CHAPTER-7

A GENERIC FRAMEWORK OF DOCUMENT SEARCH ENGINE BASED ON TEXT MINING

7.1 INTRODUCTION

Searching a large set of documents to find out desired documents is always a hectic task if there is no specific retrieving technique. Hence proposed document-weighting and document-ranking algorithms provide an immensely handful technique for these tasks. The developed architecture for document search-engine system provides the base for larger kind of systems to be built on this kind of architecture in near future. The document search-engine incorporates the proposed algorithms with quality assurance activities to provide a complete package.

7.1.1 Basic Concepts:
The research work revolves around some of the important terms which are already known to most of the people associated with the field of computer engineering. The fundamental concepts involved in the work are Data Mining, Text-mining, Intelligent agents, Document-weighting, Document-ranking, Algorithms, Performance and quality assurance. All these concepts and terms are the main base of the research work and used frequently.

7.1.2 Data-Mining:
Data exploration is the evaluation of (frequently substantial) observational data models to uncover unsuspected relationships and to review the data in innovative methods are both clear and the to useful data proprietor. Quite simply data exploration is a procedure of discovering formerly unknown, lucrative and use patterns concealed in data, without any prior theory. Yet another description is "Data exploration is the procedure of finding significant new correlations, patterns, and tendencies by browsing through huge quantities of information saved in information warehouses and by making use of pattern recognition systems as well as statistical and numerical methods." Data exploration offers an easy method of extracting previously unfamiliar, actionable info in the bottom of obtainable data in data warehouses to produce competitive advantages of businesses.
Business intelligence is derived by it in the information warehouse by making use of complex analytical methods for example inductive thinking, sensory networks heuristics, and fluffy reasoning. Data mining programs are backed with some algorithmic methods utilized to remove the important associations within the information. Probably the most often utilized strategies are: organization, sequence-based evaluation, clustering, categorization, and approximation. The data exploration procedure itself is arranged into four main steps: data choice, data change, data mining, and outcome meaning.

7.1.3 Text-Mining:
Text mining is a variance on the discipline called information mining. As text information mining, about comparable to text analytics, pertains normally to the procedure of removing fascinating and non-trivial patterns or understanding from unstructured text files it sometimes instead known. As an expansion of information exploration or understanding finding from (ordered) directories or unstructured text, it may be seen. Text exploration is thought to possess a potential greater than that of info exploration, because probably the most basic kind of saving info is text. In reality, a current research showed that 80% of the business’s advice is included in text files. Entails working with text information which are fundamentally unstructured and unclear, text exploration, nevertheless, can also be a considerably more complicated undertaking (than information exploration) because it. Text mining is a multidisciplinary area, including information access, text evaluation, information removal, clustering, categorization, visualization, database technologies, machine understanding, and info mining (Tan).

Text exploration or understanding discovery from text (KDT) deals with all the equipment supported evaluation of text. It using the calculations and KDT of approaches, data exploration, machine learning and data. them utilizes methods from information access, information removal too as natural language processing (NLP) and links. Therefore, 1 chooses a comparable process just like the KDT process, where not information generally, but text files are in concentrate of the evaluation. From this, fresh concerns for that utilized info exploration approaches appear. One difficulty is the fact there are to cope with difficulties of in the data modeling perspective unstructured data models (Hotho).
Text-mining is termed as the combination of various fields, as shown in Fig.7.1, namely Data Mining, Information retrieval (Extraction), Web Mining, Computational Linguistics & NLP and Statistics.

- **Data Mining**: Data Mining is the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information.

- **Information Retrieval (Extraction)**: Information extraction is the field encompassing the extraction of facts from structured texts such as databases or unstructured texts such as text documents.

**Figure 7.1: Conceptual study of Text-mining**
• Web Mining: Web mining is the field which uses the data mining techniques to automatically discover and extract information from web documents and services such as online repositories (Kosala).

• Natural Language Processing (NLP): The general goal of NLP is to achieve a better understanding of natural language by use of computers. Others include also the employment of simple and durable techniques for the fast processing of text, as they are presented (Hotho).

• Statistics: Statistics is the study of collection, organization and interpretation of data. When this is used in accordance to text mining the data is unstructured text.

7.1.4 Intelligent Agents:
Brokers, i.e. particular kinds of computer programs, are becoming ever more popular in processing globe in modern times. Some of the causes for this recognition are their versatility, modularity and basic usefulness to a broad variety of issues.

An intelligent computer software broker could be thought as: “A software entity which functions continuously and autonomously in a particular environment... able to carry out activities in a flexible and intelligent manner that is responsive to changes in the environment... Ideally, an agent that functions continuously in an environment over a long period of time would be able to learn from its experience ... an agent that inhabits an environment with other agents and processes to be able to communicate and cooperate with them ... perhaps move from place to place in doing so.”

Smart agents perform the part of personnel by enabling supervisors to delegate function they might did for the broker computer software. Broker technologies is obtaining its way in to many fresh methods, including decision support methods, where they execute many of the crucial decision support duties once consigned to a distinctively individual exercise. Software agents are helpful in automating repeated tasks, discovering and blocking information, wisely outlining complicated information, and etc, but more significantly, only like their individual counterparts, clever agents may have ability to understand in the supervisors and actually make suggestions for them regarding a special strategy. Agents possess several typical features, for example their capability to collaborate, convey, and organize with additional agents in a numerous agent method. Each broker is able of behaving autonomously, cooperatively, and jointly to attain the group aim of the program. The
co-ordination ability helps control issue solving so that co-operating agents interact like a group.

As a result of the intense improvement of info supply accessible on the Web and on the authorities, company, and medical sources, it is becoming very crucial for your customers to use automatic and smart resources to remove understanding from them. Smart brokers may assist creating personal computers simpler to use, permit discovering and blocking information, customizing opinions of information and automating function. An Intelligent broker is computer software that helps functions and individuals for the kids. Smart agents work by enabling individuals to delegate work they might did for the broker computer software. In line with the ability and brains to do duties the brokers may be classified in to four unique degrees:

Level 0: Agents who take straightforward orders *i.e. Robots*
Level 1: Agents who perform user initiated search by keywords *i.e. Search-engines*
Level 2: Agents who have user profiles and perform actions according to the privileges given to that user *i.e. “Software agents”*
Level 3: Deductive or considering capacities and brokers who have understanding capacities

*i.e. Autobots – kind of robots who can perform tasks by their own learning and understanding capabilities. (Like the one shown in *Transformers movie* series).*

7.1.5 Algorithms:
The term algorithm originates from the title of a textbook was written by a Persian author who on arithmetic. In computer executive the phrase 'Algorithm' has a particular importance plus it has come to reference a procedure you can use by a computer for your remedy of a difficulty. This really is what makes algorithm diverse from conditions for example procedure, method, or approach. The term ‘Algorithm’ can be defined as:

“A finite set of instructions that, if followed, accomplish a particular task.”

An algorithm comprises a limited set of measures, operations may be required one or more by each of which. The chance of the pc performing these operations requires that particular limitations be positioned on the sort of operations an formula may contain (Horowitz). One is usually applying a process of fixing an issue which includes been formerly invented, when one produces a pc system. This technique is frequently independent of the specific pc to be used: it's likely to be similarly suitable
for most computers. Regardless, it is the process, maybe not the pc plan itself, which should be examined to understand the way the difficulty will be bombarded. The definition of formula is globally employed in computer technology to explain problem-solving systems acceptable for execution as computer applications. A lot of work should proceed into comprehending and identifying the issue to be fixed, managing its intricacy, and decomposing it in to smaller subtasks that may be readily implemented (Sedgewick), when a really substantial computer plan is to be created.

Record – Weighting (Buckley) : since the data is introduced that good weighting methods are far more significant than characteristic selection procedure for your good operation in information collection, The value of great weighting practices in details access is extreme critical. You will find few techniques to compute ad-hoc weights of the records such as tf*idf approach, and meaning feedback technique. These strategies are derived from two significant variables; one is statistical event info and still another is a brief history of how nicely this characteristic has done before. In lots of scenarios, it's difficult to get background info and therefore initial dumbbells tend to be based only on record info.

TF*IDF Method:
The tf*idf method (term frequency times inverse document frequency) assigns weight wik to phrase (word) Tk in document Di, and in inverse percentage to the amount of files to that the term is designated.

Importance Feedback:
In this approach, an issue is thought to signify a continuing info require, and there have been numerous files previously observed for each issue, some part which has been evaluated important. Then the official TREC routing operate and assigns weights centered on likelihood of events in non and related - relevant files.

Document-Ranking:
Ranking algorithms are mainly used to rank or index the documents based on some deciding factor. A position is a connection between some products so that, for almost any two products, the very first is possibly 'ranked greater than', 'ranked less than' or 'ranked corresponding to' the 2nd. The positions themselves are completely purchased and have the ability to assess complicated details based on particular requirements.

There are few existing document ranking algorithms such as Google’s PageRank, RankNet, AdaRank, Coordinate ascent etc. These algorithms work in the internet
environment and are highly optimized for getting best possible results. Google’s PageRank is very famous algorithms and the search engine developed by Google Inc., named ‘Google’ itself, works on the basis of this PageRank algorithm (Dang). Idea behind PageRank (Brin) is “The document collection is analyzed in terms of links between documents – link analysis. This graph is regarded as Markov Chain* on which a random walk is performed. Score of each web page is read from the stationary distribution of the underlying transition matrix – the left eigenvector.”

7.1.6 Quality Assurance:
Quality has been the important factor in each and every product or services that human-being uses. Quality plays an important role in convincing people to buy particular products or to use particular services. For ex., if one wants to buy a new mobile phone then apart from the features quality of the material used in manufacturing that mobile phone, quality of the accessories provided with the mobile phone, quality of the system software(s) installed into mobile etc. also plays an important role. There are many definitions exist for the term ‘Quality’, and they all are given by respective organizations or people. Let’s take a look into few definitions (Fitzpatrick).

"Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements".
- **German Industry Standard DIN 55350 Part 11**

"Quality is the totality of features and characteristics of a product or a service that bears on its ability to satisfy the given needs".

"The totality of features and characteristics of a software product that bear on its ability to satisfy given needs: for example, conform to specifications… The degree to which software possesses a desired combination of attributes… The degree to which a customer or user perceives that software meets his or her composite expectations… The composite characteristics of software that determine the degree to which the software in use will meet the expectations of the customers".

Today, the quality of the code and created algorithm performs an essential function in calculating the operation of both the algorithm and code because both the algorithm and code happen to be the essential a part of the object - oriented methods.
7.2 PERCEPTION, STUDY AND LITERATURE REVIEW

A literature review is a text composed by somebody to think about the crucial points of present understanding including purposeful results along with methodological and theoretical efforts to a special issue. Books evaluations are as, and secondary sources, don't statement any fresh or unique fresh work. Lots of work has been completed in these areas by scientists. This part of study enlightens briefly on some of work done by those researchers. The work from various books, papers, articles, journals has been referred for this purpose. It is humble approach to thank those researchers whose work done will be referred in this research. Some of them are mentioned in this article.

7.2.1 Introduction to the research work:

Many of the large and complex data storage systems have become the integral part of our lives. It is like human-being cannot be completely detached from the data storage systems. We use many systems in our day-to-day life varying from personal-use systems to commercial-use systems such as Banking system, Railway and Air-ticket booking system, Product purchasing system, Information finding system, Video browsing system, Email system and also importantly Document-finding system or as we call it “Search Engine” system etc. In all this systems we need to interact with the huge amount of data and there could be possibility that a normal user can get confused using these systems while searching for his/her relevant information. She/he may not have the knowledge about how to get the relevant information from the huge amount of data.

As we all know, we everyday use document finding system or so-called Search-Engine system in one or other way. Now, suppose if the “Ranking Algorithms” are not there then imagine what would happen to the Search-Engine system. People type in few things to search for but the results will come in some random fashion. Then we need to go through all the results one by one and to look for whether it is relevant for us or not. Hectic work and we are confused and frustrated..!! That is where a text-mining approach and ranking of documents come into picture. 'Text Mining' is a form of the discipline called information mining. As text information mining, closely equal to text analytics, describes the procedure for drawing high-quality info from unstructured text it sometimes instead known. ‘Ranking algorithms’ are mainly used to rank or index the documents based on some weight-factor or relevance.
The research work evolves on designing and implementing a new ‘Document Ranking Algorithm’ and providing a framework for an agent-based text mining approach.

7.2.2 Literature Review:
Text mining, also called text information exploration or understanding finding from textual databases, relates usually to the procedure of removing fascinating and non-trivial patterns or knowledge from unstructured text files. As an expansion of info exploration or information discovery from (ordered) directories, it may be seen.

(1) Tan (1999)

The writer says that the absolute most basic kind of keeping info is text, text exploration is thought to possess a potential greater than that of information exploration. In reality, a recent research showed that 80% of the business's info is included in text files. Entails working with text information which are fundamentally unstructured and unclear. text exploration, nevertheless, can also be a more complicated undertaking (than data exploration) because it. Text mining is a multidisciplinary area, including information access, text evaluation, information removal, clustering, categorization, visualization, database, technologies, machine understanding, and information mining.

In today's situation several businesses, academic organizations, universities, company businesses, hospitals, study facilities and so on. cope with large number of information that's in kind of text files. So finding out important information and getting important patterns and commercial as well as business details is very important task and there it comes, the field of text mining. As mining is multidisciplinary area combining many other disciplines including text analytics, info access, categorization, machine studying and data exploration, the research of the disciplines can be very important to fully comprehend text - mining - text.

(2) Hotho et al. (2005)

The amazingly big quantity of info saved in unstructured texts can not merely be utilized for additional running by text is typically handled by computers, which as basic sequences of character strings. Therefore, particular (pre) - processing approaches needed are and calculations to be able to take out useful patterns. Text mining pertains normally to the procedure of removing interesting info and
understanding from unstructured text. Now, as this unstructured text must be pre-
processed before obtaining utilized by computers for more processing, different pre-
processing methods including, tokenization, coming, speech tagging parsing and so on., could be utilized. There are several existing algorithms which can extract useful
patterns from the text documents as well.

(3) Gupta and Lehal (2009)
The writer provides that text mining resembles data mining, except that data
exploration resources are made to take care of organized data from directories, but
text mining may use unstructured or semi-structured data models such as emails, full-
text records and HTML documents etc. As text mining is a considerably better answer
for businesses, a result. Up to now, nevertheless, improvement efforts and most
research have dedicated to information exploration efforts making use of organised
information. The difficulty launched by text exploration is apparent: natural language
was created for individuals to report info and to communicate together, and computers
are quite a distance from understanding natural language. Data mining normally deals
with the structured data. It may not be utilized with the unstructured information for
example text files, emails, HTML documents and so on. In today’s world, most of the
companies and organizations deal with both the structured and unstructured data, to
deal with the unstructured data text-mining is an essential technique. There is so much
research and development work going on data mining and text-mining to get more
effective solutions for problems regarding extracting useful patterns from structured
and unstructured text.

(4) Tudor et al. (2009)
The author deals with text data from different organizations is an intellectual property
of their own and they need some techniques and people to manage and handle this
intellectual property. In a aggressive world, modern businesses concentrate on
finding, keeping, moving and effectively to be able to better handle their mental
capital making use of their own advice. To manage this processed information, that is
knowledge, is not an easy task. Tons of text data is to be created, processed,
transmitted to different locations every day, to manage this knowledge one needs to
take extensive efforts while assuring their intellectuality. So managers, who are
managing all these data and knowledge, have to find out different techniques and
methods to deal with the security of intellectuality of data.
On another hands Agents, I.e. particular kinds of computer programs, are becoming ever more popular in processing globe in modern times. A few of the causes for this recognition are their versatility, modularity and basic usefulness to a broad variety of issues (data blocking and evaluation brokering, information, condition checking and alarm generation, workflow management, private aid, simulation and gaming). As a result of the intense improvement of info supply accessible on the Web and on the authorities, company, and medical sources, it is becoming very crucial for your customers to use automatic and smart resources to take out understanding from them. Smart brokers may assist creating personal computers simpler to use, permit discovering and blocking information, customizing opinions of information and automating function. An Intelligent broker is computer software that helps functions and individuals for the kids. Smart agents work by enabling individuals to delegate work they might did for the broker computer software.

The chief factors behind growing recognition of brokers, that is applications auto -bots, are their versatility, basic and versatility and modularity usefulness to a broad variety of difficulties in health-related technology areas and various architectural. One more reason is the very large chunk of data availability from which users need to find out interesting and important information, and to do it manually is the hectic and time consuming work. Therefore, because the outcomes brokers have obtain recognition within the area of info access in the numerous sorts of unstructured and organised info resources.

(5) Rajan and Sarvanan (2008)
The writers must have sufficient site understanding of the data mining program and suggest something where exploration methods and Information accessing are becoming increasingly crucial. The writer also suggest a generalized construction of the Information exploration program making use of independent Smart Agents. Huge advantages can be provided by automation in price and time. Our particular aims within this document contain growth of the versatile design for an automatic information exploration program making use of smart brokers. Additionally using the aid of an smart agent picking the best formula and approaches for the directories become simple.

Now, as agents are becoming very popular for retrieving useful and important information from structured and unstructured data sources there is a growing need of
framework or underlying architecture based on which these agents work autonomously. And in the above paper, authors have suggested a framework for such kind of system.

(6) Buckley (1993)
The author suggest to develop and implement good document weighting methods which are extremely important for retrieval of information. Proof demonstrates that the great weighting systems works nicely and provides better outcomes than the function selection procedure and it is implied that two require to proceed hand-in-hand to be powerful. File weights derive from a variety of guidelines, two might be statistical event info and the background of how nicely this characteristic have done within the previous. From the early days of computer engineering, algorithms have been the important field and they have the special significance in implementing system programs. Algorithms are used by us every day. For instance, a formula for making 'daal-rice' is an formula. Most applications, except for some intelligence programs, contain calculations. Creating refined algorithms, algorithms which are straightforward and need the actions possible, is among the main problems in encoding.

The writer clarify that basically, TF-IDF works by identifying the comparable regularity of phrases in a particular document when compared with the inverse percentage of that term within the whole document corpus. Intuitively, this computation establishes how applicable a specified term is in a special file. Words these are common in one or perhaps a little number of files generally have higher TFIDF numbers than typical phrases including prepositions and articles. The proper process of implementing TF-IDF has some slight variations over-all its apps. The method utilized in the study function for file-weighting is influenced from the TF-IDF approach. The only distinction is that the event of 'keyword' in each and every record, where it seems, is measured and then to compute the weight of any special record, the count (frequency of 'keyword' in that record) is increased by the logarithm of the department of absolute number of documents by the number of documents containing 'keyword'.
(8) Dang and Bruce (1997)

The authors offer an efficient rating platform that is surely the primary part of any info access (IR) program. Several rating models have already been offered, the hottest that are BM25 and the language modeling framework. These versions utilize only a few attributes including phrase frequency, inverse document frequency and document size. They create relatively great outcomes and have the benefit to be rapidly. Because it necessitates a critical change within the fundamental design when more functions become obtainable, nonetheless, integrating them into these designs is generally challenging. Like, the BM25 model was changed to contain PageRank as a preceding or to integrate term distance information. The ranking method proposed in the research work based on the concept of quick sort algorithm. Quick sort algorithm explains: “To sort the numbers given in ascending order, first select a pivot element and then put the numbers smaller than pivot element before pivotal position and put the numbers greater than pivot element after pivotal position. Then again select the pivot element from the list of greater numbers and repeat the procedure.”

(9) Upstill (2005)

Record access on the Internet (WWW), perhaps the world's biggest group of files, is a difficult and significant endeavor. The dimensions of the Internet is astounding, consisting of no less than five million freely visible web files dispersed on millions of server(s) world-wide. The main element of the document access in Internet situation is the document position function. The capabilities of contemporary research methods often include several kinds of file proof. A few of this proof, including textual info, is gathered locally for each file within the corpus. Additional proof, such as outside document explanations or suggestions, is accumulated through an assessment of the circumstance of a document inside the internet data.

To rank the documents available on World Wide Web (WWW) is immensely difficult and time-consuming task as WWW consists of at least ten-to-fifteen billion web documents available publically. Because of this, the ranking algorithms working in WWW scenario considers many factors to effectively rank the web documents such as following:

- Most frequently used website and Ranking of website/page rank
- No. of times the ‘keyword’ is used in title/URL etc.

7.2.3 Problem Definition:
Too much fragmentary details and non-contextual data exists in way too many locations. This makes the job of finding important info on a specific matter somewhat like getting a hook within the proverbial haystack. It's now apparent that a crucial problem is to develop approaches that enable this info to become effortlessly recovered and quickly used by individual customers and the remedy to this is to produce appropriate file weighting and position calculations. So the proposed document search-engine will incorporate a novel approach to document-weighting and document-ranking which will improve users’ experience of finding the relevant documents from a large dataset (document set).

7.2.4 Objectives, Goals and Scope of Research Work:
The chief aim of the study function is to supply an agent-based text-mining platform and to create and design two calculations among which may be utilized for document-weighting and yet another will be used for document-position. The challenge lies in defining the factor on which the algorithms work and hence the ‘weight of the document’ has been defined as the deciding factor and so in effect of that the scope of this research work is confined to Reuters-21578 Dataset and to the ‘Document Search Engine’.

This research work is intended to design and develop document search engine programs that provide relevant results in the ranked form to users for their inputted query words.

Main objectives of the research work are:

a) To design and develop algorithms for document-weighting and document-ranking and incorporate them into document search-engine, this strives to give users relevant documents for their respective searches.

b) To make critical decisions faster and with great ease doing by automation rather than doing them by manually.

c) Users can see the producing details (text or series) in a suitable (e.g. internet additionally) format.

7.2.5 Principles:
The proposed agent-based text mining framework might works in cooperation of three different agents:

i) Key-order and Keyword based agent  

ii) Word-sense based agent  

iii) Intelligent agent
This research work talks about the third agent (Intelligent agent) whose work is to provide a novel approach to document-weighting and document-ranking. This whole thesis work evolves on:

i. How the documents to be searched in a huge dataset of documents based on given keyword?
   - The inputted query word will be compared to the content of articles and articles consist of that query word will be grouped together.

ii. How the documents would be weighted?
   - Frequency (occurrences in a document) of inputted word in the particular document will be counted and then (tf*idf) equation will be applied to calculate weight of that document.

iii. How the documents would be ranked?
   - The concept of quick sort algorithm will be applied to rank the weighted documents.

This work practically considered the Reuters-21578 dataset from Reuters Corporation which consist of ‘News Articles’. The Reuters-21578 dataset files (news articles) are in SGML format and the dataset consist of five basic categories: People, Places, Orgs, Exchanges and Topics.

7.2.6 Methodology:
The methodology describes how I need to go for the overall work completion and reach towards the end with satisfactory results.

- First, the literature survey would be completed by studying reference papers and books, so the overall concept is to be known about ‘what is text mining?’, ‘What are Agents?’, ‘How they works?’ etc.
- Analysis phase would be completed in which a particular domain for research work needs to be decided. All the requirements would be identified and the risk identification, mitigation and monitoring would be performed.
- Style stage might include various UML diagrams for example Use-Case diagram, Sequence diagram, Cooperation diagram, Part diagram, and Implementation diagram and creating of algorithms.
• Implementation phase would consist of coding (programs) for algorithms developed in the design phase.
• Testing phase would encompass test cases and quality assurance activities. Test cases are required to check whether the important functionalities are working correctly or not.
• Results discussion and analysis would be the next phase of this dissertation work and that would discuss about results obtained while applying algorithms to Reuters-21578 news articles dataset.

7.2.7 Benefits of Proposed Work:
• Individuals use various Se's for example Yahoo, Yahoo, and Msn etc, * As all of us understand, within the Web globe. But in the developing countries like India Internet access is not available to everywhere and to everyone who knows a little about computers. But there will be people who are seeking for proper information related to their queries despite of not having access to Internet.
• So the research work is mainly proposed for ‘Document Search Engine’ which will work on a novel ‘Document-ranking’ algorithm.

7.2.8 Research & Development Challenge
The research work has been carried out to propose an agent-based text-mining approach and to provide insights to the novel way to weight and rank documents. The objective of the proposed work is purely based on document search engine and to provide a framework for that. The main challenge was to define the factors on which the algorithms work. Hence it had been considered to take the ‘weight’ as the main and important factor. The weight of the documents was calculated based on the frequency of the word inputted as a query to the program. Another challenge was while writing the programs for the algorithm because the Reuter-21578 dataset files are in SGML format, and it was difficult to read and scan them directly. Hence two different utility programs have been used to read those SGML files. Among those to one was used to convert SGML files into XML files and another was used to create to tree-node structure such that SGML tags would be used as nodes and the content in between those tags would be used as values.

7.2.9 Overview of the Solution:
The work basically proposes an agent-based text-mining framework incorporating document-weighting and document-ranking algorithms. The quality assurance
practices have also been performed to assess the quality of work done. The working steps are as follows:

i) Users have to enter a word (constrained to the dataset, because they are news articles).

ii) Based on inputted word ‘Document-finding Algorithm’ groups together the documents (articles) in which inputted word appears.

iii) For those, documents which are grouped together in the previous step, the weight value have been calculated using ‘Document-weighting Algorithm’.

iv) Then the ‘Document-ranking Algorithm’ will be applied and the documents will be ranked and presented to the user.

Document-Finding Algorithm Steps:

i) Takes input word from user.

ii) Initialize the counter variable to zero, which will indicate the no. of documents in which the inputted word appears.

iii) Look Search for the files, within the dataset, by which the inputted word appears.

iv) Increment the counter by 1 when the word found in the document and add the document to the final list which will consist of final resulted documents.

v) Display the list to the user.

Document-Weighting Algorithm Steps:

i) Take the final list consist of final resulted documents.

ii) Count the occurrences (frequency) of the inputted word in each and every document of the list.

iii) Then apply the (tf*idf) equation to calculate weight of that document. Here (tf) signifies term regularity and (idf) represents inverse record regularity.

iv) Create one array consisting of weight of the resulted documents which are stored in final list.

v) Display the list of documents and their respective weight values to the user.

Document-Ranking Algorithm Steps:

i) Find the average weight value by adding the weights of all documents in ‘final list’ and dividing that by ‘total documents in which word appears’.

ii) For all the documents in ‘final list’, compare the weight of document with the ‘avg. weight’.
iii) The documents which are having less weight than the ‘avg. weight’ will be added to another list ‘M’ and their weights are added to another array ‘weight1’ and the documents which are having more weight than ‘avg. weight’ will be added to another list ‘N’ and their respective weights are added to another array ‘weight2’.

iv) If there are more than one documents in the list ‘N’ then recursively call the Algorithm with parameters related to documents in list ‘N’. If all the documents in the list ‘N’ are having same ‘weight value’ then there is no need to repeat the process because every time the average weight will be same and thus it will go into the infinite loop. And if there is only one document in the list ‘N’ then add that document in the ‘final’ list.

v) If there are more than one documents in the list ‘M’ then recursively call the Algorithm with parameters related to documents in list ‘M’. If all the documents in the list ‘N’ are having same ‘weight value’ then there is no need to repeat the process because every time the average weight will be same and thus it will go into the infinite loop. And if there is only one document in the list ‘N’ then add that document in the ‘final’ list.

vi) Display the documents in the list ‘final’ to the user.

Thus ‘final’ will have all the ranked documents with their title and data which will be displayed to user who had given the search query using particular keyword.

The quality assurance has been done at three different phases of SDLC:

a) Quality Assurance at Requirements Analysis Phase
b) Quality Assurance at Design Phase
c) Quality Assurance at Implementation Phase

And more on following tasks will be performed at each level.

a) Traceability Analysis:

Traceability analysis is an important task to be carried out because it provides the traceability of design and code to the functional requirements. Generally, traceability matrixes are to be prepared showing the traceability of design and code to the functional requirements of the project.

If the traceability matrix is of design-functional requirements then functional requirements can be taken as different columns and design modules as
different rows and then one should check out whether for the given requirement particular design module is available or not. If yes then put a mark over there otherwise just leave it empty.

b) Checklist Document

The checklist document contains the type of questionnaire regarding different issues focusing on different components at three important phases of SDLC, namely requirements analysis, design and implementation. The checklist document will be prepared by making list of various questions which are important at that particular phase of the SDLC such as functional specification, non-functional specification at requirements analysis phase, design feasibility and reliability at design phase, interfaces code and data feasibility and reliability at implementation phase.

7.3 SYSTEM ANALYSIS

System study elaborates about the architecture of the system and the tasks to be completed to make system working. The main objective of the research work is to provide an agent-based text-mining framework and to design and develop two algorithms one of which will be used for document-weighting and another will be used for document-ranking.

7.3.1 System Architecture:

The program structures is the conceptual design that determines the arrangement, and perspectives of the machine. The system design proven in beneath fig. 7.2 is the general portrayal of the program and it determines the practical parts of the system, e.g. Brokers. Though the concern of the research work is only related to novice techniques for document-weighting and document-ranking (Agent 3 in the diagram), the architecture should be considered as the proposed model based on which the overall Document-Search Engine system can be built. Overall system can be decomposed into three functional parts as they are agents. Agent 1 is key-order and key-phrase based agent whose work is to find out the important keyword from the inputted text. Agent 2 is word-sense and context based agent which decides the context of the keyword and find out documents come into that context. The concern of the research work is on the working of the agent 3 which is responsible for weighting the documents based on weighting-algorithm designed (which uses
basically $tf^*idf$ method) and then ranks those documents using ranking-algorithm designed.

![Architecture of the proposed system (web application)](image-url)

Figure 7.2: Architecture of the proposed system (web application)
To develop this system, the work is divided into three project management tasks which are as follows:

1) Pre-Requisite System Management Tasks.
2) Primary System Management Tasks.
3) Supportive System Management Tasks.

The work evolves on developing two novel algorithms for document-weighting and document-ranking and thus providing novel document search engine architecture based on text-mining concepts.

7.3.2 Pre-Requisite System Management Tasks
The pre-requisite system management tasks are those tasks which are required prior to start with the development of the system. This will include:

I. Information Gathering
II. Defining Requirements

7.3.2.1 Information Gathering
The first step in system to start with would be information gathering. This is the activity which encompasses various techniques for gathering information about the research work to be done. The different tasks are:

- Gathering and reading documents regarding previous work-done.
- Review existing systems or frameworks
- Interview different people who are supposed to use the system or software
- Sort out the requirements

These all tasks are really important in understanding the existing things and deciding own objective that is what exactly we need to do. For this research work several research papers and several algorithms on weighting and ranking of documents are referred. This work focuses on the text-mining concepts and tried to provide another text-mining framework.

7.3.2.2 Defining Requirements:
After gathering enough details one needs to decide the requirement for his own research work. Each and every requirement should be stated clearly with clear idea in mind what will be achieved through that requirement. Based on requirements defined, the clear objectives to be achieved throughout the research work can be stated. Once the person has clear requirements for his/her system work it would be easy to go on
with the further tasks such as analysis, design and modeling, implementation or coding, testing and analyzing results.

7.3.3 Primary System Management Tasks
Primary project management tasks are those tasks which can be called as the core of project. These tasks include:

i) Concept Development
ii) Analysis and Design
iii) Implementation (Coding)
iv) Testing (Program)

7.3.3.1 Concept Development:
Theory development is really a procedure powered by a set person wants and goal item specs, that are subsequently changed into some conceptual styles and possible technical options. These options represent an rough explanation of operating concepts, kind, and item attributes. Frequently, these ideas are combined with commercial style versions and fresh prototypes that aid to make final choices. Concept development is an important task to understand about user needs, the need to develop a system or software, to get an insight of what exactly needed as an outcome from developing the system or software or framework or algorithm. To provide effective and innovative solutions to the existing real world application problems concept development is essential.

7.3.3.2 Analysis and Design:
Evaluation is the procedure of splitting a complex issue or substance into smaller components to get a much better comprehension of it. Evaluation can be used in many areas for example technology, biochemistry and architectural, company, cleverness, linguistics, math and many more. In the computer science and engineering field the tasks performed in analysis phase are Requirements analysis, Development methodology analysis, and Concept design and implementation analysis. Design refers to the outline of the work to be done in form of graphical based representations. Design has been done for products or system to be developed. It would layout the possible aesthetic look and feel of the product to be developed. Because it signifies the complete program or item in a method design is the significant job in any area; such areas are new design, style design, product design, executive design, interaction design, visual design and so many more. In computer science and engineering field
the design phase consists of design of user interface for systems or software which can be depicted by the use of UML diagrams that consist of class diagrams, use-case diagrams, sequence diagrams, deployment diagram, object diagrams, component diagram etc. Design also includes developing new algorithms or providing new approaches to problem solution. In-fact, developing an algorithm or algorithms comes under modeling that is the subpart of design.

7.3.3.3 Execution
Execution is the real performance of the strategy, style, design or standards. In computer engineering field it can be termed as the actual program for the algorithm developed that can be executed on any computer system in the world despite of its underlying architecture. Many implementations may exist for the same algorithm as there are so many programming languages available such as JAVA, .NET, VB, ASP, PHP, PERL etc. and many more. In IT industry there would be basically two kinds of concern: User concerns and Designer concerns. Examples of both of these concerns are as follows.

Person Issues:
- May the machine provide the information needed?
- How quickly the data can be accessed?
- How easily the data can be retrieved?
- How much office assistance is going to be needed to input information into the system?

Custom Worries:
- How several it is willed by lines of program code require to do this function?
- Now may we reduce processor period?
- What would be probably the most effective means of saving this information?

7.3.3.4 Testing:
Testing is an investigation to assure about whether the algorithms implemented works properly or not and give proper results as required or not. Testing can also be termed as the method of assessing the quality of the code developed or system to be delivered.
Program screening may be said while the procedure of validating and confirming the program created earlier in the day in execution stage and supply following information about:

- Whether it fulfills the conditions that guided its development and design; and
- Whether it functions not surprisingly.

7.3.4 Supportive system Management Tasks:
Supportive system management task is:

- Quality Assurance activities at Requirements Analysis, Design and Implementation phases of the SDLC.

Quality assurance is an important task to carry out to ensure the quality of the work done. In the proposed research work while developing a system the quality assurance activities are being performed at Requirements Analysis, Design and Implementation phases of the SDLC. The checklist document consisting of the questionnaires regarding different important issues at all the three phases and also traceability analysis matrix to trace the design and implementation modules to the requirements are being prepared.

**7.4 REQUIREMENTS ANALYSIS**
Requirements architectural investigation holds these jobs that go in to determining the demands or conditions to match for a brand new program or applications, getting accounts of the potentially conflicting needs of the numerous stakeholders. The research work deals with the proposal of process or method rather than developing a product. The proposed architecture for the text-mining system and the algorithms, which have been developed, are part of the software development paradigm. The main objective of the research work has been explained earlier. Conditions could be classified as Practical, Behavioral, New and Non-functional. These demands are described as below.

7.4.1 Requirements:

i) Architectural Requirements
   - The system components should work in accordance with each other.
   - All the three agents should be placed in one architectural environment.

ii) Behavioral Requirements
   - All the three agents should coordinate their work properly.
• Agents should respond to the users’ queries properly.
• There should not be confusing operations on users’ end.
• All tasks to be performed with ease without giving much more stress on users’ side.

iii) Functional Requirements:

An agent should:

• Has to determine the ‘weighting factor’ for documents.
• Has to perform weighting of documents based on previously decided ‘weighting factor’.
• Has to rank the documents and put them in proper order.
• Present ranked documents to users.

To achieve these functional requirements, mainly two types of algorithms need to be designed / developed: I) Document-weighting algorithm II) Document-ranking algorithm

iv) Non-functional requirements

• Results from intelligent agent should be optimized.
• Performance of the agent based system should not be compromised in contrast to achieve functionality.
• Agents should be secure enough from mal-functioning.

v) Resource requirements

Resource requirements specify the hardware and software resources required to carry out the practical work related to dissertation. These are basically the hardware requirements and software requirements as shown as below.

Hardware Requirements

a) Processor : Pentium IV or higher
b) RAM : Minimum 256 MB
c) Hard Drive : 40 GB
d) Input system : Regular Keyboard
e) Output device : VGA and High Res Monitor

Software Requirements

a) Operating System : Windows XP or above
b) JDK: Java Development Kit
c) Web browser
d) Apache-Tomcat server 6

These are the underlying requirements for carrying out the research work and may be they can be stated as the primary requirements of the research work.

7.4.2 Feasibility Study

The system might not be feasible with some development components. Therefore it is necessary to evaluate feasibility of the system at the earliest stage. The feasibility of system is based on following properties:

a) Platform and Technology

This kind of feasibility comprises research of restrictions, functionality, and capabilities that could change the capability to attain a suitable program. This system is technically feasible for implementation. No extra hardware is required while developing this system. The web application can easily run on a single machine with Windows-XP or a higher version of Operating System.

b) Economical Feasibility

Cost-effective feasibility is the assessment of the improvement cost considered against the earnings or gains based on the program. Because the computer software doesn't demand any extra equipment or any extra supporting technologies which provides no extra price for the program this technique is monetarily possible. Thus the cost is only for the development and hence system is financially feasible. The system is developed basically for study purpose so economical feasibility is not a major issue.

7.4.3 Risk Management

Risk into software project can affect achieving desired goals or might cause significant degradation in the performance or might provide an unusual expected results. In order to avoid risks it is to be categorized and find a solution to avoid such risk. For research purpose the risk is classified in two categories.
a) System Management Risks
Possible causes for hazards to happen are financial, schedule, employees, source, client, and conditions issues, and it's likely that the prices will grow and the machine schedule will slide, when they become actual. An essential program danger is, program management. Because, software is intangible, it's quite hard to manage and track a software program. The progress report is assessed on regular basis to avoid errors.

b) Platform and Technology risks
Technologies and system risks problem possible layout, execution, interfacing, screening, and care difficulties. Standard method is followed so that unambiguous specification are performed.

7.4.3.1 Threat Identification:
Threat Id is a methodical effort to pin down risks for the machine strategy. By determining the foreseeable and known dangers, a first step is taken by the system manager toward preventing them when feasible and managing them when needed. One approach for determining dangers would be to produce a risk item record which may be utilized for risk recognition. The item checklist is prepared on following subcategories: product size, Technical approach and technology to be built.

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Risk Category/ Risk item</th>
<th>Risk Description</th>
<th>Risk Checklist Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Product Size Risks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Input size</td>
<td>Maximum input file size used</td>
<td>1.26 MB</td>
</tr>
<tr>
<td>2</td>
<td>Program Size</td>
<td>Estimated size of the product in Line of Code.</td>
<td>350 LOC</td>
</tr>
<tr>
<td>II</td>
<td>Technical risk</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Software requirement analysis</td>
<td>What is specific output by software requirement analysis?</td>
<td>Functional and non-Functional specification</td>
</tr>
<tr>
<td>2</td>
<td>Software planning</td>
<td>What specific method used for planning the project?</td>
<td>Project Schedule chart</td>
</tr>
<tr>
<td>3</td>
<td>Software Quality</td>
<td>How Software quality is achieved in Development</td>
<td>By performing QA tasks,</td>
</tr>
</tbody>
</table>
### Table 7.1: Risk Identification Checklist

<table>
<thead>
<tr>
<th></th>
<th>Technology Risk</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Technology</td>
<td>Are you new to development technology?</td>
</tr>
<tr>
<td>2</td>
<td>Interface</td>
<td>Is there Interface necessity between hardware and software?</td>
</tr>
<tr>
<td>3</td>
<td>Compatibility</td>
<td>Is system compatible with platform?</td>
</tr>
</tbody>
</table>

7.4.3.2 Risk Projection:

Risk projection, also called risk appraisal, tries to price each risk in two ways—the chance or probability that the risk is actual and the outcomes of the difficulties connected with all the risk, should it happen. As shown in table 7.2 risk projection is prepared for the system. First column describes risk summary that is analyzed in the system. Each hazard is classified within the 2nd line. The chance of incidence of each hazard is joined in the next line of the table. Next, the effect of every hazard is evaluated. According to standards risk impact defined in four categories as follows.

a. Catastrophic
b. Critical
c. Marginal
d. Negligible

The classes for each one of the four threat parts—functionality, help, price, and program—are averaged to decide a general effect worth.
<table>
<thead>
<tr>
<th>Risk Description</th>
<th>Risk Category</th>
<th>Risk Probability</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement not clear</td>
<td>System management risks</td>
<td>10%</td>
<td>2</td>
</tr>
<tr>
<td>As per schedule system completed in time</td>
<td>System management risks</td>
<td>10%</td>
<td>2</td>
</tr>
<tr>
<td>Sufficient budget not available</td>
<td>System management risks</td>
<td>5%</td>
<td>1</td>
</tr>
<tr>
<td>Expected Resource not available</td>
<td>Project management risks</td>
<td>15%</td>
<td>3</td>
</tr>
<tr>
<td>Training on used tool not sufficient</td>
<td>Platform and Technology risks</td>
<td>15%</td>
<td>3</td>
</tr>
<tr>
<td>Technology not compatible with system</td>
<td>Platform and Technology risks</td>
<td>5%</td>
<td>1</td>
</tr>
<tr>
<td>System is not scalable</td>
<td>Platform and Technology risks</td>
<td>5%</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 7.2: Risk Projection**

7.4.3.3 Risk Monitoring and Mitigation

The risks given in the above table can be monitored in the following ways to restrict their involvement into system failure.

i) As per schedule system completed in time

This risk can be eliminated by the formulation of a system plan, which can be checked by taking desk review time to time.

ii) Requirement not clear

This risk can be eliminated by preparing questionnaire depending on market survey.

iii) Technology not compatible with system

This risk can be prevented by careful analysis of the system requirements and feasibility before coding starts.

iv) Training on used tool not sufficient

This is a risk that is due to inexperience developers. To remove this risk, training is to be taken.

5. Expected Resource not available

This risk can be prevented by making the clear requirement of expected resources.
6. Planning and Scheduling

Table 7.3 shows activity chart which contain task, begin date, and End date following activity.

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Information Gathering</td>
<td>5/2/2012</td>
<td>7/3/2012</td>
</tr>
<tr>
<td>3</td>
<td>Concept Development</td>
<td>27/4/2012</td>
<td>5/5/2012</td>
</tr>
<tr>
<td>4</td>
<td>Planning and Scheduling</td>
<td>6/5/2012</td>
<td>9/5/2012</td>
</tr>
<tr>
<td>5</td>
<td>Requirements analysis</td>
<td>11/5/2012</td>
<td>23/5/2012</td>
</tr>
<tr>
<td>6</td>
<td>Risk identification and monitoring</td>
<td>25/5/2012</td>
<td>5/6/2012</td>
</tr>
<tr>
<td>7</td>
<td>Design and modelling</td>
<td>6/6/2012</td>
<td>24/6/2012</td>
</tr>
<tr>
<td>8</td>
<td>Design review and refinement</td>
<td>25/6/2012</td>
<td>2/7/2012</td>
</tr>
<tr>
<td>9</td>
<td>GUI design</td>
<td>3/7/2012</td>
<td>15/7/2012</td>
</tr>
<tr>
<td>10</td>
<td>Implementation</td>
<td>17/7/2012</td>
<td>29/7/2012</td>
</tr>
<tr>
<td>11</td>
<td>Review and suggestions for Implementation</td>
<td>30/7/2012</td>
<td>31/8/2012</td>
</tr>
<tr>
<td>12</td>
<td>Testing and Quality Assurance</td>
<td>1/9/2012</td>
<td>28/9/2012</td>
</tr>
<tr>
<td>13</td>
<td>Review and suggestions for Testing and QA</td>
<td>29/9/2012</td>
<td>12/10/2012</td>
</tr>
<tr>
<td>14</td>
<td>Refined QA activities</td>
<td>13/10/2012</td>
<td>12/11/2012</td>
</tr>
</tbody>
</table>

Table 7.3: Activity Chart
7.5 DESIGN AND MODELLING

The design and modelling has been the important part of the development of kind of system in any field. Make clear how the proposed system will look and feel and how well the design accommodates the requirements identified at the earlier stages of system development. For example, if an architecture planning to develop a commercial or residential building then he/she would first design the layouts and model the proposed architecture onto small scale modelling. In computer engineering also before commencing the actual development of the software system the team of designers would create the design layouts and required UML diagrams to understand how exactly the system will look how exactly they need to follow in implementation to achieve those designs correctly. The work includes developing UML diagrams for the proposed system and designing algorithms for document-weighting and document-ranking.

7.5.1 Proposed System (Document Search-Engine) Design – UML Diagrams

The Unified Modelling Language (UML) can be used to define, assemble, visualize and record the artefacts of an object-oriented software-program under development or under suggestion. For that planned program within this study function, UML diagrams including Use-Case Diagram, Sequence Diagram, Cooperation diagram, Component Diagram and Implementation Diagram are created.
7.5.1.1 Use-Case Diagram

Figure 7.3: Use-Case Diagram
Use case diagram depicts how the user interacts with the proposed system. As shown in Fig. 7.3 each and every use-case is explained as below:

Use-case 1: Type keyword
User has to type-in word which can be considered as the keyword. Keyword should have been inputted from the list of available keywords from all the documents in the dataset.

Use-case 2: Find Documents
Based on inputted keyword documents, containing inputted keyword, from the Reuters-21578 dataset will be found.

Use-case 3: Weight Documents
Those documents are then weighted by first counting the occurrences of inputted keyword in a document and then applying (tf*idf) equation thus by applying document-weighting algorithm.

Use-case 4: Rank Documents
Then the documents are ranked using a novel approach to document-ranking that is proposed in ranking algorithm and then the ranked documents are displayed to user.

7.5.1.2 Sequence Diagram:
Interaction is shown by a sequence diagram among like a two dimensional graph things. The graph is examine all the way through. As containers mounted on a vertical dashed line the things taking part in the conversation are proven towards the top of the graph. Inside the container, the title of the item is created using a colon splitting up it from the name of the course, and both the name of the course and object are underlined. The items showing in the best symbolize if the employment case delivery was started that the item currently existed. Nevertheless, if some item is produced throughout the delivery of the use situation and participates in the conversation (e.g. a procedure contact), then the object must be proven at the proper area on the plans where it's produced. The vertical dashed point is known as the item's lifeline. The lifeline signifies the lifestyle of the item at any special level of time. The rectangle drawn on the life is called the service image and signifies that the item is lively so long as the rectangle exists. As an arrow between the lifelines of two items each information is suggested. The emails are shown in date sequence from the best towards the underside. That's, studying the plans from the best to the base might display the series by which the communications happen. Each message is tagged.
using the message title. Some manage info may also be contained. As shown in Fig. 7.4, the user enters the word or sentence to find out the relevant documents. That request is forwarded to text-miner agent 1 which performs the tokenization and parsing and decides the important keyword. That keyword is forwarded to intelligent agent who in turn forwards it to text-miner agent 2 to decide the corresponding context for inputted keyword. Relevant contexts for inputted keyword are given back to intelligent agent. Then the intelligent agent finds out the documents from dataset containing keyword. Then it calculate the weight of each document by first counting the occurrences of keyword in that document and then applying the (tf*idf) equation. In (tf*idf) equation 'tf' stands for phrase regularity and 'idf' stands for inverse record regularity. Then the documents are ranked utilizing a new strategy to document-ranking which is suggested in position formula and the lead ranked documents are proven to person.
Generic Algorithm with Mathematical Evaluation for
A Generic Framework of Document Quality Assurance of Real Time Applications
Search Engine Based on Text Mining

Figure 7.4: Sequence Diagram (for proposed System)
7.5.1.3 Collaboration Diagram:
A collaboration plans is an instance of the relationships among software items in the Unified Modelling Language (UML). It's also called a conversation plans or interaction plans. Cooperation images are ideal for the representation of easy relationships among fairly little amounts of items. Items are demonstrated as rectangles with identifying labels inside. These labels are beat by colons and could be underlined. As lines linking the rectangles the associations between the items are demonstrated. As arrows linking the important rectangles along with the message sequencing that is defined by labels the communications between items are shown. A cooperation plans can be hard to see, because the amount of communications and items develops. Fig. 7.5 shows the collaboration diagram for the proposed system. The user enters the word or sentence to find out the relevant documents. That request is forwarded to text-miner agent 1 which performs the tokenization and parsing and decides the important keyword. That keyword is forwarded to intelligent agent who in turn forwards it to text-miner agent 2 to decide the corresponding context for inputted keyword. Relevant contexts for inputted keyword are given back to intelligent agent. Then the intelligent finds out the documents from dataset containing keyword. Then it calculate the weight of each document by first counting the occurrences of keyword in that document and then applying the (tf*idf) equation. In (tf*idf) equation 'tf' stands for phrase regularity and 'idf' stands for inverse record regularity. Then the documents are ranked utilizing a new strategy to document - ranking which is suggested in position formula and the lead ranked documents are proven to person.
Figure 7.5: Collaboration Diagram (for proposed System)
7.5.1.4 Component Diagram:

It reveals the advanced packed construction of the signal itself. Dependencies among elements are proven, including supply code elements, binary code elements, and executable elements. The component plan includes parts and dependencies. Parts signify the physical packaging of the component of signal. The dependencies between your elements display how modifications made to one element might change the additional components within the machine. Dependencies in a factor plans are symbolized with a dashed line between several elements. The interfaces can be also shown by component diagrams utilized by the parts to speak to one another.

![Figure 7.6: Component diagram showing software parts of the planned system](image)

Fig. 7.6 shows the required software components, namely Text-miner agent 1, agent 2 and agent 3 and dataset consisting of documents, for the proposed agent based system.
7.5.1.5 Deployment Diagram:
It exhibits the arrangement of run-time program parts and time processing components, procedures, and items that reside to them. Software component examples signify run-time demos of signal models. The implementation plan includes nodes and contacts. A node typically signifies an item of equipment within the machine.

![Deployment Diagram](image)

**Figure 7.7: Deployment Diagram showing hardware component for the proposed system**

Fig. 7.7 shows the deployment diagram for the proposed system. There are three different processors for each of the agent, namely Text-miner agent 1, intelligent agent and text-miner agent 2, and the required dataset.

7.5.2 Algorithms Design:
Algorithm design is an important for any system before starting the implementation of the system. Once we have an algorithm we can easily understand the core working of the system and if the algorithms have been designed with proper efforts then implementation or coding part becomes easy and hence the coding can be taken as the mechanistic development.
7.5.2.1 Document-weighting Algorithm:

Algorithm WeightDocuments (list d, int num, string “w”)
{
  // ‘d’ is the set of files where the needed keyword appears.
  // “w” is the string word (keyword) to be counted in the documents contained in ‘d’.
  // ‘num’ represents the no. of documents in which inputted word ‘w’ appears.
  int i, j, count, m, n, doc_weight := 0; // variable to store the weight of particular document
  int dw[]; // ‘dw’ is an array of weight of documents which will be used for ranking algorithm.
  count:=0;
  n := 21578; // n represents the total no. of documents scanned from the dataset
  string str;
  char a[];
  for i:=0 to num do // to go through all the documents in the collection ‘d’
  {
    while j!=EOF
    {
      m:=0
      label: for k:=m to k!=NULL do // to pick up a word from document
      {
        str := a[k]; // character array is stored in a string
      }
      if strcmp (w,str) then
        count++;
        m:=strlen(str) + 1; // addition of 1 because after end of string there is a
          // white space, so to get to the next word we need to
          // increment by 1.
        go to label;
        doc_weight = count * log(n/num) // (weight = tf*idf equation)
          // where tf signifies term regularity and
          // idf represents inverse document-frequency
      } // End of loop for variable j
      dw.add (doc_weight); // add the weight of the document to the array dw,
        // which will contain the weight of all documents.
    }
  }
} // End of Algorithm
Since evidence is offered that good weighting methods are far more significant than characteristic selection procedure for your good efficiency in information access, the significance of great weighting techniques in details access is extreme critical. You will find few approaches to compute ad-hoc weights of the records such as tf*idf technique, and importance feedback technique. These approaches are derived from two significant variables; one is statistical event info and yet another is a brief history of how nicely this characteristic has done previously. In several scenarios, it's difficult to get background info and therefore initial dumbbells tend to be centered only on record info. The file-weighting algorithm suggested within the study functions about the notion of tf*idf approach. The first tf*idf method functions the following: "The tf*idf method (term frequency times inverse document frequency) assigns weight to phrase (word) Tk in document Di, and in inverse percentage to the amount of files to that the term is designated."

The following the notion for evaluating files within the proposed formula can be, these actions display how an individual file is heavy based on proposed formula:

i) The regularity (no. of events in the report) of inputted 'key word' is computed in the report which is called the 'term regularity' (tf).

ii) Then the amount of files, which includes the inputted 'key word', is measured and that's referred to as the file-frequency (df).

iii) Then the total no. of available documents in the dataset is counted and that is termed as 'n'.

iv) Then the tf*idf equation is applied to calculate the weight of the document, for example, if the document to be weighted is d1, then

\[
\text{Weight (d1)} = \text{tf(d1)} \times \text{idf} \\
= \text{tf(d1)} \times \log(n/df)
\]

Where, tf(d1) - term frequency in document d1

n – Total number of documents

df – no. of documents in which the term appears.

idf – inverse document frequency

d1 – documents to be weighted
7.5.2.2 Document-ranking Algorithm:

Algorithm RankDocuments (list l, int dw[], int num) {
    // ‘l’ is the listing of files in which inputted word appears.
    // ‘dw’ is the array of ‘weight values’ of documents in list ‘l’.
    // ‘num’ represents the no. of documents in which inputted word appears.
    int avg; // ‘avg’ is the ‘average weight value’ of all resulted documents.
    int v := 0, k; // variables used for document counting
    avg := FindAvg(l, dw[], num);
    int j;
    for j:=1 to num do {
        if (dw[j] <= avg) then{
            M.add(l(j)); // document is added into another list in which
            // weight of all documents are less than average weight.
            weight1.add(dw[j]); // ‘weight’ of document is added into
            // another array.
            k++; // counting the no. of documents in list ‘M’.
        }
        else{
            N.add(l(j)); // document is added into another list in which
            // weight of all documents are greater than
            // average weight.
            weight2.add(dw[j]); // ‘weight’ of document is added into
            // another array.
            v++; // counting the no. of documents in list ‘N’.
        }
    } // End of loop for variable j
    if v!=0 and v!=1 then
    {
        if(weight of all documents in list N are same) then
        {
            Final.add(all in N); // this is to be done because, if weights of all the
            // documents in the list N are same then there will not be
            // any requirement to calculate ‘avg weight’, coz
            // ‘avg. weight’ will be always same.
            RankDocuments(M, weight1, v); //Recursive call to function itself with
        }
    } // End of if
}
Generic Algorithm with Mathematical Evaluation for
A Generic Framework of Document
Quality Assurance of Real Time Applications
Search Engine Based on Text Mining

// parameters related to list M.
}
else
{ RankDocuments(N,weight2, v); // Recursive call to function itself.
}
}// End of If block.
else
{
    Final.add( l(j).title, l(j).data ); // add highest ranked document to ‘final’
    // list.
    if k!=0 and k!=1 then
    { if(weight of all documents in list M are same) then
        {
            Final.add(all in M); // this is to be done because, if weights of
            // the documents in the list M are same then
            // there will not be any requirement to calculate
            // ‘avg weight’, coz ‘avg. weight’ will be always
            // same.
            }
        else
        {
            RankDocuments(M,weight1, k); // Recursive call to function
            // itself.
        }
    }
    else
    { Final.add( l(j).title, l(j).data ); // add highest ranked document to ‘final’
        // list.
    }
}// End of else block.

display(final); //display all ranked documents with title and data.
} // End of Algorithm

Function FindAvg (l, dw[], num)
{

int num; // no. of documents in list ‘l’
int i, total weight:=0;
for i:=1 to num do{
    total weight: = total weight + dw[i];
}
avg:= total weight/num;
return avg;
}

Ranking algorithms are mainly used to rank or index the documents based on some deciding factor. A position is a connection between some products so that, for almost any two products, the very first is possibly ‘ranked greater than’, ‘ranked less than’ or ‘ranked corresponding to’ the 2nd. The positions themselves are completely purchased and have the ability to assess complicated details based on particular requirements. There are few existing document ranking algorithms such as Google’s PageRank, RankNet, AdaRank, Coordinate ascent etc. These algorithms work in the internet environment and are highly optimized for getting best possible results. Google’s PageRank is very famous algorithms and the search engine developed by Google Inc., named ‘Google’ itself, works on the basis of this PageRank algorithm. The proposed Document-ranking algorithm works on the basis of ‘quick sort’ method. The idea behind quick sort is as following:

“Given the list of numbers to be sorted in ascending/descending order, first select a pivot element, which is generally the mid-positioned element of the given list of the numbers, and then place all the numbers, which are smaller than pivot element, on the left of the pivot element and place all the elements, which are greater than the pivot element, on the right of the pivot element. Then take the list of numbers which are on the right side of the pivot element, which are greater than the pivot element, and repeat the procedure, at the end One will have the given numbers sorted either in ascending or descending order (as he/she wants).”

The idea for ranking documents in the proposed algorithm is as follows, these steps show how a single document is ranked according to proposed algorithm:

i) Find the average weight value by adding the weights of all documents in ‘final list’ and dividing that by ‘total documents in which word appears’.

Thus the pivot value is been decided that is the ‘avg. weight’
ii) For all the documents in ‘final list’, compare the weight of document with the ‘avg. weight’.

iii) The documents which are having less weight than the ‘avg. weight’ will be added to another list ‘M’ and their weights are added to another array ‘weight1’ and the documents which are having more weight than ‘avg. weight’ will be added to another list ‘N’ and their respective weights are added to another array ‘weight2’.

iv) If there are more than one document in the list ‘N’ then recursively calls the Algorithm with parameters related to documents in list ‘N’. If all the documents in the list ‘N’ are having same ‘weight value’ then there is no need to repeat the process because every time the average weight will be same and thus it will go into the infinite loop. And if there is only one document in the list ‘N’ then add that document in the ‘final’ list.

v) If there are more than one document in the list ‘M’ then recursively calls the Algorithm with parameters related to documents in list ‘M’. If all the documents in the list ‘M’ are having same ‘weight value’ then there is no need to repeat the process because every time the average weight will be same and thus it will go into the infinite loop. And if there is only one document in the list ‘M’ then add that document in the ‘final’ list.

vi) Display the documents in the list ‘final’ to the user.

Thus ‘final’ will have all the ranked documents with their title and data which will be displayed to user who had given the search query using particular keyword.
7.6 IMPLEMENTATION

Execution of any suggested program is an essential move to make exhibits how just the system has been applied depending on these thoughts and since it implies the sense and general appear of the proposed system. In this section the system being developed based on document-weighting and document-ranking algorithm is explained with some snapshots of the system and how the system works.

7.6.1 Snapshots of the System

This portion includes the different snapshots of the browser-oriented document search-engine system. The code has been developed using JAVA language and the concepts of JSP/Servlet. The apache server is used as the web server for the web-services and the ‘Google Chrome’ browser is used to execute the search-engine system. Fig. 7.8 shows the ‘Homepage’ of the search-engine system. It consists of a textbox for accepting input word from the user and a ‘Search’ button.

![Figure 7.8: Homepage of the System](image)

Fig. 7.9 shows the output of resulted ranked documents for the inputted word ‘Madrid’. As it can be seen from the snapshot that the total documents found for inputted word ‘Madrid’ are 13 and the response time is 2769 milliseconds. As also, the ranked documents are displayed in the table-format consisting columns named, Sr. No., Weight, Title and Content.
Fig. 7.10 shows the output of resulted ranked documents for the inputted word ‘Protest’. As it can be seen from the snapshot that the total documents found for inputted word ‘Protest’ are 67 and the response time is 2699 mili-seconds.
Figure 7.10: Output Screen (Showing results for keyword ‘Protest’)

Fig. 7.11 shows the output of resulted ranked documents for the inputted word ‘India’. As it can be seen from the snapshot that the total documents found for inputted word ‘India’ are 70 and the response time is 2668 mili-seconds. Note the response time for the results shown.
Fig. 7.12 shows the output of resulted ranked documents for the inputted word ‘Ford’. As it can be seen from the snapshot that the total documents found for inputted word ‘Ford’ are 105 and the response time is 2652 mili-seconds. Note the response time for the results shown.
Figure 7.12: Output Screen (Showing results for keyword ‘Ford’)

After looking at all the above snapshots, particularly at resulted documents table, one might be thinking that how the documents with the same weight values have been ranked. But, it is simply listed those documents as how they have been appeared during the scanning of the Reuters dataset and that concept is included in the Ranking algorithm.

Fig 7.13 shows the output for the inputted word ‘Gallardo’. As the inputted word does not belong to the documents, which are specifically News articles contained in the dataset, the result screen just shows the total documents found details with ‘zero’ and response time with 2746 mili-seconds.
Figure 7.13: Output Screen (Showing results for keyword ‘Gallardo’)

7.7 CODING DESCRIPTION
Programming or coding is a kind of mechanistic process if the algorithms have been
designed properly and correctly. Many implementations may exist for the same
algorithm as there are so many programming languages available such as JAVA,
.NET, VB, ASP, PHP, PERL etc. and many more. For the system developed in the
research work the programming is done in JAVA language.

7.8 TESTING AND QUALITY ASSURANCE
In this area functionality based quality and screening assurance actions at Needs
Evaluation, Style and Execution period of the program development lifecycle are
described.

7.8.1 Testing
Software testing can be an essential element of the software development life cycle as
it provides the way to test for the functionality of the software developed as well as
provide an insight into quality of the software developed.

Software testing may be said while the procedure of validating and confirming that
the software program/application/product:
a. Meets the conditions that guided its design and development;
b. Works as anticipated; and
c. Can be applied with the same features.

7.8.1.1 Objectives of Software Testing
a. The aim of screening is to discover flaws before clients find them away.
b. The aim of testing is to display that the system functions correctly.
c. Most of the complicated computer software methods may include problems,
which induce the machine to fail from time to time. As problems are found
and repaired while executing more and more assessments, the disappointment
fee of the program normally reduces.
d. The extremely anticipated aim of performing assessments would be to create
powerful test cases.

7.8.1.2 Confirmation and Validation
Two similar ideas associated with computer software testing often utilized by
professionals are approval and confirmation. The two ideas are described as follows:

1. Confirmation: This sort of action assists us in assessing a computer software
program by identifying whether the item of confirmed growth periods fulfills
the needs established before the beginning of that period. It's possible to notice that a product is an advanced product, including requisite standards, layout standards, code, user guide, and even the ultimate item. Activities that assess the correctness of a growth period are called verification actions.

ii) Validation: Activities of the type assist us in affirming that an item fulfills its meant use. Approval actions goal at verifying that the product fulfills its customer's expectations. In other words, validation activities concentrate on the ultimate item, that is commonly analyzed in the client perspective. Validation determines if the merchandise meets general expectations of the customers. The affirmation procedure determines the communication of an execution period of the program development procedure with its standards, whereas validation establishes the correspondence between something and users' expectations.

7.8.1.3 Software Testing Types

A. Unit Testing

Unit testing describes testing software models in seclusion. A few examples of generally recognized models are functions, processes, or approaches. Being a software component actually a course within an object-oriented programming terminology can be viewed. The perform done by a course device might not need an immediate connection with a system-level perform. Therefore, a plan device might be looked at like a bit of signal applying a "low" degree perform. You will find two good reasons for screening a device in a stand-alone fashion. First, problems discovered throughout screening can be credited to some particular device such that it can be readily repaired. Furthermore, unit testing removes dependencies on additional software models. Second, during device testing it's desired to check that each different delivery of the software unit creates the anticipated consequence. In phrases of signal particulars, a distinct execution describes a distinct route within the system. Preferably, all potential or around feasible different accomplishments should be looked at during device testing. This demands cautious choice of enter information for every delivery. A developer has immediate use of the input signal vector of the system by running a plan unit in seclusion. This immediate entry makes it simpler to run as several different pathways as appealing or feasible.
Unit screening is conducted by the program unit is written by the programmer who because the programmer is closely acquainted with the inner information on the device. The aim for the developer is to become happy that the system functions as expected. Following example demonstrates the unit testing performed for the functionality of document counting containing inputted word.

Example:
For finding out the documents containing inputted keyword, the logic is to scan whole document/news article by taking up one word at a time and compare it with the inputted word. If the word matches then add that document into the list of found documents otherwise not.

```pascal
for i:=1 to n do {
  if D(i).body!=NULL && D(i).length>0 && w (- D(i).body) then
    { count++;
    L.add(D(i)); // document is added in the resulted list.
    }
}
```

7.8.1.4 Test Cases:
“A test situation has parts that explain an input, an anticipated result and activity or event, to ascertain if a characteristic of an application is functioning right.” Preparing test cases is the most relevant activity while performing functionality based software testing. Test cases are the properly formatted document containing functionality to be tested, input to the test case, output from the test case, Feature pass/fail etc. Test cases are properly formatted documents and the format is as follows:

- Check situation identifier: A special name or number.
- Goal: What attributes, route and so on are being analyzed?
- Expected Result: Expected result from the successful execution of the functionality.
- Procedure: Test procedure describes the steps for performing test.
- Actual Result: Actual result from the successful execution of the functionality.
Feature pass/Fail: A requirements that explains under what conditions a characteristic can be viewed as "passed" (Test has failed) or "failed" (Test has succeeded).

To test the functionality of the developed system test cases are prepared. It also comprises 'Test Cases' for your various benefits and the facts about quality guarantee activities completed in the needs evaluation, layout, and execution periods of the program development life cycle and how just these activities have already been done.

<table>
<thead>
<tr>
<th>Test Case ID:</th>
<th>TC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose:</td>
<td>‘SearchDoc’ functionality to be tested.</td>
</tr>
<tr>
<td>Input:</td>
<td>Keyword – to search documents</td>
</tr>
<tr>
<td>Expected Result:</td>
<td>All the documents containing inputted keyword</td>
</tr>
</tbody>
</table>
| Procedure(Steps): | • Run the application  
• Input the keyword in the textbox and click ‘Search’.  
• System should display all the documents containing inputted keyword. |
| Actual Result: | All the documents containing inputted word has been listed. |
| Feature Pass/Fail: | Pass |

Table 7.4: Test Case for ‘SearchDoc’ functionality

<table>
<thead>
<tr>
<th>Test Case ID:</th>
<th>TC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose:</td>
<td>‘WeightDoc’ functionality to be tested</td>
</tr>
<tr>
<td>Input:</td>
<td>Keyword – to search documents and then weight them</td>
</tr>
<tr>
<td>Expected Result:</td>
<td>List all the documents containing inputted keyword with their respective weights.</td>
</tr>
</tbody>
</table>
### Table 7.5: Test Case for ‘WeightDoc’ functionality

<table>
<thead>
<tr>
<th>Test Case ID</th>
<th>TC3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purpose:</td>
<td>‘RankDoc’ functionality to be tested – that is ranking of documents based on ranking algorithm proposed.</td>
</tr>
<tr>
<td>Input:</td>
<td>Keyword – to search documents, weight them and rank them.</td>
</tr>
<tr>
<td>Expected Result:</td>
<td>List all the documents containing inputted keyword with their respective weights and their respective ranks according to ranking algorithm.</td>
</tr>
</tbody>
</table>
| Procedure(Steps): | • Run the application  
  • Input the keyword in the textbox and click ‘Search’.  
  • System should calculate rank, using ranking algorithm proposed, of all the documents containing inputted word.  
  • System should display all the documents containing inputted keyword with their respective weights and their respective ranks according to ranking algorithm. |
Generic Algorithm with Mathematical Evaluation for A Generic Framework of Document Quality Assurance of Real Time Applications Search Engine Based on Text Mining

| Actual Result: | All the documents containing inputted word has been listed with their respective weights and their respective ranks according to ranking algorithm. |
| Feature Pass/Fail: | Pass |

**Table 7.6: Test Case for ‘RankDoc’ functionality**

| Test Case ID: | TC4 |
| Purpose: | Random Input to the system to be checked. |
| Input: | Random keyword, number, un-meaningful text. |
| Expected Result: | The system should give an error message or warning message for randomly inputted un-meaningful text or number. |
| Procedure(Steps): | • Run the application  
• Input the un-meaningful keyword or any random number in the textbox and click ‘Search’.  
• System should display error or warning message. |
| Actual Result: | All the documents containing inputted keyword or number has been listed with their respective weights and their respective ranks according to ranking algorithm. |
| Feature Pass/Fail: | Fail |

**Table 7.7: Test Case for Random Input Check**

7.8.2 Quality Assurance

Quality has been the important factor in each and every product or services that human-being uses. Quality plays an important role in convincing people to buy particular products or to use particular system or services. E.g. if one wants to buy a new mobile phone then apart from the features quality of the material used in manufacturing that mobile phone, quality of the accessories provided with the mobile phone, quality of the system software(s) installed into mobile etc. also plays an
important role. In this system, the quality assurance activities at three phases of SDLC are performed:

a) Quality Assurance at Requirements Analysis Phase
b) Quality Assurance at Design Phase
c) Quality Assurance at Implementation Phase

7.8.2.1 Quality Guarantee at Needs Evaluation Period

Quality assurance is chiefly performed onto two sorts of mentioned specifications:

i) Functional requirements specification
ii) Non-Functional requirements specification

i) Functional requirements specification

Functional requirements specification is an important document in the SDLC because it provides the insight to the overall functionality related issues of the system. So, performing quality assurance related tasks at this stage provide more controlled, managed and refined way to achieving good software quality. It can simply review each and every requirement and its different aspects onto software development and then prepare quality assurance checklist document, containing questionnaire regarding functional requirements at different levels.

a) Pre-requisites

Pre-requisites are the important resources and tools required before the actual development starts. Pre-requisites for this system work are:

- Reuters-21578 News articles dataset.
- JDK 1.5 (or above) is required at the development stage to run the JAVA programs.
- Apache web server is required to run the developed JSP/Servlet APIs.
- Any web-browser is required to run the developed JSP/Servlet APIs.

b) Input level requirements

These are the requirements regarding inputs to the system developed.

- Reuter 21578 SGML/XML News Articles Dataset.
- Keywords/words from the Dataset files.

c) Output level requirements
These are the requirements regarding outputs from the system developed.

- Documents (News articles) containing the ‘searched keyword’.
- Weights of the documents found, containing ‘searched keyword’.
- Ranked documents according to weights
- All these should be in a table-format display.

d) Procedural (Workflow) requirements

These are the procedural or workflow steps for the system developed.

- The system should take the input (in form of keyword)
- The system should find out the documents (news articles) in which inputted word/keyword appears.
- The system must use the Reuters-21578 news articles dataset as the resource to find out the documents.
- The system should perform the task of ‘weighting documents’ for the resulted documents.
- The system should perform the task of ‘ranking documents’ for the resulted documents.
- The ranking is performed based on weights of the documents.
- Ranked documents are presented to the user in form of table-form display.

ii) Non-Functional requirements specification

Non-functional requirements specification is a handful document, in accordance with functional requirements specification, as it addresses the issues regarding non-functional but vital requirements. Here the research work mainly deals with the constraints at different levels of the system development. It can simply review each and every non-functional requirement and its different aspects onto software development and then prepare quality assurance checklist document, containing questionnaire regarding functional requirements at different levels.

a) Dataset Constraints

- Currently the system works on the Reuater-21578 news articles dataset, but in future, It can take some other datasets or take this system into Internet environment.
• Currently the dataset used in the system as a resource is in SGML/XML form. In future, other formats may also be supported.

b) Time Constraints
• The whole work needs to be accomplished within the specified time period.
• Proper plan had been prepared to accomplish the tasks in time.

This document is used as baseline for inspection. Quality assurance activities are performed while working on above phase by reviewing each requirement with strictly checkmarks. System functionalities are checked and added addressed suggestions to minimize defect or clarification against the Functional Specification document. Checklists document for the web-application is prepared which provides a complete list of items to be verified and also provide space for documenting findings of the checks performed.

<table>
<thead>
<tr>
<th>Sr.No.</th>
<th>Topic/Requirement</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>The Document</td>
<td></td>
<td></td>
</tr>
<tr>
<td>i)</td>
<td>Is the document prepared according standard template?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>ii)</td>
<td>Are there any ambiguities in the prepared document?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>iii)</td>
<td>Is prepared document complete?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>iv)</td>
<td>Does the record tackle the problems regarding practical and non-functional needs of the system?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>Functional Specification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>i)</td>
<td>Are all the required functionalities are properly defined?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>ii)</td>
<td>Are the functional requirements defined are clear and unambiguous?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>iii)</td>
<td>Are the inputs to the system conforming to functional requirements?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>iv)</td>
<td>Are GUI interfaces are fully defined and addressed properly?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>v)</td>
<td>Is the flow of system conforming to the stated functional requirements?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>vi)</td>
<td>Are desired time complexity and space complexity</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>
calculations have been performed for the algorithms developed?

vii) Is the ‘weighting document’ task is properly performed? *

viii) Is the ‘ranking document’ task is properly performed? *

ix) Is the developed system supports any other dataset currently? *

III Non-Functional Specification

a) Does the ‘Time line chart’ or ‘Time-schedule’ plan have been prepared for the system development? *

b) Does the system development task is feasible to complete within defined time limit? *

c) Does the system development conform to dataset constraints? *

d) Does the system require assistance from any other resources rather than those define earlier? *

<table>
<thead>
<tr>
<th>Table 7.8: Checklist Document at Requirements Analysis Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>From table 7.8, containing checklist document at requirement analysis phase, following things can be stated:</td>
</tr>
<tr>
<td>i) Various issues are addressed by the checklist document associated with non and practical - functional needs of the machine making use of survey type template.</td>
</tr>
<tr>
<td>ii) As well as it also addresses the issues regarding achieving good quality at requirement analysis level using the form of questionnaire prepared to conform to quality measures.</td>
</tr>
<tr>
<td>iii) There are few ‘NOs’ and mostly ‘YESs’, so the checklist document provides a positive outcome to the good quality achievement plan.</td>
</tr>
</tbody>
</table>

7.8.2.2 Quality Assurance at Design Stage

The design stage of the SDLC is a really essential phase because it particulars the performance focused design and offers the critical understanding onto how really the issues works in the execution stage. The design stage provides the designs of the user
interfaces of the program as well as offers the insights onto practical parts of the system at style degree. The design engineers normally need to run the check on traceability of design with requirements specified. The primary quality assurance activity during the design phase could be the formal review of the preliminary and detailed design documents. These documents are verified for their consistency, completeness, and correctness within themselves and with the requirements specification document.

Two activities have been defined for the quality assurance at design phase of the research work:

i) Design review by checklist document

ii) Traceability analysis matrix.

1) Design review by checklist document:

The checklist document contains the type of questionnaire regarding different issues focusing on different components of the design phase. The checklist document for design phase, is shown as below:

<table>
<thead>
<tr>
<th>Sr.No.</th>
<th>Topic/Requirement</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Requirement Traceability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a)</td>
<td>Does the design adhere to the requirements specified?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>b)</td>
<td>Does the design have any ambiguities respective to</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>design requirements?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c)</td>
<td>Does the design artifacts traceable to requirements?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>Functionality</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a)</td>
<td>Do the design artifacts adhere to functionality issues?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>b)</td>
<td>Are all the design artifacts clear and concise</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>according to functionality demanded?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c)</td>
<td>Do design artifacts provide a new kind of architecture</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td></td>
<td>for the system?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d)</td>
<td>Does the design support any other operating</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>environment than suggested?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>e)</td>
<td>Are all the user interfaces, data flows and control</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>flows are properly addressed?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>Design Centered</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
a) Do the design artifacts provide a new design arena that leads the system to easy and concise user interface? *  

b) Does the design provide clear and concise understanding of how the system could be workable? *  

c) Does the design support any other operating architecture than suggested? *  

d) Does the design follow standard techniques to describe system? *  

IV Design Feasibility and Reliability  

a) Does the design adhere to time, cost and effort feasibility constraints of the system? *  

b) Do the time and space complexity of the designed algorithms related issues have been addressed? *  

c) Does the impact of undesired events, such as system restart or system failure, have been considered? *  

<table>
<thead>
<tr>
<th>Table 7.9: Checklist Document at Design Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>The checklist document prepared above shows that all the required issues related to design are considered and addressed properly resulting into the good quality at design phase.</td>
</tr>
</tbody>
</table>
2) Traceability Analysis Matrix:

Traceability analysis shows the trace of design to the documented functional requirements. Traceability matrix at this level is prepared by taking requirements of the project as columns and design modules as rows.

<table>
<thead>
<tr>
<th>Textbox(for input keyword)</th>
<th>Keyword Acceptance</th>
<th>Find Documents</th>
<th>Weight Documents</th>
<th>Rank Documents</th>
<th>Display Documents</th>
<th>Agents Coordination</th>
<th>V&amp;V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design of FindDoc()</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Count frequency of keywords in documents</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Customized (tf*idf) method</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Design of Ranking algorithm</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Table-formed display</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Design for Agents-Coordination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
</tr>
</tbody>
</table>

Table 7.10: Traceability Analysis matrix at Design level

7.8.2.3 Quality Guarantee at Execution Period

Execution phase of SDLC is the phase where the particular code has been done or to be particular the actual applications is produced through coding based on conditions gathered and styles produced during earlier SDLC stages. Before the actual implementation or coding starts all the requirements are collected, reviewed and documented properly and same as all the design artifacts have been created according to stated requirements and documented properly. Therefore, execution can be viewed because the more on like physical job instead of the job. Various kinds of Active Layout Evaluations - Reviews, Code - Inspections, Code and Walkthroughs have been discovered to be quite helpful in enhancing the quality of applications. Signal assessments should look for technical correctness and completeness of the signal, and
to ensure good code methods and requirements, and confirm that it tools the prepared style are utilized. Two activities have been defined for the quality assurance at implementation phase of this work:

   i) Implementation review by checklist document
   ii) Traceability analysis matrix.

i) Implementation review by checklist document

The checklist document contains the type of questionnaire regarding different issues focusing on different components of the implementation phase. The checklist document for implementation phase, is shown as below:
<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Topic/Requirement</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Interfaces ,Code and data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a)</td>
<td>Does the implementation adhere to design artifacts?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>b)</td>
<td>Does the code (programs) are traceable to design artifacts suggested?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>c)</td>
<td>Does the coding have been performed according to standards, such as placing comments so that anyone can understand the code properly?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>d)</td>
<td>Are there any ambiguities in code regarding the design artifacts?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>e)</td>
<td>Does the code include all the required libraries or packages?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>f)</td>
<td>Do the function calls and returns work properly in the code?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>g)</td>
<td>Are demands to spend memory examined for achievement before trying to create to that memory?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>h)</td>
<td>Exist any unknown leaps or coils contained in the code?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>i)</td>
<td>Are the required variables declared properly?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>j)</td>
<td>Are the required variables initialized at the beginning of the main () function or other functions?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>k)</td>
<td>Does the system perform required functions properly?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>l)</td>
<td>Does the system provide required output in the required format?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>II</td>
<td>Feasibility and Reliability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a)</td>
<td>Does the design adhere to time, cost and effort feasibility constraints of the system?</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>b)</td>
<td>Does the backup have been maintained for developed code?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>c)</td>
<td>Does the output provided by the system is reliable?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>d)</td>
<td>Is the code developed is portable onto other computer systems?</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>e)</td>
<td>Does the impact of undesired events, such as code malfunctioning or system failure, have been considered?</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

**Table 7.11: Checklist Document at Implementation Phase**
The checklist document prepared shows that all the required issues related to implementation are considered and addressed properly resulting into the good quality at implementation phase.

ii) Traceability Analysis Matrix
Traceability analysis shows the trace of code to the design modules. Traceability matrix at this level is prepared by taking design modules as columns and code modules of the system as rows.

<table>
<thead>
<tr>
<th>Design of FindDoc()</th>
<th>Design of Ranking algorithm</th>
<th>Table-formed display</th>
<th>Design for Agents-Coordination</th>
<th>V&amp;V</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI code containing ‘Textbox’ and ‘Search’ button</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Yes</td>
</tr>
<tr>
<td>Code for FindDoc() function</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Yes</td>
</tr>
<tr>
<td>Code for WightDocs algorithm</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Yes</td>
</tr>
<tr>
<td>Code for RankDocs algorithm</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>Yes</td>
</tr>
<tr>
<td>Code for Agents-Coordination</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 7.12: Traceability Analysis matrix at Implementation level
7.9 RESULTS DISCUSSION AND ANALYSIS

The algorithms are implemented into JAVA programming code and tested on the Reuters-21578 news articles dataset. Java archive (JAR) file is created and executed using command prompt and the results are obtained for different inputted words. Resulted weighted documents and ranked documents for different inputted words are presented here, both in the form of table and charts.

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Weight</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.49443021503156</td>
<td>TREASURY, FOREIGN RESERVES ARE JORDAN'S PRIORITIES</td>
</tr>
<tr>
<td>2</td>
<td>7.49443021503156</td>
<td>GULF ARAB MINISTERS DISCUSS ECONOMIC COOPERATION</td>
</tr>
<tr>
<td>3</td>
<td>14.9888604300631</td>
<td>MIDDLE EAST CURRENCY MARKET SEES KEY CHANGES</td>
</tr>
<tr>
<td>4</td>
<td>7.49443021503156</td>
<td>ARAB BANKER SAYS TOO SOON FOR SINGLE CURRENCY</td>
</tr>
<tr>
<td>5</td>
<td>14.9888604300631</td>
<td>BAHRAIN INTRODUCES NEW MONEY MARKET REGIME</td>
</tr>
<tr>
<td>6</td>
<td>7.49443021503156</td>
<td>BAHRAIN TREASURY BILLS YIELD AVERAGE 6.00 PCT</td>
</tr>
<tr>
<td>7</td>
<td>7.49443021503156</td>
<td>U.S.-SDR RATE MOVES TO 1.28435</td>
</tr>
<tr>
<td>8</td>
<td>7.49443021503156</td>
<td>ECONOMIC SPOTLIGHT - KUWAITI ECONOMY</td>
</tr>
<tr>
<td>9</td>
<td>14.9888604300631</td>
<td>YUGOSLAV ECONOMY WORSENED IN 1986, BANK DATA SHOWS</td>
</tr>
<tr>
<td>10</td>
<td>22.4832906450946</td>
<td>CREATIVE MONETARY POLICY TO SPUR KUWAITI ECONOMY</td>
</tr>
<tr>
<td>11</td>
<td>14.9888604300631</td>
<td>WINDOW FOR BANK AID IN KUWAIT REMAINS SHUT</td>
</tr>
<tr>
<td>12</td>
<td>22.4832906450946</td>
<td>KUWAITI DINAR RATES FIRM, AID WINDOW OPEN</td>
</tr>
</tbody>
</table>

Table 7.13: Weighted Documents for inputted word ‘Dinar’
Table 7.13 shows the documents with their respective weights for inputted word ‘Dinar’. Fig. 7.14 shows the chart for the same. In chart, vertical dimension shows the weights and horizontal dimension shows the titles of the documents. When the documents are weighted, they are displayed in a way they have been appeared during the weight-calculation process and that can be seen from the above chart.

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Weight</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22.4832906450946</td>
<td>CREATIVE MONETARY POLICY TO SPUR KUWAITI ECONOMY</td>
</tr>
<tr>
<td>2</td>
<td>22.4832906450946</td>
<td>KUWAITI DINAR RATES FIRM, AID WINDOW OPEN</td>
</tr>
<tr>
<td>3</td>
<td>14.9888604300631</td>
<td>MIDDLE EAST CURRENCY MARKET SEES KEY CHANGES</td>
</tr>
<tr>
<td>4</td>
<td>14.9888604300631</td>
<td>BAHRAIN INTRODUCES NEW MONEY MARKET REGIME</td>
</tr>
<tr>
<td>5</td>
<td>14.9888604300631</td>
<td>YUGOSLAV ECONOMY WORSENED IN 1986, BANK DATA SHOWS</td>
</tr>
<tr>
<td>6</td>
<td>14.9888604300631</td>
<td>WINDOW FOR BANK AID IN KUWAIT</td>
</tr>
</tbody>
</table>
Table 7.14: Ranked Documents for inputted word ‘Dinar’

<table>
<thead>
<tr>
<th>Rank</th>
<th>Score</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>7.4943021503156</td>
<td>REMAINS SHUT</td>
</tr>
<tr>
<td>8</td>
<td>7.4943021503156</td>
<td>TREASURY, FOREIGN RESERVES ARE JORDAN’S PRIORITIES</td>
</tr>
<tr>
<td>9</td>
<td>7.4943021503156</td>
<td>GULF ARAB MINISTERS DISCUSS ECONOMIC COOPERATION</td>
</tr>
<tr>
<td>10</td>
<td>7.4943021503156</td>
<td>ARAB BANKER SAYS TOO SOON FOR SINGLE CURRENCY</td>
</tr>
<tr>
<td>11</td>
<td>7.4943021503156</td>
<td>BAHRAIN TREASURY BILLS YIELD AVERAGE 6.00 PCT</td>
</tr>
<tr>
<td>12</td>
<td>7.4943021503156</td>
<td>U.S.-SDR RATE MOVES TO 1.28435</td>
</tr>
<tr>
<td></td>
<td>7.4943021503156</td>
<td>ECONOMIC SPOTLIGHT - KUWAITI ECONOMY</td>
</tr>
</tbody>
</table>

Figure 7.15: Chart for Ranked Documents (inputted word ‘Dinar’)

- 264 -
Table 7.14 shows the ranked documents with their respective weights for inputted word ‘Dinar’. Fig. 7.15 shows the chart for the same. In chart, vertical dimension shows the weights and horizontal dimension shows the titles of the documents. Documents are displayed in a way they have been ranked by ranking algorithm and that can be seen from the above chart.

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Weight</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34.91431376</td>
<td>APPLE COMPUTER &lt;AAPL&gt; UPGRADES MACINTOSH LINE</td>
</tr>
<tr>
<td>2</td>
<td>20.94858825</td>
<td>APPLE COMPUTER &lt;AAPL&gt; HAS NEW MACINTOSH MODELS</td>
</tr>
<tr>
<td>3</td>
<td>6.982862751</td>
<td>LOTUS &lt;LOTS&gt; INTRODUCES NEW SOFTWARE</td>
</tr>
<tr>
<td>4</td>
<td>20.94858825</td>
<td>APPLE &lt;AAPL&gt;, AST &lt;ASTA&gt; OFFER MS-DOS PRODUCTS</td>
</tr>
<tr>
<td>5</td>
<td>6.982862751</td>
<td>SOFTWARE COS SUPPORT NEW APPLE &lt;AAPL&gt; PRODUCTS</td>
</tr>
<tr>
<td>6</td>
<td>6.982862751</td>
<td>COMPUTER COMPANIES FORM NETWORKING GROUP</td>
</tr>
<tr>
<td>7</td>
<td>13.9657255</td>
<td>FCOJ SUPPLIES SIGNIFICANTLY ABOVE YEAR AGO-USDA</td>
</tr>
<tr>
<td>8</td>
<td>13.9657255</td>
<td>BERTELSMANN TO MARKET APPLE SOFTWARE IN GERMANY</td>
</tr>
<tr>
<td>9</td>
<td>6.982862751</td>
<td>MOTOROLA (MOT) SEES CONTINUED GROWTH FOR CHIPS</td>
</tr>
<tr>
<td>10</td>
<td>6.982862751</td>
<td>TECHNOLOGY/DESKTOP PUBLISHING</td>
</tr>
<tr>
<td>11</td>
<td>20.94858825</td>
<td>APPLE &lt;AAPL&gt; EXPANDS NETWORK CAPABILITIES</td>
</tr>
<tr>
<td>12</td>
<td>6.982862751</td>
<td>RJR &lt;RJR&gt; UNIT SELLS FOUR TOBACCO BRANDS</td>
</tr>
<tr>
<td>13</td>
<td>6.982862751</td>
<td>NYNEX &lt;NYN&gt; TO SELL NEW IBM &lt;IBM&gt;</td>
</tr>
</tbody>
</table>
Table 7.15: Weighted Documents for inputted word ‘Apple’

<table>
<thead>
<tr>
<th>Document</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>WALL STREET STOCKS/COMPAQ</td>
<td>6.982862751</td>
</tr>
<tr>
<td>COMPUTER &lt;CPQ&gt;</td>
<td></td>
</tr>
<tr>
<td>THORN-EMI WINS U.S. RULING IN</td>
<td>13.9657255</td>
</tr>
<tr>
<td>BEATLES SUIT</td>
<td></td>
</tr>
<tr>
<td>DIGITAL COMMUNICATIONS &lt;DCAI&gt;</td>
<td>6.982862751</td>
</tr>
<tr>
<td>INTRODUCES ITEMS</td>
<td></td>
</tr>
<tr>
<td>ACTIVISION &lt;AVSN.O&gt; RELEASES</td>
<td>6.982862751</td>
</tr>
<tr>
<td>ENHANCED PROGRAM</td>
<td></td>
</tr>
<tr>
<td>SAN FRANCISCO, NOT REGION,</td>
<td>6.982862751</td>
</tr>
<tr>
<td>HURT BY RESTRUCTURING</td>
<td></td>
</tr>
<tr>
<td>FEMALE FRUIT FLY MAY LEAD TO</td>
<td>6.982862751</td>
</tr>
<tr>
<td>CALIFORNIA SPRAYING</td>
<td></td>
</tr>
<tr>
<td>SUN MICRO'S &lt;SUNW.O&gt; TOPS</td>
<td>20.94858825</td>
</tr>
<tr>
<td>UPGRADERS SOFTWARE</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.16: Chart for Weighted Documents (inputted word ‘Apple’)

Table 7.15 shows the documents with their respective weights for inputted word ‘Apple’. Fig. 7.16 shows the chart for the same. In chart, vertical dimension shows the weights and horizontal dimension shows the titles of the documents.
<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Weight</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34.91431376</td>
<td>APPLE COMPUTER &lt;AAPL&gt; UPGRADES MACINTOSH LINE</td>
</tr>
<tr>
<td>2</td>
<td>20.94858825</td>
<td>APPLE COMPUTER &lt;AAPL&gt; HAS NEW MACINTOSH MODELS</td>
</tr>
<tr>
<td>3</td>
<td>20.94858825</td>
<td>APPLE &lt;AAPL&gt;, AST &lt;ASTA&gt; OFFER MS-DOS PRODUCTS</td>
</tr>
<tr>
<td>4</td>
<td>20.94858825</td>
<td>APPLE &lt;AAPL&gt; EXPANDS NETWORK CAPABILITIES</td>
</tr>
<tr>
<td>5</td>
<td>20.94858825</td>
<td>SUN MICRO'S &lt;SUNW.O&gt; TOPS UPGRADES SOFTWARE</td>
</tr>
<tr>
<td>6</td>
<td>13.9657255</td>
<td>FCOJ SUPPLIES SIGNIFICANTLY ABOVE YEAR AGO-USDA</td>
</tr>
<tr>
<td>7</td>
<td>13.9657255</td>
<td>BERTELSMANN TO MARKET APPLE SOFTWARE IN GERMANY</td>
</tr>
<tr>
<td>8</td>
<td>13.9657255</td>
<td>THORN-EMI WINS U.S. RULING IN BEATLES SUIT</td>
</tr>
<tr>
<td>9</td>
<td>6.982862751</td>
<td>LOTUS &lt;LOTS&gt; INTRODUCES NEW SOFTWARE</td>
</tr>
<tr>
<td>10</td>
<td>6.982862751</td>
<td>COMPUTER COMPANIES FORM NETWORKING GROUP</td>
</tr>
<tr>
<td>11</td>
<td>6.982862751</td>
<td>MOTOROLA (MOT) SEES CONTINUED GROWTH FOR CHIPS</td>
</tr>
<tr>
<td>12</td>
<td>6.982862751</td>
<td>TECHNOLOGY/DESKTOP PUBLISHING</td>
</tr>
<tr>
<td>13</td>
<td>6.982862751</td>
<td>RJR &lt;RJR&gt; UNIT SELLS FOUR TOBACCO BRANDS</td>
</tr>
<tr>
<td>14</td>
<td>6.982862751</td>
<td>NYNEX &lt;NYN&gt; TO SELL NEW IBM &lt;IBM&gt; COMPUTERS</td>
</tr>
<tr>
<td>15</td>
<td>6.982862751</td>
<td>WALL STREET STOCKS/COMPAQ</td>
</tr>
</tbody>
</table>
Table 7.16: Ranked Documents for inputted word ‘Apple’

Table 7.16 shows the ranked documents with their respective weights for inputted word ‘Apple’. Fig. 7.17 shows the chart for the same. In chart, vertical dimension shows the weights and horizontal dimension shows the titles of the documents.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Document</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>COMPUTER &lt;CPQ&gt;</td>
<td>6.982862751</td>
</tr>
<tr>
<td>17</td>
<td>DIGITAL COMMUNICATIONS &lt;DCAI&gt;</td>
<td>6.982862751</td>
</tr>
<tr>
<td></td>
<td>INTRODUCES ITEMS</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>ACTIVISION &lt;AVSN.O&gt; RELEASES</td>
<td>6.982862751</td>
</tr>
<tr>
<td></td>
<td>ENHANCED PROGRAM</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>SOFTWARE COS SUPPORT NEW APPLE &lt;AAPL&gt; PRODUCTS</td>
<td>6.982862751</td>
</tr>
<tr>
<td>20</td>
<td>SAN FRANCISCO, NOT REGION, HURT BY RESTRUCTURING</td>
<td>6.982862751</td>
</tr>
<tr>
<td>21</td>
<td>FEMALE FRUIT FLY MAY LEAD TO CALIFORNIA SPRAYING</td>
<td>6.982862751</td>
</tr>
</tbody>
</table>

Figure 7.17: Chart for Ranked Documents (inputted word ‘Apple’)

- 268 -
Table 7.17: Weighted Documents for inputted word ‘Porsche’

Table 7.17 shows the documents with their respective weights for inputted word ‘Porsche’. Fig. 7.18 shows the chart for the same. In chart, vertical dimension shows the weights and horizontal dimension shows the titles of the documents.

Figure 7.18: Chart for Weighted Documents (inputted word ‘Porsche’)

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Weight</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.7997906462794</td>
<td>&lt;PORSCHE AG&gt; RAISING U.S. PRICES</td>
</tr>
<tr>
<td>2</td>
<td>7.89989532313973</td>
<td>DAIMLER OUTPUT COULD BE AFFECTED BY DISPUTE</td>
</tr>
<tr>
<td>3</td>
<td>7.89989532313973</td>
<td>PORSCHE CARS NORTH AMERICA REPORTS LOWER SALES</td>
</tr>
<tr>
<td>4</td>
<td>23.6996859694191</td>
<td>PORSCHE RECALLS 892 OF ITS 1987 MODEL CARS</td>
</tr>
<tr>
<td>5</td>
<td>23.6996859694191</td>
<td>PORSCHE HALF-YEAR EARNINGS CALLED SATISFACTORY</td>
</tr>
<tr>
<td>6</td>
<td>31.5995812925589</td>
<td>PORSCHE EXPECTS IMPROVEMENT IN U.S. SALES</td>
</tr>
<tr>
<td>7</td>
<td>15.7997906462794</td>
<td>PORSCHE MARCH SALES FALL</td>
</tr>
<tr>
<td>8</td>
<td>15.7997906462794</td>
<td>GERMAN CAR FIRMS SEEN PRODUCING SOLID 1987 RESULTS</td>
</tr>
</tbody>
</table>
Table 7.18: Ranked Documents for inputted word ‘Porsche’

Table 7.18 shows the ranked documents with their respective weights for inputted word ‘Porsche’. Fig. 7.19 shows the chart for the same. In chart, vertical dimension shows the weights and horizontal dimension shows the titles of the documents.

Figure 7.19: Chart for Ranked Documents (inputted word ‘Porsche’)

So, from the tables and charts for different inputted words, it can be said that, both, the document-weighting and document-ranking algorithms are working correctly and give satisfactory results. Hence they can be easily incorporated into document search-engine system and provides an easier way to search for relevant documents.