Rough Set Based Text Summarization

A dissertation
submitted in partial fulfilment of
the requirement for the award of degree
of

MASTERS OF TECHNOLOGY
in
COMPUTER APPLICATIONS

Submitted by
Nidhika Yadav
2005JCA2419

under the guidance of
Dr. Niladri Chatterjee

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY, DELHI
May 2007

Certificate

This is to certify that the dissertation entitled "Rough Sets based Text Summarization" submitted by "Nidhika Yadav", 2005JCA2419, for partial fulfilment of the requirements for the award of degree of Master of Technology in Computer Applications, at Indian Institute of Technology, Delhi is a record of bonafide work carried by her under my supervision and guidance. The work carried out in this project has not been submitted elsewhere, wholly or in part for the award of any other degree or diploma.

Dr. Niladri Chatterjee
Department of Mathematics,
Indian Institute of Technology, Delhi.

Acknowledgment

I would like to express sincere gratitude to my project guide, Dr. Niladri Chatterjee, Department of Mathematics, IIT Delhi, for his valuable guidance, support and constant encouragement throughout the course of the work. With his timely guidance, I was able to solve most problems that I faced in the project. He goes through my work with full patience and has always motivated me to think positive whenever the results were not coming up to mark. Without his constant guidance and help, this project would not have been successful.

I would also like to express thanks to my family members who have been inspiring and motivating all through my work.

I take this as an opportunity to thank all the people who are directly or indirectly involved in my work.

Nidhika Yadav

2005JCA2419

Contents

ABSTRACT

Rough Set is a mathematical tool to discover patterns hidden in data. It deals with the "decision" of "selecting" discerning attributes and objects. Text Summarization, on the other hand, is an NLP application that deals with the "decision" of "selecting" the main idea covered in the text. Typically, automatic text summarization is carried out through sentence extraction. Sentence extraction involves the selection of important sentences from the text, producing them as it is in the summary. Consequently, this can be viewed as a decision-making problem in which we have to decide which sentences will be part of the summary. The project aims at exploring the applications and development of Rough Set based models for Text Summarization.

Chapter 1

Introduction

Summarization is the process of condensing a source text into a shorter version preserving its information content. It involves determination of the main theme being conveyed in the text, selection of parts of the text that convey the main theme, and presenting the most important content to the user in a condensed form. Summarization systems are becoming important with the increase in amount of text data available through the web. It enables us to capture the relevant contents of text according to user needs. A user may want general view, central view or some specific view of the text. Various summarization approaches have been studied for different text data such as articles, mails, research papers. These approaches have been classified [3] as:

1. **Extraction and abstraction based**

   Extraction based methods refer to selection of important sentences based on some heuristics and producing them as it is in the summary. Abstraction based methods reconfigures the selected sentences using parsing techniques; it requires stronger understanding of the text, use of deep linguistic knowledge for rephrasing the sentences so as to keep them meaningful as well as preserving the information contents.

2. **Top-down and bottom-up approach**

   Top-down approach is keyword driven. Here keywords given by users need to be considered and sentences are extracted based on these words. Bottom-up approach,

involves finding the main concept covered in the text, then applying the top-down approach to it. This corresponds to the generic view of the text.

3. **Informative and Indicative approach**

   Informative summaries give complete and minimal set of key points covered in text while indicative summaries gives glimpse of the text key points that give the most essential part. Indicative summaries are in general short in length and are used in web browsing, cellphones etc.

1.1 Existing Summarization Techniques

Summary generation is a time consuming process and demands lot of efforts. Human beings cannot make summaries of large number of documents that are to be tackled in day-to-day work. So efficient techniques need to be developed and used. Some of the existing summarization techniques are:

1. **Mutual Reinforcement Principle for Summary Generation [45]**

   This method uses sentence links to cluster sentences. Further, term and sentence saliency scores are used to find the central theme being covered in each of the clusters.

2. **QR Decomposition Method for Summary Generation [3]**

   This method considers term-sentence matrix and selects the most important sentence using column pivoting. The relative importance of the remaining sentences changes, because some of the sentences may be carrying the same information content as the selected one. The method is iterative and stopping criterion depends on the degree of summarization.

3. **Lexical Chain Based [19, 33]**

   This method takes into account the semantics of the text. It is very efficient to form the central idea of the text. Different senses of each term present in the text

are considered. For each different sense different chain is formed, the largest chain represents the concept being covered in the text.

4. **Summarist Summarizer [7]**

   It combines concept level world knowledge (ISI, University of Southern California) SENSUS, dictionaries with other robust NLP processing techniques (using information retrieval techniques). It works in three steps: (i) Word level information retrieval techniques of topic spotting using SENSUS to perform concept counting. (ii) Identifies important topics in text. Concept based topic fusion (interpretation) to find summarizing concepts. (iii) Lists the keywords and generate the summary.

5. **Copernic Summarizer [4]**

   This uses statistical algorithms and creates a list of important concepts. It generates summaries composed of most important sentences. Concepts that the user feels are less relevant can be removed from the list of important concepts, and the new summary can be generated.

6. **TXTRACTOR [5]**

   TXTRACTOR is implemented in java and uses sentence-selection heuristics. It ranks the text segments, producing summaries that contain a user-specified number of sentences. The summarization process takes place in three steps: (i) Sentence evaluation, (ii) Segmentation or topic boundary identification, (iii) Segment ranking and extraction. Jaccard's coefficient is used as the similarity measure.

7. **WebSumm Text Summarizer [41]**

   WebSumm Text Summarizer generates summaries via algorithm generated by Inderjeet Mani. Details of implementation are not known to us.

1.2 Aim and Scope of the Project

The aim of the project is to design efficient text extraction based summarizer using mathematical concept of rough sets, described in Chapter 2. The work is based on rough

set based text document retrieval systems.

In this work we have developed four rough set based models for sentence extraction. The first one is a supervised model. It takes from user the relevant keywords. These keywords are used to determine the categorization of the text using the indiscernibility relation, described in Chapter 2. Different keywords lead to different categorization of the same text. This scheme then selects sentences using the approximations of rough sets.

Rest of the methods create generic summaries. The user can enhance the performance of these modules by providing keywords and important sentences. However, the method can work without these requirements as well. Key terms present in the text are formed using the knowledge reduction concept of rough sets. Membership measure for each sentence is calculated with respect to each key term so found. Individually each term represent a different view of the text, the algorithm then takes the aggregate membership of each sentence over all the key terms present. The higher the value of membership measure the more is the importance of the sentence.

We have applied the algorithms on many data sets, compared their results with standard summarizers WebSumm Text Summarizer [41], MS Word Autosummarize [20] and found to be highly encouraging.

1.3 Organization of Thesis

The thesis is organized as follows: Chapter 2 discusses rough sets, reduction, dependencies in knowledge, various existing rough set based classification schemes, discretization and semantics . In Chapter 3 we discuss our models for sentence extraction. Finally, Chapter 4 talks about results, evaluations, observations and future work.

Chapter 2

Rough Sets and Text Data

The amount of information in the world is increasing at a very high rate. With this increases the problem of efficiently and effectively using the abundant information. Rough Set is a mathematical tool to discover patterns hidden in data. It deals with the "decision" of "selecting" suitable attributes and objects. There are collections of objects which may not be defined using a given knowledge, Rough Sets allows us to view them *approximately*. We have discussed the problem of *suitable* attribute and object selection applied to text data using the Rough Set Theory. Rough Sets are highly used in various domains like medical data analysis and disease diagnosis [22], information retrieval [23, 31, 35, 36], text mining [1, 2, 16, 17, 29], economic and financial prediction [39]. The advantage of using Rough Sets is that given a knowledge base the facts hidden in data are extracted out. There is no need of additional information like "cut-offs(thresholds)", "interval-lengths", etc. Further, once the knowledge base is formed the problem becomes domain independent. Thus, this is a *closed-world* problem [28].

2.1 Definition

Rough set theory is an extension of set theory, and was proposed by Palwak [24] in 1982. The basis of rough sets is classification of the universe of objects into disjoint concepts. Let U be the universe of objects. A subset X of U will be called a concept in U. Typically,

it describes properties common among these set of objects. Let S be a classification of U, i.e. S partitions U into equivalence classes. We shall deal with a family of classifications, R over U. A family of classification is called a knowledge base over U. We shall refer to knowledge base as the system K = (U, R). Let P $\subseteq R$, then P induces an equivalence relation called *indiscernibility* relation, denoted by IND(P) defined as

$$[x]_{IND(P)} = \bigcap_{S \in P} [x]_S$$

where $[x]_S$ denotes the equivalence class in S containing the element x $\in U$.

We denote the equivalence classes of IND(P) by U/P and these classes are referred as concepts of P. IND(K) is the family of all equivalence relations defined in K. Let X $\subseteq U$, and R be an equivalence relation in IND(K). Then X is *R-definable* or *R-exact* if X can be defined in terms of elements of U/R; else X is *R-Rough* or *R-undefinable*. R-definable sets are those subsets of U which can be defined crisply in knowledge base K.

2.2 Approximation of Sets

Rough sets cannot be defined using available knowledge, hence are approximated by using two sets as follows:

1. *Lower Approximation* is the set of all elements of U, which can be with certainty classified as elements of X, in the knowledge R. Mathematically, $\underline{R}X = \cup\{Y \in U/R : Y \subseteq X\}$.

2. *Upper Approximation* is the set of elements of U, which can be possibly classified as elements of X, in the knowledge R. Mathematically, $\overline{R}X = \{Y \in U/R : Y \cap X \neq \emptyset\}$. We note that X is R-rough if and only if $\underline{R}X \neq \overline{R}X$.

These classifications divide the universe of objects into three regions as follows:

1. *R-positive region* of X is defined as $\mathrm{POS}_R(\mathrm{X}) = \underline{R}X$. This region consists of collection of elements of universe, which can be classified with full certainty as members of set X using knowledge *R*.

2. *R-boundary region* of X is defined as $\mathrm{BN}_R(\mathrm{X}) = \{\ \overline{R}X - \underline{R}X\ \}$. This is the undecidable region of elements of universe. This region consists of collection of elements of universe which cannot be classified to be part of X or U - X using the knowledge *R*.

3. *R-negative region* of X is defined as: $\mathrm{NEG}_R(\mathrm{X}) = \{\ \mathrm{U} - \overline{R}X\ \}$. This is the collection of elements of universe which can be confirmedly classified as members of set U-X, i.e. without any ambiguity we can say that these elements does not belong to set X, using knowledge R.

Membership Measure

In set theory an element either belongs to a set or it does not belong to it. So the corresponding membership takes values 0 or 1. In rough set theory the definition of a set is not crisp. So to measure the membership of an element in a set a rough membership function is used. It measures the degree of overlap of the set and the equivalence class to which it belongs. It is defined as follows [10]:

$$\mu_R(X)(x) = \frac{|[x]_R \cap X|}{|\ [x]_R\ |}$$

2.3 Knowledge Representation

Knowledge Representation System(KRS) is a data table labeled by attributes corresponding to equivalence relations and rows labeled by objects of the universe. We define KRS as the pair (U, A), where A is a non-empty finite set of attributes and U is the universe of objects. For every subset of attributes B $\subseteq$ A there is a binary relation, denoted by IND(B), called indiscernibility relation and defined as IND(B) = {(x,y) $\in U^2$: $\forall$ a $\in$ B, a(x) = a(y) }, IND(B) is an equivalence relation; also there is a one-to-one correspondence between knowledge base and knowledge representation system. Let C, D $\subseteq$ A be

two subsets of attributes called condition and decision attributes respectively. A KRS with distinguished condition and decision attributes is called a decision system.

Example 2.1. Let R_1 and R_2 be two classification of universe U of five elements { x_1, x_2, x_3, x_4, x_5 }, where $R_1 = \{\{x_1, x_2, x_5\}, \{x_3, x_4\}\}, R_2 = \{\{x_1, x_5\}, \{x_2, x_3, x_4\}\}$. Then, IND(R) = $\{\{x_1, x_5\}, \{x_3, x_4\}, \{x_2\}\}$. Let X = $\{x_2, x_4\}$, the R-lower approximation of X is $\{x_2\}$, R-upper approximation of X is $\{x_2\} \cup \{x_3, x_4\}$, R-boundary region of X is $\{x_3, x_4\}$ and the R-negative region of X is $\{x_1, x_5\}$. This is described in the following figure.

2.4 Reduction in Knowledge

The problem described in this section is whether all the attributes are necessary to define concepts available in the considered knowledge. This is an important issue that arises in many practical knowledge bases.

2.4.1 Reduct and Core of Knowledge

Reduct of knowledge is sufficient to define all basic concepts occurring in the considered knowledge where core is its most important part.

Let R be a family of equivalence relations and let R $\in R$. Then R is *dispensable* in R if IND(R) = IND(R - R) else R is indispensable in R. The family R is independent if for each R $\in$ R is indispensable in R, otherwise R is dependent. Q $\subseteq$ P is reduct of P if Q is independent and IND(Q) = IND(P). The set of all indispensable relations in P is called core of P denoted as CORE(P). Also, CORE(P) = $\bigcap$ RED(P), where RED(P) is the family of all reducts of P.

2.4.2 Relative Reduct and Relative Core of Knowledge

Let P and Q be equivalence relations over U. P-positive region of Q is the set of all objects of the universe U which can be properly classified to classes of U/Q by employing the knowledge expressed by classification U/P. Mathematically, $POS_P(Q) = \bigcup\{ \underline{P}X : X \in U/Q\}$. For family of equivalence relations P and Q, R $\in$ P is Q-dispensable in P if $POS_{IND(P)}(IND(Q)) = POS_{IND(P-R)}(IND(Q))$. Otherwise R is Q-indispensable in P. If every R in P is Q-indispensable then P is Q-independent. The family S $\subseteq$ P is Q-reduct of P if S is Q-independent subfamily of P and $POS_P(Q) = POS_S(Q)$. This is the minimal subset of knowledge P that provides the same classification of objects to categories of knowledge Q as the whole knowledge P. All the Q-indispensable relations in P form the Q-core of P. This is the essential part of knowledge P that cannot be eliminated without disturbing the ability to classify objects to concepts of Q.

Example 2.2. Let R = $\{A_0, A_1, A_2, A_3, A_4\}$ and let P = $\{A_0, A_1, A_2\}$ and Q = $\{A_3, A_4\}$.

U/A_0 = { $\{x_1, x_4, x_5\}$, { $x_2, x_3\}\}$
U/A_1= $\{\{x_4, x_5\}, \{x_2, x_3\}, \{x_1\}\}$
U/A_2 =$\{\{x_1, x_3, x_4$ $\}, \{x_2, x_5$ $\}\}$
U/A_3=$\{\{x_1, x_2, x_4\}, \{x_3, x_5\}\}$
U/A_4= $\{\{x_3, x_5\}, \{x_1, x_4\}, \{x_2\}\}$

Then,
U/P = $\{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$

$U/\mathrm{Q} = \{\{x_1, x_4\}, \{x_3, x_5\}, \{x_2\}\}$ And,
$\mathrm{POS}_P(\mathrm{Q}) = \{x_1, x_2, x_3, x_4, x_5\}$

Now we compute core and reduct of P w.r.t. Q.
$U/$ P-$A_0 = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$
$\mathrm{POS}_{P-A_0}(\mathrm{Q}) = \{x_1, x_2, x_3, x_4, x_5\} = \mathrm{POS}_P(\mathrm{Q})$, so A_0 is Q-dispensable.

$U/$ P-$A_1 = \{\{x_1, x_4\}, \{x_2\}, \{x_3\}, \{x_5\}\}$
$\mathrm{POS}_{P-A_1}(\mathrm{Q}) = \{x_1, x_2, x_3, x_4, x_5\} = POS_P(\mathrm{Q})$, so A_1 is Q-dispensable.

$U/$ P-$A_2 = \{\{x_1\}, \{x_2, x_3\}, \{x_4, x_5\}\}$
$\mathrm{POS}_{P-A_2}(\mathrm{Q}) = \{x_1\} \neq \mathrm{POS}_P(\mathrm{Q})$, so A_2 is Q-indispensable.

Therefore Q-core of P = $\{A_2\}$.

$U/A_2 = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$
$\mathrm{POS}_{A_2}(\mathrm{Q}) \neq \mathrm{POS}_P(\mathrm{Q})$, so A_2 is not Q-reduct of P.
$\mathrm{POS}_{A_0 \cup A_2}(\mathrm{Q}) = \{x_1, x_2, x_3, x_4, x_5\} = \mathrm{POS}_P(\mathrm{Q})$,
$\mathrm{POS}_{A_1 \cup A_2}(\mathrm{Q}) = \{x_1, x_2, x_3, x_4, x_5\} = \mathrm{POS}_P(\mathrm{Q})$,

Hence there are two reducts $\{A_0, A_2\}$, $\{A_1, A_2\}$.

2.5 Dependencies in Knowledge

Let K = (U, R) be a knowledge base and let P, Q $\subseteq$ R. Knowledge Q is derivable from knowledge P, if all concepts of Q can be defined in terms of some concepts of knowledge P denoted as P $\Rightarrow$ Q. Q depends on P if and only if IND(P) $\subseteq$ IND(Q). Q and P are equivalent if Q $\Rightarrow$ P and P $\Rightarrow$ Q. Knowledge Q and P are independent if neither Q $\Rightarrow$ P nor P $\Rightarrow$ Q.

Example 2.3. Let U/P = $\{\{x_1, x_2\}, \{x_5, x_8\}, \{x_3, x_4\}, \{x_6\}, \{x_7\}\}$ and U/Q = $\{\{x_1, x_2\}, \{x_5, x_8, x_7\}, \{x_3, x_4, x_6\}\}$, then IND(P) $\subseteq$ IND(Q), hence P $\Rightarrow$ Q.

2.5.1 Partial Dependencies in Knowledge

The dependency need not always be complete it may be partial, i.e. only part of knowledge Q is derivable from knowledge P. The partial derivability is defined using the notion of positive region of knowledge. Knowledge Q depends in degree k ($0 \leq k \leq 1$) from knowledge P,

$$P \Rightarrow_k Q, \quad if\ k = \gamma_P(Q) \ = \frac{|POS_P(Q)|}{|U|}$$

If $\gamma_P(Q) = 1$ then Q depends totally on P and elements of the universe can be classified to categories of Q using knowledge P. If $0 < \gamma_P(Q) < 1$ then Q roughly depends from P. Here only those elements of universe, which belong to the positive region, can be classified to categories of Q using knowledge P and when $\gamma_P(Q) = 0$ then Q is totally independent from P and hence none of the elements of universe can be classified to categories of Q using knowledge P. There may be decision classes which can be fully characterized by P, while others may be characterized only partially. Also the measure,

$$\gamma_P(X) \ = \frac{|\underline{P}(X)|}{|X|}$$

where X $\in U/$Q, corresponds to how the elements of each class of $U/$Q can be classified employing knowledge P.

2.6 Rough sets and Classifications in Text Documents

The basis of rough sets is classification. It allows us to classify objects into sets of equivalent members based on their attributes. We can then examine any combination of objects or even attributes using resultant classification. In a text document the classification can be performed on the basis of terms present in it or on the basis of sentences present in it. We shall discuss both the models and henceforth present our proposed model in the Chapter 3.

2.6.1 Classification Based on Terms

The universe here consists of the terms present in the text. It groups terms into equivalence classes based on synonymy relation. Typically, this method cannot work without semantic knowledge. It considers sentences and keyword list as set of terms. We shall require the relation that exists between these sets. Hence, in this section we shall study rough equalities, inclusions etc. of sets.

In Set Theory two sets are equal if they have exactly same elements. But in rough set concept we may not be able to answer the equality of sets, using the available knowledge. Two sets may have close features, which are assumed approximately equal.

Definition 2.1. Let K be a knowledge base and let X, Y $\subseteq U$ and R $\in$ IND(K). Then X and Y are bottom R-equal, X $\approx_R$ Y if $\underline{R}X = \underline{R}Y$, X and Y are top R-equal X $\approx^R$ Y if $\overline{R}X = \overline{R}Y$, further X and Y are R-equal if X $\approx_R$ Y and X $\approx^R$ Y. X is bottom R-included in Y, X $\subseteq_R$ Y if $\underline{R}X \subseteq \underline{R}Y$ and X is top R-included in Y, X $\subseteq^R$ Y, if $\overline{R}X \subseteq \overline{R}Y$.

Definition 2.2. A concept that is fully represented in a sentence is definitely discussed in the sentence. A concept that has no representation in the sentence is definitely not discussed in the sentence and a partial representation tells that a sentence possibly discusses the concept.

Some of strategies to compare two sets S and K are:

1. S = K, S and K are exactly equal.
2. S $\approx$ K, S and K are approximately exactly equal, i.e. S and K are R-equal.
3. S $\approx_R$ K, S and K are $\underline{R}$-equal.
4. S $\approx^R$ K, S and K are $\overline{R}$-equal.
5. S $\subseteq_R$ K, S is $\underline{R}$-included in K.
6. S $\subseteq^R$ K, S is $\overline{R}$-included in K.

7. S $\subseteq_R^R$, K is R-included in S.

8. K $\subseteq_R$ S, K is $\underline{R}$-included in S.

9. K $\subseteq^R$ S, K is $\overline{R}$-included in S.

10. K $\subseteq_R^R$ S, K is R-included in S.

Sentences are ranked according to the level of importance. Various similarity measures (e.g. in [23, 36]) are used for this. The model requires synonymy relation for classification and there can be a problem of multiple vocabulary views that have to be handled before we can classify the terms into a single concept. This generally require efficient techniques like clustering. We shall discuss our proposed model in chapter 3.

Example 2.4. Let $U = \{t_1, ..., t_{11}\}$, consists of all the possible terms in a text documents, and let R be the relation that partitions the universe as $U/\mathrm{R} = \{\{t_1,\ t_2,\ t_3\}, \{t_4,\ t_5\}, \{t_6,\ t_7,\ t_8\}, \{t_9,\ t_{10}\}, \{t_{11}\}\}$ where R is synonym relation. This partitions U into sets of terms such that the terms within a set are similar. Let sentence S be represented by terms $\{t_1, t_2, t_4, t_5, t_6\}$ and keyword list K by $\{t_4, t_{10}\}$. The concepts that are fully discussed in S are $\{t_4, t_5\}$ and those that are possibly discussed are $\{t_1, t_2, t_3\}$, $\{t_6, t_7, t_8\}$ and the concepts that are not at all discussed are $\{t_9, t_{10}\}$, $\{t_{11}\}$. Similarly no concept is surely discussed in K, the possibly discussed concepts are $\{t_4, t_5\}$, $\{t_9, t_{10}\}$.

2.6.2 Classification Based on Sentences

The universe here is all the sentences present in the text. It partitions the set of all sentences into disjoint concepts. It can also work without semantic knowledge. Further, it uses the indiscernibility relation. We have in our models I, II and III, performed classification on the basis of sentences present in the text. Related work has been done by Wong and Ziarko [35], Shailendra Singh [32]. We shall discuss this approach in Chapter 3.

2.7 Dimensionality Reduction

The terms chosen to represent the text play an important part in the decision table generation. A major problem that arise in dealing with text data is the high dimensionality of the attributes (terms) present. Approaches for dimensionality reduction for a general data set involves either selecting a subset of the original attributes and/or transformation of attributes. The attributes after transformation are not of the same type as the attributes in the original attribute set. They are obtained by combinations or transformations of the original ones. Stemming, lemmatizing, truncation cause transformation of the terms of text data. We shall be performing dimentionality reduction via attribute selection. *Attribute Selection* as defined by Alexios Chouchoulas [1] is a process that attempts to select a subset of features, satisfying a combination of application and methodology-dependent criteria, minimising the cardinality of the feature subset, ensuring classification accuracy does not significantly decrease; and approximating the original class distribution with the class distribution of the selected features.

Attribute reduction methods can be domain dependent or domain independent. Domain dependent methods like the elimination of stop words, removal of special characters and works on "attribute elimination" scheme. Other domain dependent methods like use of stemming, lemmatizing algorithms, morphological analysis come under "attribute transformation" schemes.

For further reduction in the number of attributes with the aim of minimum loss of knowledge, domain independent methods are used. These methods work on "attribute selection" scheme. Attribute selection from a text data require a function that has the capability to measure the importance of terms present in the text. This function can be a measure of term frequency, mutual information, information gain, gini index [42, 12] etc. For instance, consider information gain(IG), it measures the number of bits of information obtained for class prediction by knowing the presence or absence of a term in the text. IG for term t_i where there are k decision classes $c_1, c_2, ..., c_k$ is given as follows:

$$IG(t_i) = \sum_k P(c_k|t_i) * log\frac{P(c_k|t_i)}{P(c_k)} + \sum_k P(c_k|\overline{t_i}) * log\frac{P(c_k|\overline{t_i})}{P(c_k)}$$

Further, the strategies to use these measures may vary depending on the relative effect of terms on each other. Best individual attributes method evaluates all the terms individually according to a given measure and selects the best k (pre-specified) terms. These methods are fast, efficient and simple. However, they evaluate each term separately and completely ignore the existence of other terms and the manner in which the terms affect each other. The methods considering the dependencies within a set of terms(at a time) are based on forward selection and backward elimination schemes. Forward selection algorithms start with an empty set of attributes and add attributes one at a time until the final attribute set is reached. Backward elimination algorithms start with an attribute set containing all attributes and removes less significant attributes one at a time. Addition (deletion) of attributes is performed on the joint effect of the attributes as opposed to best individual attributes method.

2.7.1 Rough Sets Based Attribute Selection

We have already discussed about core and reducts in section 2.4 and about dependencies in knowledge in section 2.5. In this section, we shall discuss how these concepts can be used for attribute selection. The concept of reducts and core are important and highly used since they involve searching for attribute subset that are minimal, satisfying some consistency criterion. Some applications are: (i) Reducing size of decision table, (ii) Evaluating relative importance of attributes, (iii) Comprehending the meaning of data in decision table [28], (iv) Generating classifiers for data [2], (v) Rule generation [18], (vi) Decision tree generation [6].

For a given decision table reducts are not uniquely described. Reduct generation algorithms can be broadly classified as follows [28]:

1. Exact Algorithms

   These algorithms are capable of generating all reducts from a decision table. It involves finding all possible subsets of the attribute set. Then retaining those subsets

that spans the attribute space and are minimal. Further, one or many of these may be chosen depending on the application.

2. Approximate Algorithms

   These algorithms do not generate all possible reducts. An approximate solution is a set of attributes that achieves or exceeds some required consistency level. This means that the computed single solution may be a superset or a subset of the actual solution.

We are concerned with approximate algorithms. A general approximate attribute selection algorithms using measure function M is described as follows (a slight variation of [12]):

Step 1 Initialization of the set S, the set of terms selected by the algorithm. Typically, it is initialized to empty set or some minimum number of attributes which surely belong to the final output S.

Step 2 Pre-Computation for all terms t not in S, compute M(t).

Step 3 First Word Selection. Selecting term that optimizes M(t).

Step 4 Remaining Term Selection. For all s $\in$ S compute the cumulative measure M(s,t) for each term t not in S. Repeat until the desired number of terms are selected or some satisfactory level is reached.

The function M can be information gain, mutual information, degree of dependency etc. This algorithm is a forward selection algorithm but its main drawback is the stopping criterion which is user dependent. We have a variant of the quick reduct algorithm given by Li [16] in which we have used the core as the starting set instead of the empty set. The algorithm is given below.

Quick Reduct Algorithm

Input: The set of all attributes A,
The set of all decision attributes Q.

Output: The reduct R, R $\subseteq$ A

Initialize R to CORE.
T $\leftarrow$ R.
Do
For each x in A−R
/*For each attribute not in R see whether it can belong to the reduct*/
If $\gamma_{R\cup\{x\}}(Q) > \gamma_T(Q)$
/*Add x if the combined dependency of T $\cup$ $\{x\}$ is greater that of T alone and in this case this value is chosen to be maximum of all possible x in A−R */
Then T = R $\cup$ {x}
R $\leftarrow$ T
Until $\gamma_R(Q) = \gamma_A(Q)$
/* proceed till all the elements classified by original knowledge base are classified */
return R.

2.8 Discretization

A number of real life problems involve data table consisting of numerical values. Rough set based algorithms require categorical data. A transformation is required to convert the numerical attributes of the data table into categorical attributes preserving the meaning of data.

Discretization is a process of determining partition of attribute domain into intervals such that a consistent decision table is obtained. The discretization step determines how thickly we want to view the data. It involves finding cut-off points that define intervals and hence we convert the decision table into the one with categorical attributes. Any set of cuts define a partition of numeric values into intervals in such a way that values from one interval are not discernible. Discretization is necessary and helpful for a number of problems that exist naturally. The various Discretization methods are:

1. Local versus Global methods: Local methods discretize localized regions of the universe of objects like in decision trees. While in global methods each attribute value set is partitioned into intervals independent of other attributes.

2. Supervised versus Unsupervised methods: Unsupervised methods do not make use of decision values while the other one does.

3. Static versus Dynamic methods: Static methods perform one discretization pass for each attribute and determine the maximal number of cuts for this attribute independent of other attributes. Dynamic methods are realized by searching through the family of all possible cuts for all attributes simultaneously.

2.8.1 Non Rough Set Based Discretization

Some of the known Discretization techniques are Equal Width, Equal Frequency, Holte's 1R Discretizer, Entropy Methods [21], MD-Heuristic Algorithm [21, 24]. They are described as follows:

1. Equal Width Interval Discretization

   In this method the domain of values, $[v_{min}, v_{max}]$ of an attribute a is divided into k_a intervals where k_a is a user given parameter. The interval width is $\delta = (v_{max} - v_{min})/k_a$. The interval boundaries i.e. the cut points are c(i) $= v_{min}$ + i δ, i = 1, 2, ...,$k_a - 1$.

   Criticism:

The method is unsupervised and does not consider decision attributes. Hence may lead to inconsistent decision tables. The method require the user to specify number of intervals. Further, when the difference between the parameter and the distinct values of the attributes is significant, the resultant discretization would not be encouraging.

2. Equal Frequency Interval Discretization

   In this method values of attribute a, (total distinct values for attribute a being n(a) say v_1, v_2,...,$v_{n(a)}$) are sorted and divided into k_a intervals, where k_a is a user given parameter. Each interval contains $\lambda = \lceil (n_a/k_a) \rceil$ values. The cut points are c(i) = ($v_{i\lambda}$ + $v_{i\lambda+1}$) / 2, i = 1,.., k_a-1.

   Criticism:

   It is an unsupervised method. The method also requires the user to specify number of intervals. The method is better than the previous one as it considers the distinct values taken up by each attribute.

3. Holtes One Rule Discretization

   In this method the attribute domain is divided into intervals greedly. Each interval contains majority of objects from one decision class, with the constraint that the interval must contain atleast n (user given parameter) values, this is the minimum number of values in the intervals.

   Criticism:

   The method takes into consideration the decision classes, still, it requires a user given parameter at the initial step. The stopping criterion is to be decided using some optimization technique.

2.8.2 Rough Set Based Discretization

Keeping in mind the problems discussed above, we are using Maximal Discernibility (MD) Heuristic Algorithm [21, 24, 31], a Rough Set based method which not only discretizes but

also finds the most discerning attributes. Attributes in our problem are terms filtered from the given text data. The algorithm searches for minimal set of cuts discerning maximal number of attributes. It does not require any user given parameter and is supervised in nature, leading to consistent decision tables. The cuts for MD-Heuristic algorithm are defined below:

Definition 2.3. A *cut* corresponding to an attribute a and an interval $[i_1, i_2)$ is defined as pair (a,c) where c is the middle point of the interval.

Definition 2.4. *Binary discernibility matrix*(DM) is constructed from the data table as follows: The rows of the DM consist of pair of objects with different decision classes. Objects in our problem are sentences present in the given text data. The columns of the DM consists of all possible cuts corresponding to the attributes present in the data table. The value for the cut (a,c) and discerning objects (i, j) is v, where v = 1, if the cut has the capability to discern the object i and j, i.e. min{a(i), a(j)} < c < max{a(i), a(j)} otherwise v = 0.

The algorithm is described below:

MD-Heuristic Algorithm

Step 1. Arrange the distinct values for each attribute in ascending order.

Step 2. Find cuts for all interval pairs and for all attributes found in the previous step.

Step 3. Construct the binary *Discernibility Matrix*.

Step 4. Choose a coloum from the Discernibility Matrix, i.e. from the set of all possible cuts, the cut discerning the maximum number of pairs of objects from different decision classes is chosen.

Step 5. Delete from the matrix the coloum chosen in step 4 and all the rows marked in this coloum by one. These rows correspond to the pair of objects which can be discerned with this cut.

Step 6. If Discernibility Matrix is non-empty i.e. if the all rows of the matrix are not covered by some attribute. Then repeat the step 4 and 5 for the reduced matrix else minimal set of cuts are found.

Step 7. The numeric data table is converted into discretized data table as follows:

The attributes that are present in the minimal cut set are considered remaining ones are discarded. For each attribute a in the cut set, arrange all corresponding cut values for this attribute in ascending order and assign value 0 to all objects with numerical value less than first cut value c_0, assign value 1 for all objects with values in range $[c_0, c_1)$ and continue this for all the cuts.

2.8.3 Example and Effect of Change of Data

In this section we shall discuss the effect of variation of data in the data table. The data is subject to the change in the values of conditional attributes and change in decision attributes. These changes effect the attributes and cuts extracted from the discernibility matrix. It also reveals that the important concepts of a text being extracted are a function of the knowledge base considered.

Consider the following data table.

	A_0	A_1	Decision
Object 0	1.2	3.3	1
Object 1	1.3	7.8	0
Object 2	2.5	8.8	0
Object 3	1.5	4.0	1
Object 4	1.8	4.8	1

Table 1: Initial Data Table

The attributes present in the decision table are A_0 and A_1. The values taken up by the attribute A_0 are 1.2, 1.3, 1.5, 1.8, 2.5 and by A_1 are 3.3, 4.0, 4.8, 7.8, 8.8. The set of intervals obtained are: [1.2, 1.3), [1.3, 1.5), [1.5, 1.8), [1.8, 2.5), [3.3, 4.0), [4.0, 4.8), [4.8,

7.8), [7.8, 8.8). The cuts are (0, 1.25), (0, 1.4), (0, 1.65), (0, 2.15), (1, 3.65), (1, 4.4), (1, 6.3), (1, 8.3). The pair of objects with different decision classes are (0, 1), (0, 2), (1, 3), (1, 4), (2, 3), (2, 4). The discernibility matrix constructed is as follows:

	(0, 1.25)	(0, 1.4)	(0, 1.65)	(0, 2.15)	(1, 3.65)	(1, 4.4)	(1, 6.3)	(1, 8.3)
(0, 1)	1	0	0	0	1	1	1	0
(0, 2)	1	1	1	1	1	1	1	1
(1, 3)	0	1	0	0	0	1	1	0
(1, 4)	0	1	1	0	0	0	1	0
(2, 3)	0	0	1	1	0	1	1	1
(2, 4)	0	0	0	1	0	0	1	1

Table 2: Discernibility Matrix

The cut obtained after applying the algorithm is (A_1, 6.3), this is the attribute value pair required to discern the decision classes and maintaining the consistency of the decision table. The objects are classified as: for values in range 3.3 - 4.8 of attribute A_1 with decision class 1 and for values 7.8 - 8.8 of attribute A_1 with decision class 0.

We noticed that the final cuts and discretization depends and varies with the data in decision table. With the same decision classes and attributes but different data values the cuts and output of the discretized values changes. This is illustrated below:

Consider the following data table.

	A_0	A_1	Decision
Object 0	1.2	3.3	1
Object 1	1.3	7.8	0
Object 2	2.5	3.3	0
Object 3	1.5	4.0	1
Object 4	1.8	4.8	1

Table 3: Initial Data Table

The discernibility matrix constructed is as follows:

	(0 , 1.25)	(0 , 1.4)	(0 , 1.65)	(0 , 2.15)	(1 , 3.65)	(1 , 4.4)	(1 , 6.3)
(0 , 1)	1	0	0	0	1	1	1
(0 , 2)	1	1	1	1	0	0	0
(1 , 3)	0	1	0	0	0	1	1
(1 , 4)	0	1	1	0	0	0	1
(2 , 3)	0	0	1	1	1	0	0
(2 , 4)	0	0	0	1	1	1	0

Table 4: Discernibility Matrix

The cuts obtained after applying the algorithm is (A_0, 1.4), (A_1, 3.65), this is the attribute value pair required to discern the decision classes and maintaining the consistency of the decision table. The objects are classified as: for values in range A_0 (1.2 - 1.4) $\wedge$ A_1 (3.3 - 3.65) $->$ decision class 1 and for A_0 (1.4 - 2.5) $\wedge$ A_1 (3.65 - 9.8) $->$ decision class 0. The final discretized table is as follows:

	A_0	A_1	Decision
Object 0	0	0	1
Object 1	0	1	0
Object 2	1	0	0
Object 3	1	1	1
Object 4	1	1	1

Table 5: Discretized Data Table

This concept is used by the different term weighting and normalization schemes possible. The decision classes remains the same and the data table changes and this change is reflected in change of the cuts extracted by the algorithm.

Consider the following data table.

	A_0	A_1	Decision
Object 0	1.2	3.3	0
Object 1	1.3	7.8	1
Object 2	2.5	3.3	1
Object 3	1.5	4.0	1
Object 4	1.8	4.8	0

Table 6: Initial Data Table

The discernibility matrix constructed is as follows:

	(0 , 1.25)	(0 , 1.4)	(0 , 1.65)	(0 , 2.15)	(1 , 3.65)	(1 , 4.4)	(1 , 6.3)
(0 , 1)	1	0	0	0	1	1	1
(0 , 2)	1	1	1	1	0	0	0
(0 , 3)	1	1	0	0	1	0	0
(1 , 4)	0	1	1	0	0	0	1
(2 , 4)	0	0	0	1	1	1	0
(3 , 4)	0	0	1	0	0	1	0

Table 7: Discernibility Matrix

The cuts obtained after applying the algorithm is (A_0, 1.25), (A_0, 1.65), (A_0, 2.15). These are the attribute value pair required to discern the decision classes and maintaining the consistency of the decision table. This shows that change in decisional attribute affect the output of the algorithm. The decision attribute can be generated via various techniques. We have used heuristics for it and this can be done using a supervised version as in IR techniques.

Now, we shall see how cuts extracted depends on the relation between the conditional and decisional attributes. The main aim of the MD heuristic algorithm is to maintain the consistency of the decision table. The decision table itself can be inconsistent. Inconsistency arise due to the classifiers or the user ratings used. Two sentences may state different views, still their vector representation may be identical (depends the terms considered as candidate for decision table attributes). We shall illustrate this with the following

example.

	A_0	A_1	Decision
Object 0	1.2	3.3	0
Object 1	1.3	7.8	1
Object 2	2.5	3.3	1
Object 3	1.8	4.8	1
Object 4	1.8	4.8	0

Table 8: Initial Data Table

Note that last two objects have the same conditional attributes but different decisional attributes. The decision table is inconsistent. There is an entry in disernibility matrix corresponding to last two rows but no cut can discern these two objects.

The cuts obtained from the data table are:(0, 1.25), (0, 1.55), (0, 2.15), (1, 4.05), (1, 6.3). The Discernibility Matrix is

	(0 , 1.25)	(0 , 1.55)	(0 , 2.15)	(1 , 4.05)	(1 , 6.3)
(0 , 1)	1	0	0	1	1
(0 , 2)	1	1	1	0	0
(0 , 3)	1	1	0	1	0
(1 , 4)	0	1	0	0	1
(2 , 4)	0	0	1	1	0
(3 , 4)	0	0	0	0	0

Table 9: Discernibility Matrix

The algorithm considered will enter an infinite loop (Step 6, all rows of discernibility are not covered). Rough Sets have its own way to deal with inconsistencies [25]. We have modified the algorithm suitably.

2.9 Semantics and Text Data

Text carries part of the semantics. For example, in a news article, each sentence tell a part of the news. Text data processing beyond syntactic analysis (these methods do not take into account the links that exists between the semantically related terms) require semantic information. In order to carry out semantic study of the text certain prerequisites like lexicons and other knowledge bases are required.

Lexical Chains [15, 19, 33] are set of terms that are semantically related and have a sense flow. Lexical chains are computed from a text source by collecting a set of terms that are semantically related via the relations like identities, synonyms, hypernyms, hyponyms. Synonyms are different terms with identical meanings and can be used interchangeably. For example, student and pupil. A term a is a *hypernym* of another term b if a's meaning cover the meaning of b like vehicle is a *hypernym* of train, airplane, car. A hyponym is a term whose semantic range is included within that of another word as white, blue, purple are hyponyms of color. Each chain represent an individual concept covered in text.

The chaining algorithms are domain (semantic vocabulary) dependent. The study of Lexical chains is important for resolving term ambiguity in a particular text and for providing information for determining the meaning of text. Lexical chains are an intermediate representation of a text. Typically, they are not used directly in any application. We have used these chains in two ways (i) as weighting measure for terms, (ii) for generating synonymy relation. The chaining algorithms can be greedy chaining algorithm where each addition of a term t to a chain c is based only on those words that occur before it in the text. Non-greedy algorithm assigns a term to a chain only after looking at all possible combinations of chains that can be generated from the text. Terms having multiple senses are added into different chains. Each term is kept in only one chain, the one to which it contributes the most. Non-Greedy Algorithm by Silber, McCoy [30, 33] to build Lexical Chains:

Lexical Chaining Algorithm

Step 1. Select the set of candidate terms.

Step 2. Build all possible chains.

- For each term t_i find its all possible senses.
- For each sense s_{ij} place the term into every chain for which it has an identity, synonym, hypernym, hyponym relation and assign a weight associated with the relation(a higher weight for a higher lexical similarity).

Step 3. For each term t_i keep the chain to which it contributes the most and hence selecting the best possible chain for each term.

The semantic lexicon used is WordNet. It groups English words into sets of synonyms called synsets, provides short definitions and records the various semantic relations between these synonym sets.

With this theoretical background we are going to describe our proposed models in the following Chapter.

Chapter 3

Design of the System

In this Chapter we shall describe the Rough Set based models that we have proposed and developed. We have developed four rough set based models for text extraction: Model I, Model II, Model III and Model IV and experimented with their variants.

3.1 Description of modules of Model I

This model gives user-specific extracts. It uses the keywords given by the user as the basis of classification. Ones the sentences are classified we find the approximations and determine the extracts. The basic steps involved in this model are as follows:

Step 1: *Text preprocessing.*
Step 2: *Formation of information system.*
Step 3: *Formation of decision system.*
Step 4: *Selection of attributes.*
Step 5: *Finding the approximations.*
Step 6: *Generation of extracted summary.*
The steps are described below:

3.1.1 Text Preprocessing

This is a key step in summarization. We take the input of the text and separate sentences, from sentences we used only nouns, as nouns give the "aboutness" of the text. We used QTAG Parser [27] which is a probabilistic parts-of-speech tagger. All terms are converted to singular. For example "butterflies" → "butterfly", "trees" → "tree", "boxes" → "box".

3.1.2 Forming the Information System

The need of forming the information system is to represent the knowledge base and effectively apply the theory of rough sets on to it. The information system is I = (U, A). U is the universe of all sentences present in the text. A is the set of all attributes, which initially consists of all the terms in the text as it is terms in the text that describe it. The entry for attribute (here term) j and for sentence i depends on the presence or absence of the term j in sentence i. The entry will be 1 if the term is present else it will be 0.

3.1.3 Forming the Decision System

The decision table is (U, A ∪ decision attribute). The decision attribute is a binary valued attribute. Heuristics used for determining decision attribute are:

- **Sentence positions** [5, 7]: The first and the last sentences are considered important and are always included in the decision system though in many cases they are not extracted by our models, as they may not turn out to be comparatively important.
- **Presence of Keyword:** The presence of keywords directs the summary and the sentences containing keywords are considered important. Apart from the sentences containing keywords the sentences that contain the other terms present in such sentences also become important as this leads to links or connectivity between the sentences.

- **Presence of emphasizing terms** [5, 7]: Terms like *therefore, concluding, thus, significant, observe, notice, however, design, implementation, significantly, future, note, finally, effect, henceforth, conclude* etc. emphasize on some concepts. Presence of such terms signifies some important points, which are being covered and talked about in the text.

- **Sentence Importance:** In order to improve our decision-making we have incorporated in our model user based preferences. There are some sentences that the user may feel being important these sentences are marked important.

- **Lexical connectivity** [17]: Lexical connectivity is the number of terms shared with other sentences. Sentences that share more term with other sentences are more important. If a sentence is considered as important then the importance level of all the terms present in the sentence also increases. We in our system have a count measure of the level of importance of a sentence. A sentence that is important has all its terms as candidates for being important. If these terms are present in other sentences the importance level of those sentences are increased. The sentence that shares the maximum number of important terms is most important.

3.1.4 Selection of Attributes of the Knowledge Base

The information table so formed is highly sparse. As the size of the document (in particular the number of terms) increases the sparseness increases. If all the terms present in the information system are considered for classification, then the equivalence classes so formed are either singleton sets or are sets of finite small orders and is not able to represent the relationships that exist between the sentences. Different terms when taken in isolation reflects different classification of the same text. In the first model we have therefore incorporated a user based classification scheme in which the text is classified according to the keywords K entered by the user.

3.1.5 Finding the Approximations

(U, K) gives the classification of the sentences into equivalence classes. The system finds the lower approximation that will always be the part of the summary. Then finding the boundary region. Selecting the sentences from the boundary region depending on the rough membership measure and the degree of summarization. The membership function used is:

$$\mu_R(X)(x) = \frac{|[x]_R \cap X|}{|\ [x]_R\ |}$$

It describes the degree of membership of a sentence to a decision class.

3.1.6 Generation of summary

The sentences within an indiscernibility class are arranged according to the lexical connectivity of the sentences and the important sentences are retrieved and displayed in the order of their occurrence in the original text, to maintain coherence in summary.

Model I uses keyterms given by the user. In the following models we shall be using keyterms generated automatically. Hence the summaries produced are generic summaries.

3.2 Description of modules of Model II

This model gives generic summaries. It looks at the overall influence of all the discerning terms present in the text. This model can be made user specific via keyword, and user based sentence preference module. This allows the system to improve the decision making about certain sentences. This model is based on an information retrieval model, (see [31]), which takes the user based preferences and classifies the document to appropriate class. We have adapted this model to suite our requirements of sentence extraction process. We assign roughness measures to each sentence, rather than assigning classes, as is done in

usual information retrieval problems. This has been done so as to arrange the sentences according to the level of importance. The classes on their own would not be able to distinguish between the sentences which are in the same class so would lead to ambiguity. The steps involved in this model differ in step 4 and step 5 of Model I, which now looks as:

Step 4: *Reduction of Knowledge by finding discerning terms.*

Step 5: *Calculating aggregate membership measure.*

Other steps remain the same. The steps of Model II that differ from Model I are discussed here.

3.2.1 Reduction in Knowledge

We find the discerning terms using reducts. There are many effective attribute selection methods discussed before. We shall use rough set theory for reducing the dimensionality of the knowledge base. Since the core is contained in every reduct, we used it as a basis for construction of reducts. Hence used the reduct algorithm given in section 2.7.

3.2.2 Aggeragate Roughness Measure for Sentences

Once the features are selected, we can use these features for rule generation via rough sets. Each term in the reduct is now used to measure the sentence and its importance. Each term partitions the universe in a particular way. The membership measure of each sentence is computed with respect to each discerning term. However to get a generic view of the text, aggregate effect of all these discerning terms should be considered. Hence we use the aggregate rough membership measure defined as follows:

$$\mu(X)(x) = \sum_{R \in REDUCT} \frac{|[x]_R \cap X|}{|\,[x]_R\,|}$$

Finally the extracted sentences are displayed in order of occurrence in the text to preserve coherence.

3.3 Description of modules of Model III

The study of rough set depends a lot on the knowledge base. The better the knowledge base the more we can infer from the decision table. The decision table considered so far was binary decision table, in which its (i,j)th entry t_{ij} is 1 if term j appears at least once in the sentence i else it is 0. We have not utilized the fact that other weighting schemes are also possible. The advantage of using the binary weights was that the decision table had categorical attributes and hence the rough set based techniques could be directly applied onto it. We have experimented on the some weighting techniques [11, 44].

3.3.1 Weighting Schemes

The weighted term frequency vector corresponding to sentence i is L(t_{ij}) * G(t_{ij}) where L(t_{ij}) is Local Weighting for term j in sentence i and G(t_{ij}) is Global Weighting for term j in sentence i, [44]. Local Weighting Schemes for a sentence takes into the account the local importance of a term. Some of the local weighting schemes are:

1. Binary Weight: L(t_{ij}) = 1 if term appears atleast once in the sentence.
2. No Weight: L(i) = tf(t_{ij}), where tf(t_{ij}) is the number of times term i occurs in the sentence.
3. Logarithmic Weighting: log(1 + tf(t_{ij})).

Global Weighting Scheme for text is the Inverse Document Frequency Weight here G(i) = log(N/n(i)) where N is the total number of sentences in the document and n(i) is the number of sentences that contain the term. For example if a term is present in all the sentences then its global weight will be log(N/N), which is zero, it should be since a term which is present in all the sentences have no discriminating power. Suitable normalization has been done everywhere.

The local weight of term frequency favors terms that frequently appear in a particular sentence while the global weights direct towards the importance in the whole text. For

every sentence the weight of all terms are normalized. Hence longer sentences will not be given extra preference over the shorter sentences because of the number of terms in them.

3.3.2 Semantic Weighting Scheme

The weighted term frequency vector corresponding to sentence i is L(t_{ij}) * G(t_{ij}) as described in section 3.3.1. This section deals with some local weighting schemes taking into account the semantic knowledge. Related work, [30], has been modified to suite our problem.

- For each sentence compute the sum of the weights of its terms and the total weight of terms in the document. The semantic weight in the table corresponding to term j present in sentence i is given by:

$$t_{ij} = w_j * \frac{\sum_{term\ k\ in\ sen\ i} w_k}{\sum_{over\ all\ sentences} \sum_{term\ k\ in\ sen\ i} w_k},$$

  where w_j is the lexical weight of term j.

The algorithm differs from the previous algorithm only in the attribute selection stage. Attributes are selected using MD heuristic algorithm discussed in section 2.8.2.

3.4 Description of modules of Model IV

This model is based on Summarization via *Term Based Classification Scheme*. It can give generic or user driven summaries. It is based on information retrieval model, (see [23, 36]). We have adapted this model to suite our requirements of sentence extraction process. It considers all *sentences* and *keyterms* as a collection of objects(terms in our case). Further, rough set based inclusion, intersections (section 2.6.1) can be applied.

Step 1: *Text preprocessing*: Universe here consist of the terms present in the text. All the terms are extracted in this step. Further, the sentences are considered as set of terms.
Step 2: *Formation of synonymy relation*: This step is discussed below.
Step 3: *Finding keyterms*: This step involves finding the terms that describe the text. This can be user-defined or based on some attribute selection algorithm. We have used quick-reduct algorithm to generate the keyterms present in the text.
Step 4: *Finding rough similarity measures of sentences.*
Step 5: *Generation of extracted summary.*

A knowledge base is assumed. This scheme does not require decisional attribute. It requires synonymy relation. The synonymy relation can be constructed via techniques like:

1. Clustering: It finds groups of similar terms. Terms within a group should share some common behavior. Different groups are distinct, as much as possible.
   Criticism: It requires user given parameters like number of clusters, threshold etc.

2. Build-in knowledge base: The knowledge base consists of a number of files. Each file consisting of synonymous terms.
   Criticism: The method is *static*. It does not take into account the senses of words. The senses are domain dependent. For example, the term play can mean (i) play - a theatrical performance of a drama (ii) play - a deliberate coordinated movement requiring dexterity and skill.

We have used knowledge base constructed via lexical chains. We note the property that the chains formed from the text are such that each term is present in one chain only. This accumulates similar terms together.

This model can be made user specific via user given keywords. It gives relative importance of sentences in comparison of *keyterms*.

We assign similarity measures to each sentence. This has been done so as to arrange the sentences according to the level of importance. Let R be the knowledge base considered.

Before defining the similarity measures used we define the lower similarity and upper similarity. The lower similarity for a sentence S and keyword K is:

$$\underline{SIM}(S,K) = \frac{|\ \underline{R}S \bigwedge \underline{R}K\ |}{|\ \underline{R}S \bigvee \underline{R}K\ |}$$

Similarly, the upper similarity for a sentence S and keyword K is:

$$\overline{SIM}(S,K) = \frac{|\ \overline{R}S \bigwedge \overline{R}K\ |}{|\ \overline{R}S \bigvee \overline{R}K\ |}$$

We have used two different similarity measures. The first one is a slight modification of the similarity measure by Das-Gupta [23]. It is given as follows::

$$SIM_1(S,K) = \frac{(\underline{SIM}(S,K) + \overline{SIM}(S,K))}{2}$$

The second one is our proposed similarity measure. It is given as follows:

$$SIM_2(S,K) = \frac{2}{(1/\underline{SIM}(S,K)) + (1/\overline{SIM}(S,K))}$$

We have proposed this measure, keeping in mind that sentence S_1 is more important than sentence S_2 if it has higher lower as well as higher upper similarity. The proposed similarity measure is high when both the lower and upper similarity measures are high. This is in contrast to the usual mean used by Das-Gupta, which is high when any one of the constituents are high.

Chapter 4

Evaluation and Results

4.1 Evaluation of Summaries

There can be many candidate summaries of a given text. Evaluating the goodness of a summary is not a well-defined and well-understood problem. This problem can be viewed as *macro* or a *micro* level problem. At macro level sentences are considered as units of comparison, while at micro level, individual terms of the summary are considered. We list here some of the measures that we considered along with their merits and demerits.

1. Objective evaluation by comparing with already-generated summary. This is done by comparing the number of common sentences by two processes. However this measure depends upon the available summary, which may vary from algorithm to algorithm.

2. A human reader can evaluate the summaries. This gives good results as not only does it check for coverage of a topic, but also the coherence and problems like pronoun resolution, which are present in most of the extract based summaries. Evidently, this has a subjective component in the final evaluation score.

3. Summaries can be evaluated in terms of standard measures of precision and recall. This can be performed both at macro and micro levels. These methods have good

graphical interpretation. Precision is the ability of the algorithm to retrieve sentences that are relevant. Recall is the ability to find the relevant sentences that are retrieved. These measures are highly used in NLP. The precision and recall when comparing summary $SUMM_2$, generated by algorithm to reference summary $SUMM_1$ is given as:

$$P = \frac{|SUMM_1 \cap SUMM_2|}{|SUMM_2|}, \quad R = \frac{|SUMM_1 \cap SUMM_2|}{|SUMM_1|}$$

A higher recall and a lower precision returns most relevant sentences but has excess of irrelevant sentences. A higher precision and a lower recall returns relevant sentences but misses some importance sentences. F-measure takes into account the relative importance of both precision and recall, it is given as:

$$\frac{2 * P * R}{P + R}$$

A good summary should have a higher value of F-measure.

4. Evaluation requires finding appropriate definition of intersection, as can be seen from recall and precision. Similarity score can be used for scoring a generated-summary with respect to a standard-summary. Similarity score is proportional to number of matching terms. Matching terms in a right order gives a higher score. There are unigram and n-gram measures to calculate the similarity in terms of intersection. This is a micro level solution since the generated-summary is viewed as collection of *ordered* terms. Related work has been done in the area of Machine Translation [13, 14].

Hence in our scheme we proceed as follows: Evaluation is done by comparing the number of common sentences with Standard summarizers e.g. WebSumm Text Summarizer [41] and MS Word Autosummarize [20] and using standard recall, precision and F-measures.

4.2 Results

We have experimented on various data sets including CNN news articles. The combinations of local and global weighting schemes that we have studied are:

1. $t_{ij} = L(t_{ij}) * G(t_{ij}) / \sum_{t_{ij} \in Sentence} t_{ij}$, $L(t_{ij})$ = binary weight, $G(t_{ij})$ = inverse document frequency.

2. $t_{ij} = L(t_{ij}) * G(t_{ij})) / \sum_{t_{ij} \in Sentence} t_{ij}$, $L(t_{ij}) = tf(t_{ij})$ frequency weight, $G(t_{ij})$ = inverse document frequency.

3. $t_{ij} = L(t_{ij}) * G(t_{ij}) / \sum_{t_{ij} \in Sentence} t_{ij}$, , $L(t_{ij}) = log(1 + tf(t_{ij}))$, $G(t_{ij})$ = inverse document frequency.

4. $t_{ij} = L(t_{ij}) * G(t_{ij}) / \max_{t_{ij} \in Sentence} t_{ij}$, , $L(t_{ij}) = log(1 + tf(t_{ij}))$, $G(t_{ij})$ = inverse document frequency.

4.2.1 *Experiment 1*

The results for the following data are being discussed in this section.

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Supply Chain Management is supported by modular software applications that integrate activities across organizations, from demand forecasting, product planning, parts purchasing, inventory control, manufacturing, product assembly to product distribution. Supply Chain Management is a social or soft system which has goals, components, processes and boundary. Goals of Supply Chain Management to reduce inventory cost, to increase sales to improve the coordination and the collaboration with suppliers, manufacturers and distributors. Components of Supply Chain Management System - The components of an supply chain system consists of supply chain software and hardware,

supply chain business processes and users of Supply Chain Management system. Supply Chain Management Software and Hardware - The core of an Supply Chain Management system is Supply Chain Management software. Supply chain software is module based application. Each software module automates business activities of a functional area in the supply chain. UNIX is the most common operating system for running Supply Chain Management software Business Processes - Business processes of supply chain includes supply chain planning, execution and collaboration and operational control. Users - The users of Supply Chain Management systems are workers of supply chain participants at all levels. Processes of Supply Chain Management: Demand Planning and Forecasting - A critical success factor to supply chain management is accurate demand forecasting. Supply chain software systems utilize sophisticated mathematical models for predicating future demand from historical data. Procurement - This is the process of choosing the suppliers that will deliver the goods and services you need to manufacture or assembly your products or create your services. It involves price negotiation, receiving, and verifying the shipments. Manufacturing and Assembly - Raw components are assembled into final products or raw materials are manufactured into finished goods. Distribution - This is the process of delivering your products or services to consumers. Distribution involves warehousing, delivering, invoicing and payment collection. Return - Return and refund are important parts and also the problem parts of supply chain management. The boundary of a Supply Chain Management system is the boundary of extended enterprize which includes the company, suppliers, partners, distributors and customers.

The keyterms extracted via various methods are as follows:

QUICK REDUCT: software, system, planning, enterprize, supply.

WEIGHT I: software, system, planning, supply.

WEIGHT II: software, system, planning, supply.

WEIGHT III: worker, system, planning.

WEIGHT IV: software, system, planning, enterprize.

SEMANTIC WEIGHT: software, user, supply.

Summary produced by Model I (with keywords: system, planning, software) with 30 percent compression rate is:

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Components of Supply Chain Management System - The components of an supply chain system consists of supply chain software and hardware, supply chain business processes and users of Supply Chain Management system. Supply Chain Management Software and Hardware - The core of an Supply Chain Management system is Supply Chain Management software. UNIX is the most common operating system for running Supply Chain Management software Business Processes - Business processes of supply chain includes supply chain planning, execution and collaboration and operational control. Processes of Supply Chain Management :Demand Planning and Forecasting - A critical success factor to supply chain management is accurate demand forecasting. Supply chain software systems utilize sophisticated mathematical models for predicating future demand from historical data.

Summary produced by Model II with 30 percent compression rate is:

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Supply Chain Management is supported by modular software applications that integrate activities across organizations, from demand forecasting, product planning, parts purchasing, inventory control, manufacturing, product assembly to product distribution. Components of Supply Chain Management System - The components of an supply chain system consists of supply chain software and hardware, supply chain business processes and users of Supply Chain Management system. (5) Supply Chain Management Software and Hardware - The core of an Supply Chain Management system is Supply Chain Management software. UNIX is the most common operating system for running Supply Chain Management software Business Processes - Business processes of supply chain includes

supply chain planning, execution and collaboration and operational control. (11) Supply chain software systems utilize sophisticated mathematical models for predicating future demand from historical data.

Summary produced by Model III (Weighting Scheme I) with 30 percent compression rate is:

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Supply Chain Management is supported by modular software applications that integrate activities across organizations, from demand forecasting, product planning, parts purchasing, inventory control, manufacturing, product assembly to product distribution. Components of Supply Chain Management System - The components of an supply chain system consists of supply chain, software and hardware, supply chain business processes and users of Supply Chain Management system. Supply Chain Management Software and Hardware - The core of an Supply Chain Management system is Supply Chain Management software. UNIX is the most common operating system for running Supply Chain Management software Business Processes - Business processes of supply chain includes supply chain planning, execution and collaboration and operational control. (11) Supply chain software systems utilize sophisticated mathematical models for predicating future demand from historical data.

Summary produced by Model III (Weighting Scheme II) with 30 percent compression rate is:

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Supply Chain Management is supported by modular software applications that integrate activities across organizations, from demand forecasting, product planning, parts purchasing, inventory control, manufacturing, product assembly to product distribution. Components of Supply Chain Management System - The components of an supply chain system consists of supply chain, software and hardware, supply chain business processes

and users of Supply Chain Management system. Supply Chain Management Software and Hardware - The core of an Supply Chain Management system is Supply Chain Management software. UNIX is the most common operating system for running Supply Chain Management software Business Processes - Business processes of supply chain includes supply chain planning, execution and collaboration and operational control. (11) Supply chain software systems utilize sophisticated mathematical models for predicating future demand from historical data.

Summary produced by Model III (Weighting Scheme III) with 30 percent compression rate is:

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Supply Chain Management is supported by modular software applications that integrate activities across organizations, from demand forecasting, product planning, parts purchasing, inventory control, manufacturing, product assembly to product distribution. Components of Supply Chain Management System - The components of an supply chain system consists of supply chain, software and hardware, supply chain business processes and users of Supply Chain Management system. Supply Chain Management Software and Hardware - The core of an Supply Chain Management system is Supply Chain Management software. UNIX is the most common operating system for running Supply Chain Management software Business Processes - Business processes of supply chain includes supply chain planning, execution and collaboration and operational control. Processes of Supply Chain Management :Demand Planning and Forecasting - A critical success factor to supply chain management is accurate demand forecasting.

Summary produced by Model III (Weighting Scheme IV) with 30 percent compression rate is:

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Supply Chain Management is supported by modular software applications that in-

tegrate activities across organizations, from demand forecasting, product planning, parts purchasing, inventory control, manufacturing, product assembly to product distribution. Components of Supply Chain Management System - The components of an supply chain system consists of supply chain, software and hardware, supply chain business processes and users of Supply Chain Management system. Supply Chain Management Software and Hardware - The core of an Supply Chain Management system is Supply Chain Management software. UNIX is the most common operating system for running Supply Chain Management software Business Processes - Business processes of supply chain includes supply chain planning, execution and collaboration and operational control. Supply chain software systems utilize sophisticated mathematical models for predicating future demand from historical data.

Summary produced by Model IV (Similarity I) with 30 percent compression rate is:

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Supply Chain Management is a social or soft system which has goals, components, processes and boundary. Supply Chain Management Software and Hardware - The core of an Supply Chain Management system is Supply Chain Management software. UNIX is the most common operating system for running Supply Chain Management software Business Processes - Business processes of supply chain includes supply chain planning, execution and collaboration and operational control. Users - The users of Supply Chain Management systems are workers of supply chain participants at all levels. Return - Return and refund are important parts and also the problem parts of supply chain management.

Summary produced by Model IV (Similarity II) with 30 percent compression rate is:

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Supply Chain Management is a social or soft system which has goals, components, pro-

cesses and boundary. Goals of Supply Chain Management to reduce inventory cost, to increase sales to improve the coordination and the collaboration with suppliers, manufacturers and distributors Supply Chain Management Software and Hardware - The core of an Supply Chain Management system is Supply Chain Management software UNIX is the most common operating system for running Supply Chain Management software Business Processes - Business processes of supply chain includes supply chain planning, execution and collaboration and operational control. The boundary of a Supply Chain Management system is the boundary of extended enterprize which includes the company, suppliers, partners, distributors and customers.

Summary produced by Human is:

Supply Chain Management is a business system of enterprize strategies, business processes and information technologies for improving the planning, execution and collaboration of material flows, information flows, financial flows and workforce flows in the supply chain. Supply Chain Management is supported by modular software applications that integrate activities across organizations, from demand forecasting, product planning, parts purchasing, inventory control, manufacturing, product assembly to product distribution. Components of Supply Chain Management System - The components of an supply chain system consists of supply chain software and hardware, supply chain business processes and users of Supply Chain Management system. Processes of Supply Chain Management :Demand Planning and Forecasting - A critical success factor to supply chain management is accurate demand forecasting. The boundary of a Supply Chain Management system is the boundary of extended enterprize which includes the company, suppliers, partners, distributors and customers.

The details of results are given in Appendix A.

4.2.2 *Experiment 2*

The results for the following data are being discussed in this section.

In the past decade almost all Islamic revivalist movements have been labeled fundamen-

talistic, whether they be of extremist or moderate origin. The widespread impact of the term is obvious from the following quotation from one of the most influential Encyclopedias under the title 'Fundamentalist': " The term fundamentalist has been used to describe members of militant Islamic groups. Why would the media use this specific word, so often with relation to Muslims. Before the term fundamentalist was branded for Muslims, it was, and still is, being used by certain Christian denominations. Most of them are radical Baptist, Lutheran and Presbyterian groups. The Southern Baptist Convention is one such group, they take pride in being called the Fundamentalists. Because, according them, they have gone back to the fundamentals of Christianity. They preach absolute Biblical inerrency and Millenarianism (belief in the physical return of Christ to establish a 1000 year reign). These radical groups form only a minute minority of the total Christian population, although they may be the most vocal. They want the Church to be the only authority. This reminds the modern man of the Dark Ages in Europe when the Church was in fact supreme. What most of these groups do not realize is that the term Fundamentalist is actually derived from a series of essays published from 1910 to 1915 under the title, The Fundamentals by British and American evangelists. The purpose of this 12 - volume collection was to determine which churches, according to the authors, held up to genuine Christian doctrine and the ones that did not. Nevertheless the term Fundamentalist, in the Christian world, is synonymous with the 'Bible Thumpers' and the tele evangelists. To apply the same terminology to Muslims is neither fair nor valid. Because in the case of Islam all Muslims believe in absolute inerrency of the Quran, since it is a basic Islamic tenet. Therefore the media would have to use the word fundamentalist for all Muslims, which it does not do. It only uses the word Fundamentalist for both the extremist and terrorist groups, and the true moderate Islamic revivalist movements. Both these definitions are incompatible with each other. Using the word fundamentalist for the former may be acceptable, since it does have some parallel to the Christian definition. But if that definition is to be used, however, then using the same word to describe the latter would be erroneous and completely unacceptable. It is this dual definition that is unfair to the Islamic faith. Therefore the media should either stop using the word Fundamentalist to describe any and all Islamic organizations, or be much more careful in its usage.

Summary produced by Model I with 30 percent compression rate is:

The widespread impact of the term is obvious from the following quotation from one of the most influential Encyclopedias under the title 'Fundamentalist' : " The term fundamentalist has been used to describe members of militant Islamic groups. Before the term fundamentalist was branded for Muslims, it was, and still is, being used by certain Christian denominations. What most of these groups do not realize is that the term Fundamentalist is actually derived from a series of essays published from 1910 to 1915 under the title The Fundamentals by British and American evangelists. Therefore the media would have to use the word fundamentalist for all Muslims which it does not do. It only uses the word Fundamentalist for both the extremist and terrorist groups, and the true moderate Islamic revivalist movements. Using the word fundamentalist for the former may be acceptable, since it does have some parallel to the Christian definition. Therefore the media should either stop using the word Fundamentalist to describe any and all Islamic organizations, or be much more careful in its usage.

Summary produced by Model II with 30 percent compression rate is:

The widespread impact of the term is obvious from the following quotation from one of the most influential Encyclopedias under the title 'Fundamentalist: " The term fundamentalist has been used to describe members of militant Islamic groups. Why would the media use this specific word, so often with relation to Muslims. Therefore the media would have to use the word fundamentalist for all Muslims, which it does not do. It only uses the word Fundamentalist for both the extremist and terrorist groups, and the true moderate Islamic revivalist movements. Using the word fundamentalist for the former may be acceptable, since it does have some parallel to the Christian definition. But if that definition is to be used, however, then using the same word to describe the latter would be erroneous and completely unacceptable. Therefore the media should either stop using the word Fundamentalist to describe any and all Islamic organizations, or be much more careful in its usage.

Summary produced by Model III with 30 percent compression rate is:

The widespread impact of the term is obvious from the following quotation from one of the most influential Encyclopedias under the title 'Fundamentalist: " The term funda-

mentalist has been used to describe members of militant Islamic groups. Why would the media use this specific word, so often with relation to Muslims. Therefore the media would have to use the word fundamentalist for all Muslims, which it does not do. It only uses the word Fundamentalist for both the extremist and terrorist groups, and the true moderate Islamic revivalist movements. Using the word fundamentalist for the former may be acceptable, since it does have some parallel to the Christian definition. But if that definition is to be used, however, then using the same word to describe the latter would be erroneous and completely unacceptable. Therefore the media should either stop using the word Fundamentalist to describe any and all Islamic organizations, or be much more careful in its usage.

Summary produced by Model IV with 30 percent compression rate is:
In the past decade almost all Islamic revivalist movements have been labeled fundamentalistic, whether they be of extremist or moderate origin. The widespread impact of the term is obvious from the following quotation from one of the most influential Encyclopedias under the title 'Fundamentalist: " The term fundamentalist has been used to describe members of militant Islamic groups. Most of them are radical Baptist, Lutheran and Presbyterian groups. These radical groups form only a minute minority of the total Christian population, although they may be the most vocal. This reminds the modern man of the Dark Ages in Europe when the Church was in fact supreme. What most of these groups do not realize is that the term Fundamentalist is actually derived from a series of essays published from 1910 to 1915 under the title, The Fundamentals by British and American evangelists. Nevertheless the term Fundamentalist, in the Christian world, is synonymous with the 'Bible Thumpers' and the tele evangelists.

The details of results are given in Appendix B.

4.3 Observations

The key terms and the summary generated by Weighting Scheme I and Weighting Scheme II are same. The first one use the binary weights and the second use the frequency weights.

The frequency weights and binary weights are comparable for sentences. Hence, the expected similarity in the generated summary. We note that the measures of weighting schemes are varying, still the F-values (see Appendix A) over all weighting schemes are comparable. This shows that though the membership measures are changing still the relative importance of sentences over different weighting schemes are not changing. Precision and recall does not reflect the changes in the order of importance of sentences. The similarity measures as seen in Table 5, Appendix A, shows that sentence 18 is more important than sentence 9 and sentence 3 is more important than sentence 17 and so on. The importance levels can be verified by looking at the text. After testing on many data sets we conclude that similarity measure II is a better measure than the measure by Das Gupta [24]. Further, the results of the various summarizers themselves do not match much but as the degree of summarization increases they begin overlapping. The results in Appendix B shows that results need not be always very high and distinct.

4.4 Future Work

- There are other ways to deal with numerical data. This include fuzzy methods. These methods can be used in place of discretization techniques. Discretization methods have some errors associated with them. These includes error in classification and error in handling inconsistent data tables.

- The methods can be improved using better classifiers. We have used heuristic based techniques for classification. Other classifiers like support vector machine can be used.

- The methods can be extended for creating abstracts.

- The method can be further improved by using some good weighting techniques and better lexical chaining algorithm. This would reduce the sparseness of the decision table.

- Another issue in summarization is efficient evaluation tools. We have considered F-values and similarity measures as our measuring parameters. More techniques as discussed in section 4.1 can prove to be beneficial.

Appendix A

Results: Experiment I

.

.

.

.

.

Appendix B

Results: Experiment II

.

.

Appendix C

ScreenShots

.

.

Bibliography

[1] Alexios Chouchoulas, Qiang Shen, *A Rough Set-Based Approach to Text Classification*, In Proc. of the Seventh International Workshop on Rough Sets, LNAI, Springer, pp 118-127, 1999.

[2] Bai Rujiang, Wang Xiaoyue, *An effective Hybrid Classifier Based on Rough Sets and Neural Networks*, Web Intelligence and Intelligent Agent Technology, ACM, pp 57-62, 2006.

[3] Conroy John, Leary Dianne, *Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition*, 2001. (available at: http://citeseer.ist.psu.edu/conroy01text.html)

[4] Copernic Summarizer Homepage http://www.copernic.com/en/products/summarizer.

[5] Daniel McDonald and Hsinchun Chen, *Using Sentence-Selection Heuristics to Rank Text Segments in TXTRACTOR*, ACM, JCDL, pp 28-35, 2002.

[6] De-Sheng, Guo-Yin Wang, *A self-learning algorithm for decision tree pre-pruning*, Proc. of Third International Conference on Machine Learning and Cybermetics, IEEE, pp 2140-2145, 2004.

[7] Eduard Hovy, Chin-Yew Lin, *Automated Text Summarization in SUMMARIST*, Advances in Automatic Text Summarization, 1999. (available at: http://citeseer.ist.psu.edu/455388.html)

[8] Inderjeet Mani, *Recent Developments in Text Summarization*, CIKM, ACM, pp 529-531. 2001.

[9] Jade Goldsteiny, Mark Kantrowitz, *Summarizing Text Documents: Sentence Selection and Evaluation Metrics*, Conference on Research and Development in Information Retrieval, ACM, pp 121-128, 1999. (available at: http://portal.acm.org/citation.cfm?id=312624.312665)

[10] Grabowski Adam, *Basic Properties of Rough Sets and Rough Membership Function*, Journal of Formalized Mathematics vol 12, pp 21-28, 2003. (available at: http://mizar.org/fm/2004-12/pdf12-1/$roughs_1$.pdf)

[11] Harry Wu and Gerard Salton, *A Comparison of Search Term Weighting*, Conference on Research and Development in Information Retrieval, ACM, pp 30-39, 1981.

[12] Jana, Malik, *Information-Theoretic Feature Selection Algorithms for Text Classification*, Proceedings of International Joint Conference on Neural Networks, IEEE, pp 3272-3277, 2005.

[13] Joseph P. Turian, Luke Shen, *Evaluation of Machine Translation and its Evaluation.*

[14] Kishore Papineni, Salim Roukos, *BLEU: A Method for Automatic Evaluation of Maschine Translation*, Proc. of the 40th Annual Meeting of the ACL, pp 311-318, 2002.

[15] Lawrence Reeve Hyoil Han, *BioChain: Lexical Chaining Methods for Biomedical Text Summarization*, SAC, ACM, pp 23-27, 2006. (available at: http://citeseer.ist.psu.edu/744948.html)

[16] Li Qiang, Hua-Jian, Shen-Gong, Li Hong, *A Rough Set-Based Hybrid Feature Selection Method for topic specific text filtering*, Proc. of Third International Conference on Machine learning and Cybernectics, IEEE, pp 1464-1468, 2004.

[17] Liu Shizhu, Hu Heping, *A hybrid Text Classification System Using Sentential Frequent Itemsets*, LNAI 3801, Springer-Verlag Berlin Heidelberg, pp 442-449, 2005.

[18] MA Yu-liang, YAN Wen-jun, *Value Reduction Algorithm in Rough Sets Based Association Rules Support*, Journal of Zhejiang Univrsity, pp 219-222, 2006.

[19] Meru Brunn, Yllias Chali Christopher and J. Pinchak, *Text Summarization Using Lexical Chains.* (available at: http://www-nlpir.nist.gov/projects/duc/pubs/2001papers/lethbridge.pdf)

[20] MS Word Autosummarize http://www.microsoft.com/education/autosummarize.mspx.

[21] Nguyen Hung Son, Nguyen Sinh Hoa, *Discretization Methods in Data Mining*, Rough Sets in Knowledge Discovery 1, pp 451-482, 1998.

[22] Ohrn Aleksander, *Discernibility and Rough Sets in Medicine: Tools and Applications*, Ph.D., Norwegian University of Science and Technology, 1999.

[23] Padmini Das-Gupta, *Rough Sets and Information Retrieval, Information Systems and Systems Engineering*, George Mason University, ACM, pp 567-581, 1988.

[24] Pawlak Zdzislaw, *Rough Sets: A Tutorial*, Int. Journal of Information and Computer Sciences, 1982.

[25] Pawlak, Zdzislaw, *Rough sets: Theoretical aspects of reasoning about data*, Springer, 1991.

[26] Polkowski Lech, Skowron Andezej, *Rough Sets in Knowledge Discovery 1*, Physica Verlag, 1998.

[27] QTAG http://www.english.bham.ac.uk/staff/omason/software/qtag.html.

[28] Robert Susmaga, *Experiments in Incremental Computation of Reducts*, Rough Sets in Knowledge Discovery 1, pp 530-553, 1998.

[29] Ruizhi Wang, Duoqian Mian, Guirong Hu, *Discernibility Matrix Based Algorithm for Reduction of Attributes*, Web Intelligence and Intelligent Agent Technology, pp 477-480, 2006.

[30] Sachin Jain, *M.Tech. Project: Text Summarization*, IIT Delhi, 2005.

[31] Shailendra Singh, Lipika Dey, *A rough-fuzzy document grading system for customized text information retrieval*, LNAI 2663, , Springer-Verlag, pp 258267, 2003. (available at: http://eprint.iitd.ac.in/dspace/bitstream/2074/1252/1/singhrou2003.pdf, http://www.springerlink.com/content/vvyvw7cd1qaanden/)

[32] Shailendra Singh, *Ph.D. Thesis: A rough-fuzzy document grading system for customized text information retrieval*, IIT Delhi, 2005.

[33] Silber H. Gregory, McCoy Kathleen F., *Efficiently Computed Lexical Chains As an Intermediate Representation for Automatic Text Summarization*, ACL, 2002.

[34] Silber H. Gregory, McCoy Kathleen F., *Efficient Text Summarization Using Lexical Chains*, IUI, ACM, pp 252-254, 2000.

[35] S.K.M Wong and W Ziarko, *A Machine Learning Approach to Information Retrieval*, Conference on Research and Development in Information Retrieval, ACM, pp 228-233, 1986.

[36] Srinivasan Padamini, Ruiz M.E., Chen Jianhua, *Vocabulary mining for information retrival: Rough Sets and Fuzzy Sets*, Information Processing and Management, pp 15-38, 2001.

[37] Parul Luthra, *M.Tech. Project: Text Summarization*, IIT Delhi, 2003.

[38] Tatsunori Mori, Miwa Kikuchi, Kazufumi Yoshida, *TermWeighting Method based on Information Gain Ratio for Summarizing Documents retrieved by IR systems*, Div. of Electrical and Computer Eng., Yokohama National University, Japan.

[39] Tay and Shen, *Economic and Financial Prediction using Rough Sets Model, European Journal of Operational Research*, 2001.

[40] Web-based Summarization, Computer-Aided Summarization Tool (CAST) http://clg.wlv.ac.uk/projects/CAST/

[41] WebSumm Text Summarizer,
http://complingone.georgetown.edu/ linguist/summarizer.html

[42] Wenqian Shang, Houkuan Huang, *A novel feature selection algorithm for text categorization*, Expert Systems with Applications 33, ScienceDirect, pp 1-5, 2007.

[43] WordNet Homepage:
http://wordnet.princeton.edu

[44] Yihong Gong, Xin Liu, *Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis*, SIGIR'01, ACM, pp 19-25, 2001.

[45] Zha Hongyuan, *Generic Summarization and Key phrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering*, SIGIR'02, ACM, pp 113-120, 2002.