Textual Similarity Analysis Using Locality Sensitive Hashing

Thesis Report

submitted in partial fulfillment of the requirements

for the award of degree of

Master of Engineering

in

Software Engineering

Submitted By

Reetika Bansal

(801431020)

Under the supervision of:

Dr. Shalini Batra

Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2016

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, ***"Textual Similarity Analysis Using Locality Sensitive Hashing"***, in partial fulfillment of the requirements for the award of degree of Master of Engineering in ***Software Engineering*** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of ***Dr. Shalini Batra*** and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Reetika

Reetika Bansal
801431020
ME(SE)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Shalini Batra)
Associate Professor,
Computer Science and Engineering Department

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S.S Bhatia)
Dean (AcademicAffairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

No volume of words is enough to express my gratitude towards my guide, **Dr. Shalini Batra**, Associate Professor, Computer Science and Engineering Department, Thapar University, who has been very concerned and has supervised the work presented in this thesis report. He has helped me to explore this vast field in an organized manner and provided me with all the ideas on how to work towards a research oriented venture.

I am also thankful to **Dr. Deepak Garg**, Head of Department, CSED and **Dr. Ashutosh Mishra**, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank the staff members and my colleagues who were always there in the need of the hour and provided with all the help and facilities, which I required, for the completion of my thesis.

Most importantly, I would like to thanks my parents, friends and the almighty for showing me the right direction out of the blue, to help me to stay calm in the oddest of the times and keep moving even at times when there was no hope.

Reetika

Reetika Bansal

(801431020)

Abstract

The rapid growth in online information requires the efficient management and usability of data. Searching similar text documents in large set of documents is one of the most challenging tasks in the era of Big data. This thesis focuses on finding similarity between documents using Approximate K-nearest neighbor (KNN) search. Shingling technique has been applied which converts text documents into sets and further the set identical (similarity) value is calculated using Jaccard similarity. Later, hash functions and shingles in each document are used to create characteristic matrix. Shingles generation has been divided in three categories: with and without stopwords, with and without blank spacing and with and without stammers. Since shingle set is quite large, the size of matrix is significant, a technique called 'minhashing' is used to reduce the size of the matrix. Minhashing creates a signature matrix which is very less in size compared to characteristic matrix with approximately same outcome. Further, finding the similarity among all pairs of column of signature matrix is still a big problem because comparing a column takes O (n^2) time. The time of comparing columns for similarity is reduced using another technique, termed as Locality Sensitive Hashing which divided the entire matrix into number of bands and buckets and similar elements lie in one bucket. The entire technique has been successfully implemented with variations in categories of shingles and size of shingles.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Approximate Nearest Neighbor
APSS	All Pair Similarity Search
BoW	Bag of Words
UH	Universal Hashing
CM	Characteristic Matrix
C2LSH	Collision Counting Locality Sensitive Hashing
D	Document
DM	Distance Measure
DS	Document Set
J-DIS	Jaccard Distance
J-SIM	Jaccard Similarity
LCS	Longest Common Subsequence
LSH	Locality Sensitive Hashing
NNS	Nearest Neighbor Search
PLEB	Point Location in Equal Balls
S	Shingle
SM	Signature Matrix
SN	Shingle Number

Chapter 1
Introduction

A basic definition of similarity search is: ‘Given a query, how to quickly find the objects most similar to the query'. The objective of similarity search, key operation in many fields of data science, is to find the k nearest neighbours for the query data in the given dataset. Many applications, including multimedia information retrieval, data mining, pattern recognition, machine learning, computer vision, biomedical databases, data compression, statistical data analysis, and etc. use similarity search in one form or any other. Advantages of using Big data analytics in similarity search is that decisions in various application and data domain that were previously based on guess work can now be made based on the available data.

Searching is the basic operation required to extract the needed information. In large collection of documents, indexing does not work well for text retrieval whereas string search is the most basic form of search operation where speed is the most significant factor. As in present scenario, with the increase in the number of applications , data objects like networks, graphs, images, videos, web pages etc. need to be searched more. So depending upon the type of application, either it makes search problem easier as additional information can be exploited or it can make search problem more complex depending upon the type of data we need to adapt to.

1.1 Similarity Search

Similarity search is the mechanism used for searching spacing of objects where the only available comparator is the similarity between any pair of objects. In a large collection of information it is becoming increasingly important, as there is no natural order in which the objects are contained. In similarity search, the similarity is quantified using pair wise distance measure.

In range of applications like pattern recognition, multimedia information retrieval, data compression, statistical data analysis, biomedical databases, computer vision and machine learning, similarity search has become a major computational task. In all the

above mentioned environments, concepts like similarity/dissimilarity are useful for searching rather than exact match.

1.1.1 Vector spaces

High dimensional vectors are used to represent data in many similarity search applications. Mostly, by comparing the features extracted between any two images or text, similarity is assessed. For example, in color histogram, a vector can be used to represent a typical image feature and where the value contained in each dimension is the density of color associated with dimension [1]. Various functions like Jaccard distance, Euclidean distance, etc .are used to measure the similarity between data, simply by calculating the distance between the two vectors, when vector spacing is used to represent the data.

1.1.2 Metric spaces

In metric space approach, user- relevant data are extracted from huge collection of items. This is done using pair-wise distance measure. Problems associated with similarity search applications are: representation of data in vector form, which is not very effective and distance measures are not effectively defined by Minkowski family functions. In each dimension, values are compared independently with others present in Minkowski functions. Now, let's consider the color histogram similarity search problem, which is better judged by the quadratic form distance [2, 3], which is not present in Minkowski family. So, in such cases metric spaces can be used to represent data. In metric space distance function need to follow the positivity, triangular inequality, identity and symmetric properties and also for representation of data no special prerequisites are required.

1.2 Classification of similarity

Similarity measure between the pair of vectors or the documents is a function measure that counts the extent of similarity. Content based and context based are the different

forms according to which similarity exists. Similarity measure between query and documents are described as follows:

- On the basis of assumed importance it is possible to rank the saved documents.
- Magnitude of retrieved set can be restrained by enforcing certain threshold over it.
- The results can be used to reformulate the original query in relevance feedback (e.g., combining a document vector with the query vector).

Figure 1.1: Law of similarity [4]

Fig 1.1 depicts the law of similarity i.e. items that are similar tends to be grouped together.

1.2.1 Lexical Similarity

In linguistics, lexical similarity of any two given languages is defined as the similarity among the word sets of these languages. The lexical similarity of 0 means there are no common words in given vocabulary and lexical similarity of 1 (or 100%) means there is a complete match between words of vocabulary.

1.2.1.1 Shingling

A document is basically a collection of characters. Therefore K-Shingles is any substring of length K found in the document. K-Shingles can be defined as the set of unique "shingles" (adjacent subsequence of tokens in a text document) that can be used to test the similarity between the documents. Shingling [5] is used to find near-duplicate items. Shingling is the general approach used to alter a document into a set. If m words are present in the document in which k- shingles are formed, then total space it takes for storage is O (km). With the increase in the count of repetitions in the document, the volume of space required goes on diminishing. For example, in a

document if the text “My name is Reetika Bansal” is present, then for k=3, the shingle set will be {“My name is”, “name is Reetika”, “is Reetika Bansal”}. Also the size of the shingles to be formed, should be large enough so that the chances of any given shingles to be present in any given document should be less. Like, K=5 is considered optimum for documents containing emails.

1.2.1.2 Bag of words

Bag of words is the method of language simplification and information retrieval in which given sentence in a text document is regarded as ‘bag of its words’ in which order of words and even grammar is not considered. However, it takes into consideration multiplicity factor. Computer- vision is the main field in which bag of word technique is used. For document classification, bag of words is common method used. Common representation used in information retrieval (IR) and natural language processing is bag of words. In this similarity method, a given document text is regarded as bag of words, taking into consideration multiplicity without considering word order and grammar used. For example, we have two text documents like (a) Seema likes to eat fruits. Mani likes fruits too. (b) Seema also likes to eat Fastfood. For the given two documents, Bag of words list formed is {“Seema”, “likes”, “to”, “eat”, “fruits”, “Mani”, “too”, “also”, “Fastfood”}.

1.3 Semantic Similarity

Semantic similarity is the mathematical tool used to measure the semantic strength i.e. how identical the given files are based upon their meaning. In a given sentence, proper meaning is found for each word, termed as Word Sense Disambiguation [6]. Finally similarity of pair of words is used to compute the similarity of sentence. Topological similarity is also used to access semantic similarity which uses various logics for guessing the span between terms or concepts. For example, a directed acyclic graph is used to represent the concepts as nodes in a semi-ordered set. In order to correlate the concepts in this scenario, is to compute the shortest- path which provide relationship

between two concept nodes. Various practices like vector space model can be used to find the semantic correspondence between words and group of words.

1.3.1 Applications of Semantic Similarity

- Biomedical informatics
- Computational Informatics
- GeoInformatics
- Natural language processing

1.4 Duplicate- Types

Over the internet, on the mirror sites, there are a lot of web pages having contents very identical with each other with just minor differences i.e. they contain duplicity. Such duplicate documents are very similar to one another, but they differ from one another by little part of text. To increase the reliability of sources on internet, it is fundamental to find and abort search results with duplicity.

1.4.1 Near Duplicate

Near Duplicate refers to finding the approximate match, whereas duplicate refers to exact match between the documents which can be distinguished with the help of fingerprints. Exact duplicate detection is one of the side-effects of Near Duplicate Detection (NDD), but is not the primary purpose, since exact duplicate can be cultivated with the much less effort. Edit –Distance measure can be used to compute syntactic similarity between text documents. Further, similarity threshold can be used to detect near duplicates like, Documents are “near duplicates’, if similarity > 80%. Similar scenes or events are also described using near- duplicates. On contrary, due to the magnified extent of near- duplicates, scanning of streaming web videos on the internet is an extreme time consuming task. Due to the diverse multimedia applications, near-duplicates can be used largely by industry, academicians and various government organizations for the detection, searching and eradication of near-duplicates for the web pages.

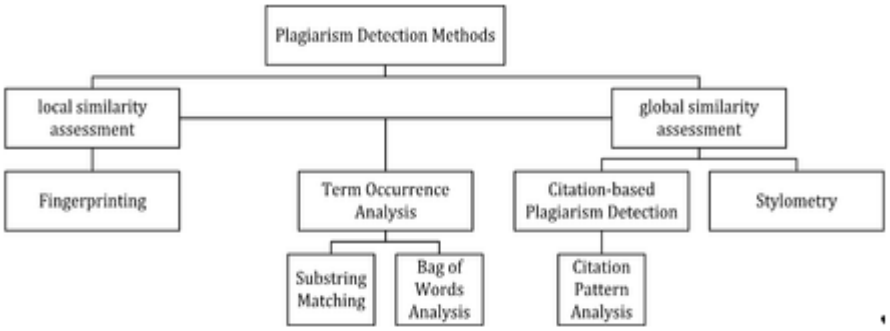

Figure 1.2: Near Duplicate [7]

1.4.2 Exact Duplicate

Exact duplicate are used to check for the given two documents, whether it is almost same or not. To check for the exact duplicates, two documents that need to be checked are compared character by character and if they differ from one another over some portion i.e. they are not similar. However, this is not the correct method to check the similarity of documents. But to erase out the redundancy in the system, it is the most acceptable method. In order to locate and delete audio, photos and documents etc., containing duplicancy, this approach is used. Figure 1.3 presents the correct duplicates of documents in the chosen folder. In the documents, duplicates are analyzed and it can be erased from the folder.

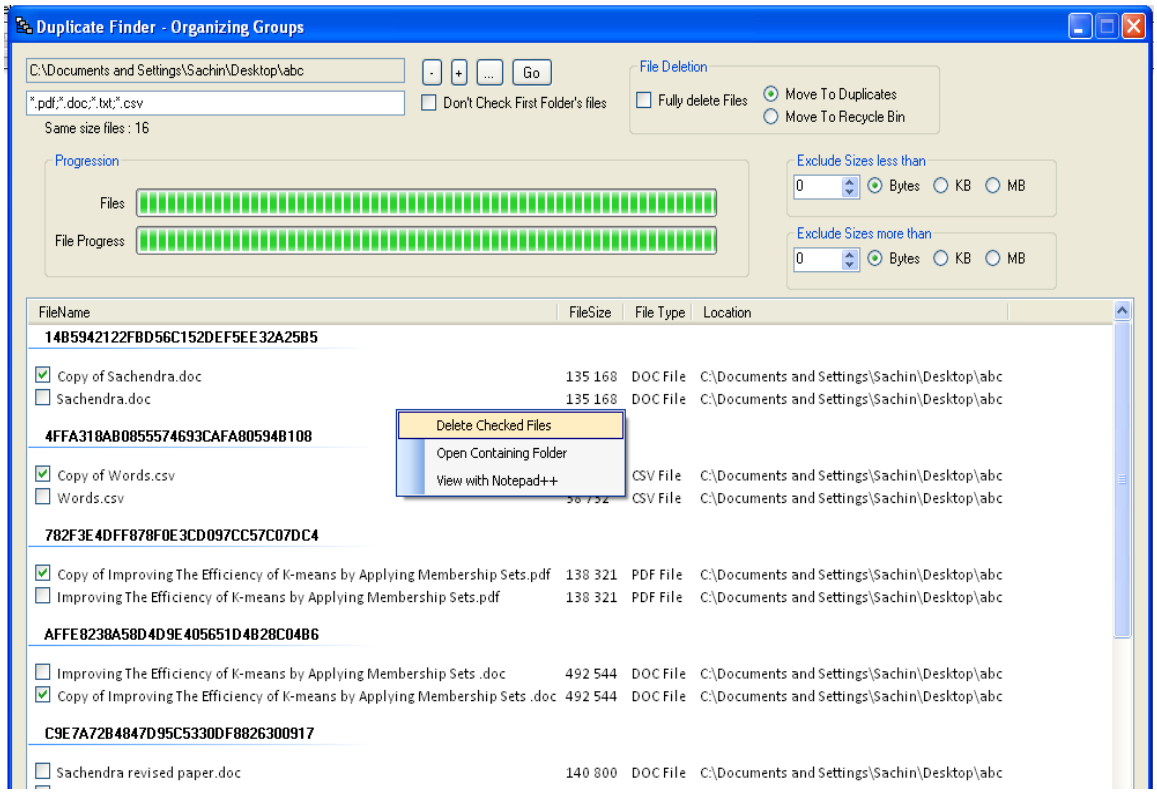

Figure 1.3: Testing the exact duplicate for documents

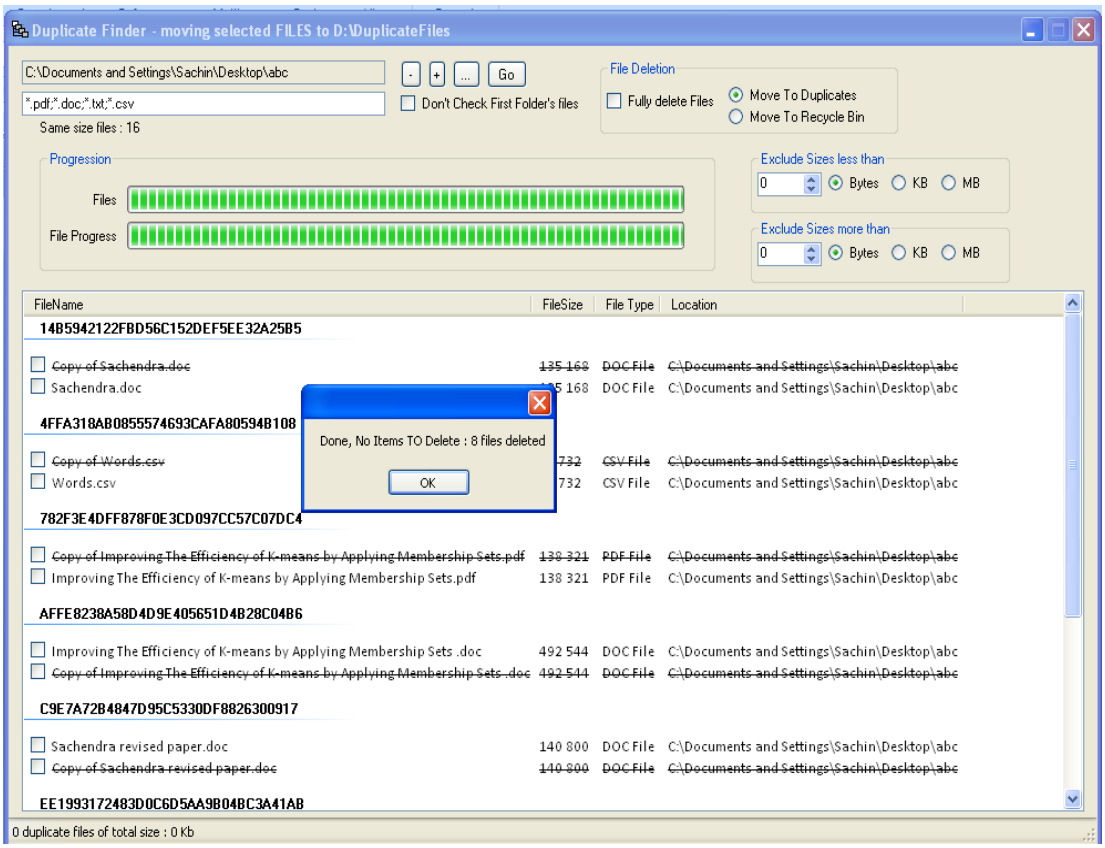

Figure 1.4: Deleting the exact duplicates of the documents.

1.4.3 All pair similarity search problem

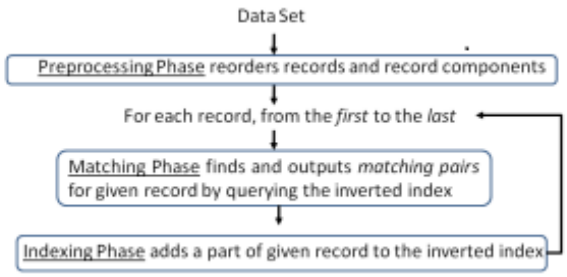

Figure 1.5: Overview of the framework common across recent exact algorithms for the all pair similarity search

It is defined as inquiry of all the pairs of items which have the correlation rate over the pre-defined value. Correlation rate is the most commonly used measure to determine

the dependence between the two variables. It is defined as the ratio of covariance of two objects with the product of their standard deviations. Countless systems related to real-world like online social networks, search engines, digital libraries and recommender system have datasets containing large number of records in millions, in a large dimensional space, that are mostly scarce. The major test is to fabricate a layout that meet the appropriate time condition.

1.4.4 k-most near similar items to query

Nearest Neighbor search or similarity search is a method to optimize the problem of finding the most similar points. Dissimilarity function is mainly used to define closeness. With a given query object and distance used to describe the correlation in a metric space, k- nearest neighbor means to find out the k- most similar objects [8]. There are a range of regression and classification tasks over which this method can be practiced. For each test data point, there are nearly N distance evaluations which results in high computation overheads during its implementation. So, to overcome this procedure and to speed up this process, discrete space partitioning techniques like K-D Trees [9] and M- Trees [10] have been used, contributing to the rapid definite K-nearest retrieval [11,12].

1.5 Distance Measure

Space is defined as regular arrangement of points. Distance measure is defined as a function dm(x1, y1) where two points are taken as input and a real number is produced as output and it should satisfy the subsequent conditions:-

- dm(x1, y1) >=0
- dm(x1, y1) = 0, contingent upon x1=y1
- dm(x1, y1)= dm (y1, x1)
- dm(x1, y1)= dm(x1, z) + dm(z, y1), Triangle Inequality

The last one is the most complicated condition. The useful distance measure, which are used in a given space to determine the proximity of the points are:

1.5.1 Jaccard Distance

Jaccard distance is specifically used to define the distance between the sets. Jaccard distance JD(x, y) between the two sets i.e. x, y is defined as one minus the Jaccard similarity JS (x, y) of the sets x and y. It can be illustrated as JD(x, y) = 1- JS(x, y). Jaccard Similarity is defined as the size of the intersection divided by the size of the union of the sample sets. It is helpful to determine the relationship between sample sets[13].

$$J(X, Y) = |X \cap Y| / |X \cup Y|,$$

$$0 <= J(X, Y) <= 1$$

Jaccard distance is a distance measure as it follows the following constraints:-

- Non-negative constraint
- Strictly positive
- Symmetric
- Triangular inequality

1.5.1.1 Properties of Jaccard Similarity

- Its help to find the similarity between the two vectors.
- Also one of the property of Jaccard similarity is, if we add constant to both the vectors, then the similarity will rise termed as 'nominal increase'.
- Clustering is also one of the properties of Jaccard similarity.
  For example: - if set X has elements {2, 5, 8, 9} and set Y has elements {5, 6, 7, 8, 9} then J(X, Y) = ½.

1.5.1.2 Jaccard with Shingles

Here we put it together i.e. shingles with jaccard similarity. Consider the (k = 3). Shingles for each D1, D2, D3, and D4:

D1: [I like you], [like you alot].

D2: [I like you], [like you and], [you and admire], [and admire you], [admire you alot].

D3: [I do not], [do not like], [not like green], [like green eggs], [green eggs and], [eggs and ham].

D4: [I do not], [do not like], [not like them], [like them and], [them and admire], [and admire them],[like you alot].

Now the Jaccard similarity is as follows:

JS(D1, D2) = 1/6 ≈ 0.166

JS(D1, D3) = 0 = 0.0

JS(D1, D4) = 1/8 = 0.125

JS(D2, D3) = 0 = 0.0

JS(D3, D4) = 0 = 0.0

JS(D3, D4) = 2/11 ≈ 0.181

1.5.2 Euclidean Distance

In the Euclidean spaces, the straight line distance between the two points p and r i.e. d (p, r) is called as Euclidean distance. It is represented as follows:

d (p, r)= d (r, p) = under root $\{(p1- q1)^2 + (p2- q2)^2 +\ldots\ldots\ldots\ldots\ldots (pn-qn)^2\}$

L2 norm is also used to define this distance.

LR-norm is another variant of this and is described below as:-

$d(p, r) = d([p1,p2\ldots. pn] , [r1, r2\ldots..rn]) = (i=1 \text{ to } n\Sigma |xi-yi|^r)^{1/r}$

L1 norm is the common measure for distance also called Manhattan distance where the distance between the points is the total of significance of the variation in each dimension.

All the formerly mentioned rules of distance measure are satisfied by Euclidean distance.

- Non- negative property is followed for Euclidean distance as square root of real number is always positive.
- Given two real numbers, then square of their difference can be 0, only in the case when both the real numbers are identical. Therefore Euclidean distance follows the positive distance constraint.
- Symmetry is always followed by Euclidean distance since $(yi - xi)^2 = (xi - yi)^2$.
- Triangular inequality is followed by Euclidean distance since in a triangle, length of the third side is always less than or equal to the total of length of any two sides of triangle.

1.5.3 Cosine Distance

If two vectors u and v are given, then the angle formed between these vectors gives the cosine distance. The total count of dimension does not affect the angle form and it mainly lies within 0 to 180 degree. Euclidean spaces are mainly used to measure cosine distance with spaces of any dimension and points in these spaces are either Boolean or integer components. Further, directions can be used to illustrate these points. The ratio of the dot product of vectors to the L2- norms of u and v gives the cosine distance.

Cosine distance also follows the axioms of distance measure since:-

- It is always a positive distance, since the values lies in the range from (0, 180).
- If the two vectors lie in the different direction, then the angle between them is not 0 otherwise vice- versa.
- Also property of symmetry is followed, by the angle between the two vectors.
- Triangular inequality axiom is also followed by cosine distance.

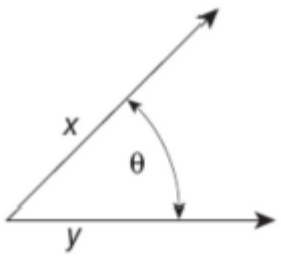

Figure 1.6: Geometric illustration of cosine measure

1.5.4 Hamming Distance

Hamming distance is used in information analysis. 0 or 1 defines the Boolean vectors and Hamming distance is defined only when it contain Boolean vectors i.e. 0 or 1. In the given binary strings, hamming distance denotes the difference between them. Given two strings of similar length, the number of positions for which the given strings differ defines the hamming distance. Distance measure properties are clearly followed by Hamming distance.

- If the vectors are equal, then it is zero.
- Hamming distance is always positive.
- Order of the vectors does not affect the equivalency of Hamming distance. So symmetric property is also followed.
- If x is the total count of elements in which a and c are not identical and x is the total count of elements in which c and b are not identical, then a and b can vary only within x+y elements.

1.5.5 Edit Distance

Edit Distance is the most commonly used concept to regulate the similarity of the two strings. It defines the minimal count of infusion, cancellation and exchange that need to be done to convert one string to another i.e. if we have two strings x1 and y1, what are the minimum deletions and insertions that need to be done to transform string x1 to y1. One of the methods to determine the edit distance is the LCS (longest common Subsequence). From the total dimension of x and y, reducing double the dimension of their LCS, we can gauge the edit distance. Distance measure properties are followed by Edit distance. It is described below:-

- The total number of inclusions and deletions required to alter string x1 to y1 cannot be negative.
- When the two strings x1 and y1 are identical, then the edit distance is nil.

- To alter string x1 to y1 and vice- versa, the total number of inclusions and deletions required is identical.
- The total count of deletions and inclusions required to alter string x1 to y1 is always less than or equal than required to alter string x1 to z1 surplus z1 to y1.

1.6 Structure of the Thesis

The remaining part of the thesis is organized in the following order:

- Chapter 2 describes the literature review in order to gain skills of information seeking and critical appraisal.
- Chapter 3 describes the problem statement, analyzing the gap between present work and the work to be done.
- Chapter 4 describes the importance of similarity search, universal hashing.
- Chapter 5 describes the implementation, results and analysis of the proposed idea of thesis.
- Chapter 6 concludes the thesis and provided the future scope of the work is i.e. further in which areas research can be conducted.

Chapter 2

Literature Review

As the number of dimensions are increasing in a given data document, this results in rise in both space and time complexity, resulting in 'Curse of dimensionality' problem. But for data with less number of dimensions, nearest neighbor search works well.

In [5], Manber presents a tool called SIF, which is used to find identical files in a given large set of file documents. In this he considered files to be identical, if they have some portion in common, even if they are dissimilar otherwise. Sif is used even for comparison of files which are not identical in size. Even using this tool, type of similarity is found by user.

In [12] Sariel Har-Peled *et al.* present a technique based on approximate solution of high degree data. For data-set of size n, with R^d complexity, an algorithm has been proposed in two steps. In first step, approximate nearest neighbor problem is covered and second one is based on the notion of locality sensitive hashing. Also they proposed locality sensitive hashing (LSH) established on data with high dimensions for comparative similarity search. Magnitude of the data input is the basic factor on which the pattern depends. In this various hash functions are used to hash the points rather than partitioning method so that the object that is identical, there probability of collision can be raised than those of dissimilar objects. The candidate pair can be traced by query points i.e. by hashing them and various points are present in different buckets and recapturing those elements containing the points. When the required points are inclined in binary Hamming space, only then this LSH scheme is applied. LSH is smooth and rapid only when the points are accessible in Hamming space. This is the major disadvantage of LSH technique. Later, they provided the extension to algorithm as data can be expanded to cover the hamming distance, but it increased the time intricacy outlay and error rate even by a huge point.

Stupar *et al*. [20] propounded an access path for execution of LSH using RankReduce approach, which is useful for data with large number of dimensions for calculating similarity search based on MapReduce framework. In order to achieve

exceptional preciseness and performance, LSH and MapReduce attributes need to be employed at the same time. When large amount of details need to be processed in an online form without transformation of query, MapReduce is normally taken into consideration.

In [22], Krauthgamer and Lee, proposed a simple deterministic data structure to maintain in a generic metric space a set of points S1, for backing range and nearest neighbor query and all the insertions and deletions *i.e.* updations in S. "Abstract dimensionality" is used to evaluate the worst- case complexity of the opted data structure. Also they provided the definition of nearest- neighbors search (NNS). There are a number of rational utilizations of such nearest- neighbor's algorithms such as machine learning, data mining, database queries, pattern recognition, *etc*. In all the above mentioned applications a typical component is that correlating a pair of items is costly and so to minimize it, the total count of distance computations should be as small as possible.

Although lot of work has been done in the area of similarity search in text data, coming up of massive data seta and data with high dimension have opened various new issues in this areas which need attention and approximation instead of exactness can serve as an alternate to improve the textual similarity. This work focus on an approximation technique called Locality sensitive hashing, a probabilistic data structure used for finding similarity among documents.

Slaney *et al.* [24] proposed different optimization approaches for recuperation of nearby neighbor in web scale quest. For the described query, most of the algorithms consider LSH as the significant perception for nearby object retrieval. In this LSH is used to represent an algorithm and it helps in parameter optimization. In this, one of the histogram is considered for describing the circulation of distances to the nearby neighbor points and for outlining the span between any count in the input data and that of the given query. In order to fit in for the achievement target with very little computational cost, LSH index and LSH parameters are considered. Mainly in the two components, LSH is partitioned i.e. in prime component, indexing of data

takes place and in the further component for a given query point, research activity is executed to look up for identical document records.

In [28], Dasgupta *et al.* proposed a transformation named as randomized Hadamard transforms to enhance the computation of Euclidean distance of d-dimensional data. Normally, the time- complexity is O (kd), whereas it gives O(dlogd+ KL), (K,L)- parameterized LSH.

In [31], entropy-based LSH scheme was proposed by Panigrahy. By checking for query offsets together with query itself, the count of hash tables mandatory can be reduced. Not only it scans the bucket analogous to the comparable query, but also the bucket comparable to perturbed variant of the query is recognized. Although, by using the above mentioned scheme the need for space requirement is minimized, but the time for querying is substantially raised.

In [35] Broder *et al.* proposed an LSH scheme which consider minhash functions to take various permutations to measure Jaccard similarity for sets. Also various deviations have been suggested to upgrade the achievement of this LSH scheme.

In [36], Dynamic collision counting is the method used for Locality-Sensitive Hashing. In this Fang *et al.* proposed collision counting LSH (C2LSH) algorithm. In this instead of building "static" compound hash function, they use a base of m simple LSH functions to build "dynamic" compound hash functions. Further, by properly choosing LSH function base m and collision threshold, C2LSH provides warranty on query quality. Also the major advantage of C2LSH is that base m is not affected by the number of dimensions of the data objects, i.e. it is highly beneficial for NN search with large number of dimensions.

In [37] Aristides Gionis *et al.* discussed the nearest or near-neighbor query which arises in all type of images, time-series databases as genome collection. According to them the problems arises due to curse of dimensionality *i.e.* as the dimensionality grows the complexity increases. It has been suggested that since the selection metric in a typical applications is rather heuristic, identifying the approximate solution rather than the exact solution will reduce the overall time and space complexity.

In [38] Parisahaghani et *al.* considered distributed k-nearest neighbor search and range query processing in high dimensional data. Their approach, based on locality sensitive hashing, has been very efficient in answering queries in centralized settings. They have considered two forms: first is K-Nearest Neighbor query where given a query point q, the goal is to find the K closest point to it. Second is range query where given a query point q and a range r the goal is to find all points within a distance r of q.

In [39] Panigrahy *et al.* proposed an LSH scheme which is entropy-based. It minimizes the number of hash tables required, by looking up a number of query offsets in addition to the query itself. In addition to considering the bucket corresponding to the query, buckets corresponding to perturbed versions of the query is also considered. Unfortunately, while the space requirements are reduced, the query time is considerably increased.

In [40] Hua *et al*. proposed an algorithm LSBF which improves approximate membership query. This algorithm uses Locality sensitive hashing functions instead of uniform and independent hash functions in bloom filter.

In [41] David Goriss *et al.* provided solution for data with large number of dimensions for approximate nearest neighbor searches by introducing χ^2 distances. To implement it, distinct hash functions are used. For image and video data, it gives improved output as compared to prime LSH.

Broder *et al*. [42] proposed shingling algorithm and Charikar*et al*. [43] propounded random projection based approach for addressing the immediate web pages with duplicity. Streaming algorithms were proposed i.e. that makes few or one passes over input data to produce an output by using limited time and storage space. Even sketching algorithms are used to estimate the similarity. Sketching algorithms means building up such functions such that produce succinct collection of objects, such that with the help of those sketches similarity can be found.

Ling, Wu *et al*. in [44] proposed a latest LSH pattern depending upon new frequency scheme defined as FBLSH where a number defining frequency threshold is stated and for hash function p-stable distribution function is considered.

Objects are recognized as nearest neighbor when their collision rate is greater than the defined threshold. This application is time inefficient but has the advantage of space competence.

Chapter 3

Problem Statement

Similarity search is the general approach which is helpful in searching of objects, which is used in range of mechanism. Nowadays it is becoming increasingly important when large amount of data is produced every second and the data repositories do not contain any natural order of the data present, like large amount of images, sounds and other sophisticated digital objects.

Similarity search has mainly two relevant subdivisions i.e. nearest neighbor search and range queries and large number of solutions have evolved over time. Partitioning algorithm fails for data with high dimensions, but works well for less dimensional data.

One of the alternates is Locality Sensitive Hashing (LSH), where the input data items are hashed and matrix generated is randomly permuted and added into buckets such that identical items are hashed to the same bucket with greater probability. LSH is most commonly applied on data with high dimensions i.e. Image databases, document collections, time-series databases, and genome databases.

3.1 Gap Analysis

- For data with high dimension, more execution time required :- All the available partition algorithms, provide the satisfying output for data with less number of features like anywhere between 15 to 25, but as the count of features rises these algorithms fail to produce the desired output.
- Efficiency :- Almost in every similarity search, heuristic techniques are used, which fail to contribute to fair level of efficiency as these provide high error rate due to presence of false negatives and false positives as these are probabilistic data structure which gives nearby i.e. approximate results, however this error rate should be cut down as low as possible.
- Space requirement: - To calculate similarity for data having large dimensions, it requires more space. However space complexity should be lowered.

3.2 Objectives

- To study and explore various similarity search techniques used for identifying similar items in massive textual datasets.
- To apply Locality Sensitive hashing for similarity matching of text data.
- To implement the proposed scheme for identifying the effects of preprocessing of the textual data sets and analyze the results achieved.

3.3 Methodology

In this thesis, the most important factor is to finds similar items and applies a mechanism which reduces storage space and time for massive data sets. Methodology to be followed is:-

- Convert the given text document (News) into shingles of varying size.
- Minhashing is applied and characteristic matrix is generated.
- LSH is used for identification of similar items.
- This technique is applied over various news report divided into various categories from different newspaper with singles with k= 5, 7, 9 respectively.
- Identify the effects of preprocessing of the textual data sets and analyze the results achieved.

Chapter 4

Importance of similarity search

Many definitions of similarity and distance have been studied so far, but the major question which comes to mind is that what are the practical applications of similarity search and what steps need to be considered to reduce the time required for identifying similar items especially if the data sets are very large. This chapter starts with the focus on major application areas of similarity search followed by the step by step details of technique followed for finding the similar documents.

4.1 Applications of Similarity search

4.1.1 Recommendation systems

Recommendation system is basically a web application that helps in predicting user responses to option. When a user buys an item, he/she is recommended other items of similar type based on his past choices. An online- news reader can be suggested with news articles based on their past experiences of reading.

Recommendation system is a big application of data mining. A number of technologies are used in recommendation system.

4.1.1.1 Applications of Recommender system

- News articles
- Movie recommendations
- Product recommendations

4.1.2 Fingerprinting

Molecular fingerprinting presents the method by which the structure of a molecule can be encoded. Fingerprints comparing help to determine, to what extent the given two molecules are identical. In fingerprinting, shingles are hashed to 64- bit integers.

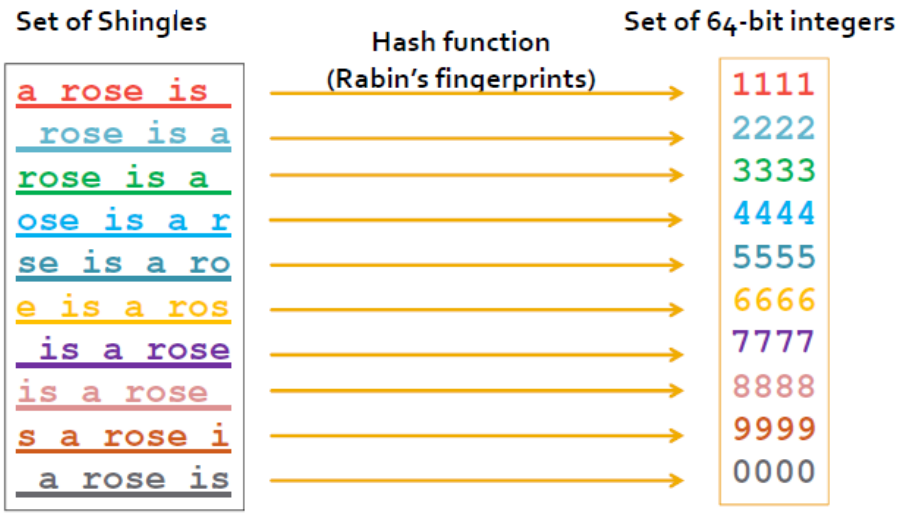

Figure 4.1: Example of Fingerprinting

4.1.3 Searching near- duplicate items

It is very important to find duplicate or near duplicate text documents from bundle of documents present on web. Nowadays as the total count of users over the internet is increasing, there is also an alarming increase in the size of the data being produced, which had increased the importance of searching an appropriate entity.

It is relevant to find mirror web pages as it helpful to avoid serving them large number of times or making indexes of them multiple times. It is helpful in finding plagiarism .For the entire above mentioned problems; we need a mechanism to locate identical objects to a query object.

4.2 Nearest neighbor problem

Closest point search or NNS is basically an optimization problem for finding the adjoining neighbors in the given space R^d. NNS is a development task to detect the most identical element from the provided input set. Dissimilarity function is commonly used to define the correlation resemblance or similarity. Triangular inequality and dissimilarity are the two properties satisfied by distance metric i.e. dissimilarity. D- dimensional vector space is taken as M, where different functions like Euclidean distance, Manhattan distance *etc*. belonging to Minkowski family is used to calculate the dissimilarity.

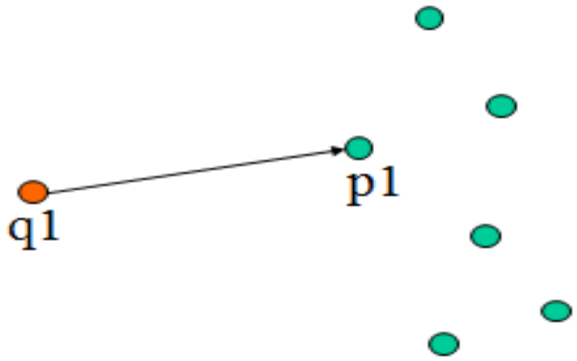

Figure 4.2: Nearest Neighbor Search [14].

4.2.1 Versions of NNS

4.2.1.1 KNN

K- Nearest Neighbor search is an algorithmic program style for organizing entities supported on nearest training *i.e.* within the feature area. When there is no previous expertise about how the data is arranged, KNN is the most significant methodology that can be considered [15-18].In KNN, when K=1 it defines NN, i.e. NN is the most elementary structure of KNN. In this technique or methodology every sample ought to be restricted similarly to its encompassing samples. Therefore, for the samples in which their classification is not known, most near i.e. its nearest neighbor are identified.

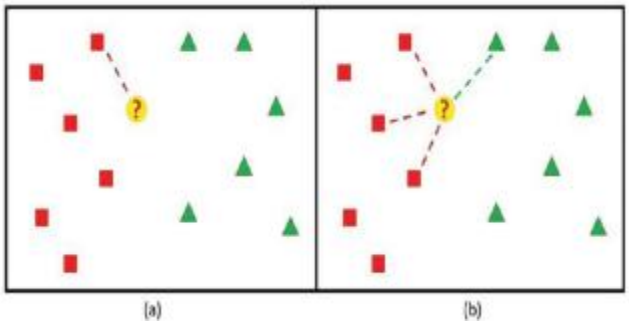

Figure 4.3: (a) The 1-NN decision rule: the point?is assigned to the class on the left; (b) the KNN [19].

The value of K chosen determines the output of K- classifier when the distance metric is applied. T he native estimate tends to be terribly poor with sparseness, ambiguity when value of K chosen is incredibly small. Even the large value of K also create problems as it makes the evaluation over smoothing and with the insertion of the outliers from different categories, classification enforcement weakens.

Applications of K-Nearest Neighbor

1) Text Mining
   - For text categorization or text mining, KNN is the most useful algorithm. In this instead of using fixed count across all classes, rather distinct count of nearest neighbors is considered for distinct classes.
2) Agriculture
   - KNN algorithm is used for everyday precipitations and alternative climate variables. For such applications, satellite mental imagery is employed, in which discrete categories are used for mapping the land cowl and land under use. It is also been successfully used for weather predictions.
3) Finance
   - KNN method is used for stock market forecasting, one of the most core financial tasks *i.e.* for estimating the cost of a stock, based on economic data and reforms used to count company's performance.
4) Medicine
   - Predict whether a patient will suffer from second heart attack after he is hospitalized due to first heart attack. All these estimations are based on analytical, dietary regimen and clinical dimensions of patient.
   - In a patient suffering from diabetes, predict the count of glucose in its blood, by checking its infrared absorption spectrum.
   - Predict danger aspects of prostate cancer depending up on clinical and analytical variables.

4.2.1.2 Approximate Nearest Neighbor Search

For many applications speed is more important factor as compared to accuracy. ANN is an inquiring or searching approach which increases speed and provide flexibility to accuracy factor. Accuracy parameter 'Ɛ' and confidence parameter 'δ' were used by Ciaccia and Patella which defines the ANN (Approximate Nearest Neighbor) as follows Definition:- " In a d- dimensional metric space X= R^d, given n1 points forming the set P1, a confidence parameter δ1 ε [0,1) and the accuracy parameter Ɛ1is defined, in this we need to search for the point in P1, which is nearest to the query point q ε X.

4.3 Universal Hashing

Hashing is the general approach that is used to minimize the content of a set by mapping the component into N bit index table. Some hash function is used, which maps some set U1 randomly into a table that lies in the dimension [0, N-1].

Let us assume, there is a universe U, having at least n1 as its cardinality, also a set S= {0, 1, 2... n-1} is defined. The group of hash function A: U–› S is termed as K- universal if for some m items y1, y2, …., ym a hash function h1 is elected evenly from A, than :-

Probability (h(y1) = ……. = h(ym)) ≤ $(n1^{m-1})^{-1}$

i.e. all m buckets hashing to the similar bucket. Some features of the universal Hashing are as follows :-

- Its output is independent of the key chosen, so performance is satisfactory in many cases, no matter which key is chosen.
- For Universal hashing to take place, there is demand of group of functions to elect from.

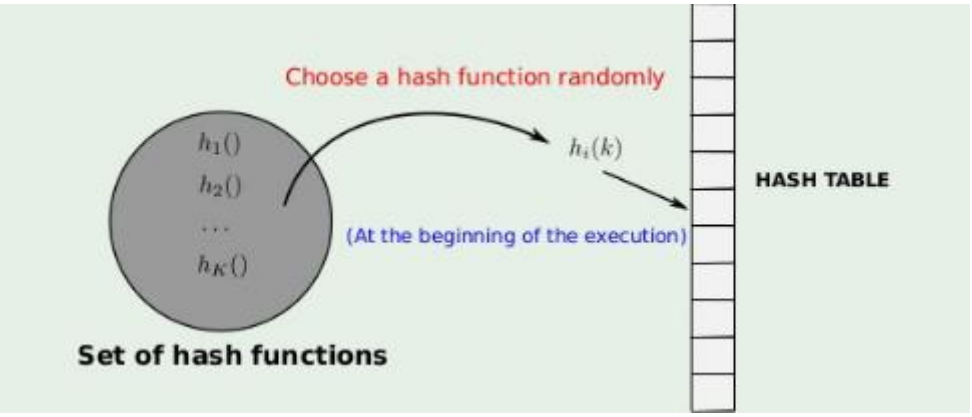

Figure 4.4: Universal Hashing

4.4 Preprocessing Steps in Textual data

4.4.1 Stemmers

It is the process of linguistic normalization in which common form is obtained from various variant forms of word. It is the common practice to improve the performance of information retrieval systems. For example:- words like 'connection', 'connections', 'connective', 'connected' and 'connecting' are all considered as stems of 'connect'.

On the basis of semantic similarity, words are grouped together in stemmers or stemming algorithms. Stemmers have a large number of advantages, and are mainly used in the field of language processing and text analysis systems. Database search systems and information retrieval are also the applications in which stemmers are used. Different types of stemming algorithms are present. Most widely used is affix removal algorithm. In affix removal, stems are produced by removing suffixes or prefixes from the words. Describing the relatedness between pairs of stemmers is the main focus of stemmer similarity analysis.

4.4.2 Stopwords

In any language, stop words are the most common words such as "the", "and", "to", *etc*. Ignoring stop words is the most common practice in many applications, since they do not provide any useful knowledge about the topic.

Uses of removing stopwords from any language:-

- In any document, stopwords counts nearly 20-30% of the total word count, so removing this will help to reduce indexing or data file size.
- It also helps to improve the efficiency as stopwords are not useful for searching or text mining
- If we remove stopwords i.e. most common words in any language, then it is possible for us to focus on important words instead.

There are different types of stopwords like:-

1) Determiners- like ‘the’, ‘a’, ‘an’, ‘another’ etc.
2) Coordinating conjunctions- like ‘for’, ‘an’, ‘nor’, ‘but’, ‘or’, ‘yet’, ‘so’.
3) Prepositions- like ‘before’, ‘in’, ‘under’, ‘towards’.

4.4.3 False positive and False negative

False positive describes a situation in which it is shown that the given condition has been fulfilled, when it’s actually not been fulfilled. Whereas, in false negative when a condition has been actually fulfilled, it is shown that it has failed. False negative is more dangerous as compared to false positive.

4.5 Essential steps for Similar documents in LSH

- Shingling process
- Minhashing
- Locality sensitive hashing

4.5.1 Shingling process

It is the main technique used to convert documents, emails into sets. K-Shingles are set of unique “shingles” (adjacent subsequence of tokens in a text document) that can be used to test the similarity between the documents. As Stupar, Michel and Schenkel *et.al*, [20]

K=9, is considered as a safe choice for large documents. In a document with m words, it takes O (km) space for k-Shingles.

4.5.1.1 Models for shingling

When defining shingle's, following models are taken into review-

- Punctuation- Shingles should be attached with punctuation symbols or not.
- Words/ characters- Shingles based on words or shingles based on characters.
- White spacing- Shingles with spacing or shingles without spacing.
- Capitalization- Shingles based on case sensitivity or not.
- Stammers- Shingles with and without stammers.
- Stopwords- Shingles with and without stopwords.
- K- Size shingles- Depending on the category of the text document, size K varies. The scope of the shingle size K should be such elected, so that collisions can be expanded. Like K=9 is relevant for research papers and K=5 is convenient for emails. For documents like web pages, blogs and news articles K value varies from 5 to 9.
- Replicas count- Shingling does not take replicas into consideration, whereas bag of words does.

4.5.2 Minhashing

In minhashing large sets are converted into small signatures while still preserving the similarity. A. Broder founded this technique and was for the first time practiced in Alta Vista search engine for locating and extracting the equivalent web pages from the search documents.In minhashing, to find whether the items are identical or not, several autonomous permutations of given columns are done. Minhashing can be used for the clustering of documents, in which set of words that are identical, based on its clusters can be formed.

In order to carry out minhashing, characteristic matrix is formed first. Characteristic matrix is a collection of sets that helps to form small signatures from large sets. Permutation of the rows is carried out in order to minhash.

Element	S_1	S_2	S_3	S_4
a	1	0	0	1
b	0	0	1	0
c	0	1	0	1
d	1	0	1	1
e	0	0	1	0

Figure 4.5: Matrix representation of four sets [21]

Matrix representation of characteristic matrix M, is (m x n) matrix where rows store the elements of universal set of shingles, while the set represent the columns of the matrix [13]. Mij= "0" such that 0<=i<=m, and 0<=j<=n, otherwise the entry Mij= "1" if the ith Shingle is present in jth set. In general, the number of rows is much greater than some of the defined hash functions. In the permuted order, the minhash value is the number in the first row, in which the column has 1.

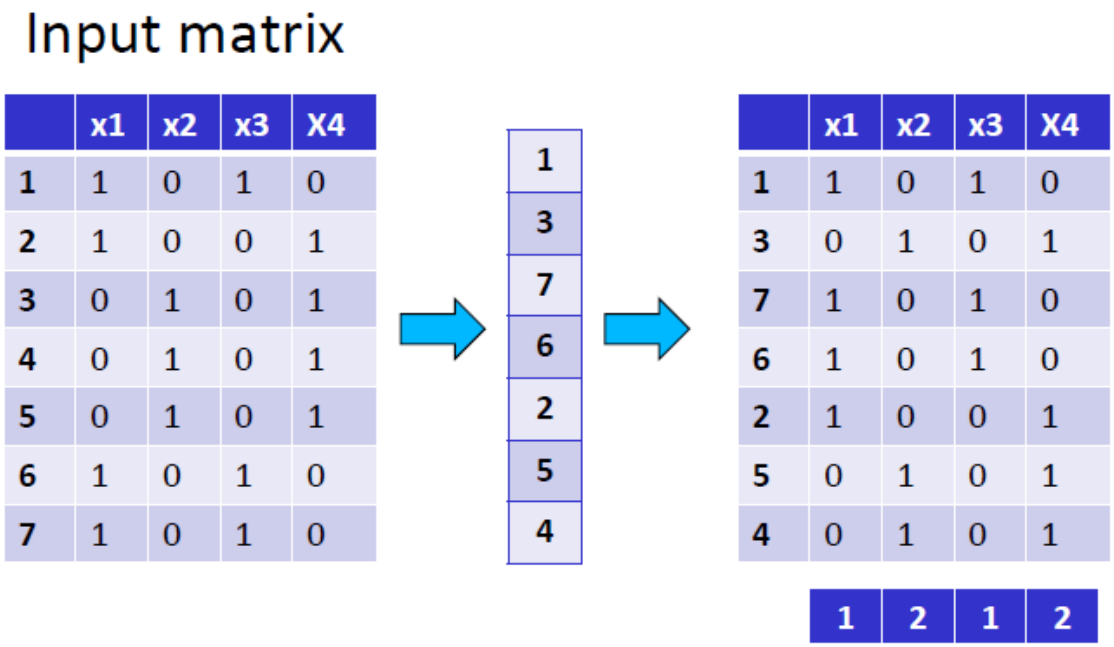

Figure 4.6: Example of Minhash signature with 1st permutation of rows [21]

In minhash signature rows of the input matrix are permuted randomly using some hash function, and the minhash value is the corresponding row number for which the column has 1. In figure 4.6, for column x1, the first 1 is present in 1st row, so it has minhash value 1. Similarly for column x2, first 1 is present in 2nd row, so it has minhash value 2. Similarly, we can compute value for x3 and x4.

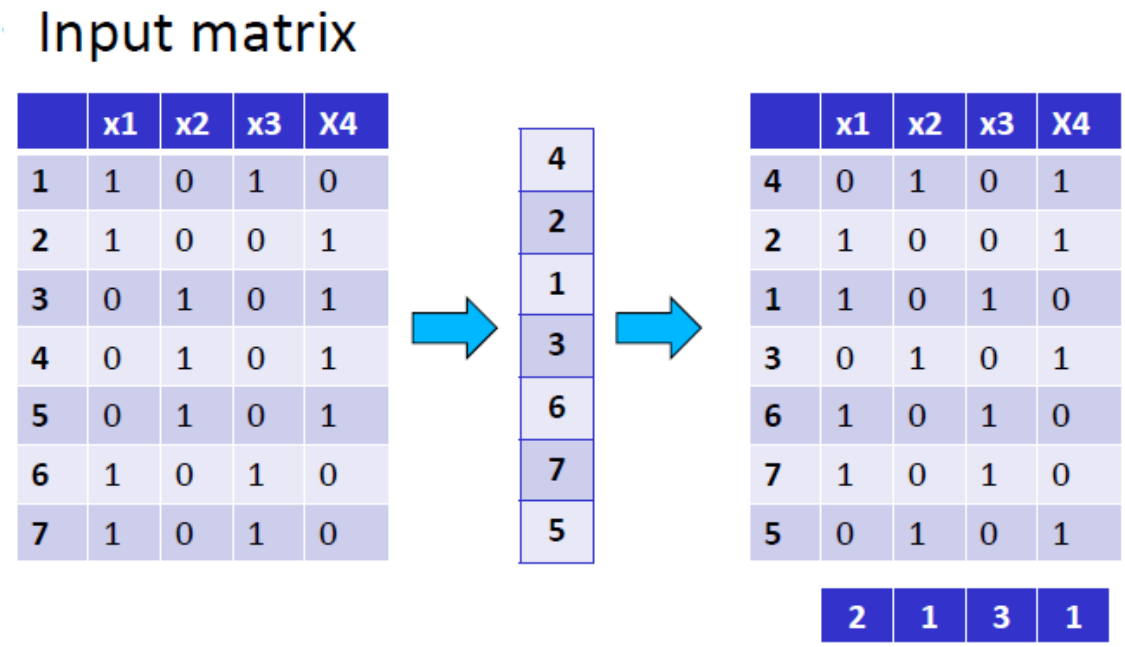

Figure 4.7: Example of Minhash signatures with 2nd permutation of rows [21]

Similarly rows of characteristic matrix are permuted large number of times and signature matrix can be computed. Later jaccard similarity is used to estimate the similarity of underlying sets in signature matrix. Jaccard similarity calculated using signature matrix is just an estimate of the true jaccard similarity.

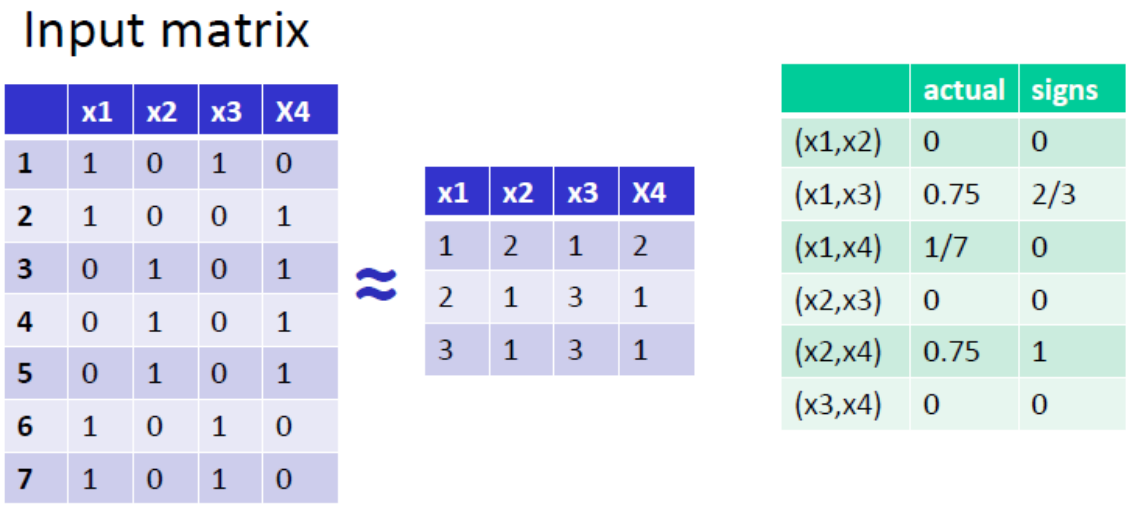

Figure 4.8: Jaccard similarity calculation using minhashing [21]

Signature matrix

Permuting a large characteristic matrix explicitly is not a feasible task. It is very time consuming and a lot of storage space is required for randomly permuting millions of rows and even sorting of rows is more time consuming making it conceptually infeasible. Therefore, following technique is used in which n randomly chosen hash functions h_1, h_2……..h_n are picked on the rows instead of picking n random permutations of the rows.

Table 4.1: Characteristic Matrix with hash functions

Shingles	DS1	DS2	DS3	DS4	DS5	DS6	HS1	HS2	HS3
0	1	1	1	1	0	1	1	2	3
1	1	1	0	1	0	0	2	3	4
2	0	1	0	1	1	1	3	4	5
3	0	0	0	1	1	0	4	5	6
4	0	0	1	1	1	1	5	6	7
5	1	0	1	0	1	1	6	7	0
6	1	0	1	0	1	0	7	0	1
7	0	0	1	0	0	1	0	1	2

HS1 ((shn+1) mod8), HS2 ((shn+2) mod8), HS3 ((shn+3) mod8) are the three hash functions used, that are connected with characteristic matrix, where shn stands for analogous shingle number.

Similarity between characteristic matrix and entries in signature matrix [22] is almost same, but it occupies very small amount of storage space.

Steps to be followed for minhash signatures:-

- The rows of Signature Matrix represent the number of hash functions 'h' and the sets represent the column of the matrix. A large number is used to initialize each entry in Signature Matrix.

	S_1	S_2	S_3	S_4
h_1	∞	∞	∞	∞
h_2	∞	∞	∞	∞

Figure 4.9: Initial matrix consisting of all infinity [21]

- Row by row scanning of characteristic matrix is done to create signature matrix. The corresponding hash values are compared with existing hash function values in signature matrix, for all the sets with entry "1" in characteristic matrix. For large value no changes are made, but signature matrix is updated for small value.
- First, we consider row 0 of Table 4.1, we can observe that hash value are HS_1 (0)= 1,HS_2 (0)= 2, HS_3 (0) = 3. Also zero number rows have 1's in the columns for set DS1, DS2, DS3, DS4 and DS6 except DS5. So value in these columns will alter, as values 1, 2, 3 are less than ∞ i.e. the hash values are renewed with the analogous values in characteristic matrix.
- Similarly all other rows are scanned one by one till the end of matrix is reached. And the final matrix obtained after scanning all rows from 0 to 7 are shown in table 4.2.

Table 4.2: Value in Signature matrix after scanning all rows of characteristic matrix

	DS1	**DS2**	**DS3**	**DS4**	**DS5**	**DS6**
HS1	1	1	0	1	3	0
HS2	0	2	0	2	0	1
HS3	0	3	0	3	0	0

- From the Table 4.2 it can be observed that column number DS2 and DS4 are exactly similar. Now Jaccard similarity can be applied on the underlying sets in the signature matrix to calculate the similarity.

4.5.3 Locality Sensitive hashing (LSH)

Generally, hashing refers to a way of converting the data item into an identical precise code having a string of bits and after hashing items are hashed into distinct sections whereas in Locality Sensitive Hashing (LSH) data items preserves their proximity even after the hashing.

- Locality- sensitive hashing (LSH) is used to find a randomized hashing structure in a high- dimensional space for productive approximate nearest neighbor search. Its definition is given depending on the LSH group H, which is basically a group of hash functions that aligns identical data items given as input with greater chance to the similar hash code, than unlike data items.
- In LSH, when banding approach is enforced, candidate pairs are those combinations that categorize to the equivalent bucket and unlike analyzing each pair in linear search, only candidate pairs are compared in LSH technique.
- The main advantage of these techniques is time rebate since these are based on parallelism. Sometimes definite outcome is required i.e. most identical pairs are required.
- Some threshold is used to describe the similarity level. Therefore in such cases the outcome needed to be achieved is defined by LSH or Nearest Neighbor Search.

When looking for identical documents, banding approach is united with group of functions known as Minhash functions so that pairs at broad distance are independent from those at shorter distance [23]. For the group of functions, following three benchmarks are characterized.

- Pairs that are near to one another should be named as candidate pairs rather than far off pairs.
- These functions should be autonomous from one another based on the factor of statistics such that the functions that have the same probability and hash to the equivalent outcome could be determined.
- Since it does not compare all pairs but only candidate pairs are compared, it take less time.

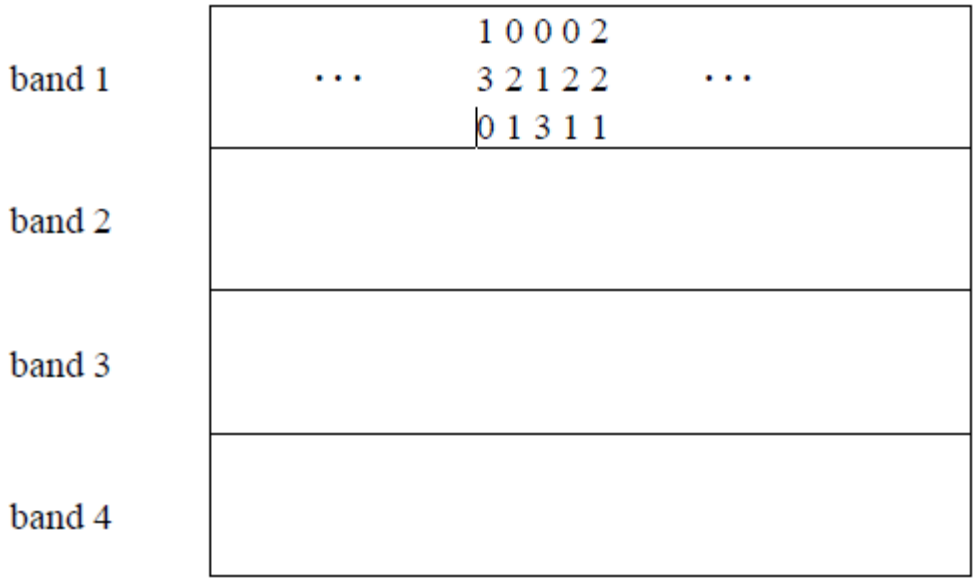

Figure 4.10: Dividing a signature matrix into four bands of three rows per band [21]

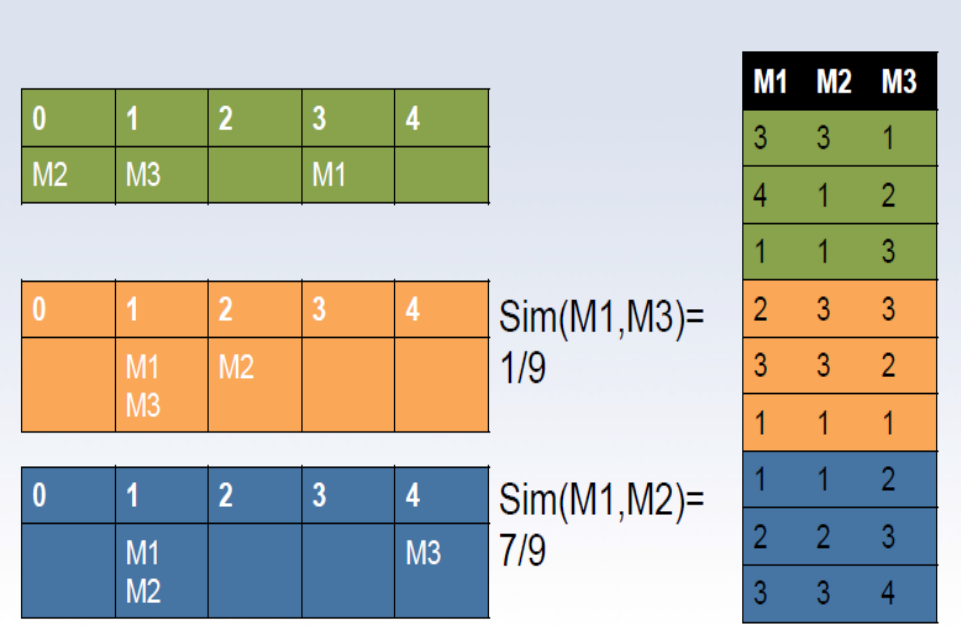

Figure 4.11: Example of banding technique to calculate similarity [21]

From Figure 4.11, one can observe that signature matrix is divided into 3 bands, containing 3 rows in each band. Now further mod5 hash function is used so that the rows can be mapped into buckets numbered 0 to 4. For first band in M1, hash function is used i.e. sum (3+4+1) mod5 which results in 3, therefore M1 is hashed into 3rd bucket for first band. Similarly hash function is used for M2 and M3, which hash to 0 and 1 bucket number respectively for first band. Since no row hashed into same bucket after applying hash function, it indicates that there is no similarity between M1, M2 and M3 for first band.

Now for the second band, M1 and M3 hash into 1st bucket therefore it indicates that there is some similarity between M1 and M3. To calculate the similarity we use the Jaccard similarity i.e. ratio of number of intersection to the number of union of M1 and M3, i.e. it gives the similarity 1/9. Similarly for third band, M1 and M2 is 7/9.

Let there be some distance dimension d and also estimate two distance criterion d1 and d2 such that (d1<d2) based on the d and also let us define the group of functions F1 and it is called (d1, d2, p1, p2) sensitive if for every f in F1:

1. If $d(x, y) \leq d_1$, then the probability that f (x) = f (y) is at least p_1.

2. If d(x, y) ≥ d2, then the probability that f (x) = f (y) is at most p2.

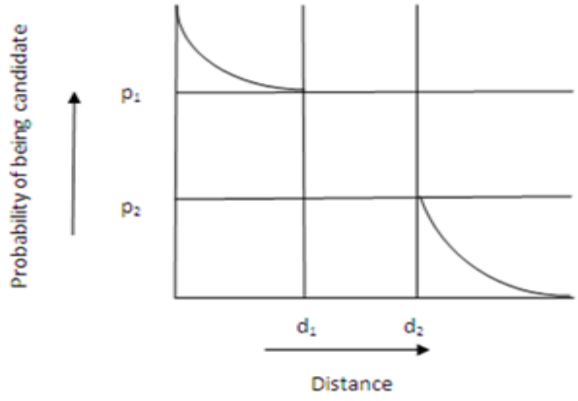

Figure 4.12: Behavior of (p1, P2, d1, d2)-sensitive function

From Figure 4.12, we observe that x- axis shows distance and being candidate pairs, its probability is represented by y- axis. Also the probability cannot be described if the point distance lies in the midway of d1 and d2. Also d1 and d2 can be near to one another but the complication is that p1 and p2 will no more be dissimilar [21].

Chapter 5

Implementation and Results

5.1 Algorithm for Similarity Search

Input: Group of specified files.

Output: Most identical files are organized together.

Rule 1: Mark i.e. select entire files

Rule 2: Obtain shingle set from text files.

Rule 3: Apply Universal hashing for shingle storage for each file. Later use large array for storage of combination of all shingles.

Rule 4: Generate characteristic matrix.

Rule 5: Now generate signature matrix.

Rule 6: Divide the matrix M into b bands containing r rows in each band.

Rule 7: For each different band, use a different hash function, to hash the columns of signature matrix in a massive bucket. Search time is improved to constant by using universal hashing for shingles storage.

5.2 Proposed Scheme

In our research work we have taken text files having dissimilar news from various categories. These files have been taken as input data and shingles have been generated using netbeans programming. These shingles are prepared on various preprocessing stages like shingles with space and without space, shingles with stop words and without stop word, shingles with stemmer and without stemmer. All these cases are being considered for similarity identification criteria.

5.2.1 Algorithm for similarity search

Step1: Input various text files which carry similar news from one category. Input the text files without stop words, without stammers and without spaces.

Step2: Read the files; convert their appropriate shingles in the matrix of string.

Step3: Remove the re-occurring shingles i.e. removes the duplicates.

Step4: prepares the universal matrix of all the shingle matrixes.

Step5: prepare the matching matrix by comparing the two matrixes.

Step6: Divide signature matrix into of n rows into b band and in every band r rows.

Step7: Prepare the matrix for those entries falls in similar band.

Step8: Calculate the matching percentage by checking the column entries.

Consider SIG(i,c) an element of the signature matrix for the hashing function i and column c. Firstly, we put SIG(i,c) equal to infinity for all the 'i' values and 'c' values. To calculate the signature matrix, we peruse the rows of characteristic matrix row by row and column by column. We calculate each row line by going through the following steps:

1) We calculate hash values for each row.
2) For each c column, we do the following steps:

a) If the column c in the row r has a value of '0', we won't take any action.
b) If the column c in the row r has a value of 1, then for each i = 1, 2,...,n if the value of hi(r) is smaller than SIG (i,c), then we will take the value of SIG(i,c) equal to hi(r).

5.3 Implementation

For the following analysis of graphs we have not considered the preprocessing time that is nearly O (n) for removing stop words, stemmers and spacing.
Dataset 1: First of all we have taken similar newsreports from newspaper denoted by s_1, s_2, s_3, s_4 to analyze the similarity among them.
Dataset 2: Further dissimilar news is taken denoted by s_2, s_3, s_4.
Dataset 3: To analyze similarity of similar news from different newspaper data set third defined by s_2, s_3, s_4 is used.
Initially, the input text documents are read and shingling technique is applied in order to

convert text document into sets. In this, results are observed for 5, 7 and 9 sized shingles. Also shingles are generated based on various cases like:

a) Shingles with and without spacing.
b) Shingles with and without stopwords.
c) Shingles with and without stammers.

Figure 5.1: Different cases of Shingle generation

Jaccard similarity is used to calculate the similarity between text documents. Stop words are removed, as they do not contribute to any relevant outcomes. 0 and 1is the range in which similarity lies. It defines how identical the pair of documents is. If the similarity of pair of documents comes out to be 0, it means the given two documents are totally distinct, whereas if it comes out to be 1 i.e. the pair of documents are totally identical to one another. Following figure 6.2 defines the path to be chosen for the results.

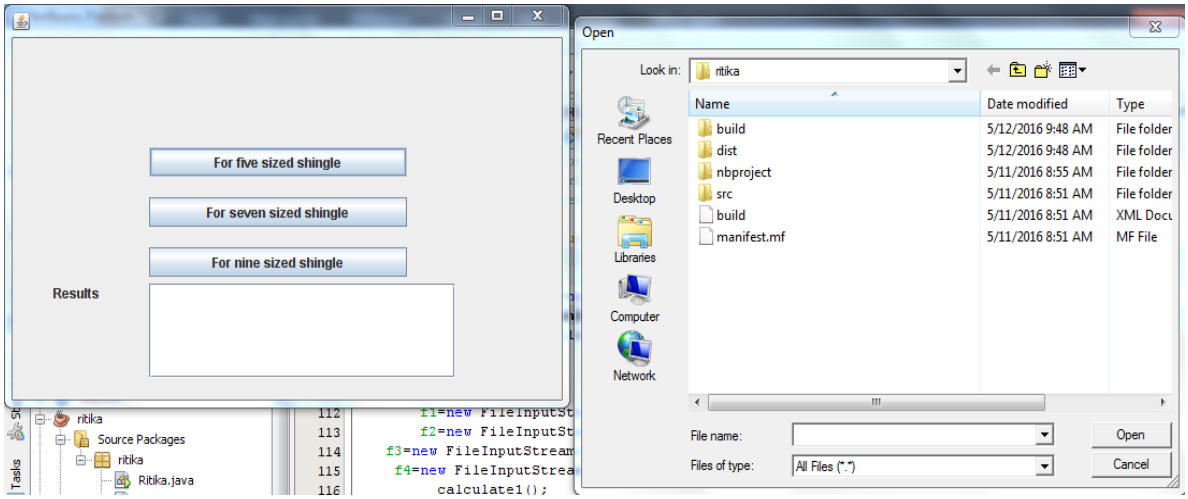

Figure 5.2: Path to be chosen for desired result

Shingles of different sizes with different test cases:-

a) Similarity analysis of 5,7 and 9 sized shingles with and without spacing :

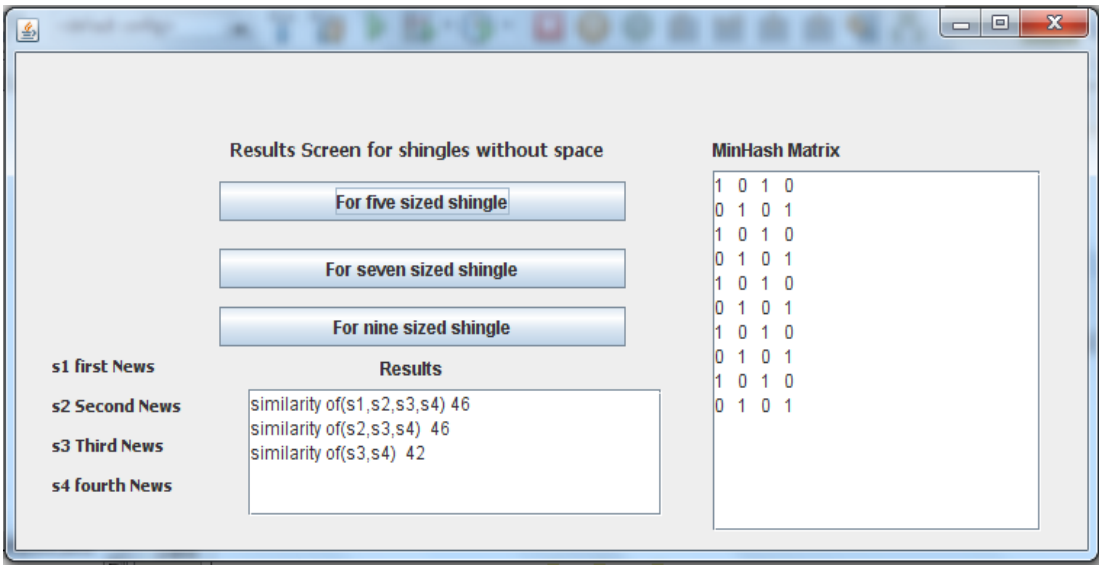

Figure 5.3: shingle without space for k=5

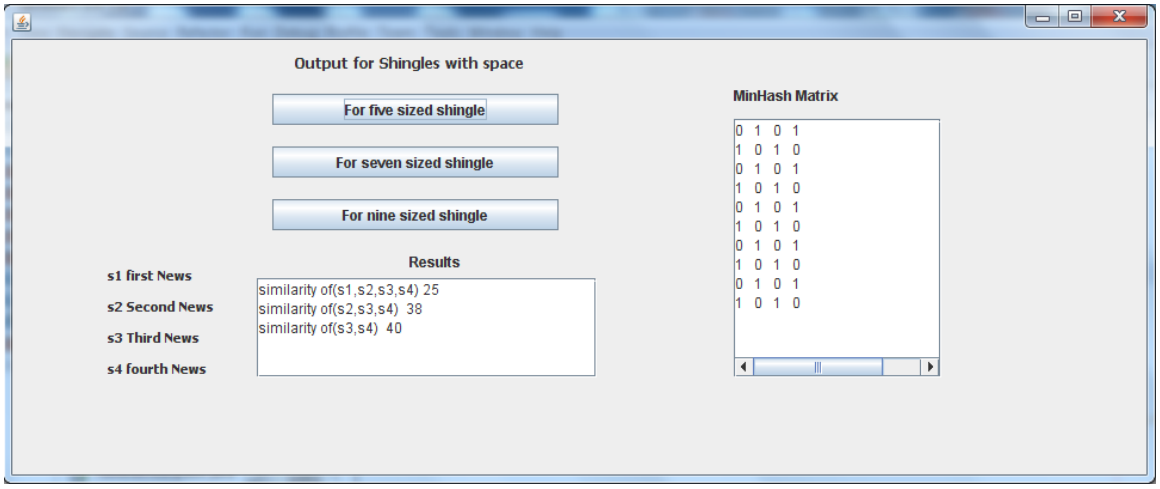

Fig 5.4: shingle with space for k=5

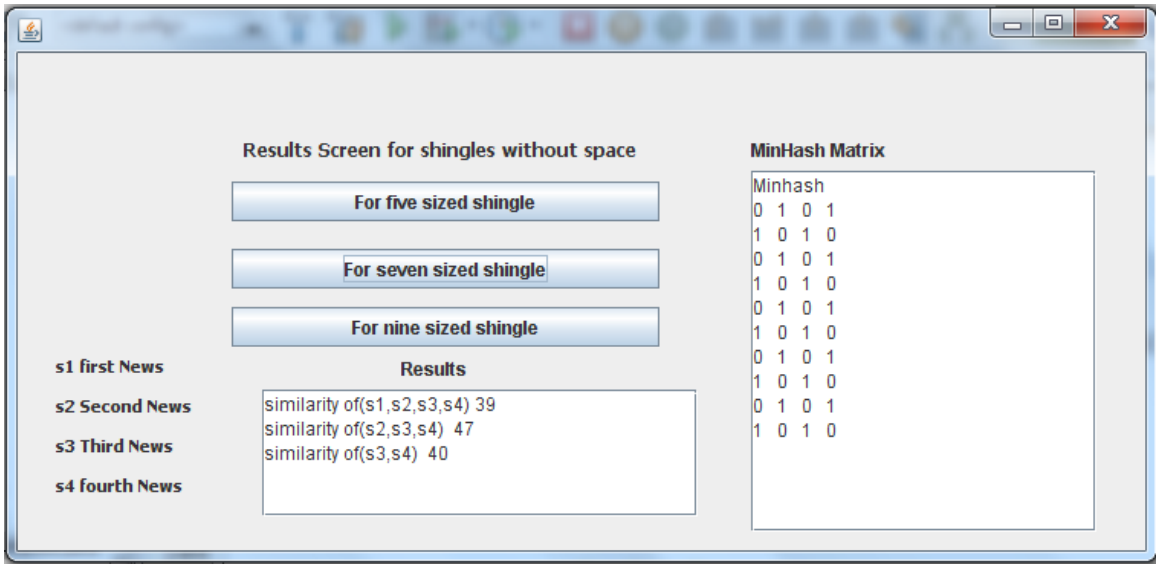

Fig 5.5: shingle without space for k=7

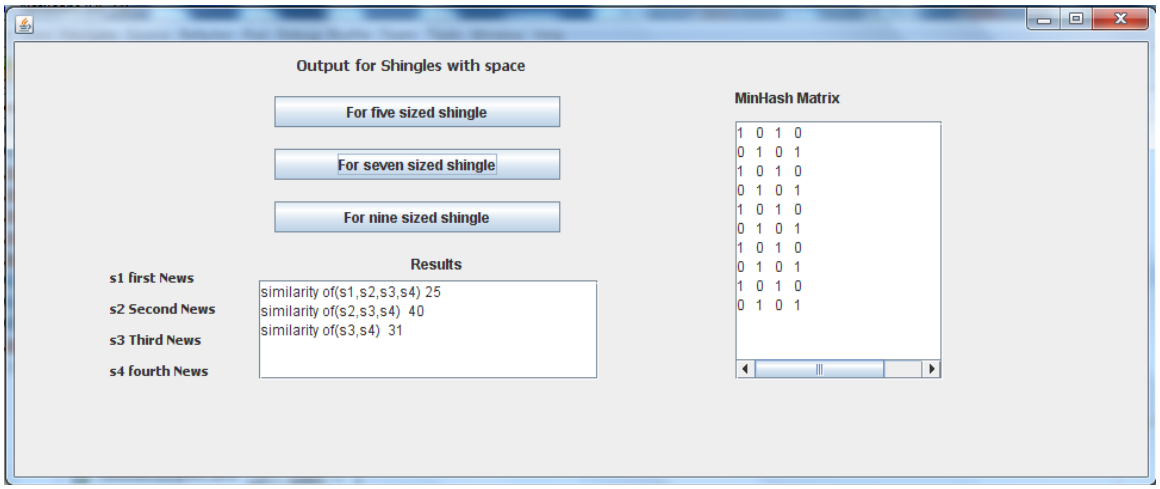

Fig 5.6: shingle with space for k=7

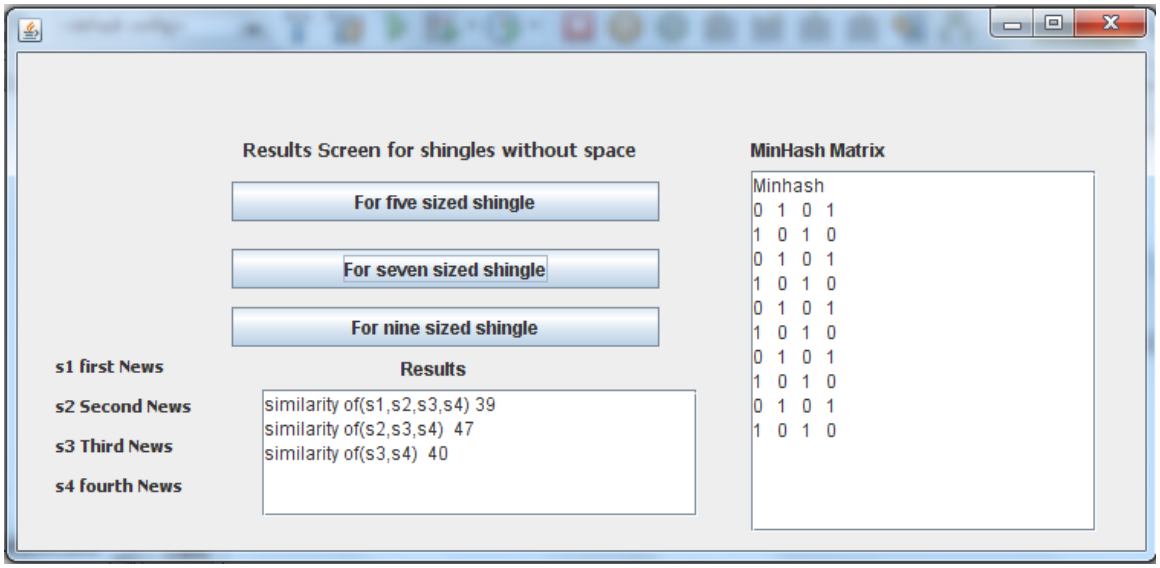

Fig 5.7: shingle without space for k=9

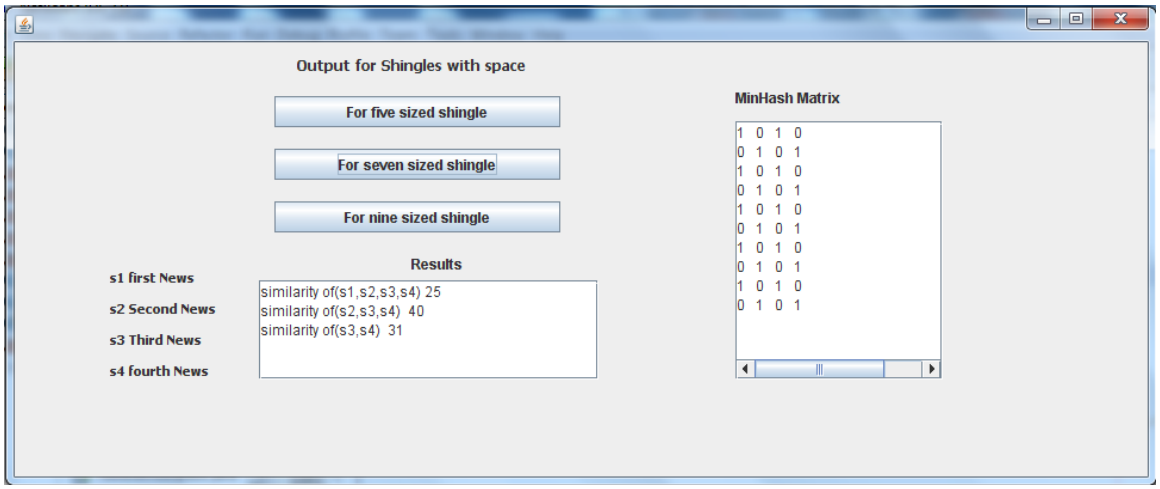

Fig 5.8: shingle with space for k=9

Analysis:

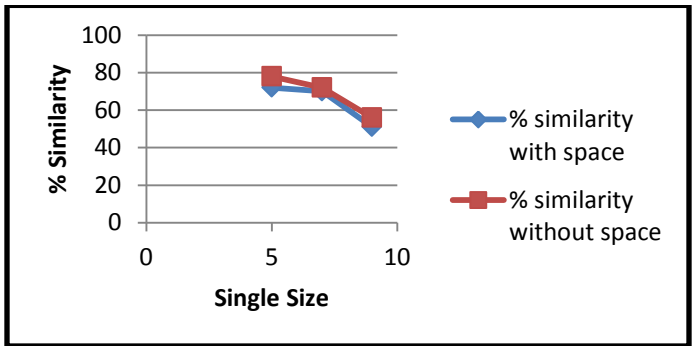

Figure 5.9: Percentage Similarity Analysis of shingle size vs. with and without spacing

Graphical representation of figure 5.9 represents the percentage match with five, seven and nine sized shingles. Relative matching percentage drops as the shingle size grows. At small shingle size matching percentage is more. Further percentage similarity is more for shingles with spacing as compared to shingles without spacing.

a) Similarity analysis with and without stemmers:

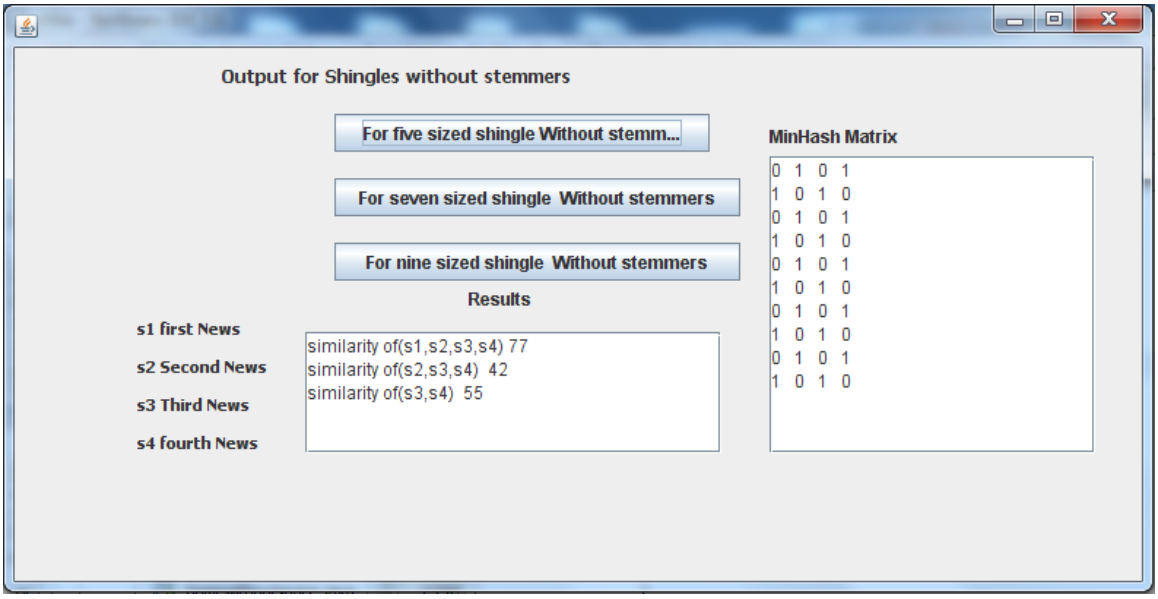

Figure 5.10: shingle without stemmers for k=5

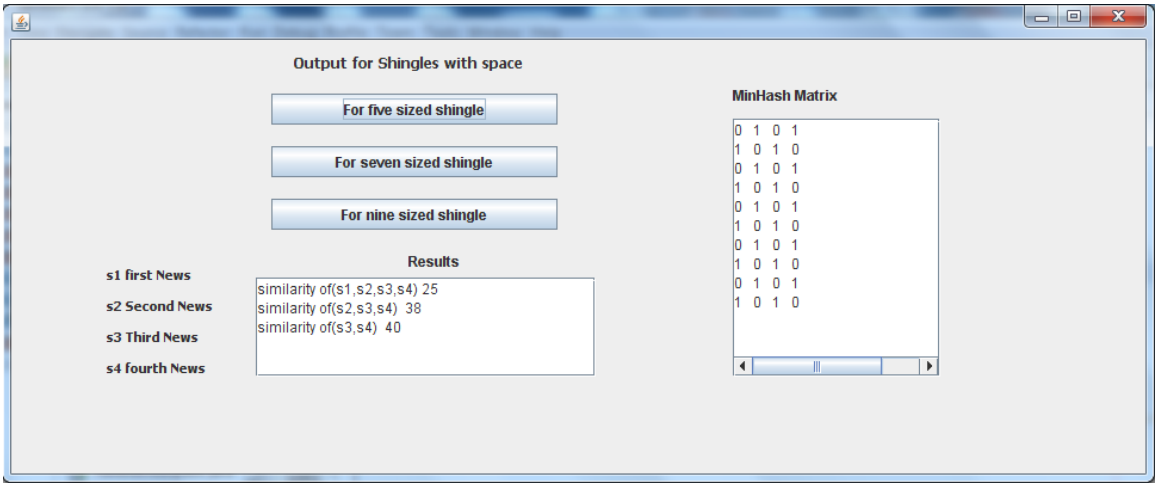

Figure 5.11: shingle with stemmers for k=5

Analysis:

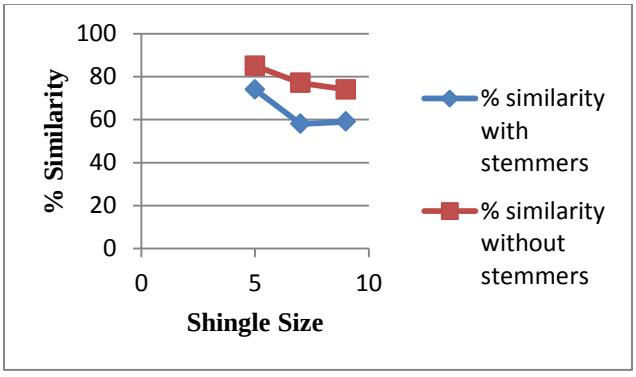

Figure 5.12: Percentage Similarity Analysis of shingle size vs. with and without stemmer

Graphical representation of figure 5.12 represents five, seven and nine sized shingle with and without stemmer has large difference in percentage matching, in terms that at nine sized shingle with stemmers has large matching percentage but without stemmers the matching percentage is less.

b) Similarity analysis with and without stopwords:

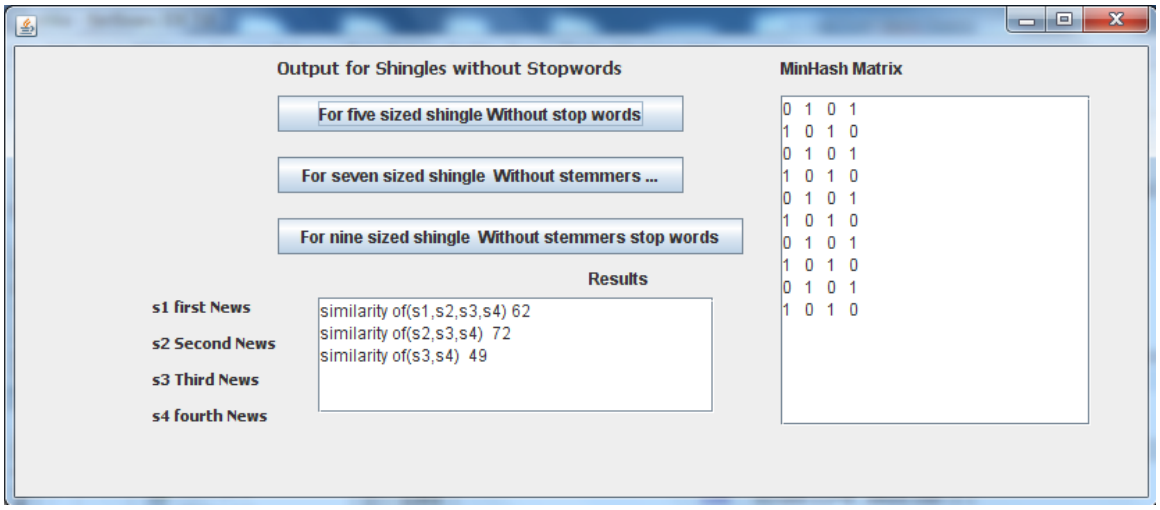

Figure 5.13: shingle without stopword for k=5

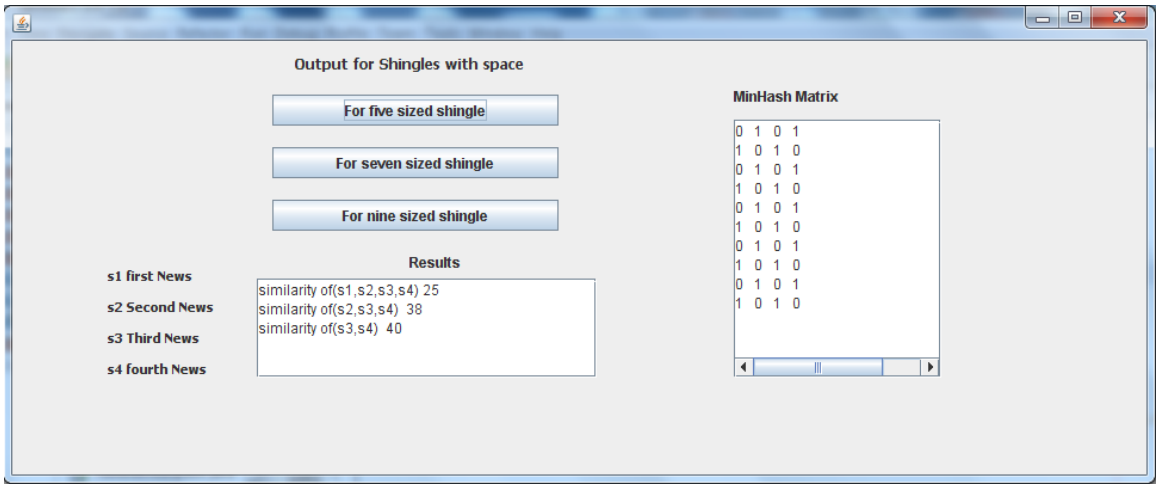

Figure 5.14: shingle with stopwords for k=5

Analysis:

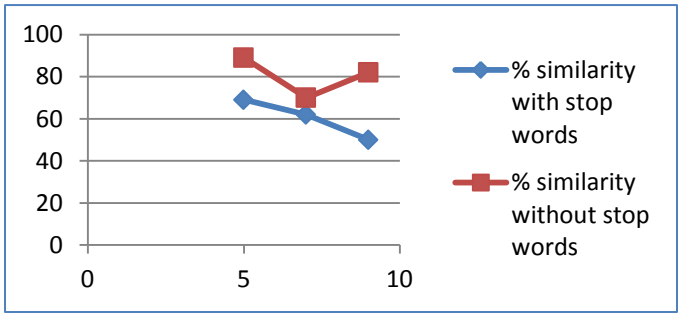

Figure 5.15: Percentage Similarity Analysis of shingle size vs. with and without stop words

Graphical representation of above figure represents that at shingle size five, seven and nine the matching percentage with and without stop words shows the large differences. The percentage similarity is more for the input text without stopwords as compared with text with stopwords. At nine sized shingles matching percentage is very less compare to matching percentage at five sized shingle.

From the above results of implementation, we can analyze that when we compare the documents for textual similarity by removing spaces, stopwords, stemmers from them, almost about 40% of the commonly used words from the text are removed, and therefore focus can be only on the important words. Therefore percentage similarity is high for text without spaces, stopwords and stemmers as compared to text with them.

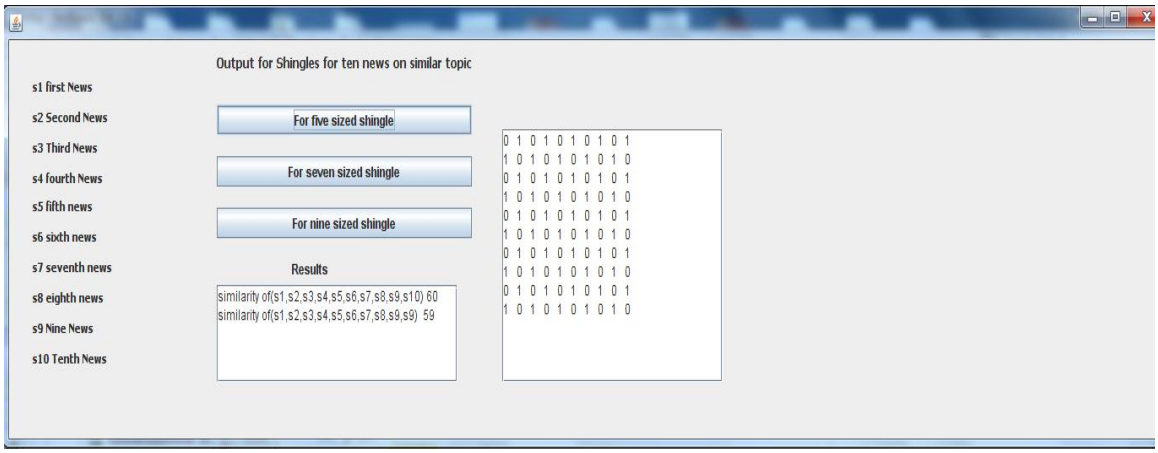

Figure 5.16: shingle for 10 news with space for k=5

Graphical line chart shows the lines for 10 dissimilar news and 10 news with 8 dissimilar news and 2 similar news. Our purpose is to show the algorithm results for big data showing similarity pattern. For 8 dissimilar news percentage similarity is more compared to 10 dissimilar news. In case of two similar news the similarity percentage is growing as the shingle size grows. But the similarity percentage decreases as the news are of dissimilar nature.

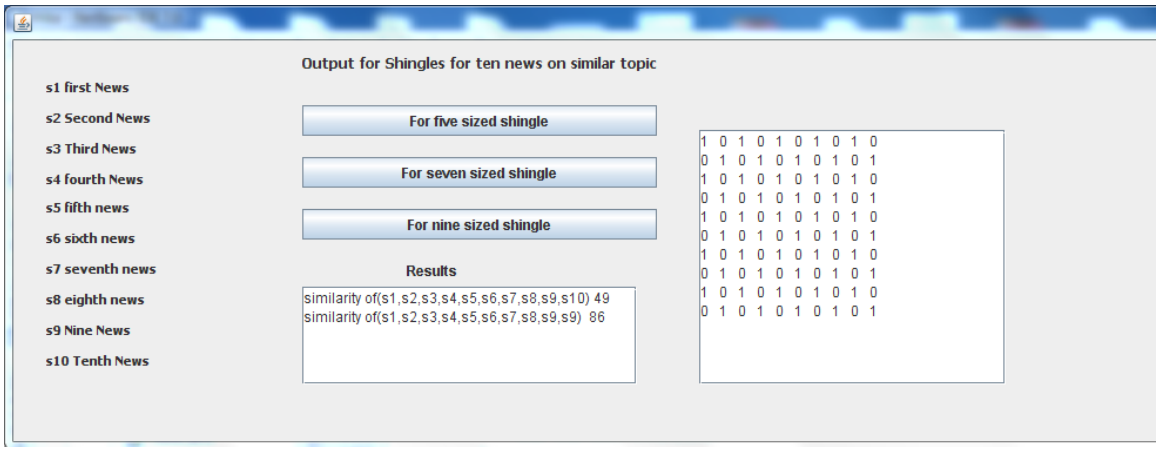

Figure 5.17: Shingle for 10 news with space for k=7

Chapter 6

Conclusion & Future Scope

6.1 Conclusion

A large number of web pages are present on WWW making it a vital source of information. As everyday new contents are produced and hosted on web, its size is continuously increasing. In order to optimize the search process, modern search engines had provided various techniques, but there is still problem of finding similar documents on the web. LSH is the general approach used to find similarity between the documents based on Approximate Nearest Neighbor Search. Signature matrix and banding technique acts as the base for Locality Sensitive Hashing (LSH), characteristic matrix and a few hash functions are used to create signature matrix. The proposed scheme has been successfully implemented and analysis has been done on advantage of using preprocessed data for similarity search.

6.2 Summary of contribution

Locality sensitive hashing has been successfully implemented on news articles and document similarity has been identified.

Effect of preprocessing of textual documents has been experimentally studied with varying single size and it has been analyzed that such steps enhance the search accuracy.

6.3 Future scope

- The work has been done only on textual data which can be extended for similarity search in graph, video image etc. Further, the dataset over which the thesis is carried out, i.e. the news report, can be scaled up to cover more areas likes citation similarity, image similarity *etc*., and with large size of dataset as there is the possibility of scalability.
- Various types of probabilistic data structures like Bloom filter, k-d trees can be applied to identify whether their processing time is better or not.

References

[1] J. R. Smith, “Integrated Spatial and Feature Image Systems: Retrieval, Analysis, and compression”, PhD thesis, Graduate School of Arts and Science, Columbia University, 1997.

[2] “Shingles and near-duplicates” [online] available at http://nlp.stanford.edu/IRbook/html/htmledition/near-duplicates-and-shingling-1.html, last accessed on 30th April, 2014.

[3] A. Bosch, X. Muñoz, and R. Martí, “Which is the best way to organize/classify images by content?”, Image and Vision Computing, vol. 25, no. 6, pp. 778–791, 2007.

[4] Dempsey Chang, Keith V. Nesbitt and Kevin Wilkins, “The Gestalt Principles of Similarity and Proximity Apply to Both the Haptic and Visual Grouping of elements”, Eighth Australasian User Interface Conference (AUIC2007), Vol. 64.

[5] U. Manber, “Finding similar files in a large file system”, Proc. USENIX Conference, 1994.

[6] Thanh Ngoc Dao, Troy Simpson. ”Measuring Similarity between sentences “ Publishedby Springer 2004, XVII, 200 p., Hardcover ISBN: 0-385-29146-6 January 2004.

[7] Maria Kashkur, Serge Parshutin,” Research into Plagiarism Cases and Plagiarism Detection Methods”, Scientific Journal of Riga Technical University Computer Science.

[8] C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer-Verlag, Secaucus, NJ, USA, 2006.

[9] J.L. Bentley, “Multidimensional binary search trees used for associative searching”, Comm. ACM, pp. 509-517, 1975.

[10] P. Ciaccia, M. Patella, and P. Zezula, “M-tree: An efficient access method for similarity search in metric spaces”, In PVLDB, 1997.

[11] G. Beskales, M. A. Soliman, and I. F. IIyas, “Efficient search for the top-k probable nearest neighbors in uncertain databases”, PVLDB, August 2008.

[12] T. Seidl and H. P. Kriegel, “Optimal multi-step k-nearest neighbor search”, SIGMOD, 1998.

[13] A. Beygelzimer, S. Kakade, and J. Langford, “Cover trees for nearest neighbours”, ICML, ACM, New York, USA, pp. 2006.

[14] Sunil Arya, David M. Mount, Ruth Silverman, “An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions”, Fifth Annual ACM- SIAMSymposium on Discrete Algorithms, 1994, pp.

[15] Devroye, "On the equality of Cover and Hart in nearest neighbor discrimination", IEEE Trans. Pattern Analysis 1981.

[16] Devroye, L., Gyorfi, L., Krzyzak, A. & Lugosi, G. ,"On the strong universal consistency of nearest neighbor regression function estimates", Ann. Statist, 22: 1994.

[17] Devroye, L. & Wagner, T.J. ,"The strong uniform consistency of nearest neighbor density estimates", Ann. Statist., 5: 536–540, 1977.

[18] Devroye, L. & Wagner, T.J. , "Nearest neighbor methods in discrimination, In Classification, Pattern Recognition and Reduction of Dimensionality", Handbook of Statistics, 2: 193–197. North-Holland, Amsterdam, 1982.

[19] S.B. Imandoust and Mohammad Bolandraftar, “Application of K-Background”, S B Imandoust et al. Int. Journal of Engineering Research and ApplicationsVol. 3, Issue 5, Sep-Oct 2013, pp.

[20] A. Stupar, S. Michel and R. Schenkel, "RankReduce - processing K-Nearest Neighbor queries on top of MapReduce", In LSDS-IR,2010.

[21] A. Rajaraman, J. Ullman, “Mining of Massive Datasets”, Cambridge University Press, December 30, 2011.

[22] R. Krauthgamer, and J. Lee, “Navigating nets: simple algorithms for proximity search”, Proc. of 15th annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA, USA, pp. 798-807, 2004.

[23] A. Andoni and P. Indyk, “Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions”, Comm. ACM, pp. 117-122, 2008.

[24] M. Datar el al., “Locality-sensitive hashing scheme based on p-stable distributions”, Proc. ACM Symposium on Computational Geometry, 2004.

[25] A. Guttman, “R-trees: A dynamic index structure for spatial searching”, SIGMOD, pp.47-57, 1984.

[26] N. Katayama, and S. Satoh, “The sr-tree: an index structure for high-dimensional nearest neighbor queries”, SIGMOD, pp. 369-380, 1997, ACM-SIAM Symposium on Discrete Algorithms.

[27] K. Ling; G. Wu, "Frequency Based Locality Sensitive Hashing", International Conference on Multimedia Technology (ICMT), pp.4929-4932, 26-28 July 2011.

[28] G. Junhao et al., “Locality-Sensitive Hashing Scheme Based on Dynamic Collision Counting”, ACM, 2012.

[29] M. Slaney, Y. Lifshits and Junfeng, "Optimal Parameters for Locality-Sensitive Hashing", Proc. of the IEEE, vol.100, no.9, pp.2604-2623, Sept. 2012.

[30] D. Knuth, “The Art of Computer Programming”, vol. 3, 1973.

[31] R. Panigrahy, “Entropy-based nearest neighbor algorithm in high dimensions”, Proc. ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, USA, pp.1186-1195, 2006.

[32] Fix, E. & Hodges, J.L., "Nonparametric Discrimination: Consistency Properties”, Randolph Field, Texas, Project 21-49-004, Report No. 4, 1951.

[33] B.H. Bloom, “Space/time tradeoffs in hash coding with allowable errors”, Comm. ACM, vol. 13, no. 7, pp. 422–426, July 1970.

[34] Fritz, "Distribution-free exponential error bound for nearest neighbor Pattern classification", IEEE Trans. Inform. Theory, 21: 552–557, 1975.

[35] A. Broder et al., "Min-wise independent permutations", Proc. Theory of computing, ACM Symposium, New York, USA, pp. 327-336, 1998.

[36] A. Dasgupta, R. Kumar, and T. Sarlós, “Fast Locality-Sensitive Hashing”, ACM conference, New York, USA, pp. 1073-1081, 2011.

[37] Aristides Gionis and Rajeev motwani, “Similarity search in high Dimension via hashing”, Department of Computer Science, Stanford University, Stanford, CA 94305.

[38] Parisa Haghani, “Distributed Similarity Search in High Dimensions using Locality Sensitive Hashing”, EPFL Lausanne; Switzerland.

[39] R. Panigrahy, “Entropy-based nearest neighbor algorithm in high dimensions,” Proc. ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, USA, pp. 1186-1195, 2006.

[40] P. Ciaccia and M. Patella, “Pac nearest neighbor queries: Approximate and controlled search in high-dimensional and metric spaces”, In ICDE, pages 244–255, 2000.

[41] D. Gorisse, M. Cord, and F. Precioso, "Locality-Sensitive Hashing for Chi2 Distance", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.34, no.2, pp.402-409, Feb. 2012.

[42] A.Broder, “On the Resemblance and Containment of Documents”, Proc. Compression and Complexity of Sequences, Washington, DC, USA, pp. 21-29, 1997.

[43] M. S. Charikar, “Similarity estimation techniques from rounding algorithms”, In Proc. of 34 annual ACM symposiums on Theory of computing (STOC '02). ACM, New York, NY, USA, pp. 380-388.

[44] Neeraj Kumar, Li Zhang, and Shree Nayar, “What is a Good Nearest Neighbors Algorithm for Finding Similar Patches in Images?”, Columbia University, University of Wisconsin-Madison.

[45] L. Qin et al., “Multi-probe LSH: Efficient indexing for high-dimensional similarity search”, Proc. VLDB, 2007.

[46] “LSH and General hashing” [online] available at http://cybertron.cg.tuberlin.de/pdci08/imageflight/nn_search.html, last accessed on 30th April, 2014.

[47] Fukunaga, K. and Hostetler, L. (1975) "k-nearest-neighbor Bayes risk estimation", IEEE Trans. Information Theory, 21(3): 285-293.

[48] Gil-Garcia, R. and Pons-Porrata, A., "A New Nearest Neighbor Rule for Text Categorization", Lecture Notes in Computer Science 4225, Springer, New York, 814–823, 2006.

[49] Guo, G., Wang, H., Bell, D., Bi, Y. and Greer, K., (2006) "Using KNN Model for Automatic Text Categorization", Soft Computing –A Fusion of Foundations, Methodologies and Applications 10(5): 423–430.

[50] Gou, J., Du, L. Zhang, Y. and Xiong, T. (2012) "A New Distance-weighted k-Nearest Neighbor Classifier", Journal of Information & Computational Science, 9(6): 1429-1436.

[51] A. Awekar and N. F. Samatova, “Fast Matching for All Pairs Similarity Search”, IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, WI-IAT '09, 15-18 Sept. 2009, Milan, Italy, pp. 295 – 300, 2009.

[52] P. Indyk and R. Motwani, “Approximate nearest neighbor: towards removing the curse of dimensionality”, Proc. Symposium on Theory of Computing, 1998.

List of Publications and Video Links

Publications

- ReetikaBansal and S. Batra, " Textual Similarity Analysis Using Locality Sensitive Hashing ",INDIACom- 2016, IEEE 3rdInternational Conference on Computing for Sustainable Global Development, ISSN 0973-7529 and ISBN 978-93-80544-19-9.[Accepted]

Video Link

https://youtu.be/qX2lfiWWAiA

Reflective Diary

January 2016 – A New Start

After understanding the basics of similarity search problem and doing a little research on what all have done in this area in the past. I could not find a very good problem to work upon. In January, I started reading a book name as 'mining of massive data set' written by D. ullman which was suggested by my guide to get the understanding of what data mining is all about. What areas comes under data mining and to read various applications like MapReduce problem, finding similar items, mining data streams, link analysis (page rank), clustering, recommendation system etc. Earlier I found very difficult to understand these applications as these are very vast areas and new to me. Later my guide suggested reading these research areas and tries to analyze them and also to find the area of your interest. After reading these problems to small extent, I found that similarity search (finding similar items) is the area of my interest. I found the importance of similarity search in daily use and how vast this area is. Then I read few more papers to understand the application i.e. Locality sensitive hashing which can be used to carry out similarity search on datasets and to find out what kind of research is going on in this area. I found a lot of research is being carried out in this area so that they suit the requirements of the application concerned. But no attention has been paid to calculate the similarity search between documents by removing stopwords, stemmers and spacing. I thought of finding out the analysis of the result by removing them. So, by the end of January, I was able to find out a good problem to work upon.

February 2016

In Search for a Solution Next, I started looking out for a probabilistic solution of the problem. Initially, I thought of applying bloom filter to reduce the space complexity but I noticed that already a lot of work has been done in this area using bloom filter. Now I started to look for a new data structure which I can use for hashing. I had a lot of discussion and brainstorming with my guide. Finally I decided universal hashing. Later I

started to look for the data set on which I can apply LSH technique. Finally I thought of taking news articles from various newspapers which may be similar and dissimilar and to carry out research on it. Also by this time my guide had told me about the conference which is going to take place in New Delhi an IEEE international conference named as INDIACOM-2016. So I started with my implementation process and had a lot of problem in implementation. Finally I completed my implementation to some extent and started writing my paper for the conference with some results and send the paper for the conference.

March 2016

Finally I received the news of paper acceptance under the title Textual similarity analysis using locality sensitive hashing. I was very happy that day. Also I was busy with my marriage preparation which is going to take place in March itself. My paper got accepted in Indiacom and a few modifications were asked by the reviewers which I did by the end of the month. Simultaneously I was busy in my new phase of life after marriage. Also I try to complete my implementation and to solve the issues which I came across while implementing.

April 2016

Completed the Implementation of text with and without stopwords, with and without spacing. And started to implement the similarity analysis of news article with and without stemmers. Also I received the email from Indiacom to upload the ppt related with my work. Now I started preparing my presentation so that it can define all the important aspects of my research and even my results upto that date. Then I took a holiday of nearly 12 days from college because I have to go to Europe for holidays after my marriage. After return from trip I again continued on my implementation work. Also by this time we received the mail from department regarding the format of thesis report and all the enclosures we have to submit.

May 2016

Because of college work and due to my interviews in various college for assistant professor, I was not able to attend the conference, and so I requested them to conduct my paper presentation online, therefore finally I presented my paper on skype. Till now I was almost done with my implementation, only few testing need to done. Also by this time I started reading papers to have knowledge about them so that I am able to write literature review of my thesis report. Also I started to look after the important areas which I should cover in my thesis report.

June 2016

The Final Touch After successful implementation and testing, I started writing text of the thesis. Writing literature review took the maximum time. Also I tried to cover all important aspects of my thesis and submitted my thesis report to guide mam for her reviews. Then, I made presentation, video, poster and wrote this diary entry. Besides this, I attended chitkara interview for assistant professor and got selected. This was all about my last semester at Thapar University. Whatever I have done and managed to achieve would not have been possible without the prudent guidance of my guide, Dr. Shalini Batra. I shall always be thankful to her