CHAPTER 4

PREPROCESSING AND FEATURE EXTRACTION

4.1 INTRODUCTION

Due to the enormous growth of data into the form of rationally numeric data, graphical web data and textual data, controlling and handling of these data are the very challenging task in the recent scenario. The issues that occur in handling these data are information overload, semantically complex documents, unstructured view and high number of possible dimensions. These issues tend to reduce the processing efficiency. Most of the researches fulfill the above issues of preprocessing techniques to reduce the dimensionality of the feature space. This research framework consists of two steps such as preprocessing and feature selection. Preprocessing technique transforms implicit structured textual data into an explicit database to facilitate the document clustering. In model construction, the feature extraction is the process of selecting predictors and variables. It is used for model simplification, enhanced abstraction, and reduced conflicting facts. These techniques are applied in online news corpus data, such as Reuters-21578, 20NG, and TDT-2 dataset.

4.2 PREPROCESSING

Pre-processing is used to transform text documents into simple word format. The preprocessing methods (Srividhya V. and Anitha R. 2010) are used
to achieve the common goal of performance, optimal quality and the context of subsequent clustering. Initially, text documents are processed as a normal string, and then the sequences of string terms are divided into a simple tokenized list of strings. It is helpful to convert the content within a document into a sequence of terms like words or phrases. The tokenized text data are applied to the two processes, namely StopWord elimination and word Stemming (Ganesh A.J. 2011).

These processes are used to remove unwanted entities and words from the document. It refines the feature vector while keeping its “rich” features as original as possible. Before clustering, text document’s feature selection has been applied to many real time domains such as Bioinformatics, image processing, and pattern recognition. The features are extracted and validated based upon the storage size, terms, over fitting, non-linear hypothesis spaces, irrelevant spaces, document dense and dimensionality. The problem in feature selection is reduced by dimensionality reduction techniques. It is used to reduce dimensionality between the text documents and enhance feature selection. The feature identification (Gabrilovich E. and Markovitch S. 2006) is useful to improve the accuracy (Sirsat S.R. et al. 2013) and model interpretability with reduced computational cost.

In addition, the text documents have great quantities of semantically corrected words as well as basic functional standards called StopWord. The StopWord lists consist of words such as “a”, “but”, “can”, “each”, “for”, “gives”, “here”, “in”, “of”, “the” and so on. These insignificant words that are eliminated from the document are called StopWord elimination or term filtering method. Moreover, it is not necessary to extract all features from the original corpus vocabulary in the data model. The term filtering method is used to improve the speed and memory consumption of text clustering.
Figure 4.1 Flowchart for Preprocessing and Feature selection
After removing, StopWord from the text documents, the Stemming process is used to reduce the number of unique terms and convert words to their stem or root form. For example, stemming algorithm (Lovins J.B. 1968), (Thangarasu M. et al. 2013) converts different word (“connection”, “connections”, “connective”, “connected”, and “connecting”) form into a similar canonical form (“connect”). The porter suffix algorithm is a standard procedure used to eliminate common morphological and inflectional endings such as “ing”, “ive” to improve the efficiency.

The figure 4.1 gives the overview of proposed preprocessing and feature extraction system. The inputs into the system are Online News Corpus such as Reuters-21578, 20 NG, and TDT-2 Dataset. The output is the preprocessed and feature selected data. The proposed system performs the process of preprocessing in three main steps (1) BOW: It selects the relevant word from the dataset corpus; (2) StopWord Elimination: It eliminates the irrelevant sentences in each document; (3) Stemming: It reduces the suffixes from the input terms. These steps are performed effectively in preprocessing. Given the inputs of preprocessing, the dimensionality reduction is applied to the dataset corpus for feature selection. After that, the Term Frequency and Inverse Document Frequency are calculated. The term significant weight produces term frequency. The term discriminating weight produced a large weight term. In the last two steps of TF/IDF, the Proposed DFT is applied in the dimensionality reduction process. Finally, the feature selected documents are retrieved. This term generated documents are applied to the document clustering.
4.2.1 Bag of Words (BOW)

BOW is a simplified representation used in natural language processing and information retrieval. Bag of Word is a simplest method for feature identification. BOW process consists of the following steps,

*Step 1:* Each document is indexed with the bag of the terms, i.e., a vector with one document for each term occurring in the whole collection. Each vector has a corresponding value (i.e.) the number of times the term occurred in the document.

*Step 2:* Each document is thus represented as a point in a vector space with one dimension for every term in the vocabulary.

*Step 3:* When a word does not appear in a given document, that particular vector is set to zero.

In document clustering process, huge number of documents that are processed by applying the standard “Bag of Words”, which order the words based on the relevance and then words are grouped in such a way that the frequent terms may be removed for maintaining the size of the corpus vocabulary (Martins C.A. et al. 2003). The vector space model is one of the common procedures to maintain bag of words for document clustering; this representation is much useful to represent the content in the same manner as the text itself does.
The process of arranging ambiguous and orthogonal less representation is called bag of words model (Boulis C. and Ostendorf M. 2005), that represents a textual document for information retrieval (Bhamidipati. N.L. and Pal K.S. 2007). Ambiguity is a process of ignoring the fact that different words can have the same meaning while the same words might have different meanings in different contexts. Orthogonal is the process of forming meaningful lexical structures and utilize relations between words to facilitate understanding against the isolated language units.

To achieve effective clustering, bag of words model is used when text documents are unambiguous and statistically related. Texts with different characteristics are abstracted into different bags, even though it might be referred using the same actual text expression. For example, the keyboard can be referred as a computer input device, musical devices, cell phone keypad, and key hanging board and each meaning is represented in a different context. Contexts have relations with each other; because they are unambiguous. Their relations can be explicitly defined.

Consider seven different text documents that are useful to differentiate the bag of words (Boulis C. and Ostendorf M. 2005), process based on the term frequency of occurrence in the text documents.

Document 1: As an example of clustering, early in Childhood, is learnt to differentiate between cats and dogs.

Document 2: Similarly, to differentiate animals and plants by continuously improving clustering schemes.
**Document 3:** The requirements of clustering in data mining include scalability, ability to deal with different attributes and the ability to deal with noise data.

**Document 4:** Some other requirements are the ability to deal with high dimensionality and constraint-based clustering.

**Document 5:** A few more requirements are interoperability and usability.

**Document 6:** To form different clusters as discussed above, different clustering algorithms have to be applied to the data objects.

**Document 7:** Different clustering algorithms includes exclusive and hierarchical structures.

These seven text documents can be represented by a Co-occurrence matrix as shown in Table (4.1).

<table>
<thead>
<tr>
<th>Individual Texts In the Document</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>Result based on the bag of word</th>
<th>Result based on the StopWord</th>
<th>Result based on the stemming/distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>20</td>
<td>6 (ivy)</td>
</tr>
<tr>
<td>Algorithms</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>2 (s)</td>
</tr>
<tr>
<td>Animals</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>10</td>
<td>2 (als)</td>
</tr>
<tr>
<td>Applied</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1 (ed)</td>
</tr>
<tr>
<td>Attributes</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>13</td>
<td>1 (es)</td>
</tr>
<tr>
<td>Cats</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>7</td>
<td>1 (s)</td>
</tr>
<tr>
<td>Childhood</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Clustering</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>54</td>
<td>7 (ing)</td>
</tr>
<tr>
<td>Constraint</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Continuously</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>10</td>
<td>1 (ous)</td>
</tr>
<tr>
<td>Data</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>22</td>
<td>3 (s)</td>
</tr>
<tr>
<td>Deal</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>Different</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>40</td>
<td>16 (ent)</td>
</tr>
<tr>
<td>Dimensionality</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>0</td>
<td>7 (ity)</td>
</tr>
<tr>
<td>Discussed</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>0</td>
<td>9</td>
<td>1 (ed)</td>
</tr>
<tr>
<td>Dogs</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Early</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Exclusive</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1 (ive)</td>
</tr>
<tr>
<td>Form</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>Hierarchical</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>1 (ical)</td>
</tr>
</tbody>
</table>

**Table 4.1 Co-occurrence matrix**
In information retrieval, Term Frequency/Inverse Document Frequency (Sirsat S.R. et al. 2013) is used to find out the bag of the words. In addition, bag of words (Salton G. and McGill M.J. 1983) is also referred as the “vector space model (VSM)” which is used for representing the documents in clustering problems, information retrieval and text classification (Ikonomakis M. et al. 2005), (Korde V. and Mahender N.C. 2012). The above seven text document phrases are expressed indirectly by following vector space model equation.

Consider all the available words in the above seven text documents,

\[ \text{Word(Text) as Word}_{n \to n-1} \ (\text{Text}) \quad \text{where n-1 is the first word in the text and n is the final word in the text.} \]

Then, Bag of Words is computed by following equation.

\[
\text{Text(Document)}_{\text{Nxm}} = \sum_{n-1}^{n} \text{Word}_{n \to n-1}(\text{Text})
\]
Here, \( M \) is Number of unique term in the text document and \( N \) is the Total Number of term available in the text (Lan M. et al. 2007). It is used to select the word based on the equation (4.2)

\[
\text{SelectWord}_{n \leq N \leq n+1}(\text{Text})
\]

(4.2)

Where “\( n \)” indicates the number of text document phrases in the corpus and \( \text{Text}_{(N \times M)} \) indicates the frequency of occurrence of the term in text document. Once an entire corpus of “\( N \times M \)” document has been transformed to a corresponding set of feature vectors \{\( \text{Text} (1)_{(N \times M)} \), \( \text{Text} (2)_{(N \times M)} \), \( \text{Text} (3)_{(N \times M)} \)\}, the document can be clustered based on the similarities between vectors. For convenience, the complete model is usually stored as a single term-document matrix,

\[
S = [\text{Text}(1)_{(N \times M)}, \text{Text}(2)_{(N \times M)}, \text{Text}(3)_{(N \times M)}, \ldots, \text{Text}(i)_{(N \times M)}] X [\text{InfoRet}_{(N \times M)}]
\]

(4.3)

From the equation (4.3), BOW approach uses features, constructed from phrases or n-grams, which contain a sequence of “\( n \)” consecutive characters that are extracted from text strings. Instead of six “Clustering” words three different word phrases called “bag of word” is formed to extract only one word “clustering”. These extracted optimized “bag of word” processes are applied to StopWord elimination.

4.2.2 StopWord Elimination Process

The StopWord removal and stemming are cognitive factors for the final document clustering results. Before feature identification, document set is normally cleaned by removing StopWord and then by applying stemming algorithm (Lovins J.B. et al. 1968) that converts different word forms into a
similar canonical form. These StopWord removal methods are described by the following procedure.

Function words are the words that are not used to identify information about its topic, but it plays an essential role in word sentence structure formation (e.g. a combination of “will”, “have”, “and”, “but”, “can”, “an”). Those words are very useful, but it does not provide meaning by themselves when it is available separately. During text preprocessing, these words should be avoided because they are meaningless, but occur frequently in a sentence structure. In addition, some words are used to make strong sentences some words such as “and”, “of”, “the” are used to make strong sentences. These words are excluded from elimination process. This type of problem is resolved by using the weighting average process. Finally, these function words are listed based on different dataset as shown in Table 4.1.

<table>
<thead>
<tr>
<th>A</th>
<th>before</th>
<th>Down</th>
<th>her</th>
<th>Into</th>
<th>not</th>
<th>she</th>
<th>then</th>
<th>very</th>
<th>Who</th>
</tr>
</thead>
<tbody>
<tr>
<td>About</td>
<td>being</td>
<td>During</td>
<td>here</td>
<td>Is</td>
<td>of</td>
<td>she'd</td>
<td>there</td>
<td>was</td>
<td>Whom</td>
</tr>
<tr>
<td>above</td>
<td>below</td>
<td>Each</td>
<td>here’s</td>
<td>isn't</td>
<td>off</td>
<td>she'll</td>
<td>there's</td>
<td>wasn't</td>
<td>who's</td>
</tr>
<tr>
<td>after</td>
<td>between</td>
<td>Few</td>
<td>her’s</td>
<td>It</td>
<td>on</td>
<td>she's</td>
<td>these</td>
<td>we</td>
<td>Why</td>
</tr>
<tr>
<td>again</td>
<td>both</td>
<td>For</td>
<td>herself</td>
<td>its</td>
<td>once</td>
<td>should</td>
<td>they</td>
<td>we'd</td>
<td>why's</td>
</tr>
<tr>
<td>against</td>
<td>but</td>
<td>From</td>
<td>he's</td>
<td>it's</td>
<td>only</td>
<td>shouldn't</td>
<td>they'd</td>
<td>were</td>
<td>won't</td>
</tr>
<tr>
<td>all</td>
<td>by</td>
<td>further</td>
<td>him</td>
<td>itself</td>
<td>or</td>
<td>so</td>
<td>they'll</td>
<td>we're</td>
<td>Would</td>
</tr>
<tr>
<td>am</td>
<td>cannot</td>
<td>Had</td>
<td>himself</td>
<td>i've</td>
<td>other</td>
<td>some</td>
<td>they're</td>
<td>weren't</td>
<td>wouldn't</td>
</tr>
<tr>
<td>an</td>
<td>can't</td>
<td>hadn't</td>
<td>his</td>
<td>let's</td>
<td>ought</td>
<td>such</td>
<td>they've</td>
<td>we've</td>
<td>You</td>
</tr>
<tr>
<td>and</td>
<td>could</td>
<td>Has</td>
<td>how</td>
<td>me</td>
<td>our</td>
<td>than</td>
<td>this</td>
<td>what</td>
<td>you'd</td>
</tr>
<tr>
<td>any</td>
<td>couldn't</td>
<td>hasn't</td>
<td>how's</td>
<td>more</td>
<td>ours</td>
<td>that</td>
<td>those</td>
<td>what's</td>
<td>you'll</td>
</tr>
<tr>
<td>are</td>
<td>did</td>
<td>Have</td>
<td>I</td>
<td>most</td>
<td>ourselves</td>
<td>that's</td>
<td>through</td>
<td>when</td>
<td>Your</td>
</tr>
<tr>
<td>aren't</td>
<td>didn't</td>
<td>Haven't</td>
<td>i'd</td>
<td>mustn't</td>
<td>out</td>
<td>the</td>
<td>to</td>
<td>when's</td>
<td>you're</td>
</tr>
<tr>
<td>as</td>
<td>do</td>
<td>Having</td>
<td>If</td>
<td>my</td>
<td>over</td>
<td>their</td>
<td>too</td>
<td>where</td>
<td>Yours</td>
</tr>
<tr>
<td>at</td>
<td>does</td>
<td>He</td>
<td>i'll</td>
<td>myself</td>
<td>own</td>
<td>theirs</td>
<td>under</td>
<td>where's</td>
<td>Yourself</td>
</tr>
<tr>
<td>be</td>
<td>doesn't</td>
<td>he'd</td>
<td>i'm</td>
<td>No</td>
<td>same</td>
<td>them</td>
<td>until</td>
<td>which</td>
<td>Yourselves</td>
</tr>
<tr>
<td>because</td>
<td>doing</td>
<td>he'll</td>
<td>In</td>
<td>nor</td>
<td>shan't</td>
<td>themselves</td>
<td>up</td>
<td>while</td>
<td>you've</td>
</tr>
</tbody>
</table>

Table 4.2 English Stop-list or Stop-words

The StopWord are listed based on the English StopWord (Table 4.2), MySQL StopWord (“as”, “between”, “consider”, “each”, “further”, “however”,

63

Figure 4.2. Illustrates StopWord Elimination Process. Some words in the above documents repeat more frequently, which is not useful in the process of text clustering. These StopWord are normally excluded from the set of index terms. Especially articles, prepositions, and pronouns are skipped during StopWord elimination. StopWord are used to understand or disambiguate the meaning of a phrase. In a document clustering process, it is generally agreed that a lot of words are indexed and irrelevant in huge number of texts. The filtering process is a process of removing irrelevant words that reduce the size as well as remove the noise from the dataset.
4.3 WORD STEMMER ALGORITHMS

In this research, Stemming is performed by suffix elimination from the text data using a Porter Stemming algorithm that is applied in information system based electronic library. The stemming is applied to text to reduce, irrelevancy, repentance, misspelled, grammatical and unwanted text. The Stemming process reduces text file size up to 40-50%. For e.g., the words, “Apply”, “Applied”, “applies”, and the words “Clustering”, “clustered”, “clusters” are reduced to a common representation “Appli” and “cluster” accordingly. The word “Appli” and “cluster” is the root of the above mentioned words. Stemming process converts words to their root or stem form. In natural language, English words such as “Appli” and “cluster” are stemmed by using the Porter Stemmer algorithm.

4.3.1 Porter Stemmer Algorithms

This algorithm is used to find out the root of a particular set of words by removing various suffixes but preserves the meaning of the word. This is useful to save memory space and time. This stemming algorithm is applied in above seven text documents which is given in Table 4.1.

4.3.2 Stemming approaches

This process is also referred as the morphological root of the word (Levent A. et al. 2009).
Pseudocode:
node ← Start
Initialize s, n, n' O,
s ← unexpected node;
 n ← before remove node;
n' ← After remove node;
O ← Overlay;
if S = null
   Expanded = nil
else if remove node (n) = not defined;
   Final text word = max (final text word (remove node))
   Condition > suffix → new_suffix;
   Condition > prefix → new_prefix;
   Removed n;
else if stem_content = input_word;
   Overlay: Overlay(source_word) = dictionary_source by input_word;

\[
\text{Stem (Overlay)} = \sum_{\text{initial input}}^{\text{input word} - 1} \left( \text{dictionary}_{\text{source}} \mid \text{input}_{\text{word}} \right)
\]

stem_content = Stem (Overlay)
else if input is noun
   stem_content = Stem (Overlay)
else if input is Verb
   Words (removed) = Overlay(source_word);
   display:stem_content = Stem (Overlay) - Overlay(before remove node(n));
else if input is Ad_Verb
   Words (removed) = stem_content(After remove node(n'))
   display: stem_content = Stem (Overlay) - Overlay(after remove node(n));
else not expand to the word as stemming
   go to feature selection

node ← End

Figure 4.3 Pseudocode for Stemming Process
If the word ends in “ed”, “ing”, “ly”, then it is removed from the stemming process. Then, “Ability” becomes “abil”, “Applied” becomes “appli” and so on. HTML tags, non-alpha characters, and unnecessary items are removed in a document. Stemming process is applied in each document in the test dataset, Reuters 21578, 20 NG, and TDT-2 Datasets. Reuters - 21578 is a collection of 21578 documents and 20 NG is a collection of 20,000 documents. There are 292 StopWord, punctuation, and numbers are removed from both document sets. Then, Porter’s stemming is done to reduce the words in their base forms in which it has 15,937 words in vocabulary. Hence, the related terms are reduced (Gupta M. and Rajavat M. 2014) before feature selection. Stemming algorithm (Lovins J.B. et al. 1968) is based on the idea that suffixes are mapped to the morphological terms in English language (approximately 1200) that are most reduced with respect to the following pseudo code in figure 4.3. This resultant stem words is input to feature selection.

4.3.3. Porter Stemmer Steps

There are a number of languages that are stemmed by different stemming concepts (Ramasubramanian C. and Ramya R. 2013). In English language, the Porter's original English stemmer is used as a Stemming algorithm (Thangarasu M. et al. 2013) to reduce the verb, noun, and other basis of normalization. It has the following steps.

Step 1: Eliminate plurals (-s) and suffixes (-ed or -ing).
Step 2: If the vowel occurs in the previous step, replace y to i on the next word.
Step 3: From the step 3, Map double suffixes to single ones (-ization, -ational).
Step 4: Additionally, reduces the suffixes like (-full, -ness) etc.
Step 5: Deducts (-ant, -ence) etc.
Step 6: If a word ends with a grammatical verb ending, then it has been removed.
Step 7: Finally, removes a (-e).

Figure 4.4 Porter Stemmer Stemming

After stemming, filtering is one of the crucial steps to remove vagueness and other vocabularies in document for effective clustering. Stemming process is given in the figure 4.4. It reduces the consumable amount of terms. From Table 4.1, the word “clustering” has the same frequency in all documents. If document features are increased, these terms are not discriminatory features for any documents. As a consequence, they do not contribute to the separation of the documents. Increased separation helps to distinguish similar documents. To overcome the issue of ambiguity and orthogonal with non-discriminatory features, it is necessary to apply a more delicate weighting scheme.
In Information Retrieval, Term Frequency and Inverse Document Frequency \textit{(Franca D. and Fabrizio S. 2003)} are used to find out the BOW. The bag of words \textit{(Salton G. and McGill M.J. 1983)} is also referred as the “Vector Space Model (VSM)” which is used for representing documents in clustering problems, information retrieval and text classification \textit{(Yu B. et al. 2008), (Wen P. et al. 2007)}. During this information retrieval, seven text document phrases (Document 1, Document 2… Documents 8) are carried to the feature weighting and feature selection.

4.4 FEATURE WEIGHTING AND FEATURE SELECTION

Feature selection is a crucial process in document clustering, which makes better accuracy, efficiency, and scalability of a text document compared to the other techniques. There are many methods available to group the text documents such as information gain, mutual information, term frequency, Chi-square process, cross entropy, the term weighting methods of text \textit{(Nivet C. et al. 2010)}, index based process. Among these methods, Information Gain, TF, DF and IDF, Chi-square (Statistical term and entropy based), term weighting methods \textit{(Franca D. and Fabrizio S. 2003)} TSW and TDW are useful methods to manage the feature selection process. The enhanced TF/IDF is used for dimensionality reduction \textit{(Lan M. and Tan C.L. 2007)} of text documents.
The feature selection and weighting methods (Franca D. and Fabrizio S. 2003) contain the following steps:

**Step 1: Describing ability of text term based on Term Frequency**

In text mining, describing the ability of the text term based on Term Frequency uses the Vector Space Model (Salton. G. and McGill. M.J. 1983). The document vector “d” is represented by,

\[ d = \{Term \times Freq_1, Term \times Freq_2, \ldots, Term \times Freq_n\} \]  

(4.4)

Where \( i = \{1, 2, \ldots, n\} \) is the term frequency for whole documents. Depending on the Vector Space Model, the weight matrix is calculated by using the matrix derivation.

**Step 2: To check the availability of particular terms in documents based on the Document Frequency.**

From the above concession, it is necessary to identify the term frequency in documents. For example, sentence based text document contains 100% functional and non-functional words. And the word “cluster” occurs frequently in documents, so the term frequency is high for the current documents. The terms that occurs in every document collection does not help to distinguish the documents from each word. Hence, term “clustering” is less valuable. To avoid these types of drawbacks, term frequency scheme focuses on the whole text document to calculate term frequency and document frequency using TF/IDF.
Step 3: To identify a particular text term difference in the text documents based on Inverse Document Frequency.

The inverse frequency / inverse document frequency is given in equation (4.5). The Term Frequency (TF) for each word is normalized by the inverse document frequency that helps to find TF/DF of the documents. Mathematically, the TF/DF describes coordinates of the term weights that are given by the term frequencies. The concept of finding term frequency and Document frequency for a given collection of documents is to calculate Cosine similarity for all vector space models. The term frequency and inverse document frequency representation of term weights defined by (Sirsat. S.R. et al. 2013) the cosine similarity equation is as follows

$$\text{CoSim}(Q,D) = \frac{\sum_i w_{Q,i} w_{i,j}}{\sqrt{\sum_i w_{Q,i}^2} \cdot \sqrt{\sum_i w_{i,j}^2}}$$  \hspace{1cm} (4.5)

Where Q refers to the query of term frequency, “I” refer to inverse document frequency, “w” refers to weight, “j” refers to the term frequency, “D” refers to the document vector.

This ratio defines the vectors in the values between 0 and 1. The cosine similarity, Sim (A, B), or CoSim are commonly used. After calculating the term frequency and document term frequency, calculate the weighted frequencies with the help of normalization function such as CoSim and the weight function such as TF and IDF.
A. Measuring document similarity and term weighting matrix

The term weighting matrix can be calculated in two ways such as a term vectors comparison of documents using Cosine Similarity and term weighting using TF/IDF. Assume that seven different documents set with average document size contains approximately 25 terms in each document. BOW and TF/IDF are computed by the representation of data dimensionality using term weighting matrix and Semantic weighting. It gives meaningful representation in terms of term weighting and semantic weighting.

The following steps are needed to compute the bag of words and TF/IDF

a. The phrases pair according to the similarity.
b. Measure the metrics value based on the document Queries.
c. Calculate term frequency, binary term weight between 0.0 and 1.0.
d. Calculate the term frequency and the inverse document frequency.
e. Compare the binary term weight with term frequency.
f. Cluster the document with respect to the appropriate cluster text with their TF/IDF.

| Term 1: Clustering | Document 1: As an example of clustering, early in Childhood, we learn to differentiate between cats and dogs. |
| Term 2: Data | Document 2: Similarly we learn to differentiate between animals and plants by continuously improving subconscious clustering schemes. |
| Term 3: Different | Document 3: The requirements of clustering in data mining include scalability, ability to deal with different attributes and the ability to deal with noise data. |
| Term 4: Ability | |
| Term 5: Deal | |
Some other requirements are ability to deal with High dimensionality and constraint-based clustering.

And a few more requirements are Interoperability and usability.

To form different clustering discussed above, different clustering algorithms have to be applied to the data objects.

Different clustering algorithms include exclusive and hierarchical.

The phrase \( p=2 \)

<table>
<thead>
<tr>
<th>P1: ability to deal</th>
<th>P2: different clustering</th>
</tr>
</thead>
</table>

Term document matrix

The requirements of clustering in data mining include scalability, ability to deal with different attributes and the ability to deal with noise data.

Some other requirements are ability to deal with High dimensionality and constraint-based clustering.

**Figure 4.5 Term Matrix Calculation**

**Step 3.1: The two documents pairing according to the similarity**

Documents pairing is done using the term ‘weighting process’. This weighting is measured by using the degree of relationship between term and document. Based on Vector Space Model, \( a_{ij} \) represents the degree of relationship between term i and document j.
Step 3.2: The metrics evaluation based on the document Queries:

<doc1 ID=id1> as an example of clustering, early in Childhood, we learn to differentiate between cats and dogs</doc1>
<doc2 ID=id1> similarly we learn to differentiate between animals and plants by continuously improving subconscious clustering schemes</doc2>
<doc3 ID=id1> The requirements of clustering in data mining include scalability, ability to deal with different attributes and the ability to deal with noise data.</doc3>
<doc4 ID=id1> some other requirements are dealt with High dimensionality and constraint-based clustering.</doc4>
<doc5 ID=id1> And a few more requirements are Interoperability and usability.</doc5>
<doc6 ID=id1> to form different clustering as discussed above, different clustering algorithms have to be applied to the data objects.</doc6>
<doc7 ID=id1> Different clustering algorithms includes exclusive and hierarchical structure</doc7>

Step 3.3: Calculate real value between 0.0 and 1.0, binary term weight

The degrees of relationship are calculated using a term weighting. It is defined as the distance between a term and a document (relation between those
terms). Consider $t_{ij}$ is the degree of relationship between term $i$ and document $j$ of Vector Space Model. While building the cluster document, documents that match the query are considered. This is formed as a group after the document matching query is identified.

The simplest association of binary weight is formed by the following lemma,

**Lemma 1: Key word occurs in the document, $t_{ij}$ is 1. Key word does not occur in the document, $t_{ij}$ is 0.**

Based on lemma 1, the binary weighting is formed using distance calculation. The following equation is used to measure the distance between the terms and individual documents.

$$t_{ij} = \sum_{i,j=1}^{t_n D_n} \left[ \text{document}_j(\text{term}_i(n)) \right]$$  \hspace{1cm} (4.6)

$t_n$ is the number of terms, $n$ varies from the entire index term count. $D_n$ is the number of terms, $n$ varies from entire document count.

**Step 3.4: Comparing the binary term weight with the term frequency**

The binary weighting produces the term frequency weighting with respect to the frequency itemset. In Table 4.3, $t_{ij}$ varies from term 1 to 5 and documents varies from 1 to 7.
The above ‘binary term weight with the term frequency’ values are calculated by two phrases such as “ability to deal” (P1) and “different clustering” (P2). From these phrases, Consider, term “clustering” that occurred in all documents frequently, exclude document 5. It indicates the absence with 0. This will be calculated from equation (4.6). After that, the final binary matrix of the above table is

\[
t_{ij} = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0
\end{bmatrix}
\]

This matrix represents the binary distance between the term and documents. Calculate TF/IDF weighting matrix using local and global term occurrences in the documents.
Step 3.5: Calculate the term frequency and the inverse document frequency

Local and global term is used to balance particular documents and find the TF/IDF weighting matrix

$$w_{i,D} = \left( tf_{i,j} \ast \log \left( \frac{N}{df_i} \right) \right)$$  \hspace{1cm} (4.7)

$tf_{i,j}$ is a term frequency and $df_i$ is document frequency. It is used to produce TF/IDF weight.

<table>
<thead>
<tr>
<th>Distance (df_i)</th>
<th>Distance (df_Clustering)</th>
<th>Distance (df_Data)</th>
<th>Distance (df_Difference)</th>
<th>Distance (df_Ability)</th>
<th>Distance (df_Deal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$\log \left( \frac{N}{df_i} \right)$</td>
<td>$\log \left( \frac{1}{6} \right)$</td>
<td>$\log \left( \frac{1}{2} \right)$</td>
<td>$\log \left( \frac{1}{5} \right)$</td>
<td>$\log \left( \frac{1}{3} \right)$</td>
<td>$\log \left( \frac{1}{2} \right)$</td>
</tr>
<tr>
<td></td>
<td>0.067</td>
<td>0.544</td>
<td>0.146</td>
<td>0.308</td>
<td>0.544</td>
</tr>
</tbody>
</table>

Table 4.4 Inverse Document Frequency

$df_i$ represents the number of documents in which term $i$ appears, and $N$ represents the total number of documents in the collection. Apply the distance value from Table 4.4 for each document and total number of document ($N=7$), to the equation (4.7). The TF/IDF weighting matrix produces the term matrix which is represented by the following matrix.

$$w_{i,D} = \begin{bmatrix}
0.067 & 0.067 & 0.067 & 0.067 & 0 & 0.067 & 0.067 \\
0 & 0 & 0.544 & 0 & 0 & 0.544 & 0 \\
0 & 0 & 0.146 & 0 & 0 & 0.146 & 0.146 \\
0 & 0 & 0.308 & 0.308 & 0 & 0 & 0 \\
0 & 0 & 0.544 & 0.544 & 0 & 0 & 0
\end{bmatrix}$$
This term weight is called the document frequency and term document matrix. This resultant matrix is called as Inverse Document Frequency. In the above, \( W_{LD} \) matrix, column vectors of \( W_{LD} \) is called document vectors, row vectors of \( W_{LD} \) is called as term vectors.

Step 3.6: Cluster the document with respect to appropriate cluster text with their TF-IDF

For example, user query of “ability to deal” is taken from the document clustering based upon the term weight. In addition to this, the similarity threshold considers documenting clustering. The similarity threshold is a lower limit for the similarity of two data records that belong to the same cluster. For example, set the similarity threshold to 0.80, data records with field values that are 85% similar are likely to be assigned to the same cluster. Let’s assume that 0.80 are the value of similarity threshold, from document 3 (figure 4.5). This document becomes the retrieved document; remaining documents are not concerned with the topic so it is ignored. These processes are represented by the following matrixes; to elaborate document cluster, the term 4 and term 5 (figure 4.5) is taken as input keywords. The available term weight for this term from the matrix is,

\[
t_{ij} = [1 \ 0 \ 0 \ 1 \ 1]^T
\]

and its document frequency is,

\[
w_{LD} = [0.67 \ 0 \ 0 \ 0.308 \ 0.844]^T
\]

The inverse document frequency is,

\[
t_{ij} X w_{LD}^T = [0.67 \ 0 \ 0 \ 0.308 \ 0.844]^T [1 \ 0 \ 0 \ 1 \ 1]^T
\]

The final query document similarity vector is,

\[
t_{ij} X w_{LD}^T = [0.80 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0]^T
\]
This matrix return,
\[ t_{ij} \mathbf{X} \mathbf{w}_{iD}^T = [0 \ 0 \ 0.308 \ 0.308 \ 0 \ 0 \ 0] \]

From the above matrix, columns three and four have the document frequency value.
\[ t_{ij} \mathbf{X} \mathbf{w}_{iD}^T = [0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0] \]

This above process is retrieved based on the column 3 and column 4. So, with respect to the query “ability to deal”, gives the document 3 and document 4 which is given in the figure 4.5.

**Step 4: To estimate the terms based on Statistical terms such as \( \chi^2 \) Test**

After TF/IDF, original dataset is available for sufficient and reliable data for document clustering. Feature selection can improve the efficiency and accuracy of text by removing redundant and irrelevant terms from the corpus in the document. Many feature selection methods are not efficient and suitable for the feature selection. \( \chi^2 \) and information gain is used in feature selection to improve the performance in document clustering. Chi-Square distance can be computed by the equation.

\[
\chi^2(Q, D) = \sum_i \frac{(q_i - d_i)^2}{q_i + d_i}
\]

(4.8)

Where \( Q \) is the number of times the significant terms occur in document \( d_i \) and the denominator is the sum of all the terms occur in document \( D \). In inverse term frequency, the equation 4.8 is used to compute the Chi-Square statistic. It can be calculated by using the following equation,
\[ Inver_{\text{Term\_Freq}}_i = \log \left( \frac{|D|}{|\{d_t \in d\}|} \right) \]  

(4.9)

TF/IDF calculated by,

\[
(Term_{\text{Freq}})_ij = (Term_{\text{Freq}}_{ij} \times Inver_{\text{Term\_Freq}}_i) + Inver_{\text{Term\_Freq}}_{ij}
\]

(4.10)

After Chi-Square statistic, frequent terms are reduced by using Inverse Document Frequency (IDF). The frequent term reduction uses two significant processes such as (i). Frequencies based optimized schema, (ii). TF/IDF. These processes produced dimensionality reduced output.

**Step 5: Estimates relevancy and distinguish the terms to document by Term Significant weight and term discriminating weight.**

Term significant weight is calculated by using Term Frequency / Inverse Document Frequency (TF/IDF). The relevancy estimation consists of these conditions. If (TF/IDF) is less than the terms, the corresponding term is called as negative terms; else the terms are called as a positive term. The proposed new weighting scheme called “term significant weight” is applied to both positive and negatives a term which is represented by the equation (4.11).

\[
\text{Term\_sig\_weight} = \text{Term\_Freq} \left( \log_2 \frac{\max \text{DP}_i}{\max \text{DN}_i} \right)
\]

(4.11)

Where \(D (P_i)\) indicates, the number of documents with positive term \((P_i)\) and \(D (N_i)\) indicates number of documents with the negative term \((N_i)\) and max_function is used to avoid division by zero error. This new weighting
scheme of TSW and TDW are considered both as the positive and negative occurrences of a term and also all the documents that contain the terms. In other hand, Term discriminating weight is used to differentiate the weight of each process. After finding the term based categorization, Term discriminating weight is calculated based on the equation (4.12),

\[
\text{Term\_discri\_weight} = \left[ \left( \frac{N}{n} \right) \left( \frac{1}{n-m} + 1 \right) \right]
\]

(4.12)

Where ‘N’ is the total number of documents, ‘n’ is the number of documents containing term ‘t’, ‘m’ is a maximum number of documents containing term t. It is formed as the original category of the term based weight.

There are two cases to form the term based categories.

Case 1: if “n” contains the term “t”, then (n-m) is large in a category.

Case 2: if “m” in the other categories contains term t, then (n-m) is small.

The individual terms are considered to calculate the frequency of occurring in the whole documents of Reuters-21578, 20 NG, and TDT-2 dataset.

The following Lemma used to create the proposed term discriminating weight based feature selection.
Lemma 2: \( n-m \) is the difference between the number of all documents \( n \) containing term t and the maximum number of documents \( m \) containing term t in a certain category.

From the above lemma, the term discriminating weight gives the following term based categorization

\[
Term = \frac{1}{\text{total number of documents except large term based categories}}
\]  

(4.13)

The proposed feature selection method combines both the term significant weight and term discriminating weight equation (4.11) and equation (4.12). These weighting schemes are efficient to select the weight based feature selection, which is derived by the equation. (4.14)

\[
a_{ik} = (f_{ik}) \left( \log \frac{N}{df_i} \right) ((\text{Term\_sig\_weight})(\text{Term\_discri\_weight}))
\]  

(4.14)

Where \( f_{ik} \) – frequency of term i in document j and \( df_i \) is document frequency of term i.

4.5 DIMENSIONALITY REDUCTION OF DOCUMENT FREQUENCY THRESHOLDING METHOD

In training documents, Document Frequency Thresholding (DFT) in figure 4.6 is used to reduce the vocabulary, vagueness and uncertainty in very large corpora with a low cost and computation complexity. To improve the
text categorization in document collections, Information Gain (IG) and Chi-Square based categorization are used.

Pseudocode:

node ← Start
Initialize TF, IDF, DF, c,
   TF ← Term Frequency;
   IDF ← inverse Document Frequency;
   DF ← Document Frequency;
   c ← threshold;
for each document in test corpus do
   Remove tags, punctuation, other language text and non-alphanumeric text
   Perform case folding
   Perform Bag of Words
   Remove stop-words
end for
for each remaining word in the dataset do
   Perform Porter Stemmer and store in a vector (Word List)
end for
for each word in the Word List do
   Calculate Modified TF/IDF and store the result in a weight matrix
end for
for each element in weight matrix
   Set the threshold ‘c’
   Calculate Document Frequency (DF) for each term
   If DF < c then
      Remove the term along with its weight from weight matrix
   End if
End for

Figure 4.6 Pseudocode for Document Frequency Thresholding

4.6 DATASETS

This research is performed on three datasets corpus such as Reuters-21578 corpus, 20 NG data corpus, and TDT-2. These three datasets corpus are collected from various resources such as web pages, newsgroup sources, and online portal reports.
One of the largest database offline Newswire articles called Reuters-21578 have many issues that are arranged by the data, time of issues under 90 categories. Each category consist of 12,902 documents, each document consists of 140 instances each with respect to single and multiple topic flow. There are 22 topics available in the Reuters-21578.

Generally, there are 20 categories available in 20 NG dataset each categories has 1000 data instances with respect to the online news collection namely UseNet postings.

Christopher C. et al. (2009) described the document organization of TDT-2. TDT-2 news document gathers new stories from various fields of sources in various formats such as text or audio of different languages. TDT-2 include story, events, and topic source segmentation and also defines samples of whole events in TDT-2.

The TDT-2 is one of the dataset corpuses, consisting of more elements, different kind of nature is ordered under single-source collection. TDT-2 dataset corpus get information from the TDT. Generally there are three types of TDT available, TDT-1 (1997), TDT-2 (1998) and TDT-3 (1999). The TDT-1contains English language based Newswire and quality transcripts of news broadcasts. There are 25 topics available in TDT-1 with 15,863 stories. Another two types of corpus both TDT-2 and TDT-3 contain Chinese and English Newswire and quality transcripts of news broadcasts. TDT-2 have 100 topics and TDT-3 have 60 topics with 116,012 stories.
4.7. PERFORMANCE MEASURES

The performance measures contain the following steps to be considered and it is developed based on the size of the data corpus such as Reuters-21578, 20 NG, and TDT-2 dataset. This is defined by the implementation of DFT. Eventually, all the preprocessing steps are compared and rated based on the individual steps. The whole content of the corpus is considered to the Feature extraction rate, data reduction rate and accuracy. Reuters-21578 collection is publicly available test collections dataset. It contains 21578 news articles. Reuters-21578 documents collection is used to categorize based on the main categories with respect to the total of 806,791 documents. Other than these categories, the subsets of 118,924 documents are categorized. These categories contain 1600 random positive examples for each Reuters-21578 dataset. From Reuters-21578 100 sample document has been considered for this research. The document classification experiment is used for reducing exact duplicates. These are executed in the news agencies such as BBC, CNN, VOA, NYT, PRI and APW. The TDT-2 corpus contains 64527 news stories. It has around 10 million words, each of which reports a major news event occurred in January 1988 to April 1988 from the Associated Press and New York Times. Here nearly 7803 documents have a unique category level. Perhaps the number of documents for different news events is very unbalanced. This research is used to retrieve this type of unbalanced news events with the help of Ontology-based document clustering. This research excluded some events, and final test sets. Because it is more unbalanced and unrelated to the existing dataset categories. TDT-2 is easy to make a cluster compared to the Reuters-21578 corpus, and 20 NG because it was grouped into 135 clusters only. The following results show which the Feature extraction for each dataset corpus and formation of cluster which are shown in the figures 4.7 to 4.9.
Figure 4.7 Feature extraction (Reuters-21578) for each preprocessing

Figure 4.7 shows the feature extraction of Reuters-21578 for each preprocessing step. It includes the corpus of documents from the top ten largest categories named as the Reuters-21578 collection. Experiments on Reuters-21578 collection show that BOW, StopWord elimination, and Porter Stemming performs much better as it involves removal of vagueness, vocabulary, and repeated words.

Figure 4.8 Feature Extraction (20 NG) for each preprocessing

Figure 4.8 shows the feature extraction of 20 Newsgroups for each preprocessing step. It includes the corpus of documents from the top ten largest categories named as the 20 Newsgroups collection. Experiments on 20 Newsgroups collection show that BOW, StopWord elimination, and Porter Stemming performs much better as it involves removal of vagueness, vocabulary, and repeated words.
Figure 4.8 shows the feature extraction for each preprocessing step that includes 20 NG corpuses preprocessing such as BOW, StopWord removal and stemming of documents from 20 NG collections. Experiments on 20 NG collection shows that BOW, StopWord Elimination, and porter stemming performs much better in removing vagueness, vocabulary, repeated word, reduction outperform better and extract the features as more in the proposed DFT.

![Feature Extraction for TDT-2 corpus](image)

**Figure 4.9 Feature Extraction (TDT-2) for each preprocessing**

The above figure 4.9 shows the feature extraction for each step which includes the initial corpus of documents from the ten largest categories from the TDT-2 collection. Experiments on TDT-2 corpus collection shows that BOW, StopWord elimination, and porter stemming performs much better to removal of vagueness, vocabulary, and repeated words.

The proposed DFT is implemented, in the performance analysis of proposed work with feature extraction, and the accuracy of different document sizes is computed using equation (4.15). The result of the implementation in terms of accuracy is recorded and shown in Table 4.5.
Accuracy

\[
\text{Accuracy} = \frac{\text{Number of corrected prediction for each preprocessing step}}{\text{Total of cases to be predicted on the dataset collection}}
\]

\[(4.15)\]

<table>
<thead>
<tr>
<th>DATASET</th>
<th>DOC SIZE</th>
<th>After BOW</th>
<th>After SWE</th>
<th>After Stemming</th>
<th>Proposed DFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuters-21578</td>
<td>20 KB</td>
<td>95</td>
<td>93</td>
<td>93</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>40 KB</td>
<td>94</td>
<td>92</td>
<td>92.5</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>60 KB</td>
<td>95</td>
<td>93</td>
<td>93.5</td>
<td>96.5</td>
</tr>
<tr>
<td></td>
<td>80 KB</td>
<td>94</td>
<td>92</td>
<td>90.5</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>100 KB</td>
<td>94</td>
<td>92</td>
<td>90.8</td>
<td>94.5</td>
</tr>
<tr>
<td>20 NG</td>
<td>20 KB</td>
<td>95.543</td>
<td>93.543</td>
<td>93.543</td>
<td>96.543</td>
</tr>
<tr>
<td></td>
<td>40 KB</td>
<td>94.543</td>
<td>92.543</td>
<td>93.043</td>
<td>96.543</td>
</tr>
<tr>
<td></td>
<td>60 KB</td>
<td>95.543</td>
<td>93.543</td>
<td>94.043</td>
<td>97.043</td>
</tr>
<tr>
<td></td>
<td>80 KB</td>
<td>94.543</td>
<td>92.543</td>
<td>91.043</td>
<td>94.543</td>
</tr>
<tr>
<td></td>
<td>100 KB</td>
<td>94.543</td>
<td>92.543</td>
<td>91.343</td>
<td>95.043</td>
</tr>
<tr>
<td>TDT-2</td>
<td>20 KB</td>
<td>96.086</td>
<td>94.086</td>
<td>94.086</td>
<td>97.11</td>
</tr>
<tr>
<td></td>
<td>40 KB</td>
<td>95.086</td>
<td>93.086</td>
<td>93.586</td>
<td>97.11</td>
</tr>
<tr>
<td></td>
<td>60 KB</td>
<td>96.086</td>
<td>94.086</td>
<td>94.586</td>
<td>97.61</td>
</tr>
<tr>
<td></td>
<td>80 KB</td>
<td>95.086</td>
<td>93.086</td>
<td>91.586</td>
<td>95.11</td>
</tr>
<tr>
<td></td>
<td>100 KB</td>
<td>95.086</td>
<td>93.086</td>
<td>91.886</td>
<td>95.61</td>
</tr>
</tbody>
</table>

Table 4.5 Feature Extraction Accuracy

Comparison of accuracy of various sizes of document and proposed DFT are shown in figure 4.10. Comparison of accuracy of various preprocessing methods and proposed DFT are shown in following figure 4.10. Accuracy is the process of extracting correct set from the available set of terms or words, depending upon the document size likely (10 KB, 20 KB, 40 KB, 60 KB, 80 KB, and 100 KB) and is taken as the individual document process which has given feature extraction accuracy with respect to the individual preprocessing step.
The above figure 4.10 shows the feature extraction for each step which includes the initial corpus of documents from the top ten largest categories named the Reuters-21578 collection and its feature selection accuracy. From figure 4.10, the proposed DFT has the higher rate of feature extraction accuracy. From the above figure, it is observed that the consideration of accuracy would be 95%, from the Reuters-21578 collection (100 samples). Moreover, the accuracy related issues are overcome by using reduction rates metrics.
The above figure 4.11 show the data reduction rate for each preprocessing step which includes the initial corpus of documents from the 100 KB document collection are taken as the test samples. The data reduction rate are calculated based on following formulae,

\[
\text{Data reduction rate}_{(\text{Average, Outliers, Fixed Grid, Regression})} = \frac{1}{\text{Accuracy}} \times 100
\]

(4.16)

From the above equation, the data reduction rate are calculated to collection of documents with the document size 100 KB, the reduction rate of data to be formed by using following criteria such as

- Average
  - Processed based on the amount of error if it is large then it is difficult to find the data reduction.
• Outliers
  o It is a document variation from the different type size documents are gathered as reduced rate varied on the different style.

• Fixed Grid
  o Process of dividing the large amount of data into the main and other categories depending on the Pre Processing Step.

• Regression
  o Reduction rate to be compared with the main categories depending upon the above mentioned criteria the reduction rate is calculated.

4.8 SUMMARY

Feature selection is a preprocessing technique commonly used on high-dimensional data. Feature selection studies shows, how to select a subset or list of attributes or variables that are used to construct models describing data. The existing feature selection method considers the frequency of word's appearance, while ignoring other factors, which may impact the word weights. The terms with higher scores are deemed to have contributed more to the text categorization than those with lower scores. This chapter focused on various processes such as dimensionality reduction, removal of irrelevant and redundant feature reducing the amount of data needed for learning, improving algorithm's predictive accuracy, and increasing the constructed models comprehensibility. In this preprocessing and feature selection processes, TF*IDF has been extensively used for term weighting and document representation. The various processing methods are discussed with respect to
preprocessing and feature selection, such as feature Identification, StopWord elimination, stemming, feature weighting, selection and dimensionality reduction. This new approach on term weighting contributes not about improvement over TF*IDF but enables different way of thinking and provides new information measure for modeling various information processes.