pages	Number of Sentences	Number of Words*
TREC (http://trec.nist.gov/data/)	1.29 GB	4,390	4,829,653	28,983,918
Gutenberg (ftp://ftp.archive.org/pub/etext/)	1.53 GB	5,643	8,935,215	72,681,213
News Articles on the Web	0.63 GB	39,349	701,520	5,429,643
A Text Archive (ftp://ftp.etext.org/pub/)	1.13 GB	3,124	6,239,154	41,624,048
Total	4.58 GB	52,506	20,705,542	148,718,822

*Number of distinct non-stop, stemmed words.

Table 3.2: Documents used for constructing least-frequent 4-grams and the correlation matrix in the fuzzy IR model

52,506 documents as shown in Table 3.2, were used to study the performance of our 4-gram copy detection approach. Out of the 200 sampled documents, 12.5%, 17.5%, 32.5%, and 37.5% of these documents were chosen from the archive sites TREC, Gutenberg, News articles on the Web, and Etext.org, respectively, and 33% of the 200 sampled documents are HTML documents in which tags are first filtered before copy detection is performed on the documents (See Section 4.1 for details.) Each pair of sentences retrieved from different documents in the sampled set, which were treated as either the same or different by using our 4-gram approach, were manually verified for *false positives* (i.e., sentences that are *different* but are treated as the *same*) and *false negatives* (i.e., sentences that are the *same* but are treated as *different*), which are then plotted on the graph for increasing number of sentences. The study shows that the aggressive increase of percentage of false positives and false negatives slows down at after 5,000 sentences and becomes steady and stable thereafter at around 16% for false positives and 12% for false negatives. (See Figure 3.1(a) for details.) We also



(a) Error analysis of the 4-gram approach

(b) Sentence distribution in HTML documents

Figure 3.1: Analysis of our 4-gram copy detection approach and sizes of HTML documents

observe that when the total number of sentences (to be compared) is less than 500, the error percentage of false positives and false negatives is below 2%, respectively, which is relatively low and acceptable for copy detection on HTML documents to which significant number of existing Web documents on the Internet belong. In fact, the average size of an HTML document, as shown in one of our surveys, is less than 60 sentences (See Figure 3.1(b) for details, which is constructed by using the 39,349 HTML documents out of the 52,506 documents as shown in Table 3.2.), and there are significant number of Web documents that are HTML documents.[2]

The major drawback of using our three least-frequent 4-gram copy detection approach is that it cannot detect similar sentences, i.e., similar sentences are treated as different, a deficiency that can be corrected by our fuzzy-set IR copy detection approach.

[2] We collected 4,389 Web documents from the Internet at one time, and almost 90% of the downloaded documents were HTML documents. This may be an isolated instance, however, since HTML is a widely used language in creating Web documents, the experimental result sounds reasonable to us.

3.2.2 Degree of Similarity in the Fuzzy-Set IR Model

Apart from the three least-frequent 4-gram approach to detect same sentences, we adopt and modify the fuzzy-set IR model [OMK91] to find similar sentences. At a high level, a sentence can be treated as a group of words arranged in a particular order. In English language, two sentences can be semantically the same but differ in structure (such as using the active versus passive voice), and matching two sentences is approximate or vague. This can be modeled by considering that each *word* in a sentence is associated with a fuzzy set that contains words with same meaning, and there is a degree of similarity (usually less than 1) between (words in) a sentence and the fuzzy set. This interpretation in fuzzy theory is the fundamental concept of various fuzzy-set models in IR. We consider the fuzzy-set IR model for copy detection, since (i) identification and measurement of similar sentences, apart from exact match, further enhances the accuracy of our copy detection approach, and (ii) the fuzzy-set model is designed and has been proved to work well for partially related semantic content, which can handle the problem of copy detection of similar, but not the same, sentences.

In the fuzzy-set IR model, a *term-term correlation matrix*, which is constructed in a pre-processing step of our copy detection approach, consists of words and their corresponding *correlation factors* that measure the degrees of similarity among different words, such as "automobile" and "car." Our fuzzy-set IR model obtains the degrees of similarity among sentences by computing the correlation factors between any pair of words from two different sentences in their respective documents. (It has been observed that words in sentences that are common to a number of documents discuss the same subject.) In the determination of the degree of similarity between two sentences using the fuzzy-set IR approach, a term-term correlation matrix with rows and columns associated to words[3] captures the degree of similarity among words. The following *word-word correlation factor*, $c_{i,j}$, defines the extent of similarity between any two words i and j in the term-term correlation matrix:

$$c_{i,j} = n_{i,j}/(n_i + n_j - n_{i,j}) \tag{3.1}$$

where $c_{i,j}$ is the correlation factor between words i and j, $n_{i,j}$ is the number of documents in a collection (such as the data set as shown in Table 3.2) with both words i and j, n_i (n_j, respectively) is the number of documents with word i (word j, respectively) in the collection.

The degree of similarity of two sentences is the extent to which the sentences match. To obtain the degree of similarity between two sentences S_l and S_j, we first compute the *word-sentence correlation factor* $\mu_{i,j}$ of word i in S_l with all the words in S_j, which measures the degree of similarity between i and (all the words in) S_j, as

$$\mu_{i,j} = 1 - \prod_{k \in S_j} (1 - c_{i,k}) \tag{3.2}$$

where k is one of the words in S_j and $c_{i,k}$ is the correlation factor between words i and k as defined in Equation 3.1.

Based on the μ-value of each word in a sentence S_i, which is computed against sentence S_j, we define the *degree of similarity* of S_i with respect to S_j as follows:

$$Sim(S_i, S_j) = \frac{\mu_{w_1,j} + \mu_{w_2,j} + \ldots + \mu_{w_n,j}}{n} \tag{3.3}$$

where w_k ($1 \leq k \leq n$) is a word in S_i, and n is the total number of words in S_i. $Sim(S_i, S_j)$ is a normalized value. Likewise, $Sim(S_j, S_i)$, which is the *degree of similarity* of S_j with respect to S_i, is defined accordingly.

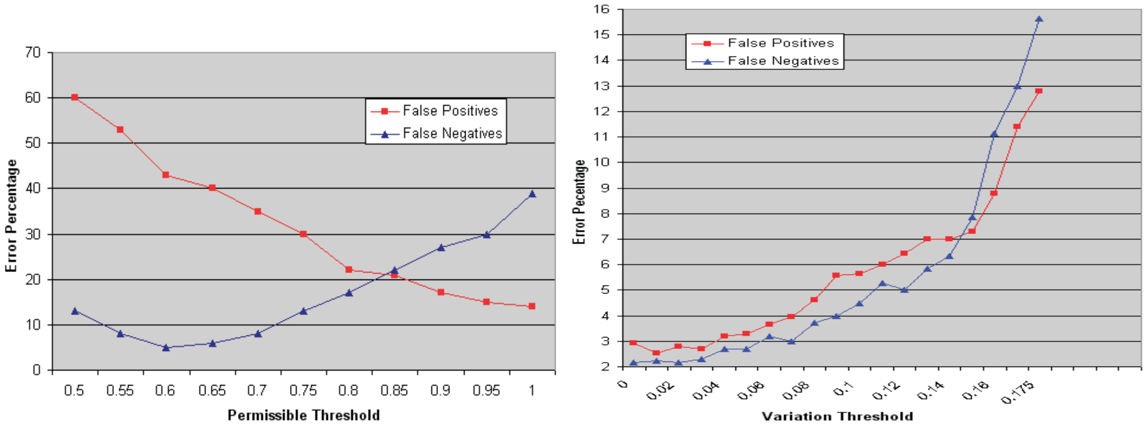
[3] From now on, unless stated otherwise, whenever we use the term *word*, we mean *non-stop, stemmed word*.

Using Equation 3.3 as defined above, we determine whether two sentences S_i and S_j should be treated the same, i.e., equal (EQ) as defined below.

$$EQ(S_i, S_j) = \begin{cases} 1 & \text{if } MIN(Sim(S_i, S_j), Sim(S_j, S_i)) \geq 0.825 \wedge \\ & \qquad |Sim(S_i, S_j) \text{ - } Sim(S_j, S_i)| \leq 0.15 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where 0.825 is called the *permission threshold value*, whereas 0.15 is called the *variation threshold value*, and $|Sim(S_i, S_j)$ - $Sim(S_j, S_i)|$ is the absolute value of the difference between the two similarity measures. The permissible threshold is a value set to obtain the *minimal* similarity between any two sentences S_i and S_j in our copy detection approach, which is used partially to determine whether S_i and S_j should be treated as equal (EQ). Along with the permissible threshold value, the variation threshold value is used to decrease the errors in determining the equality of two sentences. The variation threshold value sets the maximum, allowable difference in sentence sizes between S_i and S_j, which is computed by calculating the *difference* between $Sim(S_i, S_j)$ and $Sim(S_j, S_i)$. The threshold values 0.825 and 0.15 thus provide the necessary and sufficient conditions for estimating the equality of two sentences and were determined by testing the documents in the collection as shown in Table 3.2. We explain how to compute the threshold values below.

Using the same randomly sampled 200 documents for analyzing the performance of our 4-gram copy detection approach as discussed in Section 3.2.1, we determined the *permissible* and *variation threshold values.* According to each predefined threshold value V, which is between 0.5 and 1.0 with an increment of 0.05, we categorized each pair of sentences S_1 and S_2 in the 200 documents as equal or different, depending on V and the minimum of the two degrees of similarity measures on S_1 and S_2. S_1 and S_2 are (probably) *equal* if $MIN(Sim(S_1, S_2)$, $Sim(S_2, S_1)) \geq V$; otherwise, they are



(a) The permissible threshold value

(b) The variation threshold value

Figure 3.2: Determination of the permissible and variation threshold values in the fuzzy-set IR approach

different. Hereafter, we manually (i) examined each pair of sentences to determine the accuracy of the conclusion, i.e., *equal* or *different*, drawn on the sentences, (ii) computed the number of false positives and false negatives, and (iii) plotted the outcomes in a graph. According to the graph as shown in Figure 3.2(a), we set the *permissible threshold value* to be 0.825, which is the “balance” point of the minimum false positives and false negatives, i.e., neither the values of false positives nor false negatives dominates the copy detection errors. The curves in Figure 3.2(a) show that as the (permissible) threshold value increases, the percentage of false positives decreases, whereas the percentage of false negatives increases. This is because as the threshold value increases, the total number of sentences treated as equal decreases, since more sentences are treated as different. Hence, more documents that are similar are not detected as the threshold increases and as a result, the false negatives increase, which is opposite in the case of false positives.

To obtain the *variation threshold value*, we examined a set PS of pairs of sentences whose minimal degrees of similarity exceed or equal to the permissible threshold value.[4] The differences between the degrees of similarity of each pair of sentences in PS are first calculated and then sorted. The false positives and false negatives are then determined manually and plotted at regular intervals in sorted order. The intersection of the false positive and false negative curves is chosen as the variation threshold value, which is neither dominated by false positives nor false negatives.

Since the minimum permissible threshold value is set to 0.825, the difference in similarity between two documents can only range between 0 and 0.175. The greater the difference in similarity, the greater are the chances of error. For example, if the difference is 0.175, one of the similarity measures can be $sim(S_1,\ S_2) = 1.0$ and the other is $sim(S_2,\ S_1) = 0.825$, or vice versa. This is the case only when the set of words in S_1 is a subset of the words in S_2. For example, if sentence S_1 is "The boy goes to school every day on bus," whereas sentence S_2 is "The bus going to school passing by the lake stops for the boy at the corner every day." In this example, S_1 is a part of S_2, but not vice versa, and should be treated as different. According to Figure 3.2(b), the ideal variation threshold value between the range 0 and 0.185 is 0.15, which we adopt in our fuzzy-set copy detection approach.

We demonstrate the detection on similar, as well as different, sentences by calculating their corresponding $MIN(Sim(S_1, S_2), Sim(S_2, S_1))$ and $|Sim(S_1, S_2)$ - $Sim(S_2, S_1)|$ values using the two examples given below.

Example 1 Consider the following two sentences:

S_1: The global aid for the tsunami is on the rise.

S_2: The international community has increased help for the disaster.

and (as shown in Table 3.3) the term-term correlation matrix, which includes (i) only

[4] The purpose of using the variation threshold value is to further reduce the error percentage (i.e., false positives and false negatives) that could be introduced by the permissible threshold value.

S_2/S_1	global	aid	tsunami	rise	μ-value
international	0.800	0.496	0.579	0.177	**0.965**
community	0.430	0.771	0.117	0.334	**0.923**
increase	0.123	0.643	0.661	0.900	**0.989**
disaster	0.611	0.414	0.900	0.394	**0.986**
μ-value	**0.961**	**0.976**	**0.987**	**0.966**	[Same]

Table 3.3: The correlation matrix of Example 1 restricted to words in matching sentences

the non-stop, stemmed words in S_1 and S_2, (ii) the word-word correlation factor $c_{i,j}$ between any two words i and j in the corresponding cell, and (iii) the *word-sentence correlation factor* $\mu_{i,j}$ between word i and sentence S_j. The similarity measures of S_1 and S_2 are $Sim(S_1, S_2) = 0.973$ and $Sim(S_2, S_1) = 0.966$. Since $MIN(0.973, 0.966) = 0.966 \geq 0.825$ and $|0.973 - 0.966| = 0.007 \leq 0.15$, S_1 and S_2 are treated as the same, which is a valid conclusion based on the closeness in the content of the two sentences.

Example 2 Consider another two sentences:

S_1: Please do not hesitate to contact us.

S_2: We can be reached by phone.

and the corresponding term-term correlation matrix that include non-stop, stemmed words in S_1 and S_2 is shown in Table 3.4. The degree of similarity of the two sentences are $Sim(S_1, S_2) = 0.35$ and $Sim(S_2, S_1) = 0.67$. Since $MIN(0.67, 0.35) = 0.35 \leq 0.825$, S_1 and S_2 are treated as *different*, which is a valid conclusion based on the content of the two sentences.

S_2/S_1	please	hesitate	contact	μ-value
reach	0.059	0.038	0.683	**0.713**
phone	0.058	0.036	0.593	**0.630**
μ-value	**0.113**	**0.073**	**0.865**	[Diff]

Table 3.4: The correlation matrix of Example 2 restricted to words in matching sentences

3.2.3 Degree of Overlap

Using the *EQ* value as defined in Equation 3.4, which determines whether two sentences should be treated as the same, we derive the *degree of overlap* between documents doc_1 and doc_2, which computes the degree of overlap between doc_1 and doc_2 according to the ratios of sentences common to both documents, which is defined as follows:

$$Overlap(doc_1, doc_2) = \frac{(|doc_1 \ \cap \ doc_2|)}{|doc_1|}, \frac{(|doc_1 \ \cap \ doc_2|)}{|doc_2|} \tag{3.5}$$

where $|doc_1 \cap doc_2|$ is the number of common sentences in doc_1 and doc_2, and $|doc_1|$ ($|doc_2|$, respectively) is number of sentences in doc_1 (doc_2, respectively).

Example 3 Let the number of sentences common to documents doc_1 and doc_2 be 12, and let $|doc_1| = 32$ and $|doc_2| = 21$. Using Equation 3.5, the *degree of overlap* of the two documents is calculated as (12/32, 12/21) = (0.38, 0.57).

3.3 Evaluation of the Copy Detection Approach

Using a set of Web documents collected from various sources (as shown in Table 3.2), we evaluated the performance of our copy detection approach for Web documents. Randomly sampled pairs of sentences S_1 and S_2, which are either the similar or different with S_1 from one Web document and S_2 from another, were compared using our copy detection approach. The results, which include some pairs of sentences listed below, are shown in Table 3.5.

(1) Please do not hesitate to contact us.

We can reach the destination by tomorrow.

(2) We acknowledge receipt of your letter.

We got your letter.

(3) I consider Bob Kim's best friend.

Bob is Kim's best friend.

(4) John is the king.

The king is intelligent.

(5) The student submitted papers to the company for employment.

To get employed the graduate applied to the office with relevant documents.

(6) No one is allowed to talk to him.

Please do not hesitate to contact us.

(7) We have pleasure enclosing our updated brochure.

We have put a new brochure in this letter.

As shown in Table 3.5, the fuzzy-set IR approach outperforms the three least-frequent 4-gram approach in detecting similar sentences, since the fuzzy-set IR approach checks for similarity between completely different words, besides same words, whereas the three least-frequent 4-gram approach detects similarity between same words only. For example, "document" and "paper" are considered similar by the fuzzy-set IR approach, whereas they are declared different by the 4-gram approach.

In order to further verify the correctness of the three least-frequent 4-gram approach, as well as our fuzzy-set IR approach, we ran other test cases to compare

1. Two documents that are closely related—all the sentences in one are included

Sentence Pairs	Matched	FP (4-gram)	FN (4-gram)	FP (Fuzzy)	FN (Fuzzy)
(1)	No	No	No	No	No
(2)	Yes	No	Yes	No	No
(3)	Yes	No	No	No	No
(4)	No	No	No	Yes	No
(5)	Yes	No	Yes	No	No
(6)	No	No	No	No	No
(7)	Yes	No	No	No	No
...	...	...	...	...	...
Total Count		0%	22%	5.5%	5.5%

F(alse)P(ositive): Sentences that are *different* but are treated as the *same*
F(alse)N(egative): Sentences that are the *same* but are treated as *different*

Table 3.5: Experimental results of our copy detection approach for Web documemts on a set of randomly sampled pairs of sentences

in the other.

2. Unrelated sentences added to the subset document in (1).

3. Two documents that are highly unrelated.

4. Two documents that are (moderately) related.

5. Two documents that vary in size to demonstrate the resistance of our copy detection approach to size variations.

6. Revisions of a set of successive news reports.

The dotplot views of similar sentences detected by the three least-frequent 4-gram approach (fuzzy-set IR approach, respectively) for each test case in Tables 3.6, 3.7, and 3.8 are shown in Figures 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8.

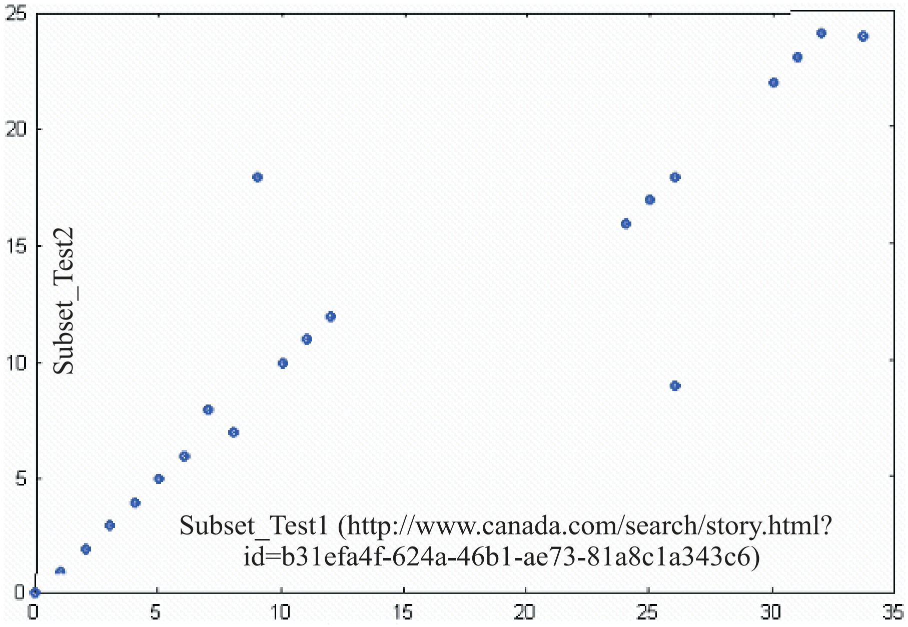
Case	Documents		Document	# Sentences		Sentence	Overlap	
	1^{st}	2^{nd}	Relationship	1^{st}	2^{nd}	Matched	1^{st}	2^{nd}
1	Subset_ Test1	Subset_ Test2	$2^{nd} \subseteq 1^{st}$	35	21	21	60%	100%
2	Subset_ Test1	Subset_ Test3	2^{nd} is 2^{nd} in case 1 + 10 unrelated sentences	35	31	21	60%	67%
3	Intel_ Stock1	Intel_ Stock2	1^{st} & 2^{nd} differ	149	133	4	3%	3%
4	9_11_1	9_11_2	1^{st} is related to 2^{nd}	87	79	47	54%	60%
5	Gandhi_1	Gandhi_2	1^{st} & 2^{nd} vary in size	425	50	15	4%	30%

Table 3.6: Copy detection on a number of Web documents using the three least-frequent 4-gram approach

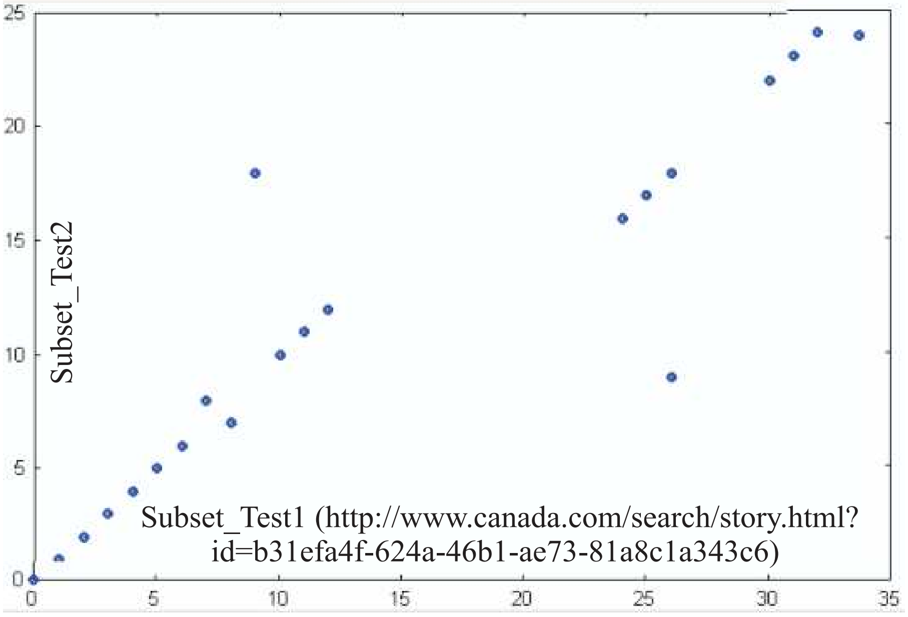
3.4 Experimental Results

To display matched sentences in two Web documents graphically, *dotplot view* [Hel96], which is originally designed for detecting patterns in languages, is an ideal choice. Dotplot views depict common sentences in two documents (see Figure 3.3, 3.4 and 3.5 for an example). The X-axis (Y-axis, respectively) represents the sentence position of the 1st (the 2nd, respectively) document. The dots in the graph represent the sentence positions where the sentences in corresponding documents match. The reason for using dotplot is three-fold. First, matched sentences can be captured graphically and are easy to trace. Second, the relative distance between the sentences can be observed. Third, the size of the documents under consideration becomes irrelevant as the axis in dotplot can be scaled to include as many sentences as possible. Moreover, patterns as shown in a dotplot reflect the relative order of common sentences in their respectively documents. For example, a dotplot view with two diagonal lines perpendicular to each other shows that sentences in the two corresponding documents are in revised order. If the direction of the move is directly up (to the left), this would correspond to aligning several sentences in one document with only one sentence in the other document. If the direction of the move is horizontal (vertical, respectively) from upper left to lower right, then there is a gap between two segments of matched sentences.

The additional test cases, along with others, confirm that both three least-frequent 4-gram and fuzzy-set IR approaches perform very well with same sentences, with only a few false positives and false negatives, and the fuzzy-set IR approach detects similar sentences significantly better than the 4-gram approach, since the latter cannot detect similar sentences. To support our claims, consider the corresponding dotplot views in Figures 3.3, 3.4, and 3.5. The corresponding pairs of dotplot views in Figures 3.3, 3.4,

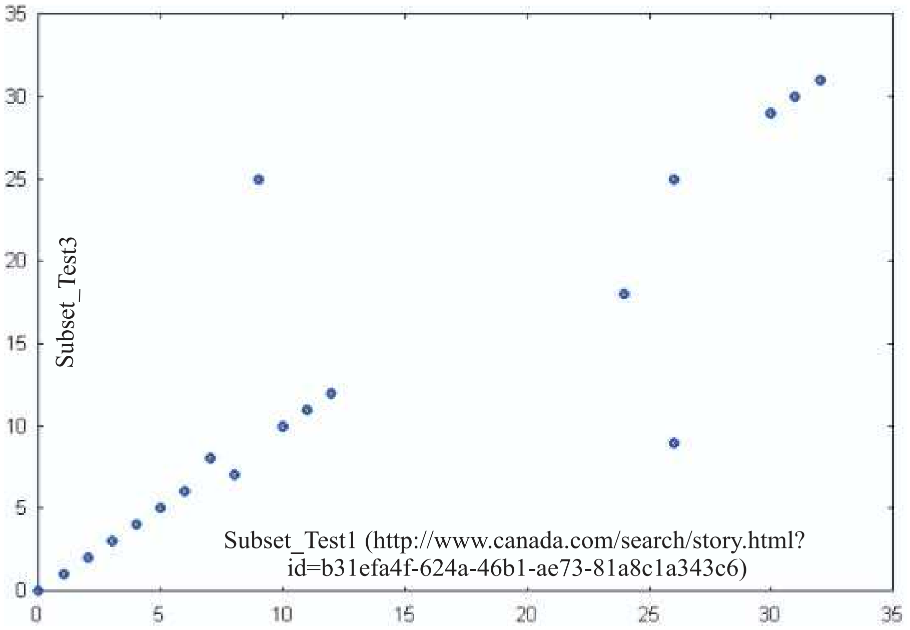


(a) The dotplot view of similar sentences in Subset_Test1 and Subset_Test2 detected by using the three least-frequent 4-gram approach

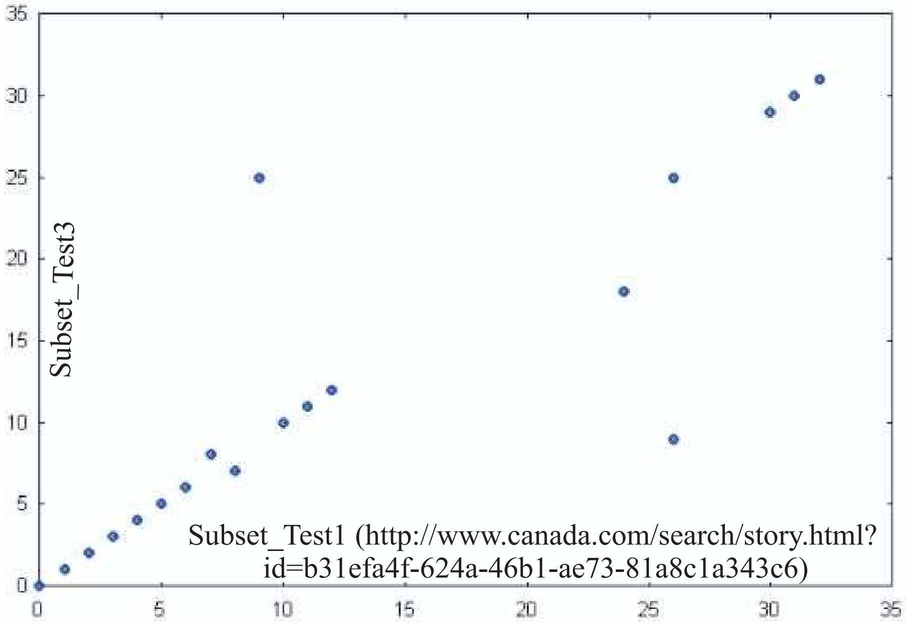


(b) The dotplot view of similar sentences in Subset_Test1 and Subset_Test2 detected by using the fuzzy-set IR approach

Figure 3.3: Dotplot views of similar sentences in Web documents Subset_Test1 and Subset_Test2 (as shown in Tables 3.6 and 3.7) detected by using the three least-frequent 4-gram and fuzzy-set IR approaches

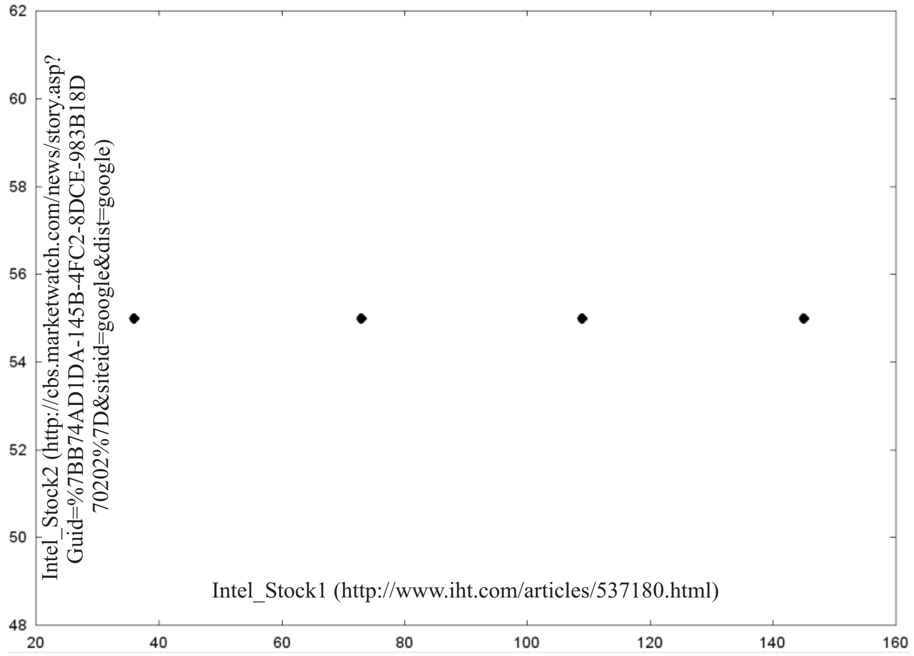


(a) The dotplot view of similar sentences in Subset_Test1 and Subset_Test3 detected by using the three least-frequent 4-gram approach

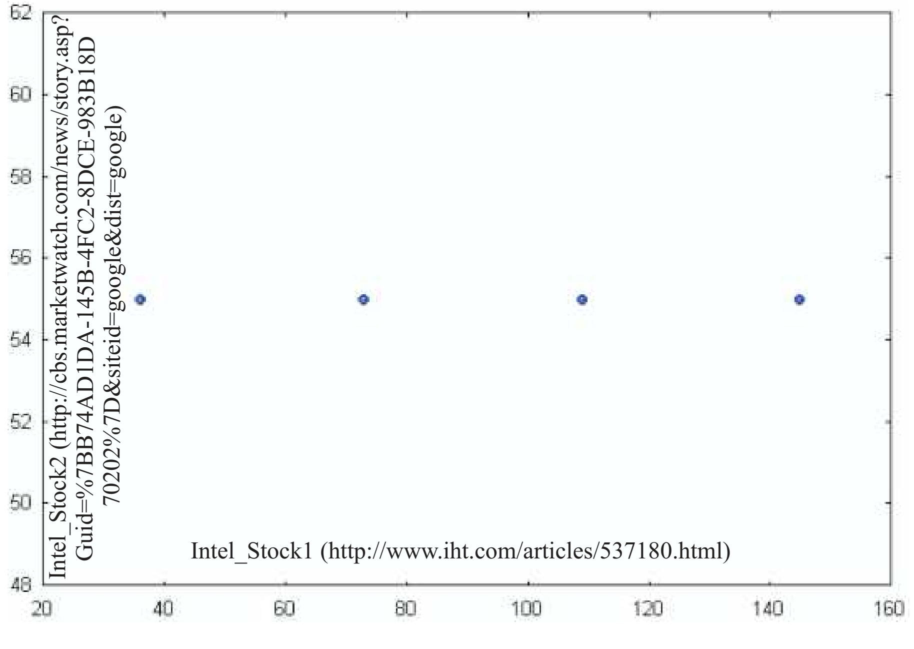


(b) The dotplot view of similar sentences in Subset_Test1 and Subset_Test3 detected by using the fuzzy-set IR approach

Figure 3.4: Dotplot views of similar sentences in Web documents Subset_Test1 and Subset_Test3 (as shown in Tables 3.6 and 3.7) detected by using the three least-frequent 4-gram and fuzzy-set IR approaches



(a) The dotplot view of similar sentences in Intel_Stock1 and Intel_Stock2 detected by using the three least-frequent 4-gram approach



(b) The dotplot view of similar sentences in Intel_Stock1 and Intel_Stock2 detected by using the fuzzy-set IR approach

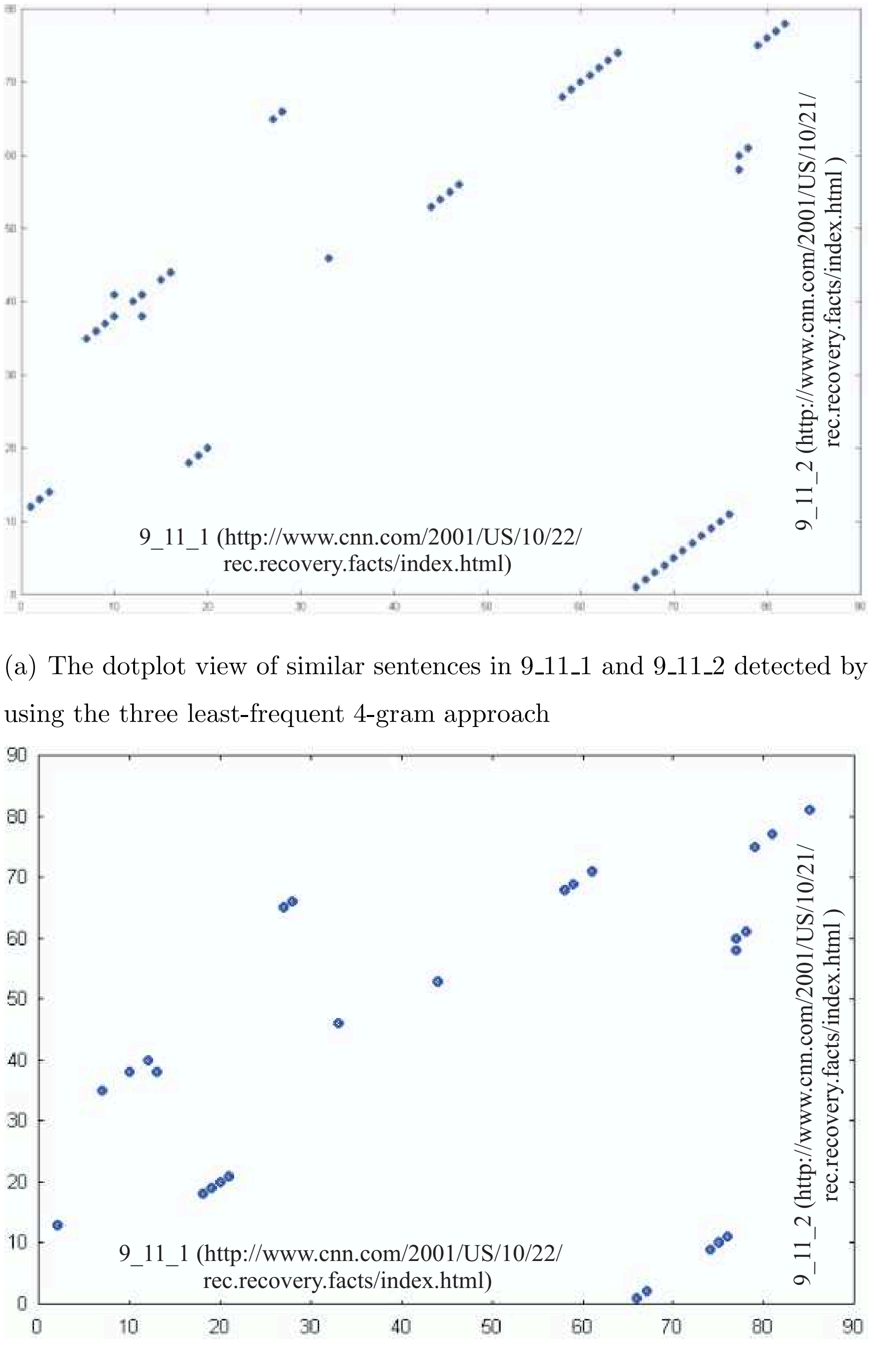
Figure 3.5: Dotplot views of similar sentences in Web documents Intel_Stock1 and Intel_Stock2 (as shown in Tables 3.6 and 3.7) detected by using the three least-frequent 4-gram and fuzzy-set IR approaches

Case	Documents		Document	# Sentences		Sentence	Overlap	
	1^{st}	2^{nd}	Relationship	1^{st}	2^{nd}	Matched	1^{st}	2^{nd}
1	Subset_Test1	Subset_Test2	$2^{nd} \subseteq 1^{st}$	35	21	21	60%	100%
2	Subset_Test1	Subset_Test3	2^{nd} is 2^{nd} in case 1 + 10 unrelated sentences	35	31	21	60%	67%
3	Intel_Stock1	Intel_Stock2	1^{st} & 2^{nd} differ	149	133	4	3%	3%
4	9_11_1	9_11_2	1^{st} is related to 2^{nd}	87	79	27	31%	34%
5	Gandhi_1	Gandhi_2	1^{st} & 2^{nd} vary in size	425	50	20	5%	40%

Table 3.7: Copy detection on a number of Web documents using the fuzzy-set IR approach

Case	Documents		Number of Sentences in Documents		Computed by **4-gram**			Computed by **Fuzzy-set**		
	1^{st}	2^{nd}			Sentence Matched	Overlap %		Sentence Matched	Overlap %	
			1^{st}	2^{nd}		1^{st}	2^{nd}		1^{st}	2^{nd}
1-2	Rus_1	Rus_2	99	83	42	42	51	49	50	59
2-3	Rus_2	Rus_3	83	90	36	43	40	37	45	41
3-4	Rus_3	Rus_4	90	101	39	43	39	43	48	43
4-5	Rus_4	Rus_5	101	104	77	76	74	77	76	74
5-6	Rus_5	Rus_6	104	101	96	92	95	96	92	95
1-6	Rus_1	Rus_6	99	101	42	42	42	46	47	46

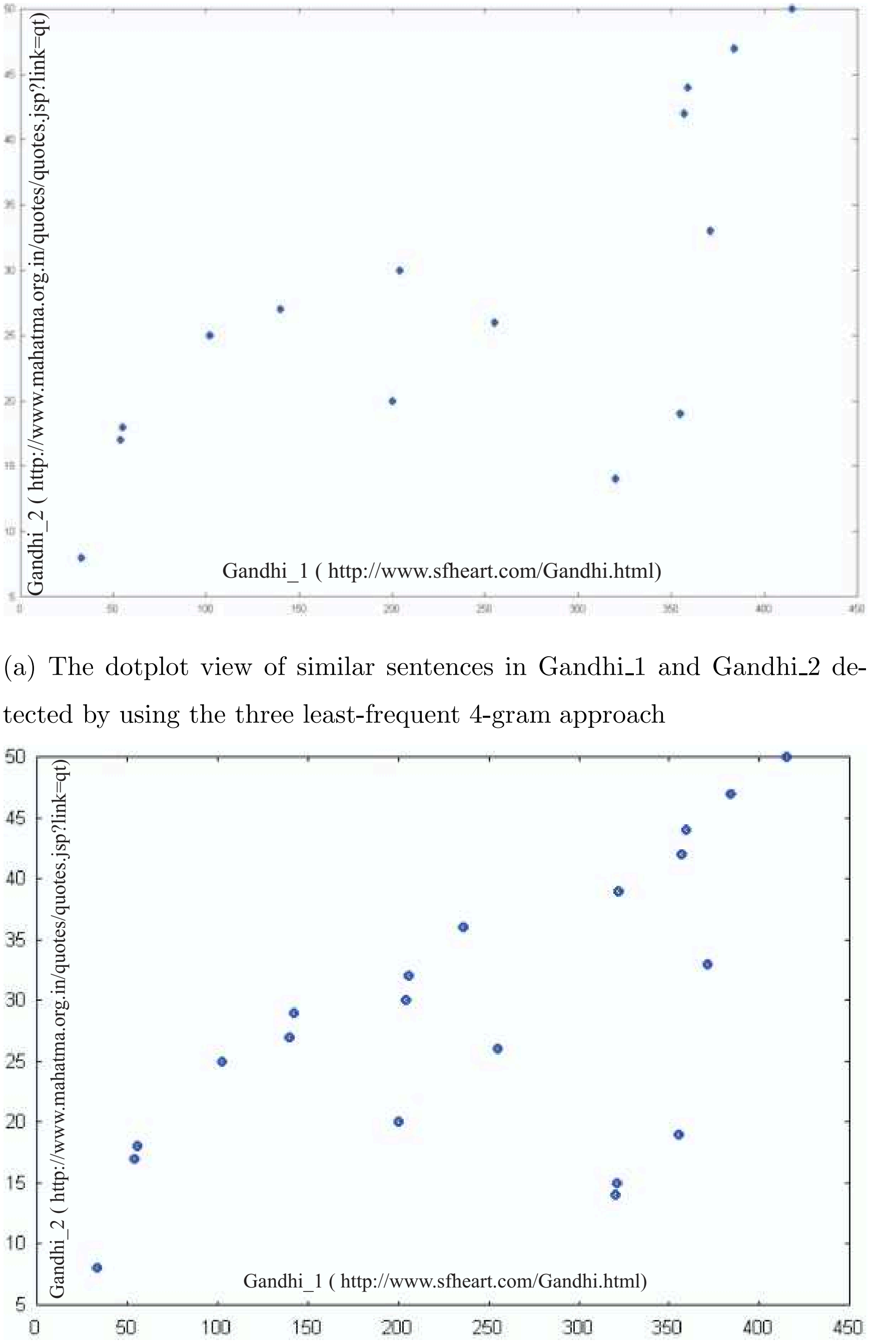
Table 3.8: Comparisons of successive news reports on Russian hostages taken with Rus_1 reported on 09/01/2004, Rus_2 and Rus_3 on 09/02/2004, and Rus_4, Rus_5 and Rus_6 on 09/03/2004 (http://www.cnn.com/2004/WORLD/europe/09/01(02/03)/rus-sia.school/index.html) using the 3 least frequent 4-gram and fuzzy-set IR approaches



(a) The dotplot view of similar sentences in 9_11_1 and 9_11_2 detected by using the three least-frequent 4-gram approach

(b) The dotplot view of similar sentences in 9_11_1 and 9_11_2 detected by using the fuzzy-set IR approach

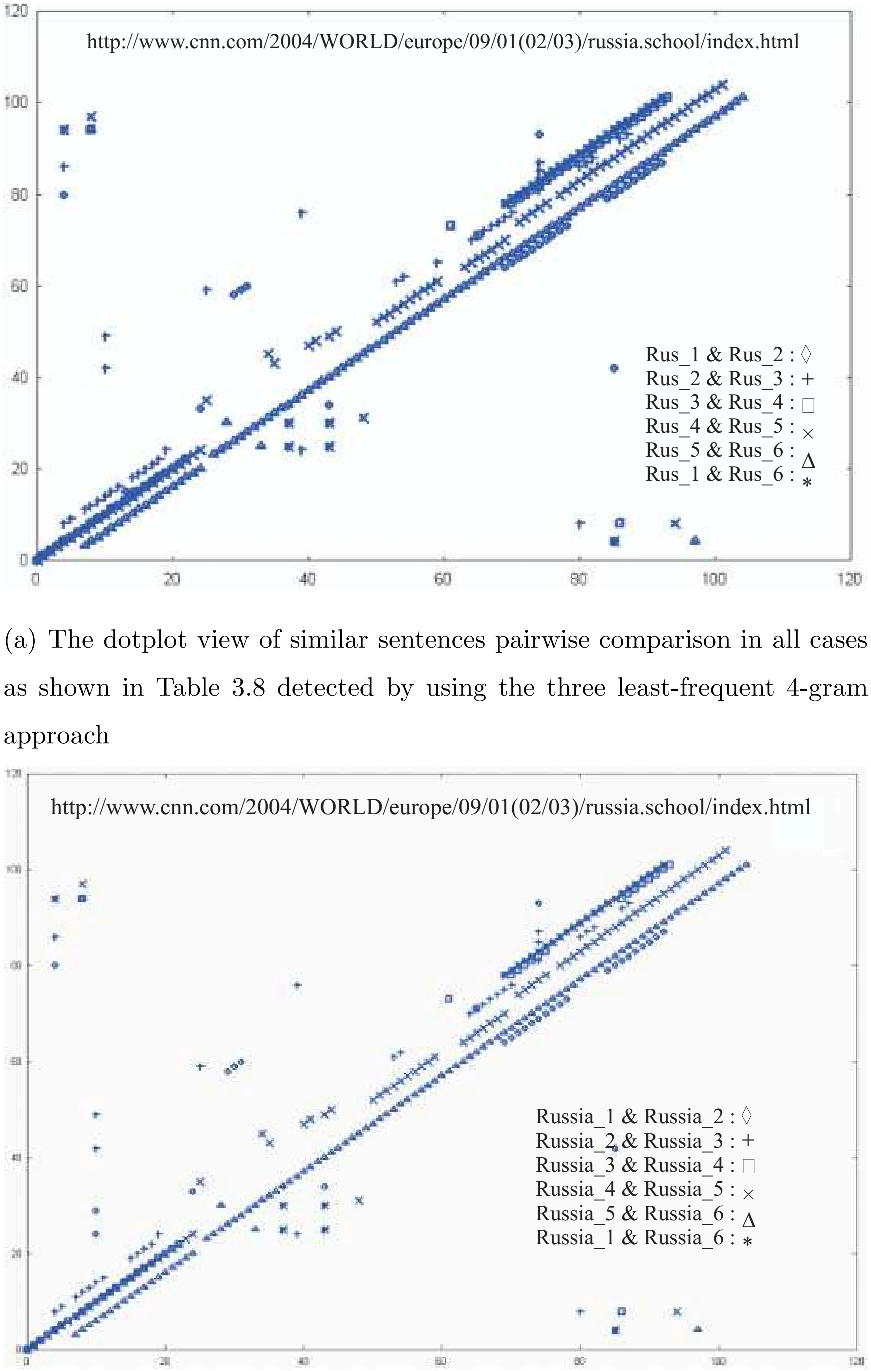
Figure 3.6: Dotplot views of similar sentences in Web documents 9_11_1 and 9_11_2 (as shown in Tables 3.6 and 3.7) detected by using the three least-frequent 4-gram and fuzzy-set IR approaches



(a) The dotplot view of similar sentences in Gandhi_1 and Gandhi_2 detected by using the three least-frequent 4-gram approach

(b) The dotplot view of similar sentences in Gandhi_1 and Gandhi_2 detected by using the fuzzy-set IR approach

Figure 3.7: Dotplot views of similar sentences in Web documents Gandhi_1 and Gandhi_2 (as shown in Tables 3.6 and 3.7) detected by using the three least-frequent 4-gram and fuzzy-set IR approaches



(a) The dotplot view of similar sentences pairwise comparison in all cases as shown in Table 3.8 detected by using the three least-frequent 4-gram approach

(b) The dotplot view of similar sentences pairwise comparison in all cases as shown in Table 3.8 detected by using the fuzzy-set IR approach

Figure 3.8: Dotplot views of similar sentences in different Web documents (as shown in Table 3.8) detected by using the three least-frequent 4-gram and fuzzy-set IR approaches

and 3.5 show no difference since there does not exist any similar, which are not exact, sentences in the documents compared. We observe that the number of sentences matched (using the fuzzy-set IR approach) in Figure 3.6(b) is less than the one (using the 4-gram approach) as shown in Figure 3.6(a). This is because occasional false positives are obtained when using the 4-gram approach. Since the 4-gram approach uses three least 4-grams in a sentence S to represent S, two different sentences can be represented incorrectly to be the same, if by chance the same three least 4-grams dominate the sentences. In Figure 3.7(b) more sentences are detected (by using the fuzzy-set IR approach) compared to the 4-gram approach (as shown in Figure 3.7(a)), since the corresponding documents contain some similar sentences apart from exact sentences. Figures 3.8(a) and 3.8(b) show almost the same results, since most of the sentences detected are exact matches and there are negligible, or very, few similar sentences. Overall, the fuzzy-set IR approach out-performs the 4-gram approach in terms of (i) detecting similar sentences, which can be handled by the former, but not the latter, and (ii) generating less *false positives* in detecting similar sentences than the 4-gram approach because unlike the 4-gram approach, the fuzzy-set IR approach considers (semantically the) same words in sentences to determine similarity of sentences, and (iii) producing less *false negatives* in determining similar sentences than the 4-gram approach because the 4-gram approach cannot detect similar sentences and thus treats similar sentences as different sentences.

3.5 Complexity Analysis of Our Copy Detection Approach

The overall complexity of our copy detection approach is $\mathcal{O}(|A|)(|B|)$, where $|A|$ and $|B|$ are the number of sentences in two documents A and B, respectively. The time complexity for stopword removal is $\mathcal{O}(|N|)$, where N is the number of words in a document, and the stemming complexity is $\mathcal{O}(|M| + |P|)$, where M is the number of words searched and P is the number of iterations for converting each word into its corresponding stem. The last two measurements can be ignored, since $\mathcal{O}(|A|)(|B|)$ is the dominating factor. Also, the time complexity for applying the three least-frequent 4-gram or the fuzzy-set IR approach is $\mathcal{O}(w_1)(w_2)$, where w_1 (w_2, respectively) is the number of words in a sentence (another sentence, respectively) in document A (document B, respectively). Since $\mathcal{O}(w_1)(w_2) \leq \mathcal{O}(|A|)(|B|)$ holds, $\mathcal{O}(|A|)(|B|) \cong \mathcal{O}(|A|^2)$.

Chapter 4

Detecting Similar HTML Documents

In the previous chapter, we discussed our copy detection approaches, i.e., the three least-frequent 4-grams approach and the fuzzy-set IR model. Experimental results show that the fuzzy-set IR approach out-performs the three least-frequent 4-grams approach, after analyzing the errors generated and the accuracy obtained using the two approaches. Since the fuzzy-set IR approach is a more sophisticated tool than the three least-frequent 4-grams approach in copy detection, we adopt the former to detect the degree of similarity/dissimilarity between two HTML documents. HTML documents are chosen instead XML, other semi-structured documents, or plain text documents, since HTML documents are widely used and more prevalent on the Web.

In this chapter, we present our approach in determining to what extent the contents of different HTML documents are in common, i.e., detecting similar/dissimilar HTML documents. We first discuss *odd ratio*, which is used to precisely capture the degree of overlapping between two HTML documents according to the *degrees of resemblance* (to be defined below) between the documents. Hereafter, we introduce our *tree matching* approach that determines and depicts the overlapped portions of two HTML documents graphically using their corresponding semantics hierarchies and a dotplot view. We justify the correctness of our similar document detection approach by verifying the accuracy of the degree of similarity between any two sentences in terms of the false positives, false negatives, precision and recall, and F-measure values.

4.1 Capturing Document Content Using Semantic Hierarchies

Data in HTML documents are hierarchical and semi-structured in nature. In order to extract the content of an HTML document, the document is passed through our semantic hierarchy construction tool from where the text (not tags) of the document is extracted. The tool takes an HTML document D as input and returns a tag-less, hierarchical structure of data in D according to the HTML grammar.

During the process of constructing the semantic hierarchy portion for non-table data, HTML tags are divided into different levels, which include headings, block-level, text-level (font, phrase, form, etc.), and addresses. These tags determine the relative locations of different portions of data content in the hierarchy, and the data content are clustered, merged, and promoted during the construction process of the hierarchy. To create the semantic hierarchy portion for HTML table data, if they exist, the hierarchical dependencies (e.g., row and column order) among the data content in the table are determined using various HTML table tags. Figure 4.1(b) (Figure 4.2(b), respectively) depicts the semantic hierarchy of the HTML document in Figure 4.1(a) (Figure 4.2(a), respectively). As shown in Figure 4.1(b), the root node of the semantic hierarchy is $Sistani_1$.html, which is the name of the document, and individual nodes display the content of the document. Content of these nodes in the two documents are processed (as explained in Section 4.3) and are later compared by using the tree matching algorithm.

On computing the similarity of the two (HTML) documents D_1 and D_2, we calculate the two similarity values, i.e., similarity of D_1 with respect to D_2, and visa versa. Since it is intuitive, as well as convenient, to obtain a single value that indicates the degree of similarity between D_1 and D_2, we calculate the odds ratio (as defined in the next section) between D_1 and D_2. Note that the greater the odds ratio value of D_1 and D_2, is the more similar D_1 and D_2 are.

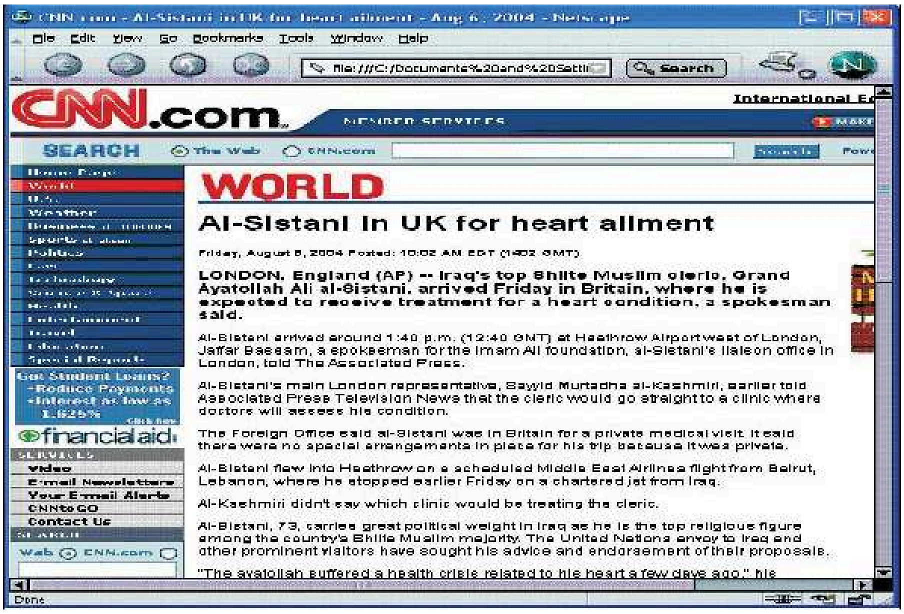
4.2 Odds Ratios

We can compute the number of similar sentences appeared in any two documents using the formula EQ in Equation 3.4. Using EQ, we define the *degree of resemblance* (RS) of document doc_1 with respect to document doc_2, which counts the total number of similar sentences in doc_1 appeared in doc_2 as follows:

$$RS(doc_1, doc_2) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} EQ(S_{d_{1_i}}, S_{d_{2_j}})}{m} \tag{4.1}$$

where $S_{d_{1_i}}$ $(1 \leq i \leq m)$ is a sentence in doc_1, $S_{d_{2_j}}$ $(1 \leq j \leq n)$ is a sentence in doc_2, and m (n, respectively) is the total number of sentences in doc_1 (doc_2, respectively). $RS(doc_2, doc_1)$ can be defined accordingly.

After obtaining the degrees of resemblance between any two documents doc_1 and doc_2, we represent these two degrees with a single value, which depicts to what extent the relative *degrees of resemblance* between doc_1 and doc_2 is. These single values of different pairs of documents, e.g., between doc_1 and doc_2, and between doc_1 and doc_3, can be used to compare the relative degree of similarity among different documents. We obtain this single value using *odd ratio*, which comes with the property that the



WORLD

Al-Sistani in UK for heart ailment

Friday, August 6, 2004 Posted: 10:02 AM EDT (1402 GMT)

LONDON, England (AP) -- Iraq's top Shiite Muslim cleric, Grand Ayatollah Ali al-Sistani, arrived Friday in Britain, where he is expected to receive treatment for a heart condition, a spokesman said.

Al-Sistani arrived around 1:40 p.m. (12:40 GMT) at Heathrow Airport west of London, Jaffar Bassam, a spokesman for the Imam Ali foundation, al-Sistani's liaison office in London, told The Associated Press.

Al-Sistani's main London representative, Sayyid Murtadha al-Kashmiri, earlier told Associated Press Television News that the cleric would go straight to a clinic where doctors will assess his condition.

The Foreign Office said al-Sistani was in Britain for a private medical visit. It said there were no special arrangements in place for his trip because it was private.

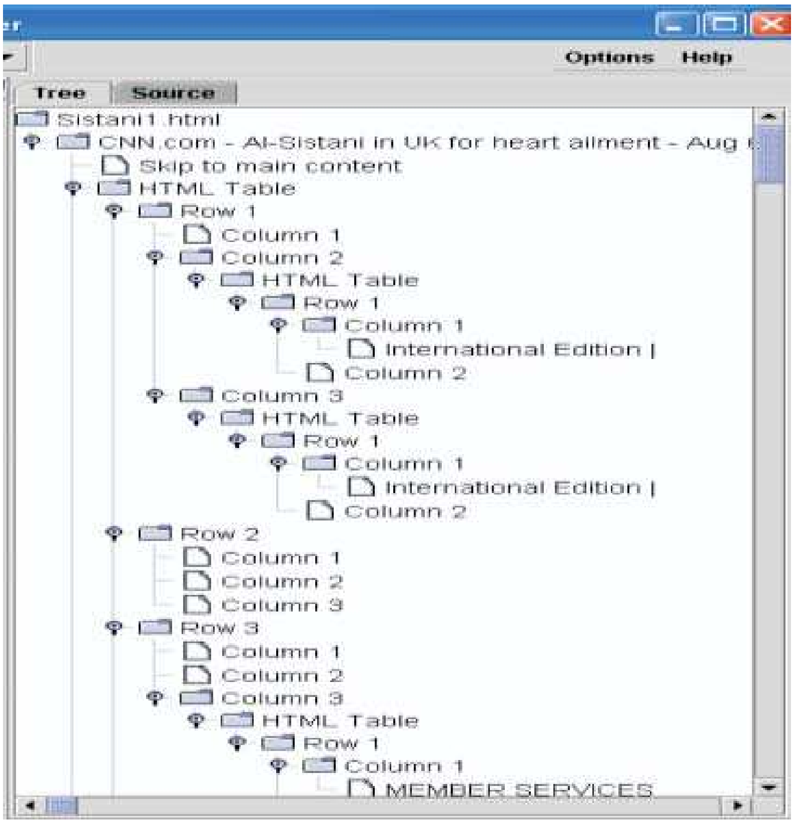
Al-Sistani flew into Heathrow on a scheduled Middle East Airlines flight from Beirut, Lebanon, where he stopped earlier Friday on a chartered jet from Iraq.

Al-Kashmiri didn't say which clinic would be treating the cleric.

Al-Sistani, 73, carries great political weight in Iraq as he is the top religious figure among the country's Shiite Muslim majority. The United Nations envoy to Iraq and other prominent visitors have sought his advice and endorsement of their proposals.

"The ayatollah suffered a health crisis related to his heart a few days ago," his

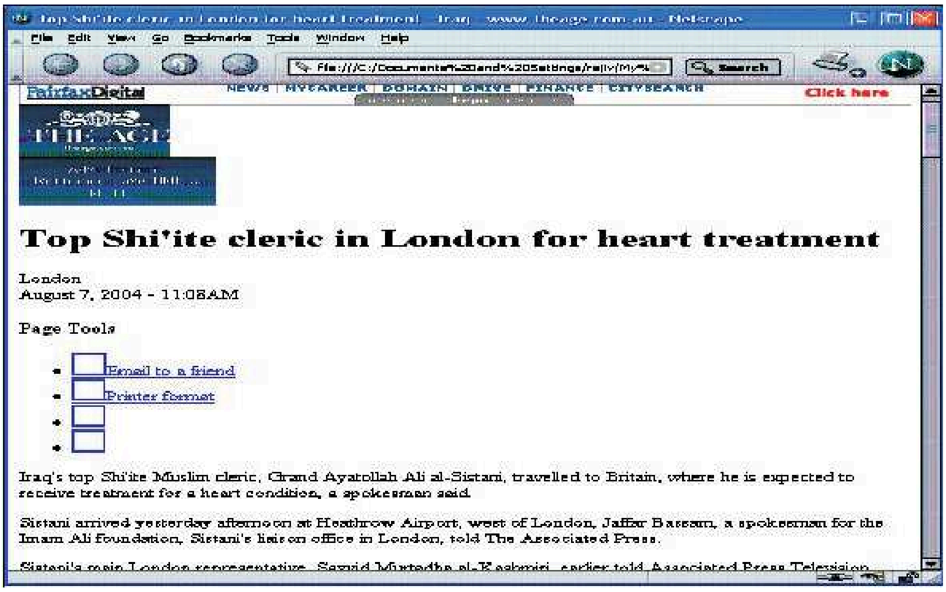
(a) *Sistani*$_1$.html (http://www.cnn.com/2004/WORLD/europe/08/06/sistani.london.ap/index.html?headline=Al-Sistani~in~UK~for~heart~ailment) posted on 08/06/04



```
Sistani1.html
  CNN.com - Al-Sistani in UK for heart ailment - Aug 6
    Skip to main content
    HTML Table
      Row 1
        Column 1
        Column 2
          HTML Table
            Row 1
              Column 1
                International Edition |
              Column 2
        Column 3
          HTML Table
            Row 1
              Column 1
                International Edition |
              Column 2
      Row 2
        Column 1
        Column 2
        Column 3
      Row 3
        Column 1
        Column 2
        Column 3
          HTML Table
            Row 1
              Column 1
                MEMBER SERVICES
```

(b) The semantic hierarchy of *Sistanil*$_1$.html in Figure 4.1(a)

Figure 4.1: An HTML document *Sistanil*$_1$.html, a news article, and its semantic hierarchy



File Edit View Go Bookmarks Tools Window Help
Search
FairfaxDigital
NEWS | MYCAREER | DOMAIN | DRIVE | FINANCE | CITYSEARCH
Click here

Top Shi'ite cleric in London for heart treatment

London
August 7, 2004 - 11:08AM

Page Tools

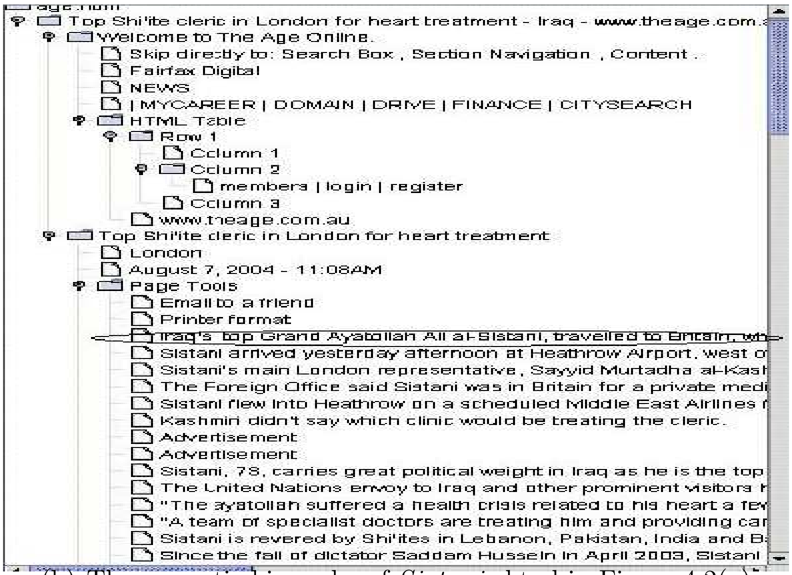
- Email to a friend
- Printer format

Iraq's top Shi'ite Muslim cleric, Grand Ayatollah Ali al-Sistani, travelled to Britain, where he is expected to receive treatment for a heart condition, a spokesman said.

Sistani arrived yesterday afternoon at Heathrow Airport, west of London, Jaffar Bassam, a spokesman for the Imam Ali foundation, Sistani's liaison office in London, told The Associated Press.

Sistani's main London representative, Sayyid Murtadha al-Kashmiri, earlier told Associated Press Television

(a) *Sistani*$_2$.html (http://www.theage.com.au/arti-cles/2004/08/07/1091732-126841.html?oneclick=true) posted on 08/07/04



```
Top Shi'ite cleric in London for heart treatment - Iraq - www.theage.com.a
    Welcome to The Age Online.
        Skip directly to: Search Box , Section Navigation , Content .
        Fairfax Digital
        NEWS
        | MYCAREER | DOMAIN | DRIVE | FINANCE | CITYSEARCH
        HTML Table
            Row 1
                Column 1
                Column 2
                    members | login | register
                Column 3
            www.theage.com.au
    Top Shi'ite cleric in London for heart treatment
        London
        August 7, 2004 - 11:08AM
        Page Tools
            Email to a friend
            Printer format
            Iraq's top Grand Ayatollah Ali al-Sistani, travelled to Britain, wh
            Sistani arrived yesterday afternoon at Heathrow Airport, west o
            Sistani's main London representative, Sayyid Murtadha al-Kash
            The Foreign Office said Sistani was in Britain for a private medi
            Sistani flew into Heathrow on a scheduled Middle East Airlines f
            Kashmiri didn't say which clinic would be treating the cleric.
            Advertisement
            Advertisement
            Sistani, 73, carries great political weight in Iraq as he is the top
            The United Nations envoy to Iraq and other prominent visitors h
            "The ayatollah suffered a health crisis related to his heart a few
            "A team of specialist doctors are treating him and providing car
            Sistani is revered by Shi'ites in Lebanon, Pakistan, India and B
            Since the fall of dictator Saddam Hussein in April 2003, Sistani
```

(b) The semantic hierarchy of *Sistani*$_2$.html in Figure 4.2(a)

Figure 4.2: An HTML document *Sistani*$_2$.html, a news article, and its semantic hierarchy

higher the odds ratio between two documents is, the more similar the two documents are. Odds ratio, which is also called *odds* in the literature, is defined as the ratio of the probability (p) that an event occurs to the probability (1 - p) that it does not, i.e.,

$$\text{Odds ratio} = p/(1-p) \tag{4.2}$$

The reasons to adopt odds ratio is three fold. First, it is easy to compute. Second, it is a natural, intuitively acceptable way to express magnitude of association. Third, it can be linked to other statistical methods, such as Bayesian Statistical Modeling, Dempster-Shafer theory of evidence, and probability analysis. In Equation 4.2, p and (1 - p) are *odds.* The ratio gives the *positive* versus *negative* value, given that p is positive and (1 - p) is negative. As (1 - p) approaches to one, the odds ratio reaches infinity. To obtain p as defined in odds ratio, we apply the *Dempster-Shafer's theory of evidence* [SG90], which combines two or more evidences or propositions to obtain a single proposition, and in our case the evidences or propositions are degrees of resemblance between two documents.

The Dempster's combination rule computes a measure of agreement between two bodies of evidence concerning various propositions discerned from a common frame of discernment [RL02]. According to this rule, which is based on *independent items of evidence*, if the probability for evidence E_1 to be reliable is P_1 and if the probability for evidence E_2 to be reliable is P_2, then probability that both E_1 and E_2 are reliable is $P_1 \times P_2$.

Applying the Dempster-Shafer rule on the degrees of resemblance $RS(doc_1,\ doc_2)$ and $RS(doc_2,\ doc_1)$ between two documents doc_1 and doc_2, we obtain $RS(doc_1,\ doc_2)$ $\times$ $RS(doc_2,\ doc_1)$, which plays the role of p in odds ratio. Hereafter, we compute *odds ratio* of doc_1 and doc_2 as follows:

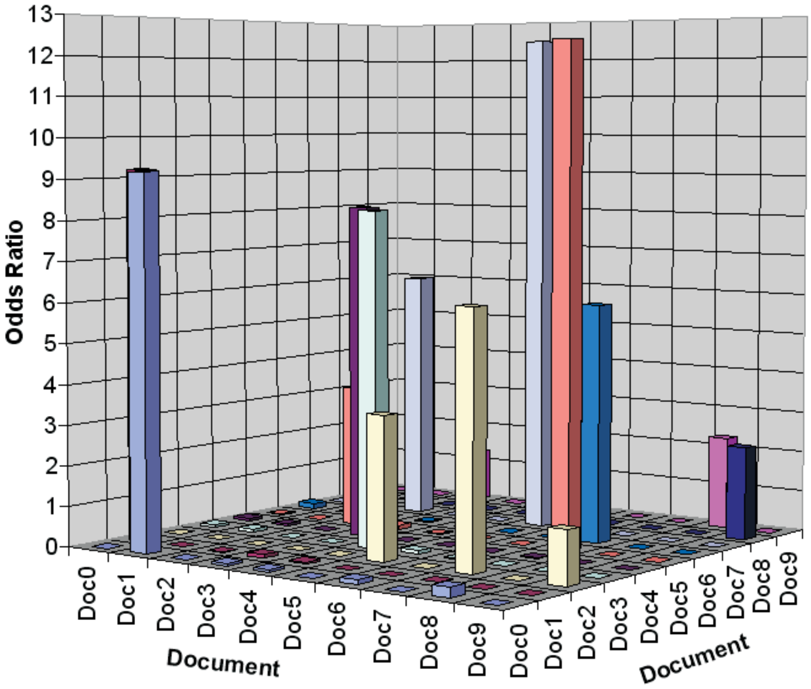


Figure 4.3: Odds ratios among Documents

$$\frac{RS(doc_1,\ doc_2)\ \times\ RS(doc_2,\ doc_1)}{1\ -\ (RS(doc_1,\ doc_2)\ \times\ RS(doc_2,\ doc_1))} \tag{4.3}$$

The odds ratio, along with the Dempster-Schafer's rule of combining evidence, serves the purpose of deriving a single measure between two documents, which reflects how similar the documents are. The higher the odd ratios of the two documents is, the lesser the difference between the documents is, and the greater the similarity between the two documents is. For example, if the degrees of resemblance of two documents are 0.9 and 0.9, then the odds ratio is 4.26, whereas if the degrees of resemblance of the two documents are 0.1 and 0.9 instead, then the odds ratio is 0.098, which is further distant apart compared with the other two documents with the odds ratio value of 4.26. Thus, the odds ratio accurately reflects the "closeness" and "difference" of the two corresponding documents. If the product of similarities is one, we assign a value 100 as odds ratio to represent *infinity*.

	Doc_0	Doc_1	Doc_3	Doc_4	Doc_5	Doc_7
Doc_0	100	**9.21**	0.064	0.058	0.01	0
Doc_1	**9.21**	100	0.09	0.051	0.001	0.021
Doc_3	0.064	0.09	100	**8.242**	0.1	0
Doc_4	0.058	0.051	**8.242**	100	0.015	0.009
Doc_5	0.01	0.001	0.1	0.015	100	**12.4**
Doc_7	0	0.021	0	0.009	**12.4**	100

Table 4.1: Odds ratios among six documents in a collection of ten documents

Example 4 We compute odds ratios on a collection of ten Web documents, which show the overall similarity between any pair of documents, to illustrate the accuracy of the odd ratios. The collection of Web documents were retrieved from the Internet by the Google search engine, using keywords such as NAT, IP Address, ONS, etc. The computed odds ratios among all these documents are shown in Figure 4.3, where we intentionally set the odd ratios of identical pairs of documents to zero (which are supposed to be 100 instead) so that the graph is more visible. We choose six out of the ten documents, i.e., Doc_0, Doc_1, Doc_3, Doc_4, Doc_5, and Doc_7, to demonstrate their relative odd ratios as shown in Table 4.1, which indicates that Doc_0 and Doc_1 (Doc_3 and Doc_4, and Doc_5 and Doc_7, respectively) are more similar to each other than to the other documents.

Consider the two documents Doc_0 and Doc_1 in the collection and as shown in Table 4.1. Doc_0 (http://www.cisco.com/warp/public/473/lan-switch-cisco.pdf), entitled "How LAN Switches work," discusses networks, switches, switching topologies, bridging, spanning trees and VLANs, whereas Doc_1 (http://www.howstuffworks.com/lan-switch.htm/printable), entitled "How Switches work," which introduces switches from a beginner's point of view, explains the functioning of switches, networking basics, traffic, fully switched networks, switch configurations, and other topics related to switches. The degrees of resemblance between Doc_0 and Doc_1 are $RS(Doc_0, Doc_1)$ = 0.91 and $RS(Doc_1, Doc_0)$ = 0.98, and their odds ratio is 8.242, which is treated as moderately high.

Further consider two other documents, Doc_3 and Doc_4, on *routers* in the collection and as shown in Table 4.1. Doc_3 (http://www.fisica.unipg.it/˜gammaitoni/fisinfo/documenti/Routers.pdf), entitled "How Routers Work," includes topics such as directing traffic, transmitting, Mac addresses, backbone of the internet, and other topics related to routers, whereas Doc_4 (http://computer.howstuffworks.com/router.htm/printable), which has the same title as Doc_3, explains what routers are and their characteristics. The degrees of resemblance between Doc_3 and Doc_4 are $RS(Doc_3, Doc_4)$ = 0.93 and $RS(Doc_4, Doc3)$ = 0.97, and the odds ratio of Doc_3 and Doc_4 is 9.21.

In order to obtain the odds ratio of two HTML documents Doc_1 and Doc_2, we compare the semantic hierarchies of Doc_1 and Doc_2, which are tree structures that represent the data contents of Doc_1 and Doc_2, in which nodes contain a sequence of sentences to be analyzed for their degrees of similarity . In the following section, we introduce the tree matching algorithm to compare two semantic hierarchies.

4.3 Tree Matching Algorithm

We present a simple tree matching algorithm to match nodes in one semantic hierarchy with nodes in another semantic hierarchy such that the matched nodes contain at least one similar sentence. The matching captures the degree of *common data content* of the corresponding HTML documents, running in $n \ log \ n$ time complexity, where log is the logarithmic base 2 function and n is the number of nodes in a semantic hierarchy.

The proposed tree matching algorithm is (i) efficient in terms of time complexity, (ii) easy to implement, and (iii) intuitive and natural in expressing matching nodes of two different semantic hierarchies.

Intuitively, given two semantics hierarchies, which are called *source tree* and *target tree*, we match nodes in the two trees by traversing the trees *inorder*. Assume that the two semantic hierarchies to be matched are S and T, where S is the source tree and T is the target tree. Nodes in T are considered one-by-one according to the order of inorder traversal. For each node n in T, S is parsed inorder to match n, i.e., determining the (total number of) sentences in each node in S that are similar to the sentences in n. When a match of n is found in S, the node data structure of n is updated with the necessary information of the matching node m in S, which include the *Node ID* of m and the sentence positions of similar sentences in m. The node structure information of m is also updated accordingly. This process is continued till the entire tree T is parsed.

We adopt the following node data structure <*D*, *SPos*, *MNInfo*, *LEFT*, *RIGHT*> for each node N in S (T, respectively) in our tree matching algorithm:

- Data (D): Sentences in N.

- Sentence positions ($SPos$): Positions of sentences S_1, …, S_j in N are maintained in the structure (L_{S_1}, …, L_{S_j}), which are obtained according to (i) the relative order of N in the inorder traversal of the tree and (ii) the relative order of sentences in N.

- Matched_Node_info ($MNInfo$): Information on each node M on the other tree that matches N, which are stored in a linked list L, and each component of L is a triple structure

$$<Node\_ID, SP, NextLink>$$

where $Node\_ID$ is the physical address of M, SP is the sentence position of a similar sentence in M, and $NextLink$ is the pointer to the next component in L.

- Left Pointer ($LEFT$): Pointer to the left subtree.
- Right Pointer ($RIGHT$): Pointer to the right subtree.

Prior to invoking the *Tree_Matching* algorithm (given below), we assign the sentence positions of sentences in the node data structure of each node in S, as well as in T, by calling the *Update_Sentence_Pos* algorithm. This task can be achieved by traversing S (T, respectively) *inorder*. A sentence position index, called *Pos*, is initialized to zero for the first call to algorithm Update_Sentence_Pos. Note that S and T, which are two semantic hierarchies to be compared, should have already been converted into *binary trees*, since a semantic hierarchy is not necessary a binary tree.

Algorithm Update_Sentence_Pos(T, **Pos*)

Input: T, a semantic hierarchy in its binary tree format, and a sentence position index **Pos*.

Output: Updated *Pos* in the data structure of each node in T.

```
BEGIN
  1. IF $T$ has a LEFT SUBTREE, call Update_Sentence_Pos($T \rightarrow LEFT$, Pos).
  2. FOR each sentence (in the given order) in $T$, DO
     (a) $T.SPos[i{+}{+}]$ = *Pos /* $i$ is initialized to 0 before the FOR Loop */
     (b) *Pos = *Pos + 1
  3. IF $T$ has a RIGHT SUBTREE, call Update_Sentence_Pos($T \rightarrow RIGHT$, Pos).
```

```
END
```

In the proposed *Tree_Matching* algorithm, pointers to the roots of the source tree S and target tree T are passed as input arguments. The algorithm traverses trees S and T in inorder, and calls procedure *Node_Comparison* to compare sentences in any node n_1 of S with sentences in any node n_2 of T. If n_1 and n_2 contain similar sentences, the total number of similar sentences found in n_1 (n_2, respectively), as well as the Node_ID of n_1 (Node_ID of n_2, respectively), are recorded in the node data structure of n_2 (n_1, respectively). Hereafter, algorithm *Graphical_Display* is called to display the result, i.e., matched nodes in the semantic hierarchies and matched sentences in a dotplot view.

Algorithm Tree_Matching(S, T)

Input: Two semantic hierarchies, source tree S and target tree T, which are binary trees.

Output: Matched nodes and sentences in S and T.

```
BEGIN
  1. IF $T$ has a LEFT SUBTREE, call $Tree\_Matching(S, T \rightarrow LEFT)$.
  2. Call $Node\_Comparison(S, T)$.
  3. IF $T$ has a RIGHT SUBTREE, call $Tree\_Matching(S, T \rightarrow RIGHT)$.
END
```

Procedure Node_Comparison(S, N)

Input: A semantic hierarchy S and a node N.

Output: Detect similar sentences between each node in S and N.

```
BEGIN
  1. IF $S$ has a LEFT SUBTREE, call $Node\_Comparison(S \rightarrow LEFT, N)$.
```

```
2. Determine similar sentences in $S$ and $N$ using the fuzzy set IR approach.
   (a) Let $M$ be $S$. /* the node being pointed to by $S$ */
   (b) Let sentences in $M$ be $m_1, \ldots, m_k$, and let sentences in $N$ be $n_1, \ldots, n_p$.
   (c) FOR $i = 1..k$, DO
         FOR $j = 1..p$, DO
           IF $EQ(m_i, n_j) = 1$, THEN
             i. $N.MNInfo{\rightarrow}Node\_ID = M$ and $N.MNInfo{\rightarrow}SP = M.SPos[i]$.
             ii. $M.MNInfo{\rightarrow}Node\_ID = N$ and $M.MNInfo{\rightarrow}SP = N.SPos[j]$.
3. IF $S$ has a RIGHT SUBTREE, call $Node\_Comparison(S{\rightarrow}RIGHT, N)$.
4. RETURN
END
```

Algorithm Graphical_Display(T, S)

Input: Semantic hierarchies T and S.

Output: Graphically display the matched sentences in S and T in a dotplot view and matched nodes in S and T.

```
BEGIN
1. IF $T$ has a LEFT SUBTREE, call Graphical_Display($T{\rightarrow}LEFT$)
2. Let $N$ be $T$
   (a) $MNInfo := N{\rightarrow}MNInfo$.
   (b) WHILE $MNInfo{\rightarrow}NextLink \neq$ NULL /* Perform Graphical Display */
     i. $M := N{\rightarrow}MNInfo.Node\_ID$
       (I) Cnt := Cnt + 1. /* Cnt is a similar sentences counter, initialized
```

```
          to zero */
          (II) Create an edge from $M$ to $N$, if there is no edge connecting $M$ and $N$.
          (III) The pair ($M \rightarrow MNInfo.SP$, $N \rightarrow MNInfo.SP$) is plotted on dotplot
             view.
        ii. $MNInfo = MNInfo \rightarrow NextLink$
      END WHILE
      (c) Shaded region of $M$ with percentage computed by $Cnt$ / $M.|SPos|$.
  3. IF $T$ has a RIGHT SUBTREE, call Graphical_Display($T \rightarrow RIGHT$)
  4. RETURN
END
```

Example 5 Consider the two documents $Sistani_1$.html (S) and $Sistani_2$.html (T) as given in Figures 4.1(a) and 4.2(a). Their semantic hierarchies, as shown in Figures 4.1(b) and 4.2(b), respectively, are first converted into binary trees before applying the *Tree_Matching* algorithm to determine the overlap between the two documents. Figure 4.4(a) shows portions of the two trees. In the figure, Node 45 in T matches Node 46 in S, and the corresponding node data structure are updated. Following is the similar sentence found in Nodes 46 (in S) and 45 (in T):

> "Iraq's top Shiite Muslim cleric, Grand Ayatollah Ali al-Sistani, arrived Friday in Britain, where he is expected to receive treatment for a heart condition, a spokesman said." (A sentence in node 46 of S)

> "Iraq's top Shi'ite Muslim cleric, Grand Ayatollah Ali al-Sistani, travelled to Britain, where treatment is expected to be received by him for a heart condition, a spokesman said." (A sentence in node 45 of T)

Note that 33% of the area in Node 45 is shaded, since one out of 3 sentences in Node 45 is similar to the corresponding sentence in Node 46 of S. Similarly, Node 49 of T

has 75% shaded area, since three of the 4 sentences in the node match with sentences (resided at different nodes) in S.

The dotplot view of the two documents is shown in Figure 4.4(b), which depicts the relative sentence positions of similar sentences in the two documents. Overall, thirteen similar sentences from both documents are found as shown in Figure 4.4(b).

Even though there are grammatical differences (indicated by the bold italicized text) in the following three pairs of sentences (out of the 13 sentence pairs), our approach successfully identifies each pair to be semantically the same.

1. In sistani1.html: Al-Sistani arrived ***around 1:40 p.m. (12:40 GMT)*** at Heathrow Airport west of London, Jaffar Bassam, a spokesman for the Imam Ali foundation, ***al***-Sistani's liaison office in London, told The Associated Press.

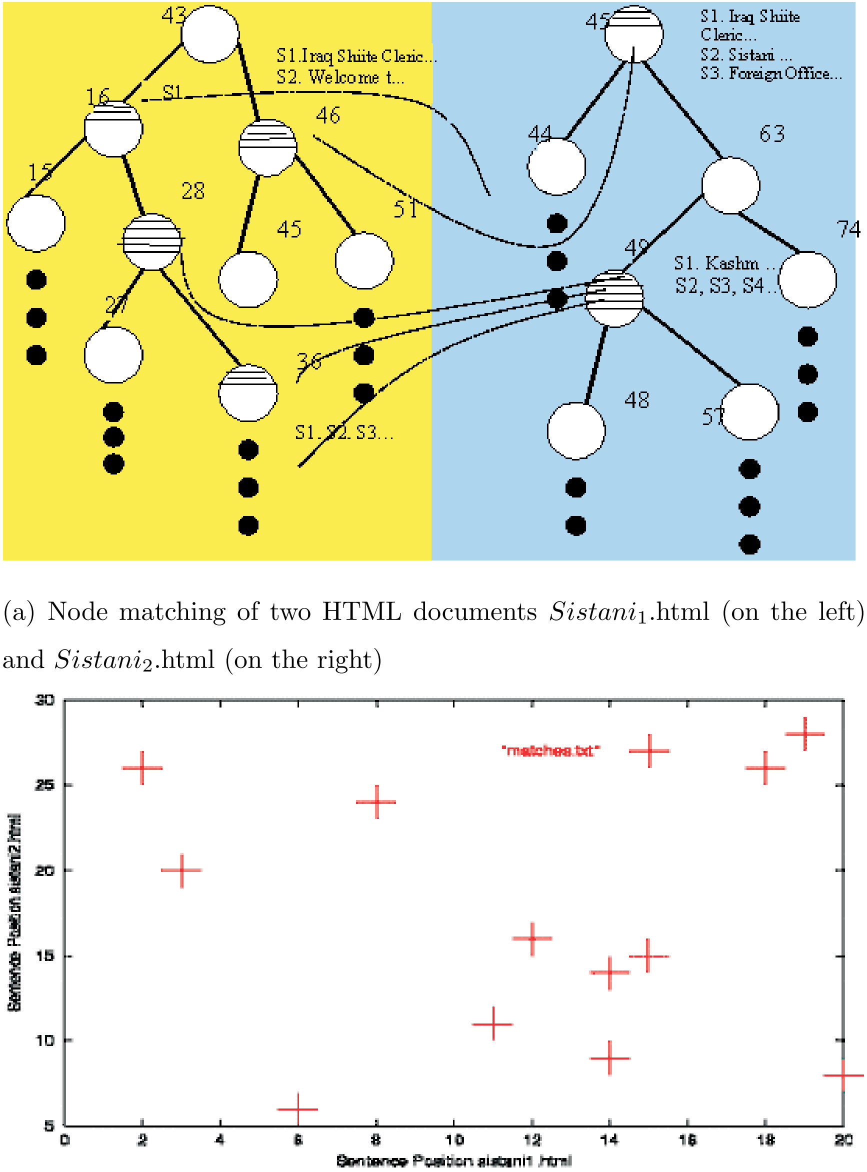
In sistani2.html: Sistani arrived ***yesterday*** afternoon at Heathrow Airport, west of London, Jaffar Bassam, a spokesman for the Imam Ali foundation, Sistani's liaison office in London, told The Associated Press.

2. In sistani1.html: ***LONDON, England (AP)*** – Iraq's top Shiite Muslim cleric, Grand Ayatollah Ali al-Sistani, ***arrived Friday in*** Britain, where ***he is expected to receive treatment*** for a heart condition, a spokesman said.

In sistani2.html: Iraq's top Shi'ite Muslim cleric, Grand Ayatollah Ali al-Sistani, ***travelled to*** Britain, ***where treatment is expected to be received by him*** for a heart condition, a spokesman said.

3.In sistani1.html: ***Al***-Sistani flew into Heathrow on a scheduled Middle East Airlines flight from Beirut, Lebanon, where he stopped earlier ***Friday*** on a chartered jet from Iraq.

In sistani2.html: Sistani flew into Heathrow on a scheduled Middle East Airlines flight from Beirut, Lebanon, where he stopped earlier ***yesterday*** on a chartered jet from Iraq.



(a) Node matching of two HTML documents $Sistani_1$.html (on the left) and $Sistani_2$.html (on the right)

(b) Sentence matching of two documents $Sistani_1$.html and $Sistani_2$.html

Figure 4.4: Matching nodes and sentences in the documents $Sistani_1$.html (S) and $Sistani_2$.html (T)

4.4 Experimental Results

To determine the similarity between any two documents, their odds ratio is computed and used to determine the extent of resemblance between the two documents. Since an odds ratio, which is derived from the well-established mathematical model Dempster Schaffer's theory of evidence [SG90], combines two *degrees of resemblance* to obtain a single measure, the next logical step in verifying the correctness of our similar documents detection approach is to verify the degrees of resemblance of two documents under consideration.

The degree of resemblance of document D_1 with respect to another document D_2 counts the number of common (i.e., similar) sentences in D_1 and D_2 i.e., the number of EQ values, each of which is partially determined by the permissible threshold value, a value set to obtain the minimal similarity between any two sentences S_1 and S_2 in our copy detection approach, and the variation threshold value, which sets the maximal, allowable difference in sentence sizes between S_1 and S_2, along with the *degrees of similarity* of S_1 and S_2. The threshold values were set according to the test set of documents as discussed in Section 3.2.2, and they have been verified to be accurate, since using the threshold values we observe minimum false positives and false negatives in determining similar sentences. The degree of similarity, on the other hand, is computed by using the well-established fuzzy set IR theory that uses the correlation factor between any pair of words in the sentences under consideration for which we justify its correctness below.

We verified the correctness of the proposed *degree of similarities* between any two sentences by choosing similar sentences manually from various Web sources as shown in Table 4.2, which have been verified to be similar by us, as well as by the document sources. In addition, different pairs of sentences have been randomly selected from various documents obtained from the Web, such as TREC, Guttenberg, etc., which include a number of similar and dissimilar sentences.

The source documents, as shown in Table 4.2, were obtained by (i) posting keyword search queries, such as "Similar sentence matching," "Similar sentence rules," etc., to the Google search engine, which is an easy way to tap into the huge resource of documents with similar sentences, and (ii) documents in text databases, such as TREC and Guttenberg, in which pairs of similar sentences, as indicated by the source documents based on English grammar [QBD04a] and [QBD04b], were extracted. Most of the similar sentences fall into one or more of the following categories of similar sentences, which have been defined by [IKL02] and [Sim]:

1. *Equivalence*: equivalent sentence pairs with minor differences in content but with similar meaning. These kinds of sentences are paraphrases. The main content of the sentences is similar in meaning, but different lexical items are used to express the same content.

2. *Anaphora*: the deliberate repetition of a word or expression (that acts as a prefix) at the beginning of several successive sentences, which make them similar.

3. *Subsumption*: one sentence is a superset of (i.e., containing more similar words than) the other.

4. *Structural*: shared content but different rhetorical structures.

5. *Context*: same event but details different emphasis.

6. *Voice*: active versus passive voices.

7. *Tense*: sentences with change in tenses.

Sources	Number of similar pairs of Sentences	Number of dissimilar pairs of Sentences
Active and passive sentences (http://www.geocities.com/fifth-grade-tpes/active-passive.html)	25	0
Anaphora (http://plato.stanford.edu/entries/anaphora/)	77	0
Sentences gleaned over 18 months from news articles (http://research.microsoft.com/research/nlp/msr-paraphrase.htm)	32	0
Active and passive sentences (http://www.primary-resources.co.uk/english/passive.htm)	22	0
Similar sentences (http://www.lavc.edu/Wcweb/active passive.html)	15	0
Others (TREC, Guttenberg etc.)	30	1,880
Total	201	1,880

Table 4.2: Sources of Web documents from where similar and dissimilar sentences were extracted

We specify in Table 4.2 the sources and the total number of sentences obtained from them. Out of the 2,081 sampling sentences, 201 sentences have been picked manually from the sources and tested for their degrees of similarity using our copy detection tool, and the rest have been randomly chosen from TREC, Guttenberg, etc., to obtain their degrees of similarity and were further verified for their similarity/dissimilarity. Tables 4.3, 4.4, 4.5, and 4.6 show some sample sentences that have been verified to be similar or different.

As shown in Tables 4.3, 4.4, 4.5, and 4.6 sentence pairs 2, 5, 6, 8, and 10 are discovered to be dissimilar sentences by our copy detection tool, whereas sentence

Sno	Sentence 1	Sentence 2	Sim (S_1, S_2)	Sim (S_2, S_1)	EQ	Similar Sentence Type
(1)	The waiter dropped the tray of food.	The tray of food was dropped by the waiter.	0.95	0.95	1	6
(2)	The Senate Select Committee on Intelligence is preparing a blistering report on prewar intelligence on Iraq.	A powerful US Congressional Committee due to the report will criticize American intelligence leading up to the war on Iraq, officials said today.	0.87	0.73	0	N/A
(3)	An autopsy found Hatab's death was caused by "strangulation/asphyxiation," Rawson said Thursday.	An autopsy found that Nagem Sadoon Hatab's death on 12th Friday was caused by "strangulation/asphyxiation, Marine spokesman Dan Rawson said thursday.	0.86	0.91	1	1

N/A: Dissimilar sentences.

Table 4.3: Sample sentence pairs 1, 2, and 3 that have been manually verified for correctness and detected as either similar or dissimilar by our copy detection approach

Sno	Sentence 1	Sentence 2	Sim (S_1, S_2)	Sim (S_2, S_1)	EQ	Similar Sentence Type
(4)	"United is continuing to deliver major cost reductions and is now coupling that effort with significant unit revenue improvement," chief financial officer Jake Brace said in a statement.	"United is continuing to deliver major cost reductions and is now coupling that effort with significant unit revenue improvement," he said.	0.85	0.83	1	3
(5)	To save time, the paper was written on a computer.	To save time and to finish on time, Kristin wrote using pen and a paper.	0.61	0.41	0	N/A

N/A: Dissimilar sentences.

Table 4.4: Sample sentence pairs 4 and 5 that have been manually verified for correctness and detected as either similar or dissimilar by our copy detection approach

Sno	Sentence 1	Sentence 2	Sim (S_1, S_2)	Sim (S_2, S_1)	EQ	Similar Sentence Type
(6)	There is a dire need in our country to provide a platform for young writers so that they can flourish and reach up to the levels of Amitav Ghosh.	Seeking to lay off workers without taking the blame, the CEO hired consultants to break the bad news.	0.23	0.1	0	N/A
(7)	He traveled to all parts of the world before reaching Italy, his hometown.	He will travel to all parts of the world before he reaches Italy, his hometown.	0.91	0.95	1	7
(8)	We are way behind when it comes to reading.	The children ate the cookies.	0.63	0.11	0	N/A
(9)	But Secretary of State Colin Powell brushed off this possibility today.	Secretary of State Colin Powell last week ruled out a non-aggression treaty.	0.85	0.97	1	2

N/A: Dissimilar sentences.

Table 4.5: Sample sentence pairs 6, 7, 8, and 9 that have been manually verified for correctness and detected as either similar or dissimilar by our copy detection approach

Sno	Sentence 1	Sentence 2	Sim (S_1, S_2)	Sim (S_2, S_1)	EQ	Similar Sentence Type
(10)	He was taken for a ride in the car.	John jumped into the air	0.25	0.12	0	N/A
(11)	The search feature works with around 8 titles from 7 publishers, which translates into some 6 million pages of searchable text.	This innovative search feature lets Amazon customers search the full text of a title to find a book, supplementing the existing search by author or title.	0.89	0.85	1	4
(12)	A Hunter Valley woman sentenced to 3 years jail for killing her four babies was only a danger to children in her care, a court was told.	As she stood up yesterday to receive a sentence of 2 years for killing her four babies, Kathleen Folbigg showed no emotion.	0.83	0.85	1	5

*N/A: Dissimilar sentences.

Table 4.6: Sample sentence pairs 10, 11, and 12 that have been manually verified for correctness and detected as either similar or dissimilar by our copy detection approach

Goups	FP(%)	FN(%)	Number of FP	Number of FN
(1)	3	13	28	13
(2)	4	9	37	9
Average	3.5	11	32.5	11

F(alse)P(ositive): Sentences that are *different* but are treated as the *same*
F(alse)N(egative): Sentences that are the *same* but are treated as *different*

Table 4.7: False positives and false negatives on the 2,081 pairs of sentences

pairs 1, 3, 4, 7, 9, 11, and 12 are detected to be similar sentences by our copy detection tool with each one of them falling into one of the seven similar sentence categories. The first pair of similar sentences in 4.3 are of type 6, i.e., sentences that are similar and only different in their voice. Similarly sentence pair 12 in Table 4.6 is of type 5, since both sentences describe the same event (a women sentence to jail) but in different contexts. Note that intuitively in sentence pair 9 in Table 4.5, $Sim(S_1, S_2)$ should have higher value, whereas $Sim(S_2,S_1)$ should have lower value, not the other way as shown in the table, since S_2 has more words than S_1. The reason for this behavior is that even though S_2 has more words than S_1 after stop words removal both sentences have equal number of words, i.e., 7, as shown below.

S_1 : Secretary, state, colin, powell, brush, possibility, today.

S_2: Secretary, state, colin, powell, week, nonaggression, treaty.

The uncommon words in S_2 have greater correlation factor with words in S_1 which gives it higher degree of similarity value when compare to S_1. Same argument can be used to explain similar behavior in sentence pair 11, i.e., $Sim(S_1, S_2)$ should be less than $Sim(S_2, S_1)$ in Table 4.6.

Based on the degrees of similarity compiled by using the 2,081 pairs of similar and

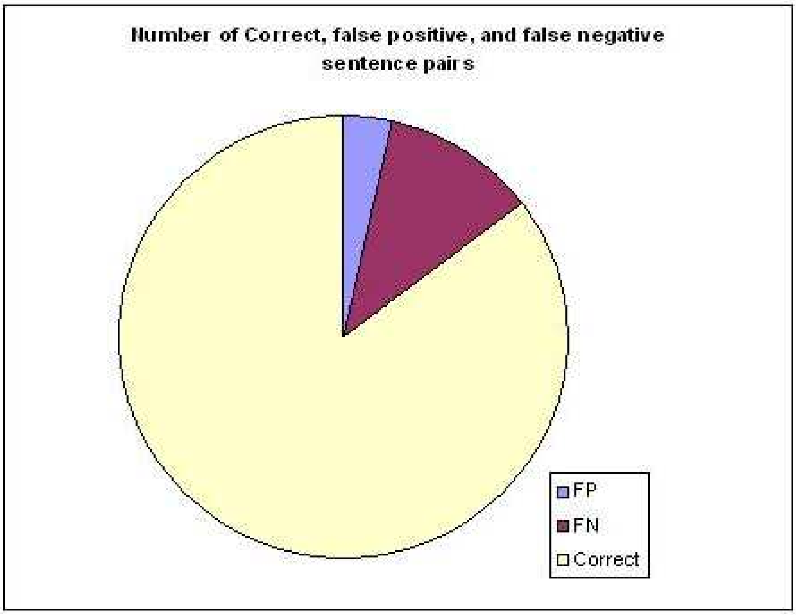


Figure 4.5: Total number of correct sentence pairs detected, false positives, and false negatives

dissimilar sentences, our task now is to verify the correctness of the computed degrees of similarity, which determine the correctness of our similar document detection approach. Upon computing the EQ values for all the 2,081 pairs of sentences using their corresponding degrees of similarity, the results are analyzed for false positives and false negatives. Again, a *false positive* occurs when a pair of sentences is dissimilar, but is detected as similar by our copy detection tool, whereas a *false negative* occurs when a pair of sentences is similar, but is detected as dissimilar by us. Prior to determining the false positive and false negative ratios among the sentence pairs, we divided the pairs of 2,081 sentences into two groups, such that Group 1 contains 1,041 sentence pairs, whereas the other group, Group 2, consists of 1,040 sentence pairs. Of the 2,081 pairs of sentences, 201 pairs that are similar are randomly divided into the Groups 1 and 2 of 100 and 101 pairs, respectively and the remaining 1,880 pairs are divided into half and combined with the 100 and 101 pairs, respectively, thus forming the two groups (i.e., test sets) of sentence pairs. The number of false positives observed in Group 1 accounts for 3%, and the number of false negatives observed yields 13%, whereas the number of false positives observed in Group 2 sums up to 4%, while the number of false negatives observed adds up to 9%.

Figure 4.5 shows the number of false positives, false negatives, and the correct sentence pairs detected among the 2,081 pairs of sentences tested for similarity. According to these experimental results, we observe that the numbers of false positives are consistently less than the number of false negatives. This behavior is attributed to the high permissible threshold value in *EQ* set during the design process so that number of similar sentences can be detected accurately. The number of false positives and false negatives can be increased or decreased by changing the permissible threshold value. If the permissible threshold value is increased, then the number false positives decreases and the number false negatives increases, whereas as the permissible threshold is decreased, the numbers of false positives increases, and the number of false negatives decreases. Thus the user of our copy detection tool has the flexibility to adjust the threshold value to control the ratios of false positives and false negatives. As shown in Table 4.7, the overall average of false positives and false negatives are 3.5% and 11%, respectively.

The false positive and false negative ratios are further converted into *precision* and *recall* ratios respectively, since (i) precision and recall are reliably and widely used in the field of IR for evaluating search strategies and (ii) a single measure, i.e., *F-Measure*, can be obtained from precision and recall ratios. F-Measure is used, since it gives an overall single measure of the errors due to precision (i.e., false positives) and (i.e., false negatives). The higher the F-Measure value is, the less error prone the system is. Note that the recall ratio used in our context is semantically different from the recall ratio used in IR in the conventional way. Recall in IR is usually referred to

Groups	Precision	Recall	F-Measure
(1)	0.97	0.88	0.92
(2)	0.96	0.91	0.93

Table 4.8: The calculated precision, recall, and F-Measure values using false positives and false negatives as shown in Table 4.7

the fraction of the relevant documents which has been retrieved; however, in our context it is referred to the fraction of the number of correctly detected similar/dissimilar sentence pairs to the total number of examined sentence pairs, which include false negative sentence pairs. Figure 4.6 shows the precision and recall ratios for the entire set of sentence pairs in Group 1, whereas Figure 4.7 shows the precision and recall ratios for the sentence pairs in Group 2. The F-Measure, which combines the false positives and false negatives, yields a single measure on the false positives and false negatives, is defined as follows:

Correct: Number of similar/dissimilar sentence pairs that are correctly identified

False Positives: A pair of sentences is dissimilar, but is detected as similar.

False Negatives: A pair of sentences is similar, but is detected as dissimilar

$\textit{Precision} = \frac{\text{Correct}}{\text{Correct + False Positives}}$

$\textit{Recall} = \frac{\text{Correct}}{\text{Correct + False Negatives}}$

$\textit{F-Measure} = \frac{2}{\frac{1}{\text{Precision}}+\frac{1}{\text{Recall}}}$, which is also called *Harmonic Mean* [BYRN99]

As indicated in Table 4.8, the F-Measure, is 0.92 for Group 1 and 0.93 for Group 2, which indicate that our copy detection approach makes very few mistakes in detecting similar/dissimilar sentences while detecting most of the sentence pairs correctly.

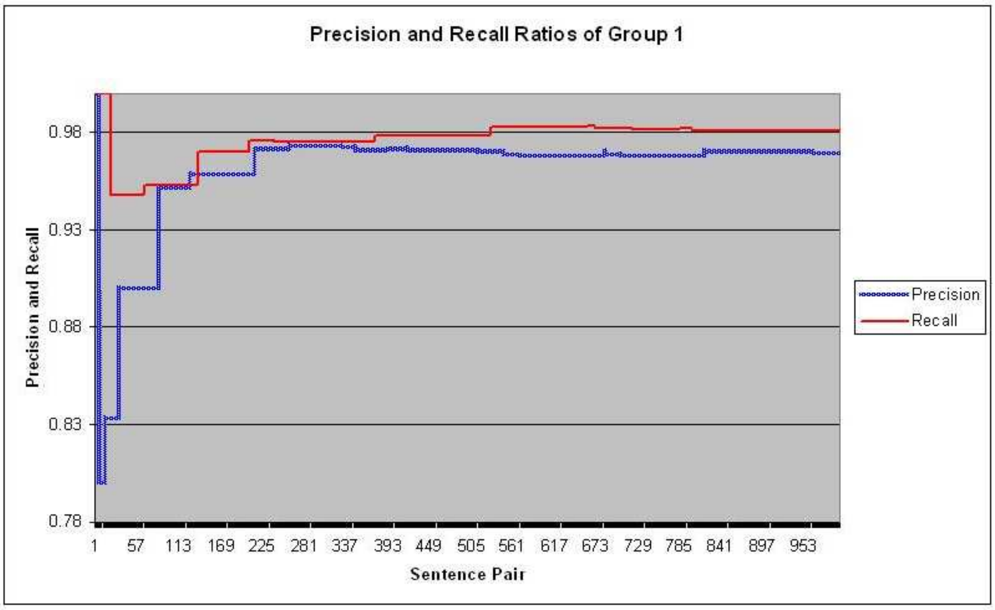


Figure 4.6: The precision and recall of the sentence pairs in Group 1

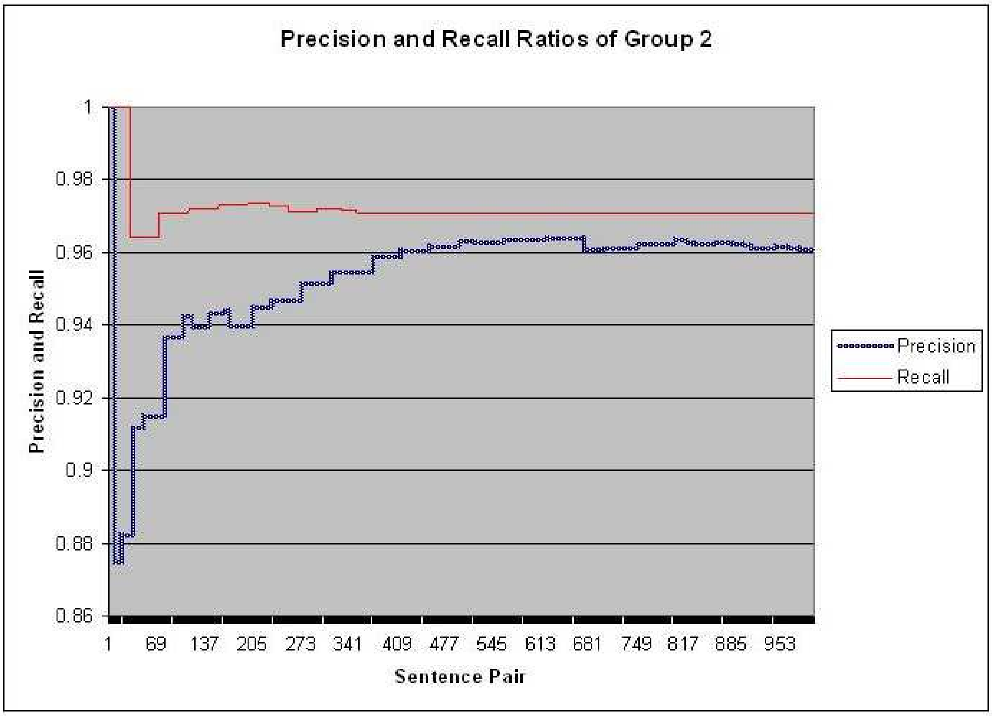


Figure 4.7: The precision and recall of the sentence pairs in Group 2

4.5 Implementation

The program that implements our copy detection approach and similar document detection approach has been written in Java on a Windows XP PC with a 3.4 GHz processor (Pentium 4) processor and a 120 GBytes hard drive. The experiments that verified the accuracy of the degrees of similarity between pairs of sentences were performed between April 20, 2005 and May 20, 2005.

4.6 Complexity Analysis

The overall complexity of our copy detection and similar HTML documents detection approaches is affected by (i) computing the degrees of similarity, (ii) finding the EQ values (i.e., equality between two sentences), (iii) determining degrees of resemblance, (iv) calculating the odds ratios, (v) the construction of tree hierarchy, and (vi) comparing the tree hierarchies, as discussed in this chapter. Since on an average the number of sentences in a document is usually greater than the number of words in a sentence, we compute the time complexity of our copy detection approach with respect to the number of sentences in a document. The time complexity for calculating the degrees of similarity, EQ, and degrees of resemblance, is $\mathcal{O}(m \times n) \cong \mathcal{O}(n^2)$, where m is the total number of sentences in a document and n is the number of sentences in another document to be computed. To determine the time complexity for detecting similar HTML documents, the time complexities for computing the odds ratios and comparing tree hierarchies are calculated. As the computation of odds ratio for any two documents is straightly determined by the degrees of resemblance of the two documents, the time complexity is $\mathcal{O}(n^2)$, where n is the number of sentences in a document. The time complexity for constructing a tree hierarchy and comparing two trees is $\mathcal{O}(m\ log\ m)$, where log is the logarithmic base 2 function and m is the number of nodes in the semantic hierarchy. Comparing two nodes in two semantic hierarchies actually requires comparing sentences resided at the corresponding nodes. Hence, $\mathcal{O}(m\ log\ m) \leq \mathcal{O}(n\ log\ n)$, where n is the number of number of sentences in a document. Note that in Chapter 3 the complexity of our copy detection approach has been analyzed to be $\mathcal{O}(|A|^2)$, where $|A|$ denotes the number of sentences in document A, which is $\mathcal{O}(n^2)$. Therefore, the overall complexity of our copy detection and similar HTML documents detection approach is $\mathcal{O}(n\ log\ n + n^2 + n^2) \cong \mathcal{O}(n^2)$.

Chapter 5

Conclusions

In this thesis we have presented an approach to detect Web documents, especially on HTML documents. We first introduced a copy detection approach, which is based on the three least-frequent 4-grams approach and the fuzzy set information retrieval (IR) model, to detect similar sentences in two documents and then introduced the odd ratio of the two documents, which reflects the degree of common content. Our copy detection approach (i) determines similar, not necessarily the same, Web documents, which can act as a filter to various Web search engines/Web crawlers to improve efficiency by indexing fewer documents and eliminating the ones that are redundant, (ii) detects similar sentences, apart from same sentences, by using the fuzzy-set IR approach on Web documents or detects same sentences using either the three least-frequent 4-gram approach and/or the fuzzy-set IR approach, and (iii) captures same (or similar) sentences in any two Web documents graphically, which displays the location of overlapping portions of the documents. Not only does our copy detection approach handle a wide range of documents (such as sports, news, science, etc.), but it is also applicable to different Web documents in different subject areas since it does not require static word lists . The time complexity for our copy detection approach is $\mathcal{O}(n^2)$, where, n is the total number of sentences in a document.

Experimental results indicate that the fuzzy-set approach outperforms the 4-grams approach for copy detection. Hence, the fuzzy-set IR approach has been chosen over the three least-frequent 4-grams for detecting similar Web documents, especially HTML documents, which (i) are abundant on the Web, (ii) are widely used to publish, and (iii) provide inter and intra-document links on the Internet. For detecting the degrees of similarities between HTML documents we use semantic hierarchy to extract data from the HTML documents, which gives a tree structure of the corresponding HTML document. We then use our tree matching algorithm to compare the two HTML documents to detect for any similarities. The time complexity for detecting similar HTML documents is $\mathcal{O}(n\ log\ n)$, where, n is the total number of sentences in a document. The overall time complexity of our copy detection and similar HTML documents detection approach is $\mathcal{O}(n\ log\ n + n^2) \cong \mathcal{O}(n^2)$.

In order to evaluate the correctness of our similar document detection approach, we have verified the correctness of the EQ value that indicates the similarity of any two sentences. This is because the EQ value is used to compute the degree of similarity between any two sentences and the sum of the EQ values of sentences in two HTML documents yields the degree of resemblance and hereafter the odds ratio between the two documents. As the odds ratios, degrees of resemblance, and degrees of similarity are supported by well-established mathematical models, that are both simple and straight forward, the verification of the EQ value to be accurate completes the evaluation procedure of our copy detection approach and similarity between documents. Our work has been published in two conference proceedings [YN05a] and [YN05b].

For future work, we would like to (i) analyze similar sentences in two Web documents using the natural-language processing approach, which could further enhance the accuracy of our copy detection approach and (ii) extend our similarity measures to handle copy detection of non-English Web documents.

Bibliography

[BDGM95] S. Brin, J. Davis, and H. Garcia-Molina. Copy Detection Mechanisms for Digital Documents. In *Proceedings of the 1995 ACM SIGMOD*, pages 398–409, 1995.

[BYRN99] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, 1999.

[Cam97] D. Campbell. A Sentence Boundary Recognizer for English Sentences. Unpublished work, 1997.

[CCB02] J. Cooper, A. Coden, and E. Brown. Detecting Similar Documents Using Salient Terms. In *Proceedings of CIKM'02*, pages 245–251, 2002.

[CCS00] D. Campbell, W. Chen, and R. Smith. Copy Detection Systems for Digital Documents. In *Proceedings of IEEE Advances in Digital Libraries*, pages 78–88, 2000.

[CP00] J.W. Cooper and J.M. Prager. Anti-Serendipity: Finding Useless Documents and Similar Documents. In *Proceedings of the 33rd Hawaii International Confernce on System Sciences*, 2000.

[Dam95] M. Damashek. Gauging Similarity with N-grams: Language-Independent Categorization of Text. *Science*, 267:843–848, 1995.

[Hel96] J. Helfman. Dotplot Patterns: A Literal Look at Pattern Languages. *Theory and Practice of Object Systems*, 2(1):31–41, 1996.

[HO82] C.M. Hoffmann and M.J. O'Donnell. Pattern Matching in Trees. *Journal of ACM (JACM)*, 39(1):68–95, January 1982.

[IKL02] A. Ibrahim, B. Katz, and J. Lin. Extracting Structural Paraphrases from Aligned Monolingual Corpora. In *Proceedings of the Second International Workshop on Paraphrasing (IWP 2003)*, pages 101– 127, 2002.

[LPF02] G.A.D. Lucca, M.D. Penta, and A.R. Fasolino. An Approach to Identify Duplicated Web Pages. In *Proceedings of COMPSAC*, pages 481–486, 2002.

[Man94] U. Manber. Finding Similar Files in Large File System. In *USENIX Winter Technical Conferences*, January 1994.

[Nev96] H. Nevin. Scalable Document Fingerprinting. In *Proceedings of the 2nd USENIX Workshop on Electroninc Commerce*, pages 191–200, November 1996.

[OMK91] Y. Ogawa, T. Morita, and K. Kobayashi. A Fuzzy Document Retrieval System Using the Keyword Connection Matrix and a Learning Method. *Fuzzy Sets and Systems*, 39:163–179, 1991.

[Por80] M.F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.

[PZ04] A.R. Pereira and N. Ziviani. Retrieving Similar Documents from the Web. *Journal of Web Engineering*, 2(4):247–261, 2004.

[QBD04a] C. Quirk, C. Brockett, and W.B. Dolan. Monolingual Machine Translation for Paraphrase Generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 201–233, 2004.

[QBD04b] C. Quirk, C. Brockett, and W.B. Dolan. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *Proceedings of COLING 2004*, pages 191–213, 2004.

[RL02] I. Ruthven and M. Lalmas. Experimenting on Dempster-Shafer's Theory of Evidence in Information Retrieval. *Journal of Intelligent Information Systems*, 19(3):267–302, 2002.

[RSF$^+$01] J. Rabelo, E. Silva, F. Fernandes, S. Meira, and F. Barros. ActiveSearch: An Agent for Suggesting Similar Documents Based on User's Prefrences. In *Proceedings of the International Conference on Systems, Men, and Cybernetics*, pages 549–554, 2001.

[SG90] D. Shafer and R. Glenn. Perspectives on the Theory and Practice of Belief Functions. In *International Journal of Approximate Reasoning*, pages 1–40, 1990.

[SGM95] N. Shivakumar and H. Garcia-Molina. SCAM: A Copy Detection Mechanism for Digital Documents. *D-Lib Magazine*, 1995. http://www.dlib.org.

[Sim] Microsoft Research. http://research.microsoft.com/research/nlp/msr_paraphrase.htm.

[TRE] Text Retrieval Conference (TREC). http://trec.nist.gov/.

[Uni] Unix Manual (Man) Pages in HTML. http://www.rt.com/man.

[WB91] I. Witten and T. Bell. The Zero Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. Technical report, University of Calgary, 1991.

[YLW$^+$99] C. Yu, K. Liu, W. Wu, W. Meng, and N. Rishe. Finding the Most Similar Documents Across Multiple Text Databases. In *Proceedings of the IEEE Forum on Research and Technology Advances in Digitial Libraries*, pages 150–162, 1999.

[YN05a] R. Yerra and Y-K. Ng. A Sentence-Based Copy Detection Approach for Web Documents. To appear in Proceedings of 2005 International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'05), August 2005.

[YN05b] R. Yerra and Y-K Ng. Detecting Similar HTML Documents Using a Fuzzy Set Information Retrieval Approach. To appear in Proceedings of IEEE International Conference on Granular Computing (GrC'05), July 2005.