CHAPTER 3

THE SUMMARIZATION WORK FLOW

3.1 INTRODUCTION

Automatic text summarization is a technique to identify the most important information from a text document, thereby omitting irrelevant information and minimizing details to generate a compact coherent summary. It is said that in order to better understand the operation of summarization systems and to emphasize the design choices system developers need to make, there are three relatively independent tasks which must be performed by virtually all summarizers (Nenkova & McKeown 2012) and Figure 3.1 depicts these tasks:

(i) Creating an intermediate representation of the input which captures only the key aspects of the text

(ii) Scoring sentences based on that representation

(iii) Selecting a summary consisting of several sentences.

![Figure 3.1 Tasks of a typical summarizer](image-url)
3.1.1 Intermediate Representation

Even the simplest systems derive some intermediate representation of the text they have to summarize, and identify important content based on this representation. Topic representation approaches convert the text to an intermediate representation, interpreted as the topic(s) discussed in the text. Some of the most popular summarization methods rely on topic representations, and this class of approaches exhibits an impressive variation in sophistication and representation power. They include frequency, TF-IDF, topic word approaches, etc.

3.1.2 Scoring Sentences

Once an intermediate representation has been derived, each sentence is assigned a score which indicates its importance. For topic representation approaches, the score is commonly related to how well a sentence expresses some of the most important topics in the document or to what extent it combines information about different topics. For the majority of indicator representation methods, the weight of each sentence is determined by combining the evidence from the different indicators, most commonly by using machine learning techniques to discover the indicator weights.

3.1.3 Selection of Summary Sentences

Finally, the summarizer has to select the best combination of important sentences to form a paragraph length summary. In the best $n$ approaches, the top $n$ most important sentences, which have the desired summary length are selected to form the summary. In maximal marginal relevance approaches (Carbonell & Goldstein 1998), sentences are selected in an iterative greedy procedure. At each step of the procedure, the sentence importance score is recomputed as a linear combination between the original importance weight of the sentence and its similarity with already chosen
sentences. Sentences that are similar to already chosen sentences are not preferred. In global selection approaches, the optimal collection of sentences is selected, subject to constraints that try to maximize the overall importance, minimize redundancy, and, for some approaches, maximize the coherence..

In this chapter, the approaches, methods and evaluation techniques that are used in this research are briefly discussed. The corpus used for the research and the feature selection and extraction done, are also introduced. The following section details the summarization workflow of the research under discussion.

3.2 THE LAYERED SUMMARIZATION MODEL

Automatic summarization is a multistage process. Any automatic text summarization system has some definite sets of prerequisites to be satisfied before processing, a unique set of algorithms applied on the input document, and its own evaluation mechanisms. Figure 3.2 shows the proposed architecture along with the techniques and processes applied in this research for satisfying the various requirements of the proposed automatic summarization method.

![Figure 3.2 Summarization workflow](image-url)
The present research proposes an automatic extractive text summarization architecture, and the summarization workflow shown in Figure 3.2 can be visualized as a layered summarization model, which is discussed in this Chapter. The layered summarization model consists of the following four-fold layers: pre-processing, feature link, business and the summary presentation layers. Figure 3.3 shows the layers of the summarization model discussed in this thesis.

![Layers of the summarization model](image)

**Figure 3.3 Layers of the summarization model**

Each of the layers performs various roles in the process of summarization. The following subsection discusses the layers of the summarization model in detail.

### 3.2.1 Preprocessing Layer

The main task of the pre-processing layer is to clean the document, by removing words that do not contain any information that uniquely identifies a sentence. This layer filters the noise away. Pre-processing includes Sentence segmentation, Tokenization, Stop word removal and Word stemming respectively.
Once, a text has been submitted as input to the system, the entire text is segmented into individual sentences. Each sentence is again tokenized into individual words. Stop words are words which appear frequently in the document, but provide less meaning in identifying the important content of the document. It is essential to remove words such as ‘a’, ‘an’, ‘the’, etc., from the text, in order to obtain a high quality text. The last step in pre-processing is word stemming. Stemming is the process of taking a word to its native state. Affixes are removed from every word, which concludes pre-processing.

3.2.2 Feature Link Layer

In any task of text mining, features play an important role. Features are attributes that attempt to represent data used for the task. The main role of the feature link layer is to select important features, and then extract the selected features that will represent the sentences in the text document.

3.2.2.1 Feature selection

Feature selection is the process of selecting a specific subset of the terms of the training set, and using only that in the prediction or classification tasks. The feature selection process takes place before the training of the predictor or the classifier. The main advantages of using feature selection algorithms, are the facts that they reduce the dimension of our data, make the training faster, and improve accuracy by removing noisy features. As a consequence, feature selection can help to avoid over-fitting. The basic selection algorithm for selecting the k best features is presented below (Manning et al 2008):
SELECTFEATURES \( (D,c,k) \)

1. \( V \leftarrow \text{EXTRACT VOCABULARY} (D) \)
2. \( L \leftarrow [\] \)
3. for each \( t \in V \)
4. \( \text{do } A(t,c) \leftarrow \text{COMPUTEFEATUREUTILITY} (D,t,c) \)
5. \( \text{APPEND} (L, \langle A(t,c), t \rangle) \)
6. return \( \text{FEATURESWITHLARGESTVALUES} (L,k) \)

Previous feature selection studies for text domain problems have been a great help in providing guidance and motivation for this study, which features a more extensive variety of metrics, a larger set of benchmark problems, and one of the best induction algorithms of late, the support vector machine. For example, the valuable study by Yang & Jan (1997) considered five feature selection metrics on the standard Reuters dataset, and OHSUMED. It did not consider SVM, which they later found to be superior to the algorithms they had studied, LLSF and kNN. With sufficient data and time, it is fine to use all the input features, including those irrelevant features, to approximate the underlying function between the input and the output. But in practice, there are two problems which may be evoked by the irrelevant features involved in the learning process.

- The irrelevant input features will induce greater computational cost.
- The irrelevant input features may lead to overfitting.

Another motivation for feature selection is that, since our goal is to approximate the underlying function between the input and the output, it is reasonable and important to ignore those input features with little effect on the output, so as to keep the size of the approximated model small. For example, (Akaike 1973) proposed several versions of the model selection criteria,
which basically are the trade-offs between high accuracy and small model size.

In experiments, that were conducted on the feature selection methods for sentiment analysis, the results showed that Information Gain gave consistent results, and the Gain Ratio performed the best on the whole, for sentimental feature selection, out of five different feature selection algorithms (Sharma & Dey 2012). In addition to Information Gain (IG) and the Gain Ratio (GR), this research uses the Chi-squared attribute selection algorithm also. In this thesis, three feature selection algorithms, namely, the chi-squared attribute selection, information gain based feature selection and gain ratio-based feature selection, have been used in selecting and ranking fifteen features. The chosen features are taken from the literature, and these features were considered important in text summarization systems. They are the title feature, sentence centrality, sentence location, sentence length, term weight, sentence to sentence similarity, proper noun, numerical data, sentence relative length, bushy node, similarity to first sentence, negative words, word frequency, thematic word, and node similarity. The following subsection discusses the feature selection algorithms used in selecting the top features for use in the task of text summarization.

3.2.2.1 Chi-squared attribute selection

Chi-Squared is the common statistical test that measures divergence from the distribution expected, if one assumes that the feature occurrence is actually independent of the class value. As a statistical test, it is known to behave erratically for very small expected counts, which are common in text classification, both because of having rarely occurring word features, and sometimes because of having few positive training examples for a concept. More specifically in feature selection we use it to test, whether the occurrence of a specific term and the occurrence of a specific class are independent.
Thus, the following quantity for each term using Equation (3.1) is estimated and ranked:

\[
X^2(D, t, c) = \sum_{e_t \in \{0, 1\}} \sum_{e_c \in \{0, 1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}
\]  

(3.1)

where \(e_t = 1\) if the training instance contains term \(t\) and \(0\) otherwise, and \(e_c = 1\) if the training instance is in class \(c\), and \(0\) otherwise. \(N\) is the observed frequency and \(E\) the expected frequency. For example, \(E_{11}\) is the expected frequency of \(t\) and \(c\) occurring together in the data, assuming that they are independent. High scores on \(X^2\) indicate that the null hypothesis (\(H_0\)) of independence should be rejected, and thus, that the occurrence of the term and class are dependent. If they are dependent then we select the feature is selected for future extraction (Forman 2003).

3.2.2.1.2 Information gain based attribute selection

Information gain is used as a feature (term) goodness criterion in machine learning based classification (Yang & Jan 1997, Tan & Zhang 2008, Wang et al 2011). It measures information obtained (in bits) for the class prediction of an arbitrary text document by evaluating the presence or absence of a feature in that text document. Information Gain is calculated by the feature’s contribution on decreasing the overall entropy. The expected information needed to classify an instance (tuple) for partition \(D\) or identify the class label of an instance in \(D\) is known as entropy, is given by Equation (3.2).

\[
\ln f o(D) = -\sum_{i=1}^{m} (P_i) \log_2(P_i)
\]

(3.2)

where \(m\) represents the number of classes (\(m=2\) for binary classification) and \(P_i\) denotes the probability, that a random instance in partition \(D\) belongs to class \(C_i\) estimated as \(|C_{i,D}| / |D|\) (i.e. proportion of instances of each class or category). A log function to the base 2 justifies the fact that we encode
information in bits. If we have to partition (classify) the instance in $D$ on some feature attribute $A \{a_1, \ldots, a_v\}$, $D$ will split into $v$ partitions set ${D_1, D_2, \ldots, D_v}$. The amount of information in bits required for an exact classification, is measured by Equation (3.3):

$$\text{Info}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{Info}(D_j)$$

(3.3)

where $|D_j|/|D|$ is the weight of the $j$th partition, and $\text{Info}(D_j)$ is the entropy of partition $D_j$; the Information gain by partitioning on $A$ is as shown in Equation (3.4):

$$\text{Information Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

(3.4)

The features ranked as per the highest information gain score are selected. We can optimize the information needed or decrease the overall entropy, by classifying the instances using those ranked features.

3.2.2.1.3 Gain ratio based attribute selection

The Gain Ratio enhances the Information Gain, as it offers a normalized score of a feature’s contribution to an optimal information gain based classification decision. The Gain Ratio is utilized as an iterative process where we select smaller sets of features in an incremental fashion. These iterations terminate when there are only a predefined number of features remaining. The Gain ratio is used as one of the disparity measures, and the high gain ratio for the selected feature implies that the feature will be useful for classification. The Gain Ratio was first used in the decision tree (C4.5), and applies normalization to the information gain score, by utilizing a split information value (Quinlan 1993). The split information value corresponds to
the potential information obtained by partitioning the training data set \( D \) into \( v \) partitions, resulting in \( v \) outcomes on attribute \( A \) as shown in Equation (3.5).

\[
SplitInfo_A(D) = - \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2 \frac{|D_j|}{|D|}
\]  

(3.5)

where high \( SplitInfo \) means the partitions have equal size (uniform), and low \( SplitInfo \) means a few partitions contain most of the tuples (peaks). The gain ratio is defined as shown in Equation (3.6).

\[
Gain \ Ratio \ (A) = \frac{Information \ Gain \ (A)}{SplitInfo(A)}
\]  

(3.6)

3.2.2.1.4 Features selected for the summarization model

Selecting appropriate features for any processing model is an important and crucial task and for selecting the set of features we have used three popular feature selection algorithms namely the chi-squared attribute selection, information gain based attribute selection and the gain ratio based attribute selection. At the end of the selection, we have chosen six features that were ranked in the top ten namely the title feature, sentence length, term weight, sentence to sentence similarity, thematic words and numerical data feature. Proper nouns are important features in text summarization since sentences containing proper nouns are informative in nature and hence such sentences can be candidates for presence in the summary. Hence we have added proper noun feature with the previously ranked six features and the set of features utilized in the proposed summarization model is discussed in detail in the following section.
3.2.2.2 Feature extraction

Feature Extraction is creating a set of features by decomposing the original data, since it extracts important and interesting features that help to represent the original text. Though there are a number of researchers working on feature based techniques for extracting features such as tf-idf, cue words, bushy node, etc., (Cigar et al 2009, Suanmali et al 2009, Geng et al 2010) an attempt is made in this research to derive a solid, unique, consistent set of features, which ensures that sentences are properly represented by these features. In this layer, seven features for each sentence, viz. sentence length, title resemblance, thematic words, numerical data, proper noun, term frequency and sentence to sentence similarity, were experimentally selected, based on the feature selection algorithms discussed in the previous section. The Feature Link layer uses a pre-processed text sent by the previous layer, namely, the pre-processing layer. The rest of this section discusses the features extracted, and methods of extraction.

a) Title feature

The number of title words in the sentence contributes to the title feature. Titles contain groups of words that give important clues about the subjects contained in the document. Therefore, if a sentence has higher intersection with the title words, the sentence is more important than others. Equation (3.7) exhibits how the value is calculated.

\[
Score (S_i) = \frac{\text{No. of Title words in } S_i}{\text{No. of Words in Title}} \quad (3.7)
\]
b) Sentence length

The number of words in a sentence gives a good idea about the importance of the sentence. This feature is very useful to filter out short sentences, such as datelines, and author’s names commonly found in articles. The short sentences are not expected to belong to the summary, and the sentence length feature is calculated by the formula given in Equation (3.8).

\[
\text{Score} \left( S_i \right) = \frac{\text{No. of Words occurring in } S_i}{\text{No. of words occurring in the longest sentence}} \quad (3.8)
\]

c) Term weight

The term weight feature score is obtained by calculating the average of the Term Frequency, Inverse Sentence Frequency (TF-ISF). The frequency of term occurrences within a document has often been used for calculating the importance of a sentence (Zamanifar et al 2008). Inverse sentence frequency helps to identify important sentences that represent the document (Saeedeh et al 2010). The term weight feature is calculated by the formula given in Equation (3.9).

\[
\text{Score} \left( S_i \right) = \frac{\text{Sum of TF-ISF in } S_i}{\text{Max (Sum of TF-ISF)}} \quad (3.9)
\]

d) Sentence to sentence similarity

Similarity between sentences is calculated as follows: for a sentence s, the similarity between s and all other sentences is computed by the cosine similarity measure. The score of this feature for a sentence is obtained by computing the ratio of the summation of the sentence similarity of a sentence S with each of the other sentences over the maximum value of the sentence similarity as given in Equation (3.10).
\[ Score \ (S_i) = \frac{\text{Sum of Sentence Similarity for } S_i}{\text{Max (Sum of Sentence Similarity)}} \] (3.10)

e) Proper noun

The proper noun feature gives the score based on the number of proper nouns present in a sentence, or the presence of a named entity in the sentence. Usually sentences that contain proper nouns are considered to be important, and these should be included in the document summary. The score for this feature is calculated by the formula given in Equation (3.11).

\[ Score \ (S_i) = \frac{\text{No. of Proper nouns in } S_i}{\text{Length } (S_i)} \] (3.11)

f) Thematic word

The number of thematic words in a sentence is an important feature, because terms that occur frequently in a document are probably related to the topic. Thematic words are words that capture the main topics discussed in a given document. We used the top 10 most frequent content words for consideration as thematic. The score for this feature is calculated by the formula given in Equation (3.12).

\[ Score \ (S_i) = \frac{\text{No. of Thematic Words in } S_i}{\text{Length } (S_i)} \] (3.12)

g) Numerical data

This feature gives a score for the number of numerical data in a sentence. The contribution made by this feature to the weight given to a sentence is significant, since a sentence that contains numerical data essentially contains important information. The idea behind finding numerical data in the text is that the sentences containing numbers and numerals are more valuable and informative to the reader than those which do not contain
numbers or numerals (Emilia, B & Marek, G 2005). The score for this feature is calculated by the formula given in Equation (3.13).

\[ \text{Score} (S_i) = \frac{\text{No. of Numerical Data in } S_i}{\text{Length } (S_i)} \] (3.13)

Feature values extracted provide significant information about the importance of the document sentences. The values of the features discussed in the previous section lie between ‘0’ and ‘1’. The categorization of values help to represent the weight of features in a presentable manner, which can be easily used for further processing and also as input to a number of algorithms, methods, etc. Based on the values, the feature values are grouped into five categories, namely, low, very low, average, and significant values - high and very high, as shown in Table 3.1 (Suanmali et al 2009).

<table>
<thead>
<tr>
<th>Feature_score</th>
<th>Class</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-0.2</td>
<td>Very Low</td>
<td>VL</td>
</tr>
<tr>
<td>0.21-0.4</td>
<td>Low</td>
<td>L</td>
</tr>
<tr>
<td>0.41-0.6</td>
<td>Average</td>
<td>A</td>
</tr>
<tr>
<td>0.6-0.81</td>
<td>High</td>
<td>H</td>
</tr>
<tr>
<td>0.8-1.0</td>
<td>Very High</td>
<td>VH</td>
</tr>
</tbody>
</table>

3.2.3 Business Layer

Once the prerequisites for summarization are done, the text is now ready for processing in the business layer. The output of the feature link layer becomes the input to the business layer. The business layer deals with various approaches used for summarization in the model. The business layer receives a unique, well-defined set of feature values that represent the sentences in the
original document. The business layer performs tasks needed for summarization through the Regression Based Summarization (RBS) model, Classification Based Summarization (CBS) model and Summary Refinement Model. The following subsections introduce and discuss the operations of the business layer in detail.

### 3.2.3.1 Regression based summarization

Multivariate statistics is a form of statistics encompassing the simultaneous observation and analysis of more than one statistical variable. The application of multivariate statistics is called the multivariate analysis. Multivariate regression is a multivariate statistical technique for examining the linear correlations between two or more Independent Variables (IVs) and a single Dependent Variable (DV) (Hair 1995). Multivariate regression is a method wherein we can obtain the pure value of a relation. This is the basic purpose of using it as the main process in our system, since it can identify the core pure value representing the relation between the text and its model summary. Hence, this method is a suitable candidate for text summarization.

Features discussed in section 3.2.2.2 are extracted for every sentence in the document given as input to the system. These features form the independent vector of the input document. The dependent vector is formed making use of the model summary provided for a given input document. The independent and the dependent vectors are given as input to the system of multivariate regressions (Raubenheimer 2004). A constant weighted value is obtained as the result of the multivariate regressions. This value represents the relation between the text and its model summary. The RBS system is trained using multivariate regressions. The RBS model, its role in summarization, and its performances on two datasets are discussed in detail in Chapter 4.
3.2.3.2 Classification based summarization

This thesis proposes Classification Based Summarization (CBS). This model performs automatic summarization of the text through classification. Summarization systems are the need of the hour, since information is overloaded in the web and extracting informative sentences from a document is highly essential. The proposed method delivers a platform to perform automatic summarization of a text, exploiting classification. The capability of classification algorithms in predicting summary sentences and comparing their prediction performance against eight well-known machine learning models, was investigated. The compared models are three tree-based classifier techniques; (i) the Iterative Dichotomiser (ID3) (ii) J48 (iii) logistic model trees (LMT), a neural networks technique; the Radial Basis Function (RBF), a Bayesian technique; the Naive Bayes (NB) and two optimization algorithms; (i) Sequential minimal optimization (SMO) and (ii) Support Vector Machine (SVM).

Based on the evaluation results, the CBS model is trained with decision trees. Decision trees are a simple but powerful form of multiple variable analysis. The approach starts with extracting features for each and every sentence and classifying them. 60 % of the documents in the DUC 2002 corpus is used for training and the remaining 40% for testing the CBS model. For every sentence present in the input test document, seven feature attributes as discussed in section 3.2.2.2 were extracted, and given as input to the classifier model. Based on the classifier model learnt in the training phase, each sentence was classified as ‘interesting’ or ‘not-so-interesting’. The generated summary consists of only interesting sentences based on the required compression ratio. An attempt is made to improve the CBS model using the multivariate approach. A detailed discussion on CBS and its improvements is given in Chapter 6.
3.2.3.3 Summary refinement model

System generated summaries are prone to contain similar sentences that convey similar meaning. Such similar sentences would have been the candidate sentences of the summary due to their similarity in features. This leads to redundancy in summaries and thereby increases the length of the summaries. Some researchers have proposed to refine a system-generated summary using filtering sentences or phrases, before they could become part of the summary. We have taken up this challenge and suggested a way of refining extractive summaries by removing redundant sentences. The rough summaries used in our refinement are those produced by supervised techniques such as regression and classification. Supervised techniques tend to produce summaries sentences by learning features of sentences provided in the training dataset. This drawback could be overcome when unsupervised techniques such as clustering are used, making the summary free of redundant sentences. However our redundancy elimination approach makes use of a rough summary generated by supervised methods. Thus in Chapter 6, the use of the binomial model to refine system-generated summaries is discussed in detail, with Classification Based Summarization (CBS) methods, Regression Based Summarization (RBS) and Fuzzy Based Summarization (FBS) model as a case study.

The summary refinement framework takes a machine generated summary as a rough copy and uses the context-weighing scheme to identify the importance of every sentence in the rough summary. The semantic similarity between sentences is identified and the sentences are removed, thereby refining the summary. A summary document which is refined, is said to produce better informative summaries. The summary refinement model was tested on rough summaries obtained from other summarization systems, and their performances in terms of precision, recall and f-measure.
3.2.4 Summary Presentation Layer

The main role of this layer is to present an extractive refined summary for a given input text document. The layer focuses on providing information rich summaries that are very close to the model summaries, and still satisfying the compression ratio required by the end user (Lin & Hovy 2002). Summaries of various compression ratios such as 10%, 20% and 30%, were produced and the performance evaluation was made for the models discussed in the business layer.

3.3 CORPUS DESCRIPTION

Two text datasets are used in this research, namely,

(i) Religious Dataset collected online

(ii) DUC 2002 dataset

The Religious Dataset consisted of about 150 text documents that discuss doctrines of different religions. It was collected online over a period of 3 months, from various websites. The summaries of these documents were generated using an online summarizer called the ‘Open Text Summarizer’ for our reference summary and is referenced in several academic publications (Yatsko, VA & Vishnyakov, TN 2007, Girma & Debele 2012). The Open Text Summarizer (OTS) is an open source tool for summarizing texts (Rotem, N 2003), which is widely being used and is usually being considered for benchmarking other text summarizers. OTS is based on extracting the most frequent terms in a text and returns the sentences that cover these terms. It ships with Ubuntu, Fedora and other Linux distributed operating systems. Open Text Summarizer supports more than 25 languages which are configured in XML files. It can be downloaded free at http://sourceforge.net/projects/libots/.
The TIPSTER program, with its two main evaluation style conference series, TREC and Document Understanding Conference-DUC (now called as Text Analysis Conference-TAC) has shaped the scientific community, in terms of performance, research paradigm and approaches. We used 567 documents from DUC2002 for generating the training dataset. Model summaries provided by DUC2002 were used to evaluate our system.

The main features of the document corpus are:

a. Each document has a minimum of 7 sentences and maximum of 33 sentences.

b. The total number of sentences in the corpus is 3960.

3.4 SUMMARY EVALUATION AND METRICS

The evaluation of a summary is perhaps the most difficult task since there is no universal standard to measure and evaluate a summary. When there is a manual summary in hand, then the automatically generated summary can be compared against it. Since human judgment is expensive, in terms of resources, and is inconsistent, an evaluation standard is a necessity.

The Layered Summarization Model is evaluated by two metrics:

(i) Compression Ratios (Lin & Hovy 2002)

(ii) Recall-oriented Understudy for Gisting Evaluation (ROUGE) metrics (Lin 2004)

3.4.1 Compression Ratios (CR)

One important property that should be considered when evaluating summaries and summarization systems is the CR, that helps to identify how
much shorter the summary is than the original document given in Equation (3.8) below.

The compression ratio is calculated as the number of sentences in the system generated summary, to the number of sentences in the original text input. The summarization model is tested and evaluated on various compression ratios such as 10%, 20% and 30%.

\[ CR = \frac{\text{length of the summary}}{\text{Length of the original text}} \]  

(3.8)

3.4.2 ROUGE Metrics

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) was proposed by Lin (2004). This system calculates the quality of a summary generated automatically, by comparing it with the summary (or several summaries) created by humans. Specifically, it counts the number of overlapping different units, such as word sequences, word pairs and n-grams between the computer-generated summary to be evaluated and the ideal summaries created by humans. ROUGE includes several automatic evaluation measures:

ROUGE-N (n-grams co-occurrence): is an n-gram recall between a candidate summary and a set of reference summaries, and is calculated as given in Equation (3.9):

\[
\frac{\sum_{\text{summary}} \sum_{\text{gram} \in S} \text{Count}_{\text{match}}(\text{gram} _n)}{\sum_{\text{summary}} \sum_{\text{gram} \in S} \text{Count}(\text{gram} _n)}
\]  

(3.9)
where \( n \) is a length of the \( n \)-gram, \( \text{gram}_n \) and count match (\( \text{gram}_n \)) is the maximum number of \( n \)-grams co-occurring in a candidate summary and a set of reference summaries.

**ROUGE-L (longest subsequence):** There is only some corpus which has summaries made by humans. It is very time-consuming and tedious, composing summaries manually. A sequence \( S = (s_1, s_2, \ldots, s_n) \) is a subsequence of another sequence \( = (x_1, x_2, \ldots, x_m) \); if there exists a strict increasing sequence \( (i_1, i_2, \ldots, i_k) \) of indices of \( X \) such that all \( j = 1, 2, \ldots, k \), then \( x_{i_j} = s_j \). Given two sequences \( X \) and \( Y \), the Longest Common Subsequence (LCS) of \( X \) and \( Y \) is a common subsequence with the maximum length. When LCS is applied in summarization evaluation, a summary sentence is viewed as a sequence of words. Hence, the longer the LCS of two summary sentences is, the more similar the two summaries \( X \) of length \( m \) and \( Y \) of length \( n \), will be, assuming that \( X \) is a reference summary sentence and \( Y \) is a candidate summary sentence.

**ROUGE-W (weighted longest subsequence):** Given two sequences \( X \) and \( Y \), LCS is called weighted, if a length is calculated using a weighted function.

**ROUGE-S (skip-bigram co-occurrence):** Skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. Skip-bigram co-occurrence statistics measure the overlap of skip-bigrams between a candidate summary and a set of reference summaries. It is shown that (Lin 2003) these types of measures can be applied for evaluating the quality of summaries generated automatically, achieving 95% of correlation of human judgments.

For each of the measures ROUGE-N, ROUGE-L, ROUGE-W, etc., ROUGE returns Recall, Precision and F-measure scores.
Recall (R) = \#(Correct) / \#(Correct + Missed) \hspace{1cm} (3.10)

Precision (P) = \#(Correct) / \#(Correct + Wrong) \hspace{1cm} (3.11)

F-measure (F) = \frac{(\alpha + 1) \times \text{Recall} \times \text{Precision}}{\text{Recall} + (\alpha \times \text{Precision})} \hspace{1cm} (3.12)

**Precision (P):** reflects how many of the system’s extracted sentences were good, as shown in Equation (3.11). **Recall (R):** reflects how many good sentences the system missed, as shown in Equation (3.10), where *correct* is the number of sentences extracted by the system and the human, *wrong* is the number of sentences extracted by the system but not by the human, and *missed* is the number of sentences extracted by the human but not by the system. F-measure shown in Equation (3.12) can be obtained using Recall and Precision.

### 3.5 CONCLUSION

Feature selection and feature extraction techniques play a very important role in systems that extract information. In this thesis, effective feature selection and feature extraction are done laying a firm foundation for the proposed summarization model to be built upon. The methods and algorithms involved in feature selection and extraction are discussed. The proposed work makes use of good evaluation metrics based on precision, recall and f-measure. The corpus made use of in the research is explained in detail in this chapter. The following Chapters namely 4, 5 and 6 introduce and discuss the automatic text summarization models proposed and utilized in this research.