ISSN No. 0976-5697

Volume 4, No. 9, July-August 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Survey on Preprocessing in Text Mining

Dr. Anadakumar. K
Assistant Professor (Selection Grade), Department of computer Application, Bannari Amman Institute of Technology, India
anandhsns@yahoo.co.in

Ms. Padmavathy. V
Research Scholar,
Department of computer Science, Dr.S.N.S. Rajalakshmi College of Arts and Science
mv.padmavathy@gmail.com

Abstract: Now-a-days information's are stored electronically in databases. Extracting reliable, unknown and useful information from the abundant source is an eminent task. Data mining and Text mining are the process for extracting unknown and useful information. Text Mining is the process of extracting interesting and non-trivial patterns or knowledge from text documents. This paper presents the related activities and focuses on preprocessing steps in text mining.

Keywords— DBMS, Data Mining, Text Mining, Preprocessing, Tokenization, Stemming, POS tagging.

I. INTRODUCTION

We have been collecting tremendous amounts of information. Initially, with the advent of computers and means for mass digital storage, we started collecting and storing all sorts of data, counting on the power of computers to help sort through this amalgam of information. Unfortunately, these massive collections of data stored on disparate structures very rapidly became overwhelming. This initial chaos has led to the creation of structured databases and database management systems (DBMS). The efficient database management systems have been very important assets for management of a large corpus of data and especially for effective and efficient retrieval of particular information from a large collection whenever needed. The proliferation of database management systems has also contributed to recent massive gathering of all sorts of information. Today, we have far more information than we can handle: from business transactions and scientific data, to satellite pictures, text reports and military intelligence. Information retrieval is simply not enough anymore for decision-making. Confronted with huge collections of data, we have now created new needs to help us make better managerial choices. These needs are automatic summarization of data, extraction of the "essence" of information stored, and the discovery of patterns in raw data.[1],[2]

With the enormous amount of data stored in files, databases, and other repositories, it is increasingly important, if not necessary, to develop powerful means for analysis and perhaps interpretation of such data and for the extraction of interesting knowledge that could help in decision-making.[3]

II. EVALUATION OF DATA BASES

Since the 1960s, database and information technology has been evolving systematically from primitive file processing systems to sophisticated and powerful database systems. The research and development in database systems since the 1970s has progressed from early hierarchical and network database systems to the development of relational database systems, data modeling tools, and indexing and accessing methods. In addition, users gained convenient and flexible data access through query languages, user interfaces, optimized query processing, and transaction management. Efficient methods for on-line transaction processing (OLTP), where a query is viewed as a read-only transaction, have contributed substantially to the evolution and wide acceptance of relational technology as a major tool for efficient storage, retrieval, and management of large amounts of data[4],[5]

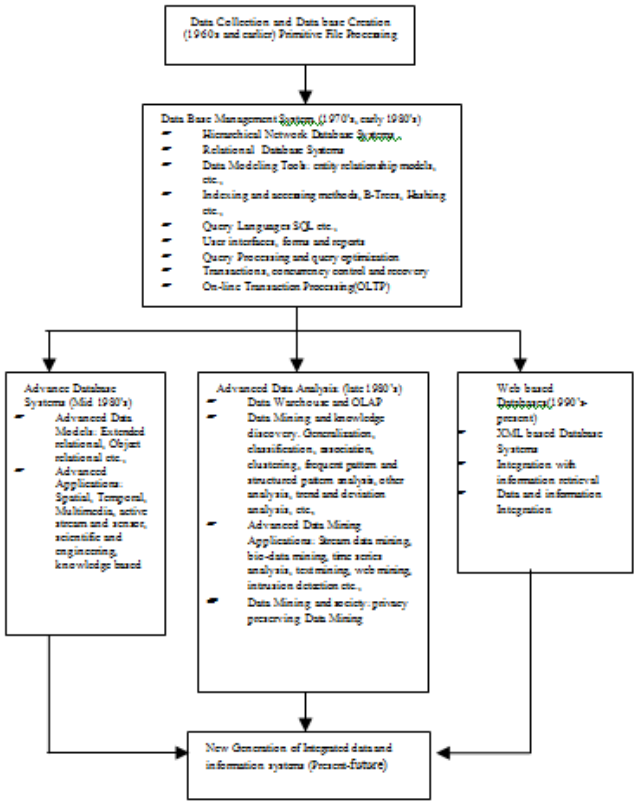

Figure 1 The evolution of database system Technology

Database technology since the mid-1980s has been characterized by the popular adoption of relational technology as shown in *fig 1* and an upsurge of research and development activities on new and powerful database systems. These promote the development of advanced data

models such as extended-relational, object-oriented, object-relational and deductive models. Application-oriented database systems, including spatial, temporal, multimedia, active, stream, and sensor, and scientific and engineering databases, knowledge bases, and office information bases, have flourished.[6][7]

Issues related to the distribution, diversification, and sharing of data have been studied extensively. Heterogeneous database systems and Internet-based global information systems such as the World Wide Web (WWW) have also emerged and play a vital role in the information industry.[8]

III. DATA MINING

Data can now be stored in many different kinds of databases and information repositories. One data repository architecture that has emerged is the data warehouse, a repository of multiple heterogeneous data sources organized under a unified schema at a single site in order to facilitate management decision making.

Data mining refers to extracting or "mining" knowledge from large amounts of data. The term is actually a misnomer. Remember that the mining of gold from rocks or sand is referred to as *gold* mining rather than rock or sand mining. Thus, data mining should have been more appropriately named "knowledge mining from data," which is unfortunately somewhat long. "Knowledge mining," a shorter term, may not reflect the emphasis on mining from large amounts of data. Nevertheless, mining is a vivid term characterizing the process that finds a small set of precious nuggets from a great deal of raw materials. Thus, such a misnomer that carries both "data" and "mining" became a popular choice. Many other terms carry a similar or slightly different meaning to data mining, such as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging.[9]

A. Definition:

Data Mining is a Non-trivial extraction of implicit, previously unknown and potentially useful information from data. Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns**. [10]**

B. Process of KDD:

Data Mining is sometimes referred to as KDD. The Knowledge Discovery in Databases process comprises of a few steps leading from raw data collections to some form of new knowledge.

The iterative process consists of the following steps:

- ***a) Data cleaning****:* also known as data cleansing, it is a phase in which noise data and irrelevant data are removed from the collection.[11]
- ***b) Data integration****:* at this stage, multiple data sources, often heterogeneous, may be combined in a common source.
- ***c) Data selection****:* at this step, the data relevant to the analysis is decided on and retrieved from the data collection.
- ***d) Data transformation****:* also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.
- ***e) Data mining****:* it is the crucial step in which clever techniques are applied to extract patterns potentially useful.
- ***f) Pattern evaluation***: in this step, strictly interesting patterns representing knowledge are identified based on given measures.
- ***g) Knowledge representation****:* is the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.[12],[13]

It is common to combine some of these steps together. For instance, *data cleaning* and *data integration* can be performed together as a pre-processing phase to generate a data warehouse. *Data selection* and *data transformation* can also be combined where the consolidation of the data is the result of the selection, or, as for the case of data warehouses, the selection is done on transformed data.

The KDD is an iterative process. Once the discovered knowledge is presented to the user, the evaluation measures can be enhanced, the mining can be further refined, new data can be selected or further transformed, or new data sources can be integrated, in order to get different, more appropriate result[14][15]

C. Applications of Data Mining:

Data mining is becoming increasingly common in both the private and public sectors. Industries such as banking, insurance, medicine, and retailing commonly use data mining to reduce costs, enhance research, and increase sales. In the public sector, data mining applications initially were used as a means to detect fraud and waste, but have grown to also be used for purposes such as measuring and improving program performance.

Data mining is emerging as one of the key features of many homeland security initiatives. Often used as a means for detecting fraud, assessing risk, and product retailing, data mining involves the use of data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. In the context of homeland security, data mining is often viewed as a potential means to identify terrorist activities, such as money transfers and communications, and to identify and track individual terrorists themselves, such as through travel and immigration records. [16],[17]

D. Limitations of Data Mining:

While data mining represents a significant advance in the type of analytical tools currently available, there are limitations to its capability. One limitation is that although data mining can help reveal patterns and relationships, it does not tell the user the value or significance of these patterns. These types of determinations must be made by the user. A second limitation is that while data mining can identify connections between behaviors and/or variables, it does not necessarily identify a causal relationship. To be successful, data mining still requires skilled technical and analytical specialists who can structure the analysis and interpret the output that is created.[18]

E. Issues in data mining:

Data mining algorithms embody techniques that have sometimes existed for many years, but have only lately been applied as reliable and scalable tools that time and again

outperform older classical statistical methods. While data mining is still in its infancy, it is becoming a trend and ubiquitous. Before data mining develops into a conventional, mature and trusted discipline, many still pending issues have to be addressed. Some of these issues are addressed below. Note that these issues are not exclusive and are not ordered in any way.

a. Security and social issues:

Security is an important issue with any data collection that is shared and/or is intended to be used for strategic decision-making. In addition, when data is collected for customer profiling, user behavior understanding, correlating personal data with other information, etc., large amounts of sensitive and private information about individuals or companies is gathered and stored. This becomes controversial given the confidential nature of some of this data and the potential illegal access to the information. Moreover, data mining could disclose new implicit knowledge about individuals or groups that could be against privacy policies, especially if there is potential dissemination of discovered information. Another issue that arises from this concern is the appropriate use of data mining. Due to the value of data, databases of all sorts of content are regularly sold, and because of the competitive advantage that can be attained from implicit knowledge discovered, some important information could be withheld, while other information could be widely distributed and used without control.[19]

b. User interface issues:

The knowledge discovered by data mining tools is useful as long as it is interesting, and above all understandable by the user. Good data visualization eases the interpretation of data mining results, as well as helps users better understand their needs. Many data exploratory analysis tasks are significantly facilitated by the ability to see data in an appropriate visual presentation. There are many visualization ideas and proposals for effective data graphical presentation. However, there is still much research to accomplish in order to obtain good visualization tools for large datasets that could be used to display and manipulate mined knowledge. The major issues related to user interfaces and visualization are "screen real-estate", information rendering, and interaction. Interactivity with the data and data mining results is crucial since it provides means for the user to focus and refine the mining tasks, as well as to picture the discovered knowledge from different angles and at different conceptual levels.

c. Mining methodology issue:

These issues pertain to the data mining approaches applied and their limitations. Topics such as versatility of the mining approaches, the diversity of data available, the dimensionality of the domain, the broad analysis needs (when known), the assessment of the knowledge discovered, the exploitation of background knowledge and metadata, the control and handling of noise in data, etc. are all examples that can dictate mining methodology choices. Moreover, different approaches may suit and solve user's needs differently. Most algorithms assume the data to be noise-free. This is of course a strong assumption. Most datasets contain exceptions, invalid or incomplete information, etc., which may complicate, if not obscure, the analysis process and in many cases compromise the accuracy of the results. As a consequence, data preprocessing (data cleaning and transformation) becomes vital. It is often seen as lost time, but data cleaning, as time consuming and frustrating as it may be, is one of the most important phases in the knowledge discovery process. Data mining techniques should be able to handle noise in data or incomplete information. More than the size of data, the size of the search space is even more decisive for data mining techniques. The size of the search space is often depending upon the number of dimensions in the domain space. The search space usually grows exponentially when the number of dimensions increases. This is known as the *curse of dimensionality*. This "curse" affects so badly the performance of some data mining approaches that it is becoming one of the most urgent issues to solve.

d. Performance issues:

Many artificial intelligence and statistical methods exist for data analysis and interpretation. However, these methods were often not designed for the very large data sets data mining is dealing with today. Terabyte sizes are common. This raises the issues of scalability and efficiency of the data mining methods when processing considerably large data. Algorithms with exponential and even medium-order polynomial complexity cannot be of practical use for data mining. Linear algorithms are usually the norm. In same theme, sampling can be used for mining instead of the whole dataset. However, concerns such as completeness and choice of samples may arise. Other topics in the issue of performance are incremental updating, and parallel programming. There is no doubt that parallelism can help solve the size problem if the dataset can be subdivided and the results can be merged later. Incremental updating is important for merging results from parallel mining, or updating data mining results when new data becomes available without having to re-analyze the complete dataset[20].

e. Data source issues:

There are many issues related to the data sources, some are practical such as the diversity of data types, while others are philosophical like the data glut problem. We certainly have an excess of data since we already have more data than we can handle and we are still collecting data at an even higher rate. If the spread of database management systems has helped increase the gathering of information, the advent of data mining is certainly encouraging more data harvesting. The current practice is to collect as much data as possible now and process it, or try to process it, later. The concern is whether we are collecting the right data at the appropriate amount, whether we know what we want to do with it, and whether we distinguish between what data is important and what data is insignificant. Regarding the practical issues related to data sources, there is the subject of heterogeneous databases and the focus on diverse complex data types.[21]

IV. TEXT MINING

Text mining is the process that turns text into data that can be analyzed. Text mining, also known as text data mining or knowledge discovery from textual databases, refers to the process of extracting interesting and non-trivial

patterns or knowledge from text documents. Text Mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. A key element is the linking together of the extracted information together to form new facts or new hypotheses to be explored further by more conventional means of experimentation.

Text Mining can not only extract information on protein interactions from documents, but it can also go one step further to discover patterns in the extracted interactions. Information may be discovered that would have been extremely difficult to find, even if it had been possible to read all the documents. This information could help to answer existing research questions or suggest new avenues to explore.

Text mining is different from what we're familiar with in web search. In search, the user is typically looking for something that is already known and has been written by someone else. The problem is pushing aside all the material that currently isn't relevant to your needs in order to find the relevant information.

In text mining, the goal is to discover heretofore unknown information, something that no one yet knows and so could not have yet written down. [22]

A. Why Text mining:

Broadly speaking there are four main reasons to embark on text mining: to enrich the content in some way; to enable systematic review of literature; for discovery or for computational linguistics research.

a. Enriching the Content:

Text mining can be used to improve the indexing of text. This is especially useful for publishers since they can create indexes more efficiently (machine-aided indexing). They can also add value in a digital environment by creating links from articles to relevant further reading. For example, mentions of gene sequences can be identified in articles and linked directly to databases such as GenBank.. This use of text mining is widespread and predicted to grow quickly. 46% of publishers in a recent study5 reported that they currently text mine their own content, of the ones that do not, a further 30% will start doing so within a year of the study.6 Third party tools have also been developed to improve the reading experience, such as Utopia Docs which identifies named entities within PDFs and build links out to related web resources in the life sciences.

b. Systematic Review of Literature:

Text mining can help a scientist to systematically review a much larger body of content and do faster. There is considerable demand for this kind of text mining in the corporate environment: why pay biologists to read biology papers when machines can do it for them, and they can concentrate on doing research instead? Furthermore, text mining can help researchers keep up with their field and reduce the risk that they miss something relevant

c. Discovery:

Text mining is used to create databases that can be mined for discovering new insights. Many people especially in the pharmaceutical world, believe that there is huge promise here and to a large extent this is driving the hype around text mining. Scholarly texts are written to communicate factual information or opinions and so it seems to make sense to try to extract this information automatically. However, there are very few published examples that show new insights as a direct result of data mining. One example identifying new therapeutic uses for thalidomide is often quoted.8 It is not clear what can be considered as a new insight. Is it the discovery of some sort of association between a gene and the literature surrounding a particular disease, or is it only an insight if the association is verified in the lab? It is probably more useful to think of text mining as machine-aided research tool that can open up additional sources of information for use in research rather than as some sort of holy grail.

d. Computational Linguistics Research:

Text mining itself is the subject of research into text mining. There is considerable work worldwide in the field of computational linguistics dedicated to improving the extraction of meaning from text. Text mining is the raw material for this research. This area appears to be driving a very large part of the current activity in text mining. Around half of the publishers recently surveyed have been approached by researchers in this field requesting permission to mine their content.6 This is also the area where the most progress is being made in developing new tools and techniques. The research in this area is often challenge driven. For example, BioCreative9 sets text mining tasks as challenges that are relevant to biomedicine and that stimulate the community to develop new methods. Challenges like this have also resulted in the development of tools for researchers in other scientific disciplines such as Reflect for the life sciences.[23]

B. Data Mining Vs Text Mining:

The difference between regular data mining and text mining is that in text mining the patterns are extracted from natural language text rather than from structured databases of facts. Databases are designed for programs to process automatically; text is written for people to read. We do not have programs that can "read" text and will not have such for the foreseeable future. Many researchers think it will require a full simulation of how the mind works before we can write programs that read the way people do. [24]

As the most natural form of storing information is text, text mining is believed to have a commercial potential higher than that of data mining. In fact, a recent study indicated that 80% of a company's information is contained in text documents. Text mining, however, is also a much more complex task (than data mining) as it involves dealing with text data that are inherently unstructured and fuzzy. Text mining is a multidisciplinary field, involving information retrieval, text analysis, information extraction, clustering, categorization, visualization, database technology, machine learning, and data mining.

C. A Framework of text mining:

Text mining can be visualized as consisting of two phases:

Text refining that transforms free-form text documents into a chosen intermediate form, and knowledge distillation that deduces patterns or knowledge from the intermediate form. Intermediate form (IF) can be semi-structured such as the conceptual graph representation, or structured such as the relational data representation. Intermediate form can be

document-based wherein each entity represents a document, or *concept based* wherein each entity represents an object or concept of interests in a specific domain. Mining a document-based IF deduces patterns and relationship across documents. Document clustering/visualization and categorization are examples of mining from a document-based IF. Mining a concept-based IF derives pattern and relationship across objects or concepts. Data mining operations, such as predictive modeling and associative discovery, fall into this category. A document-based IF can be transformed into a concept-based IF by realigning or extracting the relevant information according to the objects of interests in a specific domain. It follows that document-based IF is usually domain-independent and concept-based IF is domain-dependent. Figure 1: A text mining framework. *Text refining* converts unstructured text documents into an *intermediate form (IF)*. IF can be *document-based* or *concept-based. Knowledge distillation* from a *document-based IF* deduces patterns or knowledge across documents. A *document-based IF* can be projected onto a *concept-based IF* by extracting object information relevant to a domain. *Knowledge distillation* from a *concept-based IF* deduces patterns or knowledge across objects or concepts.[25]

D. The Working Process of Text Mining:

The prime aim of the text mining is to identify the useful information without duplication from various documents with synonymous understanding. Text Mining is an empirical tool that has a capacity of identifying new information that is not apparent from a document collection. The processing steps are shown in fig 2.

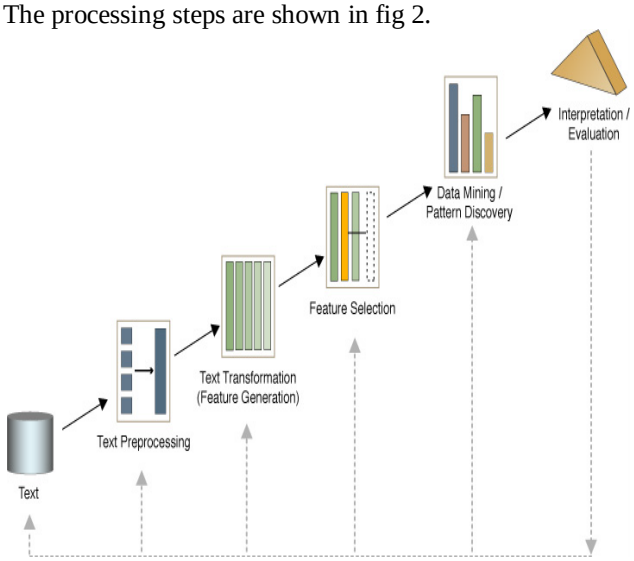

Fig 2 The Working process of Text Mining

The Process of Text Mining includes searching, extracting, categorization where the themes are readable and the meaning is obvious. Typically text mining tasks include text categorization, text clustering, Information extraction, information retrieval, sentiment analysis, document summarization and entity relation modeling

a) Information retrieval (IR) is a field that has been developing in parallel with database systems for many years. Unlike the field of database systems, which has focused on query and transaction processing of structured data, information retrieval is concerned with the organization and retrieval of information from a large number of text-based documents. Since information retrieval and database systems each handle different kinds of data, some database system problems are usually not present in information retrieval systems, such as concurrency control, recovery, transaction management, and update. Also, some common information retrieval problems are usually not encountered in traditional database systems, such as unstructured documents, approximate search based on keywords, and the notion of relevance. Due to the abundance of text information, information retrieval has found many applications. There exist many information retrieval systems, such as on-line library catalog systems, on-line document management systems, and the more recently developed Web search engines.

The most well known IR systems are search engines such as Google™, which identify those documents on the WWW that are relevant to a set of given words. IR systems are often used in libraries, where the documents are typically not the books themselves but digital records containing information about the books. This is however changing with the advent of digital libraries, where the documents being retrieved are digital versions of books and journals.

IR systems allow us to narrow down the set of documents that are relevant to a particular problem. As text mining involves applying very computationally intensive algorithms to large document collections, IR can speed up the analysis considerably by reducing the number of documents for analysis. For example, if we are interested in mining information only about protein interactions, we might restrict our analysis to documents that contain the name of a protein, or some form of the verb ‘to interact’ or one of its synonyms.[26][27]

b). Natural language processing:

NLP is one of the oldest and most difficult problems in the field of artificial intelligence. It is the analysis of human language so that computers can understand natural languages as humans do. Although this goal is still some way off, NLP can perform some types of analysis with a high degree of success. For example:

(a). Part-of-speech tagging classifies words into categories such as noun, verb or adjective

(b). Word sense disambiguation identifies the meaning of a word, given its usage, from the multiple meanings that the word may have

(c). Parsing performs a grammatical analysis of a sentence. Shallow parsers identify only the main grammatical elements in a sentence, such as noun phrases and verb phrases, whereas deep parsers generate a complete representation of the grammatical structure of a sentence

The role of NLP in text mining is to provide the systems in the information extraction phase (see below) with linguistic data that they need to perform their task. Often this is done by annotating documents with information such as sentence boundaries, part-of-speech tags and parsing results, which can then be read by the information extraction tools.[28]

c). *Information extraction (IE)* is the process of automatically obtaining structured data from an unstructured natural language document. Often this involves defining the general form of the information

that we are interested in as one or more templates, which are then used to guide the extraction process. IE systems rely heavily on the data generated by NLP systems. Tasks that IE systems can perform include:

(a). Term analysis, which identifies the terms in a document, where a term may consist of one or more words. This is especially useful for documents that contain many complex multi-word terms, such as scientific research papers

(b). Named-entity recognition, which identifies the names in a document, such as the names of people or organizations. Some systems are also able to recognize dates and expressions of time, quantities and associated units, percentages, and so on

(c). Fact extraction, which identifies and extracts complex facts from documents. Such facts could be relationships between entities or events

A very simplified example of the form of a template and how it might be filled from a sentence is shown in Figure 1. Here, the IE system must be able to identify that 'bind' is a kind of interaction, and that 'myosin' and 'actin' are the names of proteins. This kind of information might be stored in a dictionary or an ontology, which defines the terms in a particular field and their relationship to each other. The data generated during IE are normally stored in a database ready for analysis in the final stage, data mining.

When used in text mining, DM is applied to the facts generated by the information extraction phase

We put the results of our DM process into another database that can be queried by the end-user via a suitable graphical interface. The data generated by such queries can also be represented visually **[29][30]**

V. TEXT PREPROCESSING

Text Mining starts with a collection of documents; which would retrieve a particular document and preprocess it by checking format and character sets. Then it would go through a text analysis phase, sometimes repeating techniques until information is extracted, many other combinations of techniques could be used depending on the goals of the organization. The resulting information can be placed in a management information system yielding an abundant amount of knowledge for the user of that system.

"Preprocessing" is the process to distill the documents into a structured format

A. Overview of Pre-processing Steps:

a) Extract text and structure :
(eg. from Microsoft Word, HTML pages or LaTeX to XML)
b) Clean characters and encoding
c) Remove stop words (eg. remove "the", "at", "all", etc)
d) Named entity recognition (eg. find proper names)
e) Stemming (eg. extract "process" from "processing")
f) Part-of-Speech Tagging

The preprocessing phase converts the original textual data in a data-mining-ready structure, where the most significant text-features that serve to differentiate between text-categories are identified as shown in fig 3. It is the process of incorporating a new document into an information retrieval system. An effective preprocessor represents the document efficiently in terms of both space (for storing the document) and time (for processing retrieval requests) requirements and maintain good retrieval performance (precision and recall). This phase is the most critical and complex process that leads to the representation of each document by a select set of index terms. The main objective of preprocessing is to obtain the key features or key terms from online news text documents and to enhance the relevancy between word and document and the relevancy between word and category. [31][32]

The dataset is an unstructured dataset of documents which are pre-processed using the following three rules:

a) Tokenize the file into individual tokens using space as the delimiter.
b) Removing the stop word which does not convey any meaning.
c) Use stemmer algorithm to stem the words with common root word.

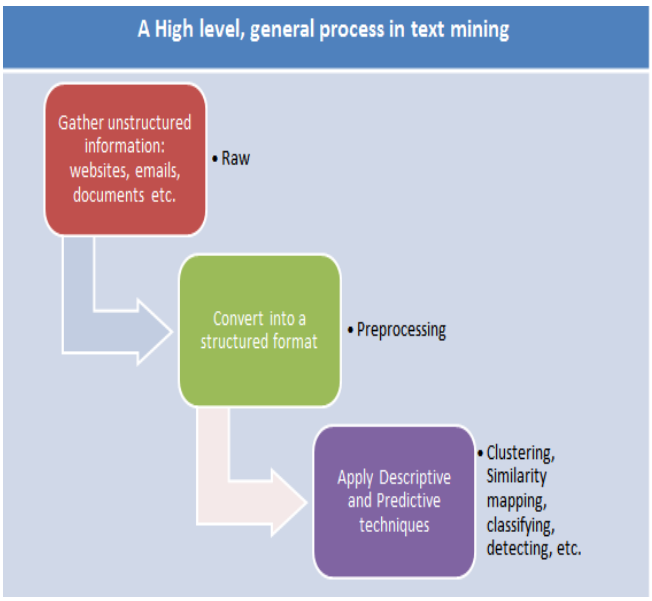

Figure 3 General Preprocessing in Text mining

B. Tokenization:

a. What is Tokenization?:

Tokenization is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences. In the process of tokenization, some characters like punctuation marks are discarded. The tokens become the input for another process like parsing and text mining.

Tokenization relies mostly on simple heuristics in order to separate tokens by following a few steps:

a) Tokens or words are separated by whitespace, punctuation marks or line breaks.
b) White space or punctuation marks may or may not be included depending on the need
c) All characters within contiguous strings are part of the token. Tokens can be made up of all alpha characters, alphanumeric characters or numeric characters only.

Tokens themselves can also be separators. For example, in most programming languages, identifiers can be placed together with arithmetic operators without white spaces. Although it seems that this would appear as a single word or

token, the grammar of the language actually considers the mathematical operator (a token) as a separator, so even when multiple tokens are bunched up together, they can still be separated via the mathematical operator.

b. Tokenization in Text Preprocessing:

Tokenization is commonly understood as the first step of any kind of natural language text preparation. The major goal of this early (pre-linguistic) task is to convert a stream of characters into a stream of processing units called tokens. Beyond the text mining community this job is taken for granted. Commonly it is seen as an already solved problem comprising the identification of word borders and punctuation marks separated by spaces and line breaks. But in our sense it should manage language related word dependencies, incorporate domain specific knowledge, and handle morpho syntactically relevant linguistic specificities. Therefore, we propose rule based Extended Tokenization including all sorts of linguistic knowledge (e.g., grammar rules, dictionaries). The core features of our implementation are identification and disambiguation of all kinds of linguistic markers, detection and expansion of abbreviations, treatment of special formats, and typing of tokens including single- and multi-tokens. To improve the quality of text mining we suggest linguistically-based tokenization as a necessary step proceeding further text processing tasks.

c. Types of Tokens:

First, raw texts are preprocessed and segmented into textual units. This step comprises cleansing and filtering (e.g., whitespace collapsing, stripping extraneous control characters) and removal of all kinds of structural or layout relevant markup (e.g., XML tags). Then, Extended Tokenization segments the plain text into appropriate processing units. Subsequent tasks like tagging are applied on the tokenized output and thus should be supported as far as possible (e.g., format normalization, consistent terminology).

d. Single Token:

The most simple form of a token is the single-token. It is defined as a character string not containing any non-printable or delimiting characters (blank, tabulator, line-feed, new line, etc.), corresponding to the traditional concept of a token.

e. Multi Tokens:

Written texts also contain more complex language constructs that do not fit into the singe-token concept. Such tokens may be specially formatted using blanks (the standard delimiter for token boundaries) or belong to semantically motivated groups of tokens. In this case the blank is part of a token chain fixed together through interpretation. We define such tokens containing token delimiter characters for formatting (e.g., blanks, tabs, new lines, line feeds) as multi-tokens.

Well known representants are composite nouns, special formats ('+43 463 2700-3531'), named entities (names, locations, institutions), and idioms (formulas). Traditionally they have been identified as a sequence of atomic tokens glued together during a later processing phase - mainly using dictionary lookup. In our approach these tokens are multi-tokens mainly through heuristic interpretation or, in other words, they are tokens through rule-based typing.

The early treatment of multi-tokens as (semantic) concepts during text preparation benefits the overall quality of data- and text-mining tasks. The representation of a text using multi-tokens leads to better intermediate results, hence structurally and (semantically) grouped tokens are treated as atomic units. If subsequent tasks do not support multi-tokens, a simple reinterpretation into standard tokens is possible.

f. Tokenization Working Method:

The tokenization algorithm starts with basic text segmentation, separating strings into single using standard delimiters (blanks, tabs, new lines, line feeds). Each identified single-token is typed using a predefined set of basic token types.

Examples of basic types and subtypes are

(a). Alphabetic

i. Ta: no letters capitalized (Ta1),
ii. first letter capitalized(Ta2),
iii. all letters capitalized (Ta3),
iv. mixed cases (Ta4) etc.

(b). Numeric's

i. Tn: plain numbers (Tn1)
ii. numbers containing periods or colons (Tn2) etc.

i. punctuation marks Tp: sentence ending markers (Tp1),
ii. pairwise marks lick brackets and quotes (Tp2),
iii. single sentence-internal marks like commas (Tp3) etc.

(c). Mixtures

i. Tm: ending with sentence end marker (Tm1),
ii. ending with hyphen (Tm2),
iii. starting with hyphen (Tm3), containing slashes/hyphens (Tm4),
iv. containing numbers (Tm5) etc.

These basic types are assigned to tokens straightforward, utilizing a classification of characters into distinct categories.

g. User Defined Tokens:

During the next step punctuation marks are identified and separated Only tokens typed as mixtures (Tm1) are investigated. A set of user-defined token types is used to reinterpret and group (basic) token types and strings . The user is enabled to define custom types to support domain-specific needs. Such token types are simply expressed through strings, which are assigned to recognized tokens. The definition of token types can be related to different sources of knowledge about the motivation for token interpretation. This includes domain knowledge (i.e., structure of an organization, knowledge about data warehouses), gazetteer knowledge (i.e., country names, river names), expert knowledge (i.e., medicine, astronomy), and pure linguistic knowledge (i.e., morphological and syntactical rules, subject of a sentence).

Examples of user-defined types are

a) stop words (U1),
b) abbreviations (U2),
c) dates and times (U3),
d) phone numbers (U4),
e) email addresses (U5),
f) a sequence of capitalized single-tokens (U6, in many cases extended keywords) etc.

These types are identified by applying two strategies: First, tokens are compared with an repository of reliable (string; token type) entries created by a human expert or any kind of (semi) automatic machinery. If no match is found, an ordered list of rules is applied to process the sequence of tokens .

The rules include regular expression matching of token), matching of token types and combinations. Example types of rules may cover morphological, syntactical, and general patterns like

a) Suffix identification of well-known endings (e.g., "-ly", "-ness").
b) Identification and reconcatenation of hyphenated words at line breaks
c) Sentence border disambiguation
d) Multi-token identification
e) Special character treatment (e.g., apostrophes, slashes, ampersand etc.)[34][35][36].

VI. STOP WORD ELIMINATION

Stop-words are words that from non-linguistic view do not carry information. They have mainly functional role and usually we remove them to help the methods to perform better. Stop words are considered useless or less relevant to search results and so they are ignored.

The most common words are in any text document does not provide meaning of the documents; those are prepositions, articles, and pro-nouns etc. These words are treated as stop words. Because every text document deals with these words which are not necessary for text mining applications. These words are eliminated. Any group of words can be chosen as the stop words for a given purpose. This process also reduces the text data and improves the system performance.

A. What are Stop words?:

Words such as - and, because, or, not and a host of other words which basically form the foundation of English language are stop words. Search engines are more accurate when you search for products and services since ignoring the keywords may not affect the concept of information provided. But when you go into personal details, blog entries and articles, you will have less accurateresults. If the search results have to be better and if we are looking forward for a semantic web (intelligent web), then Stop words are everything. Until and unless these are incorporated in the search and clustering done with a strong emphasis for stop words – Search results aren't doing to turn any better.

Like Stop words A, AN, ONE should be clustered together when the result has to produce information related to single term. Words like AND should bring two keywords closer in the cluster and NOT should separate them more by an algorithm to ensure we get a more proper result. Stemming relevant stop words to one another using multiple algorithms will help clustering better.

B. List of Stop words:

A, about, above, above, across, after, afterwards, again, against, all, almost, alone, along, already, also, although, always, am, among, amongst, amongst, amount, an, and, another, any, anyhow, anyone, anything, anyway, anywhere, are, around, as, at, back, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, below, beside, besides, between, beyond, bill, both, bottom, but, by, call, can, cannot, cant, co, con, could, couldn't, cry, de, describe, detail, do, done, down, due, during, each, eg, eight, either, eleven, else, elsewhere, empty, enough, etc, even, ever, every, everyone, everything, everywhere, except, few, fifteen, fifty, fill, find, fire, first, five, for, former, formerly, forty, found, four, from, front, full, further, get, give, go, had, has, hasn't, have, he, hence, her, here, hereafter, hereby, herein, hereupon, hers, herself, him, himself, his, how, however, hundred, ie, if, in, inc, indeed, interest, into, is, it, its, itself, keep, last, latter, latterly, least, less, ltd, made, many, may, me, meanwhile, might, mill, mine, more, moreover, most, mostly, move, much, must, my, myself, name, namely, neither, never, nevertheless, next, nine, no, nobody, none, noon, nor, not, nothing, now, nowhere, of, off, often, on, once, one, only, onto, or, other, others, otherwise, our, ours, ourselves, out, over, own, part, per, perhaps, please, put, rather, re, same, see, seem, seemed, seeming, seems, serious, several, she, should, show, side, since, sincere, six, sixty, so, some, somehow, someone, something, sometime, sometimes, somewhere, still, such, system, take, ten, than, that, the, their, them, themselves, then, thence, there, thereafter, thereby, therefore, therein, thereupon, these, they, thick, thin, third, this, those, though, three, through, throughout, thru, thus, to, together, too, top, toward, towards, twelve, twenty, two, un, under, until, up, upon, us, very, via, was, we, well, were, what, whatever, when, whence, whenever, where, where after, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, whoever, whole, whom, whose, why, will, with, within, without, would, yet, you, your, yours, yourself.[37][38]

VII. STEMMING

Stemming is a technique for the reduction of words into their root. Many words in the English language can be reduced to their base form or stem e.g. agreed, agreeing, disagree, agreement and disagreement belong to agree. Furthermore are names transformed into the stem by removing the " ' s".

A. Word Stemming:

Word stemming is an important feature supported by present day indexing and search systems. Indexing and searching are in turn part of Text Mining applications, Natural Language Processing (NLP) systems and Information Retrieval (IR) systems. The main idea is to improve recall by automatic handling of word endings by reducing the words to their word roots, at the time of indexing and searching. Recall in increased without compromising on the precision of the documents fetched.

B. Stemming Process:

Stemming is usually done by removing any attached suffixes and prefixes (affixes) from index terms before the actual assignment of the term to the index. Since the stem of a term represents a broader concept than the original term, the stemming process eventually increases the number of retrieved documents in an IR system. Text clustering, categorization and summarization also require this conversion as part of the pre-processing before actually applying any related algorithm.

The result of the removal may lead to an incorrect root. However, these stems do not have to be a problem for the stemming process, if these words are not used for human interaction. The stem is still useful, because all other inflections of the root are transformed into the same stem. Case sensitive systems could have problems when making a comparison between a word in capital letters and another with the same meaning in lower case.

C. Uses of Stemming:

a. Improving effectiveness of IR and text mining
b. Matching similar words
c. Mainly improve recall
d. Reducing indexing size
e. Combing words with same roots may reduce indexing
    a) size as much as 40-50%.

D. Working of a Stemmer:

It has been seen that most of the times the morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. Since the meaning is same but the word form is different it is necessary to identify each word form with its base form. To do this a variety of stemming algorithms have been developed. Each algorithm attempts to convert the morphological variants of a word like introduction, introducing, introduces etc. to get mapped to the word 'introduce'. Some algorithms may map them to just 'introduce', but that is allowed as long as all of them map to the same word form or more popularly known as the stem form. Thus, the key terms of a query or document are represented by stems rather than by the original words. The idea is to reduce the total number of distinct terms in a document or a query which in turn will reduce the processing time of the final output.

E. Stemming Algorithm:

The most popular stemming algorithm is arguably that of MF Porter, who first described his process in a journal called Program in 1980. It wasn't the first stemming algorithm, but it was the first simple and easily implemented one.

Previous realizations relied on extremely complicated linguistically-informed transformation rules, which worked well for regularly inflected words but tended to break on irregular or archaic inflections. Porter's algorithm still takes regular and irregular forms into account, but by and large just takes a sledgehammer to words by decapitating affixes. And, as Porter recently proved, this doesn't affect its performance – his simple stemmer works really well.

Unfortunately, Porter didn't provide an official implementation of the algorithm with his article, and lots of variations quickly arose, creating any differing fragmented versions of the 'Porter stemmer'. This was remedied in 2001, when he published an official version on his website. It's written in the Snowball programming language (created by Porter and his colleagues specifically for the original implementation) and is available for many text analysis toolkits.

Imperfect variations on Porter's original algorithm are still out there and are being used in many very popular packages. Notably, R. which brings us to this little confusing anecdote, by way of a digression. R is fucking awesome at using machine learning to do really interesting text analysis on preprocessed text (like text that's been preprocessed with the Porter stemmer). R is not that awesome at doing the text preprocessing, though lots of people still use it for that task.

Most stemming approaches are based on the target languages morphological rules (e.g., the Porter stemmer for the English language where suffix removal is also controlled by quantitative restrictions (e.g.,' ing' is removed when the resulting stem has more than three letters as in "jumping," but not in "king") or qualitative restrictions (e.g., '-ize' is removed if the resulting stem does not end with '-e' as in "seize"). Certain ad hoc spelling correction rules can also be applied to improve conflation accuracy (e.g., "running" gives "run" and not "runn"), particularly when phonetic rules are applied to facilitate easier pronunciation.

Another approach consults an online dictionary to obtain better conflation results while Xu & Croft suggest a corpus-based approach that more closely reflects the language use rather than all its grammatical rules. Few stemming procedures1 have been suggested for languages other than English.

There are two famous stemming algorithms in Farsi language:

a. Kazem Taghva algorithm:

This one is like the Porter algorithm in English, which is based on removing the suffix and prefix. Kazem taghva, Russel Beckley and Mohammad Sadeh designed this stemmer in 2005. In this algorithm Farsi language morphology and a BNF machine whit 40 step are used to remove suffix and prefix.

b. Krovetz improved algorithm in Farsi:

The second algorithm is designed by GholamReza Ghasem Sani and Reza Hesamifard . This method is based on the database's information. In the other word all the stems of the language should be saved. At first the input word should be searched in the database, if it is found, the word will be returned as a stem, otherwise the suffixes and prefixes should be removed and it should be searched again in database. This method has some problems. The database needs to be update and also the speed of the stemmer is low.[39]

F. Errors in Stemming:

There are mainly two errors in stemming – over Stemming and under stemming. Over-stemming is when two words with different stems are stemmed to the same root. This is also known as a false positive. Under-stemming is when two words that should be stemmed to the same root are not. This is also known as a false negative. Paice has proved that light-stemming reduces the over-stemming errors but increases the under-stemming errors. On the other hand, heavy stemmers reduce the under-stemming errors while increasing the over-stemming errors .

G. Classification of Stemming Algorithms:

Broadly, stemming algorithms can be classified in three groups *ref fig 3.* truncating methods, statistical methods, and mixed methods. Each of these groups has a typical way of finding the stems of the word variants. [40]

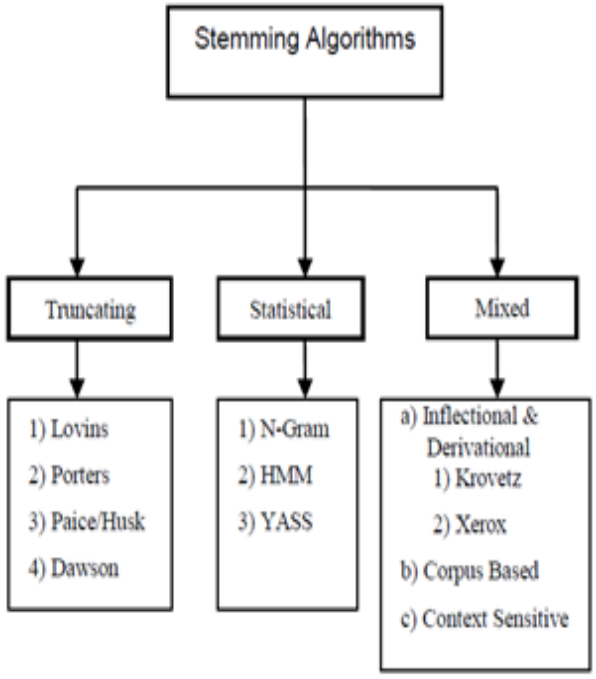

Figure 3: Types of Stemming Algorithms

a. Truncating (Affix Removal) Methods:

a) Lovins Method:

(a). Advantages:

i. Fast – Single pass algorithm
ii. Handles removal of double leters in words like 'getting' being transformed to 'get'.
iii. Handles many irregular plurals like mouse and mice etc.

(b). Limitations:

i. Time consuming.
ii. Not all suffixes available.
iii. Not very reliable and frequently fails to form words from stems.
iv. Dependent on the technical vocabulary being used by the author.

b) Porters Stemmer:

(a). Advantages:

i. Produces the best output as compared to other stemmers.
ii. Less error rate.
iii. The Snowball stemmer framework designed by Porter is language independent approach to stemming.

(b). Limitations:

i. The stems produced are not always real words.
ii. It has at least five steps and sixty rules and hence is time consuming.

c) Paice / Husk Stemmer:

(a). Advantages:

i. Simple form.
ii. Each iteration takes care of deletion and replacement.

(b). Limitations:

i. Heavy algorithm.
ii. Over stemming may occur.

d) Dawson Stemmer:

(a). Advantages:

i. Covers more suffixes than Lovins.
ii. Fast in execution.

(b). Limitations:

i. Very complex.
ii. Lack a standard implementation.

b. Statistical Methods

a) N-Gram Stemmer:

(a). Advantages:

i. Based on the concept of n-grams and string comparisons.
ii. Language independent.

(b). Limitations:

i. Not time efficient.
ii. Require significant amount of space for creating and indexing the n-grams.
iii. Not a very practical method.

b) HMM Stemmer:

(a). Advantages:

i. Based on the concept of Hidden Markov Model.
ii. Unsupervised method and so is language independent.

(b). Limitations:

i. A complex method for implementation.
ii. Over stemming may occur in this method.

c) Yass Stemmer:

(a). Advantages:

i. Based on hierarchical clustering approach and distance measures.
ii. It is also a corpus based method.
iii. Can be used for any language without knowing is morphology.

(b). Limitations:

i. Difficult to decide a threshold for creating clusters.
ii. Require significant computing power.

c. Mixed:

a) Krovetz Stemmer:

(a). Advantages:

i. It is a light stemmer.
ii. Can be used as a pre-stemmer for other stemmers.

(b). Limitations:

i. For large documents, this stemmer is not efficient.
ii. Inability to cope with words outside the lexicon.
iii. Does not consistently produce a good recall and precision.
iv. Lexicon to be created in advance.

b) Xerox Stemmer:

(a). Advantages:

i. Works well for a large document also.
ii. Removes the prefixes where ever applicable.
iii. All stems are valid words.

(b). Limitations:

i. Inability to cope with words outside the lexicon.
ii. Not implemented successfully on languages other then English. Over stemming may occur in this method.
iii. Dependence on the lexicon makes it language dependent.

VIII. PART OF SPEECH TAGGING

A. Definition of POS:

POS is defined as "The Process of assigning a part-of-speech or other lexical class marker to each word in a corpus" (Jurafsky and Martin)

B. POS Tags:

POS Tagging is the annotation of words with the appropriate POS tags based on the context in which they appear. POS tags divide words into categories based on the role they play in the sentence in which they appear. POS tags provide information about the semantic content of a word. Nouns usually denote "Tangible and intangibles things", Whereas prepositions express relationships between "things".[41][42].

Most POS tag sets make use of the same basic categories. The most common set of tags contains even different tags.(Article, Noun, Verb, Adjective, Preposition, Number and Proper Noun),. Some systems contain a much more elaborate set of tags.

Usually, POS taggers at some stage of their processing perform morphological analysis of words. Thus, an additional output of a POS tagger is a sequence of stems (also known as "lemmas") of the input words.

C. Aim of POS Tagging:

Part of Speech Tagging aims to find the part of speech of a word

a. noun, verb, adjective, adverb, conjunction, determiner, pronoun, preposition.
b. Noun: Naming word (cat, dog, Rob, television)
c. Verb: Action word (jump, fish, sell, watch, say)
d. Adjective: Qualifier for noun (big, red, expensive, hungry)
e. Adverb: Qualifier for verb (quickly, noisily, efficiently)

These are 'content' words (and hence ones to keep!)

Other four main types of part of speech:

a. Conjunction: Joining words (and, hence, then)
b. Determiner: Articles (the, a, an)
c. Pronoun: Word that stands for a noun (he, it, them, her)
d. Preposition: 'Direction' words (to, with, in, of)

These are 'function' words (and hence ones to throw away!) But POS taggers have a much larger set of classes.

D. Taggers:

90% of words only have one part of speech. So a simple dictionary lookup will get ~90% accuracy, then need a model to predict the other words. Of the words that have more than one PoS, about ½ are noun/verb, and >80% are noun/verb or noun/adjective. Could build a decision tree tagger using this information, with some simple additional rules for non dictionary words and words with multiple PoS. Brill did this, with some additional training methods to correct errors. Gets 95% accuracy using about 70 rules.

E. POS Tagging Methods:

a) Stochastic Tagger: HMM – based(using Viterbi Algorithm)
b) Rule-Based Tagger : ENGTWOL (English Two Level analysis)
c) Transformation-Based Tagger (Brill)

a. Stochastic Tagging:

a) Based on Probability of certain tag occurring given various possibilities
b) Requires a training corpus.
c) No probabilities for words not in corpus.
d) Simple Method: Choose most frequent tag in training text for each word.
   i. Result : 90% accuracy
   ii. Baseline
   iii. HMM is an example.

(a). HMM Tagger:

a) Intuition: Pick the most likely tag for this word.
b) HMM Taggers choose tag sequence that maximizes this formula:

i. P(word(tag) x P(tag | previous n tags)
ii. Let $T = t_1,t_2,\ldots t_n$
iii. Let $W = w_1,w_2,\ldots w_n$
iv. Find Pos tags that generate a sequence of words i.e., look for must probable sequence of tags T underlying the observed words W.

b. Rule Based Tagging:

a) Assign all possible tags to words.
b) Remove tags according to set of rules of type:

i. If word +1 is an adj, adv, or quantifier and the following is a sentence boundary and word-1 is not a verb like "consider" then eliminate non-adv else eliminate adv.

c) Typically more than 1000 hand written rules, but may be machine learned.

c. Transformation-Based Tagging:

a) Combination of Rule-based and Stochastic tagging methodologies

i. Like rule-based because rules are used to specify tags in a certain environment.
ii. Like stochastic approach because machine learning is used-with tagged corpus as input.

b) Input:

i. Tagged corpus
ii. Dictionary (with most frequent tags)

(a). Usually constructed from the tagged corpus.

IX. CONCLUSION

At last we conclude that, Text mining is also known as Text Data Mining or Knowledge-Discovery in Text, refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. This survey presented the useful information about the data mining, text mining and a deep flow on preprocessing steps.

X. REFERENCES

[1]. Osmar R.Zaiane, "Introduction to Data Mining" CMPUT690 Principles of Knowledge Discovery in Databases,1999.

[2]. Anupriya and Ashok Kumar, Ph.D. " Analysis on Parallelization of Apriori Algorithm in Data Mining", International Conference on Advances in Computer Application (ICACA - 2013) Proceedings published in International Journal of Computer Applications® (IJCA) (0975 – 8887).

[3]. Neelamadhab Padhy, Dr. Pragnyaban Mishra , and Rasmita Panigrahi, "The Survey of Data Mining Applications And Feature Scope", International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol.2, No.3, June 2012.

[4]. Jiawei rlan and Micheline Kamber, "Data Mining: Concepts and Techniques ",Second Edition ,APR 2012.

[5]. Aloka Arora, "Introduction to Data Mining", Indian Agricultural Statistics Research Institute, New Delhi, Jan 2012, pp 163-170 .

[6]. T. Sunil kumar and Dr. K. Suvarchala, "A Study: Web Data Mining Challenges and Application for information Extraction", IOSR Journal of Computer Engineering (IOSRJCE), ISSN: 2278-0661, ISBN: 2278-8727Volume 7, Issue 3 (Nov. - Dec. 2012), PP 24-29

[7]. "Overview of Data Mining",IASRI, Winter School on "Data Mining Techniques and Tools for Knowledge Discovery in Agricultural Dastasets", iasri.res.in/ebook/win_school_aa/notes/DATA_MINING.pdf.

[8]. Abraham Silberschatz, Henry F. Kort , "Database system concepts – ",fifth Edition, Apr 2011.

[9]. Peter Brezeny, "Data Mining – Introduction", Institute for Scientific Computing, Universität Wien, Sep 2010. SOUMEN Chakrabarti, "Mining the Web: Discovering Knowledge from Hypertext Data, Part 2", Morgan Kaufmann, 2003.

[10]. Raorane A.A., Kulkarni R.V. and Jitkar B.D., "Association Rule – Extracting Knowledge Using Market Basket Analysis", Research Journal of Recent Sciences , ISSN 2277-2502, Vol. 1(2), 19-27, Feb. (2012) Res.J.Recent Sci..

[11]. Pieter Adriaans, "Data Mining",first edition, Pearson (2002) .

[12]. Sankari. A, "USING DATA MINING TECHNIQUES FOR BUSINESS INTELLIGENCE IN WEB MINING", International Journal of Computer Science and Management Research Vol 1 Issue 2 September 2012 ,ISSN 2278-733X.

[13]. Aakanksha Bhatnagar, Shweta P. Jadye and Madan Mohan Nagar, "Data Mining Techniques & Distinct Applications: A Literature Review", International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 9, November- 2012 ISSN: 2278-0181.

[14]. Dr. S. Santhosh Baboo and S. Sasikala "A Survey on Data Mining Techniques for GeneSelection and Cancer Classification", IJCSIS) International Journal of computer Science and Information Security,Vol. 8, No. 1, April 2010.

[15]. Osmar R. Zaïane," Principles of Knowledge Discovery in Databases", CMPUT690, University of Alberta, 1999.

[16]. Jeaffrey W. Seifert, "Data Mining: An Overview", Report for Congress, Penny Hill Press, May 2003.

[17]. Jeffrey W. Seifert, "Data Mining and Homeland Security: An Overview", CRS Report for Congress Received through the CRS Web, Order Code RL31798, January , 2006

[18]. Gurjit Kaur, Lolita Singh, "Data Mining: An Overview", IJCST, Vol 2 , Issue 2, June 2011, SSN : 2229-433.

[19]. Margaret H Dunham, "Data Mining: Introductory And Advanced Topics", First edition, Pearson (2008) .

[20]. Tripti Arjariya, Shiv Kumar, Rakesh Shrivastava, Dinesh Varshney," Data Mining and It's Approaches towards Higher Education Solutions", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-5, November 2011.

[21]. Scribd , "Data Mining", Weka Tutorial, Apr 2013.

[22]. Marti Hearst, "What Is Text Mining?", SIMS,UC Berkeley, hearst@sims.berkeley.edu ,October 17, 2003.

[23]. Jonathan Clark," Text Mining and Scholarly Publishing", Publishing Research Consortium 2013.

[24]. Shirsendu Patra, "Mining frequent Key-words and Key-phrases using Apriori algorithm on horizontal data format", Jadavpur University,2010.

[25]. Ah-Hwee Tan, "Text Mining:The state of the art and the challenges", July 1997.

[26]. JISC, "Text Mining", National Centre for Text Mining and produced and edited by Judy Redfearn and the JISC Communications team, Sep 2008.

[27]. Yagmur Sengez, "Text Mining", Docstoc, Feb 2010.

[28]. Shaidah Jusoh and Hejab M Al Fawareh, "Agent-based Knowledge Mining Architecture", 2009 International Conference on Computer Engineering and Applications IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore.

[29]. Mrs. Radha Shakarmani, Nikhil Kedar and Naman Khandelwal, "Performance Assessment using Text Mining", 2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 12.

[30]. Tom Magerman, Bart Van Looy, Bart Baesens and Koenraad Debackere, "Assessment of Latent Semantic Analysis (LSA) text mining algorithms for large scale mapping of patent and scientific publication documents", Department of Managerial economics, strategy and innovation(MSI), Oct 2011.

[31]. Yanbo J. Wang, Frans Coenen and Robert Sanderson, "A Hybrid Statistical Data Pre-processing Approach for Language-Independent Text Classification", Advanced Data Mining and Applications Lecture Notes in Computer Science Volume 5678, 2009, pp 338-349.

[32]. V. Srividhya, R. Anitha, "Evaluating Preprocessing Techniques in Text Categorization", International Journal of Computer Science and Application Issue 2010, ISSN 0974-0767.

[33]. Marcus Hassler, G¨unther Fliedl "Text Preparation through Extended Tokenization" Alps-Adria University Klagenfurt.

[34]. Dipl.-Ing. Mag. Marcus Hassler, "Linguistically Enhanced Information Retrieval of Structured Documents", March 2009.

[35]. Ronen Feldman, James Sanger, "The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data", Cambridge University Press 2006.

[36]. Recruitment Technology - for tomorrow: "Text mining – Stop Words", Aug 2009, recruitmenttechnology. blogspot.com/.../text-mining-stop-words-are-ever.

[37]. A. Anil Kumar and S.Chandrasekhar, "Text Data Pre-processing and Dimensionality Reduction Techniques for Document Clustering", International Journal of Engineering Research & Technology (IJERT) ,Vol. 1 Issue 5, July – 2012 ISSN: 2278-0181.

[38]. Mr.Ben Piché "Text Mining, R, and Stemming", Posted in Natural Language Processing on March 3rd, 2013.

[39]. Ms. Anjali Ganesh Jivani "A Comparative Study of Stemming Algorithms", Anjali Ganesh Jivani et al, Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938.

[40]. Naveen Kumar, , Saumesh Kumar, and Padam Kumar , "Parallel Implementation of Part of Speech Tagging for Text Mining Using Grid Computing", Advances in Computing and Communications , Communications in Computer and Information Science Volume 190, 2011, pp 461-470.

[41]. Antony P J and Dr. Soman K P, "Parts Of Speech Tagging for Indian Languages: A Literature Survey", International Journal of Computer Applications (0975 – 8887) Volume 34– No.8, November 2011.