Fuzzy Approaches for Text Document Classification

Dissertation Submitted to Jawaharlal Nehru University in Partial Fulfillment of the Requirement for the Award of the Degree of

Master of Technology

Submitted by

Nitin Prajapati

Submitted to

Aditi Sharan

School of Computer & System Sciences
Jawaharlal Nehru University
New Delhi-110067
India
2012

जवाहरलाल नेहरू विश्वविद्यालय

School of Computer & Systems Sciences

JAWAHAR LAL NEHRU UNIVERSITY

NEW DELHI-110067, INDIA

DECLARATION

I hereby declare that the dissertation entitled "**FUZZY APPROACHES FOR TEXT DOCUMENT CLASSIFICATION**", submitted by me to the School of Computer and Systems Sciences, **Jawaharlal Nehru University**, New Delhi, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science and Technology**, is a bona fide work carried out by me under the supervision of **Dr. Aditi Sharan**.

The matter embodied in this dissertation has not been submitted to any other university or institution for the award of any other degree or diploma.

NITIN PRAJAPATI

M.TECH-CSE

SC&SS, JNU,

New Delhi-110067

जवाहरलाल नेहरू विश्वविद्यालय

School of Computer & Systems Sciences

JAWAHAR LAL NEHRU UNIVERSITY

NEW DELHI-110067, INDIA

CERTIFICATE

This is to certify that the dissertation entitled "**FUZZY APPROACHES FOR TEXT DOCUMENT CLASSIFICATION**", submitted by Nitin Prajapati to the School of Computer and Systems Sciences, **Jawaharlal Nehru University**, New Delhi, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science and Technolgy**, is a bona fide work carried out by him under my supervision.

The matter embodied in this dissertation has not been submitted to any other university or institution for the award of any other degree or diploma.

Dr. Aditi Sharan,
(Supervisor)
SC&SS, JNU, New Delhi

(Dean, SC&SS,)
JNU, New Delhi

Dedicated to all my beloveds……

Acknowledgement

I would like to gratefully acknowledge the enthusiastic supervision of **Dr. Aditi Sharan** during this work. This work wouldn`t have been possible without her constant support, valuable suggestions and comments during my whole tenure of this dissertation work. I feel privileged to work under her for my master`s dissertation. Apart from the academic guidance she has always been a great mentor of mine in encouraging me to be disciplined and well organized. I must surely say, she has given her best in providing me the infrastructure required, which led to the successful completion of my dissertation. I would take this opportunity to thank her once again for her esteemed support and I, from the bottom of my heart would like to wish her the best in all her future endeavors.

I wish to thank my colleagues Mr. Mayank Saini and Mr. Jagendra Singh for creating a home like environment in our lab to keep the stress away. I would also like to thank my best critics Anuj Kumar, Vipin Kumar and Ranjeet Kumar Ranjan for making my dissertation an attempt of producing an error free record of my work. Thank you guys!!

Finally I would like to thank the whole faculaty members of our department for clarifying my doubts throughout this work and last but not least, the JNU administration for creating such a secular and healthy environment amongst the students.

Abstract

We are living in the information age, where almost every piece of information is represented as electronic data on computer. In day to day life we have to deal with a huge number of documents to search the desired information. With the introduction of internet technology each and every piece of information is under our hands with one click but to search the right information that suits your needs is difficult, because of a large corpus of documents available on web servers. To tackle with such situation technique of text classification been found helpful.

Text classification is the task of automatically classifying the text documents into a set of predefined categories. Once you know that your desired information belongs to a particular category this reduces your search complexity to a great extent. To automate the concept of text classification we need to apply some machine learning algorithm to textual data. Soft computing techniques have been found most elegant for this purpose. Fuzzy techniques are good construct of soft computing algorithms and most suitable for text document classification.

In this approach we propose a method of text document classification based on the theory of information retrieval with fuzzy techniques used for text classification. The fuzzy logic theory seems to play a better role for document representation, category center formation and predicting class profiles. Further in this work we have discussed various issues related to structure of text classification and the performance of various similarity measures.

Table of contents

List of Figures

List of Tables

Chapter-1

Introduction

1.1 Introduction to Text Categorization

Text categorization is the task of automatically classifying the text documents into a set of predefined categories. This task which falls at the crossroads of information retrieval and machine learning, has witnessed a booming interest in the last decades from researchers and developers alike. One purpose of text document categorization could be to reduce the search space for a particular user. Text categorization can be divided into two existing approaches.

- Text classification
- Text clustering

1.1.1 Text Classification

Text classification is the task of classifying text data into predefined categories based on supervised learning approach. Supervised learning technique is used to learn a model over training data recursively, which can be applied to the test data. Labeled training data is used as input to this technique. Predicted class labels for test documents are outcome at the end.

1.1.2 Text Clustering

Text clustering is the process of categorizing text data into predefined number of clusters. Text clustering is an unsupervised learning approach. Unlabeled test data is used as input and in turn class labels are learned as outcome.

1.2 Process of Text Document Classification

Text classification is based on supervised learning approach. Supervised learning approach makes use of machine learning techniques to learn a model over given training data. Further that model is used to predict the class profiles of test data.

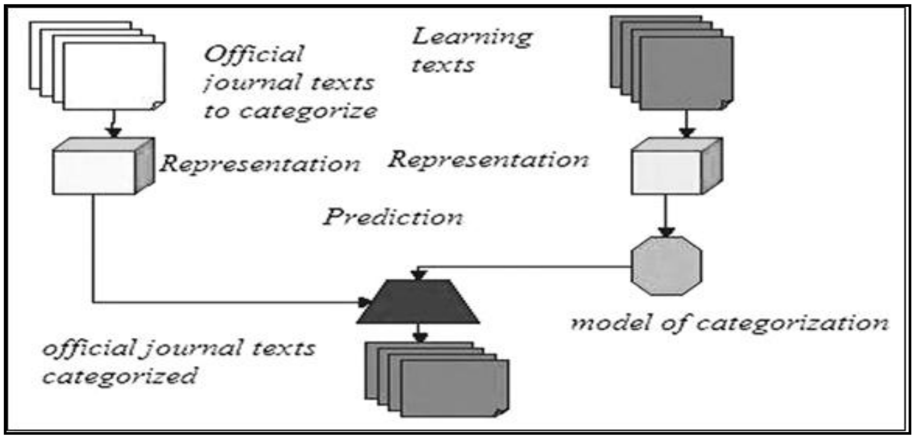


Figure 1.1 : Process of text classification

The process of text classification can be carried out in two phases.

1.2.1 Training Phase

This phase is used to build/train a model over given labeled training data by applying suitable machine learning techniques. The accuracy of the classifier depends upon the efforts made for training the model. Once the model is built, it must be trained well in order produce acceptable classification accuracy. Appropriate number of iterations must be run over the training data until the threshold value is reached. Threshold value is reached when result starts repeating itself.

1.2.2 Testing Phase

The objective of this phase is to produce classification results for testing data whose class labels are unknown. Once the built model is trained well you can pass test data to this model. This model then predicts class labels of test documents. If class labels of test data are known in advance then we can test the accuracy of classifier over test data.

The accuracy of the phases involved in text classification task, described above, depend on the methods/models used for text representation, because the major problem associated with textual data is, that it is always found in an unstructured manner so it is quite tedious to process that data. We have to deploy text documents in the format that must be suitable for text processing by the classifier and also retain the meaning of text document.

One of the famous text representation schemes is proposed by vector space model [6, 20]. VSM is one of the most famous models used for textual information retrieval task. It represents the documents as a feature vector of terms (for more information read section 4.2.2). Other models proposed by researchers are: tensor space model [21], cognitive reading indexing model [15], lexical chaining model [2] and semantic graph based model [17] etc.

1.3 Applications of Text Document Classification

Major application areas of text document classification can be: email spam filtering, categorization of newspaper articles, organizing web pages into hierarchical categories, sort journals and abstracts by subject categories. Assigning international clinical codes to patient records and mail routing etc.

1.4 Introduction to Fuzzy logic theory

We must exploit our tolerance for imprecision.

Lotfi Zadeh

Soft computing [19] technique is a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solution cost. The role model for soft computing technique is the human brain. One of the most applied soft computing techniques is Fuzzy Logic theory.

1.4.1 Fuzzy Logic

“Fuzzy logic is a mathematical logic that attempts to solve problem by assigning values to an imprecise spectrum of data in order to arrive at the most accurate solution possible. [12, 28]”

The term ‘fuzzy’ means something vague or whose boundaries are not fixed and most real world problems contain this vagueness. Initially the probabilistic approaches were not enough to deal with real time problems, hence as an attempt to deal with the vagueness or impreciseness which is inherent in real life entities, a paper on fuzzy logic [19] theory was introduced by Prof. Lotfi Zadeh in 1965, which later played a vital role in solving real time problems. Fuzzy logic theory is based on the fuzzy sets which provide a mathematical way to represent vagueness or fuzziness in humanistic system.

e.g. Take the predicate into account “it is hot today”.

The term “hot” in the above predicate is a vague term which may have imprecise or uncertain values regarding to reference. A man living in hilly area would say 30 degree Celsius is hot irrespective to that, a man living in plain area would say 40 degrees Celsius is hot. Here the term hot is having uncertainty according to geographical reference. Hence to tackle these sorts of real life problems we need fuzzy theory.

Fuzzy logics are notably represented by fuzzy sets. In contrast to the crisp sets, an element shows a partial membership of an element to fuzzy set. This partial membership is determined by membership function μ_A*: X* →[*0, 1*]. A graphical view of membership function is

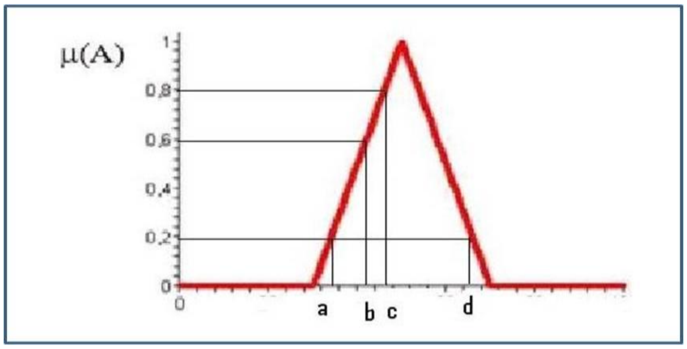


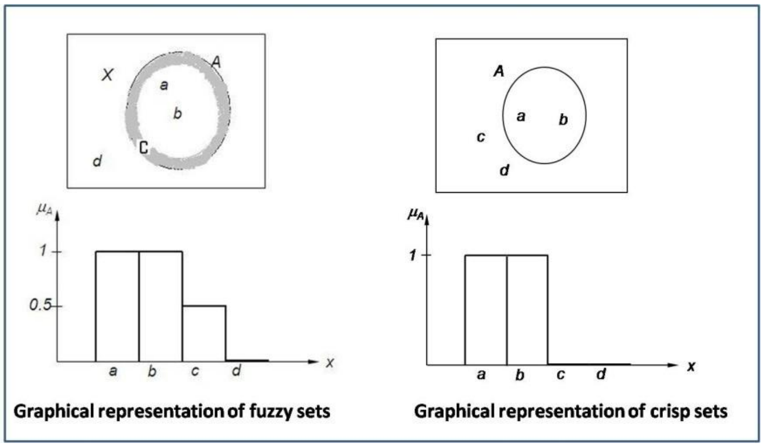
Figure 1.2 : Graph for fuzzy membership function

A fuzzy set with above membership function may be described as

$$\boldsymbol{A = \{\frac{0.2}{a}, \frac{0.6}{b}, \frac{0.8}{c}, \frac{0.2}{d}\}}$$

1.4.2 Comparison of Fuzzy sets to Crisp sets

Fuzzy sets are based on real valued logic between 0 and 1, while crisp sets are based on Boolean valued logic i.e. only 0 or 1. Following figure provides a clear cut view of both logics.



Graphical representation of fuzzy sets

Graphical representation of crisp sets

Figure 1.3 : Fuzzy sets v/s crisp sets

1.5 Background and Motivation

The solution to the classification problem has been sincerely needed since documents were used to be represented on paper and classification was used to perform manually. With the introduction of computers, documents can be easily represented on computer in electronic format, so we need to automatize the process of classification by applying computer algorithms. Now a days the problem of text classification has been widely studied by databases, data mining and information retrieval communities. A wide variety of techniques (like decision trees [11], rule based classifiers [7], SVM classifier [27], nearest neighbor classifier [16], Bayesian classifier [22]) have been designed for classification but they cannot be applied in straight forward manner for text document classification because text is a particular kind of data in which the word attributes are sparse, high dimensional and with low frequencies on words. Therefore it is critical to design classification methods which effectively account for these characteristics of text.

These limitations of standard techniques motivate us to find a text classification mechanism which gives an intuitive idea of text representation and processing, and ought to give appreciable results. Our approach is inspired from the field of information retrieval [9, 3, 16] in which VSM [6] is used for document indexing and fuzzy techniques [10, 25] for text processing. Although standard techniques could be used over VSM but they might not give good results. Further multiclass text classification can be achieved by applying fuzzy theory where classes can be assumed fuzzy.

1.6 Outline of Dissertation

The rest of this dissertation is organized as follows. Chapter 2 describes about issues and challenges related to text classification and techniques to tackle with those challenges. Chapter 3 is all about the work related to text classification and application of fuzzy techniques to field of text mining. Proposed work, experimental results and analysis part is discussed in chapter 4 followed by conclusion and future scope in chapter 5.

Chapter-2

Issues and Challenges to Text Classification

This chapter is about describing and dealing the difficulties faced by a system / person during text classification task is pursued. Starting from the concept, document as a bag of words, one has to process these documents and predict their class profiles. Processing a document simply means processing words in that document; now one has to process the words that may be from a thesaurus of billions of words. Some of these words may represent the same concept or different concepts may be represented by same word depending on the context in which these words are used. Moreover a document or a class can be viewed as a concept and these words collectively make that concept. Processing of these many words may cause problem for text classification task.

We tried to categorize the issues and challenges faced by the text classification task in following categories and tackle these issues in our work.

2.1 Text Representation

Text representation is certainly a major challenge for text processing researchers. In order to get your text processed by a computer system you have to have some mechanism to represent the textual data on a computer system. On paper we represent a document as a structured sequence of words/sentences and as a whole; document can be viewed as a concept. But neither of two ideas (structured sequence of words/sentences) is good for text representation from an algorithmic point of view.

The major problem with text is that it resides either in unstructured or semi structured form. To deal with such textual data is difficult, so a better representation technique is always required, because better representation always reduces the processing time and cost as well.

The most widely used representation model is VSM. According to this model a text document can be described as a vector whose dimension is the number of text features extracted from the corpus. This vector consists of statistical measure of terms extracted from corpora.

$$\boldsymbol{d_j = \{(t_1, 0.34), (t_2, 0.98), (t_3, 0.76), \ldots \ldots (t_m, 0.21)\}}$$

Other model used for text classification tasks are described in brief by following table [4].

Model	**Terms**	**Representation Type**	**Objective**
Antonellis and Gallopoulos	Sentences	Term-by-sentence matrices	Text mining
Blake and Pratt	Word, phrases, concepts	Association rules	Representation of medical texts
Bloehdorn et al	Words and concepts	Combination of bag-of-words and concept hierarchy	Text clustering and classification
Carenini et al	Concepts	Hierarchy	Feature extraction
Caropreso et al	Phrases	N-grams	Text categorization
Cimiano et al	Concepts	Concept hierarchy	Automotive acquisition of taxonomy
Kehagias et al	Word senses	Sense-based vector	Text categorization
Mladenic and Grobelnik	Phrases	N-grams	Text learning
Rajman and Besancon	Word and compounds	Vector	IR
Salton	Noun phrases	Tree	Book indexing
VSM	Words	Vector	IR
Vareles et al	Words	Tree	Semantic similarity of IR

Table 2.1 : Models for text representation

2.2 Stopwords

Stopwords can be assumed as frequently occurring and insignificant words in a language that help to construct a sentence but do not carry any meaning to that document. Articles, prepositions, conjunctions and some pronouns are natural elements of stopword family.

Stopwords must be removed before the document are indexed or stored because 20% to 30% words of the document are stop words and they are highly frequent over the document. If stopwords are not removed they may increase the dimension of data which needs extra processing cost and time. Normally stopwords are controlled by human input and a list of stopwords [8] could be different for different purposes.

2.3 Stemming

Stemming refers to the process of reducing words to their stems or roots. The literal meaning of the stem is root. A stem is defined as a portion of a word that is left after removing its prefixes and suffixes. In different languages a word may have various syntactic structures depending on the context it is used. In English noun have plural forms, verbs have gerund forms and verbs used in the past tense are different from the present tense. These can be considered as syntactic variation of the same root form. Such variation causes low recall for a retrieval system because a relevant document may contain a variation of query word but not the exact word itself and it may incur extra processing cost and time too.

In English most variations of words are generated by the addition of suffixes to their stem. Hence stemming in English mainly refers to the suffix removal rather than prefix removal. **e.g.** Computer, computing, computation, computers, computed, compute can be reduced to stem “comput”.

Over the decades researchers are trying to find the appropriate way of stemming that results into some standard stemming algorithms popularly used today. These algorithms are also known as stemmers. Popular stemming algorithms are:

2.3.1 Brute force method

Brute force stemmers [1] employ a lookup table, which maintains a hierarchical relation between stem and its inflected forms. To find the root of a word, every time that lookup table is queried for exact match of inflection. If a match is found root of that inflection is returned.

2.3.2 Lovin`s algorithm

This algorithm was developed by Julie Beth Lovins [26] in 1968. This algorithm is mainly for the processing of large endings which is the base of Lovins stemming algorithm. The heart of this algorithm is 35 transformation rules and 29 endings. The ending list contains the possible endings for suffix stripping itself. Each list entry is related to a condition. Therefore an ending can only be removed if the appropriate condition is true.

2.3.3 Porter stemming

This algorithm was proposed by Martin F. Porter [24] in 1980. The idea behind this algorithm was to remove all prefixes and suffixes to get the root of the word. The main application of porter stemmer is found in languages having inflections like English.

This algorithm is based on making distinction vowels and consonants in a word. Therefore the selection of the applying rules during the stemming process is based on the sequence of consonants and vowels. A word is represented by the form.

[c]vcvcvc….vc[v] ➔ [c](vc){m}[v]

e..g. sea ➔ m=0, cat ➔ m=1, garden ➔ m=2

2.4 Noisy Data

Noisy data is meaningless data. The term has often been used as a synonym for corrupt data. However, its meaning has expanded to include any data that cannot be understood and interpreted correctly by machines, such as unstructured text.

While Text processing is a growing as mature field of text mining that has great value because of the huge amounts of data being produced. Processing of noisy text is necessary because a lot of common applications may produce noisy textual data. Noisy unstructured text data [23] is mainly produced by informal settings such as online chats, text messages, e-mails, message boards, newsgroups, blogs, wikis and web pages. Also, text produced by processing spontaneous speech of natural languages using automatic speech recognition and printed or handwritten text using optical character recognition contains processing noise. Text produced under such situations is highly noisy containing spelling errors, abbreviations, non-standard words, false starts, repetitions, missing punctuations, missing letter case information, pause filling words such as “um” and “uh” and other texting and speech disfluencies. Such text can be obtained in large amounts in contact centers, chat rooms, optical character recognition (OCR) of text documents, short message service (SMS) text, etc. Documents with ancient language can also be considered noisy with respect to today’s knowledge about the language. Such text contains important historical, religious, ancient medical knowledge that is useful.

2.4.1 Techniques to deal with Noisy data

Noise from the data must be removed before it is sent as input to the computer algorithm. Following techniques can be used to remove noise from data.

- Binning
- Regression
- Clustering
- Combined computer and human inspection

2.5 Word Sense disambiguation

In text classification task a document/class can be considered as a concept. Ideally a concept matching scheme should be considered for the sake of classification. A document is assumed as a bag of words and we know that there are words which may have multiple meanings depending on context. This context can be considered by the phrases, sentences and paragraphs in which that particular word is used. WSD technique is used to tackle with this sort of problems.

In text processing/computational linguistics, word sense disambiguation [30] is an open problem of the field of NLP, which tracks the process of identifying which sense/meaning of a word is used in a sentence when the word has multiple meanings. Mainly the problem of word sense disambiguation can be categorized into two categories.

2.5.1 Polysemy (Homonymy)

Polysemy refers to the set of words from which any word may have multiple meaning depending upon the different-2 contexts.

e.g. Word “star” is a polysemy.

- Sun is the biggest star in our galaxy.

Here the sun and galaxy are contextual words which make it clear that that we are talking about astronomical star here.

- Leonardo Dicaprio is the biggest star of Hollywood.

Here Leonardo Dicaprio and Hollywood are contextual words that relate the word star to a film star.

- The share of star textiles is all time low now.

Here it is clear that the star is a company.

2.5.2 Synonymy

Synonymy is set of words which represent the same concept or sense. Synonymy words may also cause problem because traditional classification approach classifies the document based on syntactic matching of words. If another word is used to represent the same concept then it may be possible that documents are not classified correctly.

e.g. Words : photo, image, picture, snap may be used to represent the same concept, graphical view of any object.

Various supervised, unsupervised and dictionary based approaches can be used for WSD. Wordnet [31] indexing based approaches are getting popular these days.

2.6 Tagging

It has been seen that in text classification task major role is played by noun, adjective and adverb words. Remaining words of the documents can be assumed as supportive words to make grammatically correct structure of sentences. Tagging refers to the process of assigning POS tags to the words read from the text. This technique could be based upon both the word itself and its context, that is, relationship with adjacent and related words in a phrase, sentence or paragraphs.

One could ask the role of POS tagging in text classification. POS tagger tokenizes the text into part of speech words. Once POS tagging is done we can classify documents based on noun words only, because classes and documents can be accurately represented by noun words mainly.

2.7 Grammar/Syntax rule

This is certainly an issue related to text classification task. Grammar/syntax rules are used to form a conceptually right sentence in a particular language. Words on combining make sentences, sentences make paragraphs and paragraphs make documents. So a document is a structured arrangement of words that carries a specific sense as a concept. If the syntactical arrangements are disturbed, then, a document would only be a container of words. Words, those, may have specific meaning in itself but may not carry a collective concept to a document or class.

So we have to apply some technique to text classification such that the grammatical structure of sentences in documents should not be lost.

Chapter-3

Related Work

Some of the related work to text classification and application of fuzzy techniques to text mining have been discussed in this chapter.

3.1 Use of fuzzy logic theory in text classification

Since the invention of fuzzy logic theory as soft computing technique, it has found its numerous applications in most of real world problems. Starting from washing machines, dishwashers it has been implemented in huge mechanical systems. This section mainly discusses about how fuzzy theory approaches can be useful for the text document classification purpose. Mainly from starting to end point in text document classification everything seems to be fuzzy. To provide a clear view of how fuzzy technique can be helpful we will discuss the following steps.

3.1.1 Fuzzy document representation

Better document representation has always been a basic need for accurate text classification. Many standard techniques have its own way of representing documents. Here a document can be represented as a feature vector of fuzzy weighted terms. Basically VSM is used to represent documents as a feature vector of terms.

$$\boldsymbol{d_j = \{(t_1, 0.34), (t_2, 0.98), (t_3, 0.76), \ldots \ldots (t_m, 0.21)\}}$$

3.1.2 Estimating Fuzzy Category center

In text classification task one needs to represent a category with some fuzzy statistics such that is easy to calculate the similarity between category and test documents. We found that a category can be easily represented as a feature vector of terms with fuzzy weights. The weight of these terms carry fuzzy participation value of the particular term to that category. This is a similar approach to document representation and similarity can be easily calculated by applying an appropriate similarity measure over test document and category center.

$$\boldsymbol{C_j = \{(t_1, 0.34), (t_2, 0.98), (t_3, 0.76), \ldots \ldots (t_m, 0.21)\}}$$

3.1.3 Fuzzy Similarity Measure

Every text classification technique must have some way to calculate similarity between class and document in order to classify the documents. Some standard similarity measure such as cosine, Euclidian and Jaccard give appreciable results still there is scope for more, fuzzy similarity measures can provide a better way of measuring similarity. Fuzzy similarity measure uses the fuzzy t-norm [10] and t-conorm [10] operators over fuzzy category center and fuzzy test document vector, which results in fuzzy similarity score. A general formula for Fuzzy similarity [10] is given below :

$$\boldsymbol{Sim(d, C_j) = \frac{\sum_{k=1}^{n}(\mu_r(t_k, C_j) \otimes \mu_d(t_k))}{\sum_{k=1}^{n}(\mu_r(t_k, C_j) \oplus \mu_d(t_k))}} \tag{3.1}$$

Where $\mu_r(t_k, C_j)$ is term category score of term tk in category Cj and $\mu_d(t_k)$ is weight of kth term in document. ⊗ is fuzzy conjunction and ⊕ is fuzzy disjunction operator respectively.

3.1.4 Fuzzy Multiclass Distribution

In case of single label text classification, classes are crisp i.e. a document may do belong to a class or don`t, but in case of fuzzy multiclass text classification moreover classes can be fuzzy i.e. a single document may have partial participation to more than one classes. In this case a document shows a fuzzy participation to each class or classes may have fuzzy membership of documents.

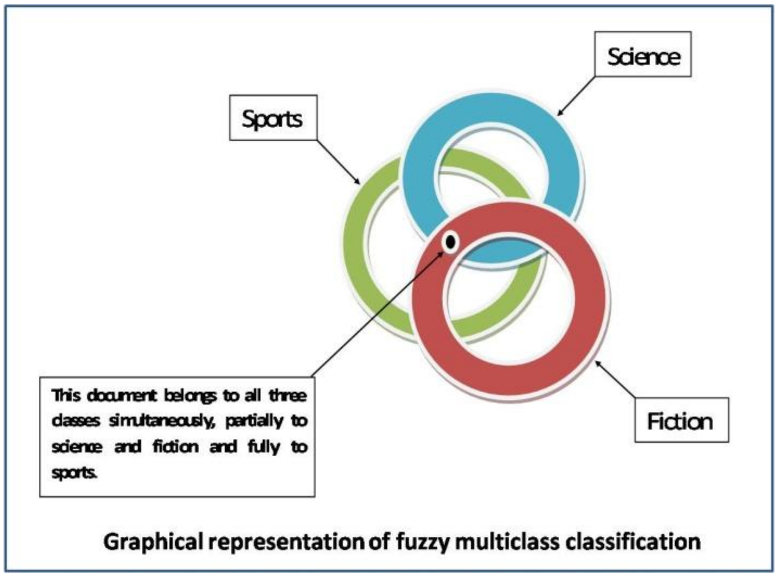

Science
Sports
This document belongs to all three classes simultaneously, partially to science and fiction and fully to sports.
Fiction
Graphical representation of fuzzy multiclass classification

Figure 3.1 : Graphical representation of fuzzy multiclass distribution

Further multiclass classification [25] problem can be reduced to the single class classification problem by applying α-cut with an appropriate threshold value over a multiclass classification score vector.

3.1.5 Dimensionality Reduction

Fuzzy approach can be applied to the document set in order to reduce the dimensionality of document feature vector. One can extract some representative terms that are able to retain the conceptual meaning of document after removing inappropriate dimensions (terms). This can be done by taking α-cut of the fuzzy document feature vector over an appropriate experimental threshold t, now the crisp set remained after taking α-cut will represent the appropriate dimensions of document vector. Dimensionality reduction will result in fast and easy processing of text data.

An estimate of threshold value can be calculated by averaging the fuzzy weight of terms. If there are m terms and n documents then threshold can be calculated as:

$$\boldsymbol{T} = \frac{\frac{\sum_{k=1}^{n} t_1}{n} + \frac{\sum_{k=1}^{n} t_2}{n} + \frac{\sum_{k=1}^{n} t_3}{n} + \cdots + \frac{\sum_{k=1}^{n} t_m}{n}}{m} \tag{3.2}$$

Once the value of threshold has been calculated we can easily decide which dimensions to cutoff by taking α-cut over threshold value T. Following figure shows that the dimensions ‘a’ and ‘d’ must be chopped off while ‘b’ and ‘c’ must be taken into consideration.

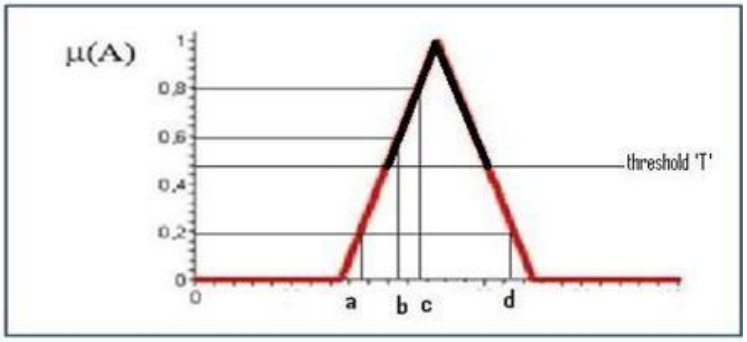


Figure 3.2 : Fuzzy membership function with threshold

3.2 Summary Generation

With the rapid expansion in the number of electronic text documents, need of automatic analysis of these documents has been sincerely aroused, because human analysis is not feasible for a huge corpus of documents. This need gave birth to the theory of automatic summary generation. Automatic summary generation [5, 18] refers to the process of generating summaries with the help of machine learning algorithms. Fuzzy logic theory plays role of machine learning technique in automatic summary generation. These text summarization techniques can be used to select the most relevant information from the various heterogeneous text sources.

Mainly automatic summary generation can be classified into two categories.

- Abstraction based Summarization
- Extraction based Summarization

Abstraction based summarization is very near to the process of human summary generation i.e. “explaining in your words”. Abstraction based techniques require heavy computational power and a good knowledge of NLP.

Extraction based methods are relatively simple. It can be just viewed as the process of selecting important sentences, phrases and words from the original text and putting them together to form summary.

The technique of automatic summary generation proposed here is based on applying human expertise in the form of fuzzy rules and set of extracted features. Specifically a parser is designed for selecting sentences based on their location, attributes, importance in the text document using the fuzzy logic inference system. The parser parses the text into sentences and recognizes the following features for each sentence as the input of fuzzy inference system.

- The number of title word in the sentences.
- Whether it is the first sentence in the paragraph.
- Whether it is the last sentence in the paragraph.
- The number of words in the sentences.
- The number of thematic words in the sentences.
- The number of bonus words.

Above feature extracted can be taken as input to the fuzzy inference system [13] in order to generate the membership value of each feature. At last important features are selected and presented as summary. Documents and classes can be summarized as a concept that makes classification an easy task.

3.3 Fuzzy Information Retrieval

IR is the field of study that intends the user to find out the desired information from a large corpus of text document set against the user query. Traditional information system assumes that the basic unit of information is the text document itself.

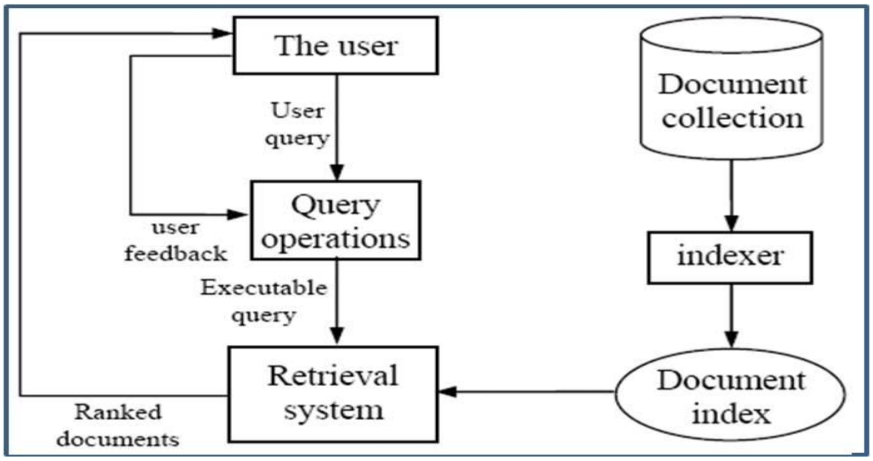


Figure 3.3 : information retrieval process [6]

Retrieving information simply means finding a set of documents that is relevant to the user query. Ranking of documents is also performed for user ease according to the relevance score of query and document.

Fuzzy IR system [9, 13, 29] makes use of fuzzy sets to represent documents, the fuzzy membership score for query terms, fuzzy logical operators to define extended queries and fuzzy similarity measure to assess the similarity score of the document to the query. Main components of fuzzy information retrieval can be described as: document representation, query representation, extended query formulation, document relevance score and similarity measure.

3.3.1 Document Representation

The document can be represented as a feature vector of terms with fuzzy weights. Fuzzy weights show the statistical importance of the term to that particular document.

$$\boldsymbol{F_d} = \left\{ \left(\frac{\mu_{\boldsymbol{f_d}}(\boldsymbol{w})}{\boldsymbol{w}} \right) \middle| \ \boldsymbol{d} \in \boldsymbol{D} \ \boldsymbol{and} \ \boldsymbol{w} \in \boldsymbol{W} \right\} \qquad (3.3)$$

3.3.2 Query Representation

A query must be represented in a computer understandable format, in order to get processed by the system but the process of formulating a query begins with user, expressing the needed information through the query. Users, however, find it difficult to specify queries directly to the system understandable format without system assistance. A query can also be represented as a feature vector of terms with fuzzy weights

A simple query can be represented as.

$$\boldsymbol{q_d} = \left\{ \left(\frac{\mu_{\boldsymbol{f_q}}(\boldsymbol{w})}{\boldsymbol{w}} \right) \middle| \ \boldsymbol{w} \in \boldsymbol{W} \right\} \qquad (3.4)$$

3.3.3 Relevance Score Calculation

The relevance score can be determined by a fuzzy similarity measure when both the document and query are represented as fuzzy sets. The degree of relevance of document ‘d’ over query ‘q’ can be calculated as set theoretic notion of fuzzy query set f_q in fuzzy document set f_d.

$$\mu_{\boldsymbol{f_{rq}}}(\boldsymbol{d}) = \frac{\sum_{\boldsymbol{c} \in \boldsymbol{C}} \boldsymbol{min}(\mu_{\boldsymbol{f_d}}(\boldsymbol{c}), \mu_{\boldsymbol{f_q}}(\boldsymbol{c}))}{\mu_{\boldsymbol{f_q}}(\boldsymbol{c})} \qquad (3.5)$$

3.4 Fuzzy Correlation between Terms

One of the major problems with the traditional IR system is that it considers only syntactic matching between terms. No semantic matching of terms or documents is being considered here. Even though the syntactic matching scheme gives appreciable results, but the sense of matching may get lost because this is only statistics based syntactic matching of words. Now the question is, what, if terms are syntactically similar but semantically different? To deal with such problems concept of morphological processing of words is proposed.

Morphological processing can be used to find the semantic relation of two terms. Fuzzy set theoretic approach may be useful in order to find such relationship. Fuzzy association rules are used to capture the relationship between different index terms.

Def-1 A fuzzy association between two finite sets $\mathrm{X} = \{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \mathrm{x}_4, \ldots.., \mathrm{x}_\mathrm{m}\}$and $\mathrm{Y} = \{\mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3, \mathrm{y}_4, \ldots.., \mathrm{y}_\mathrm{n}\}$is formally defined as a binary fuzzy relation: f: X × Y →[0,1]

Def-2 given a set of index terms as: $\mathrm{T} = \{\mathrm{t}_1, \mathrm{t}_2, \mathrm{t}_3, \mathrm{t}_4, \ldots.., \mathrm{t}_\mathrm{m}\}$.

Set of documents as: $\mathrm{D} = \{\mathrm{d}_1, \mathrm{d}_2, \mathrm{d}_3, \mathrm{d}_4, \ldots.., \mathrm{d}_\mathrm{n}\}$

$F(t_i, d_j)$is defined as degree of membership of t_i in d_j.

Def-3 fuzzy correlation formula [7] between terms t_i and t_j over corpus of documents can be established as:

$$\boldsymbol{RT(t_i, t_j) = \frac{\sum_{k=1}^{n} min\,(f(t_i,d_k), f(t_j,d_k))}{\sum_{k=1}^{n} max\,(f(t_i,d_k), f(t_j,d_k))}} \tag{3.6}$$

A simplified form of the above correlation formula can be defined as:

$$\boldsymbol{r_{i,j} = \frac{n_{i,j}}{n_i + n_j - n_{i,j}}} \tag{3.7}$$

Where $r_{i,j}$ is correlation coefficient between terms i and j. $n_{i,j}$ is the number of documents containing both terms. n_i is the number of documents containing ith term and n_j is the number of documents containing the jth term.

3.5 Rocchio Text Classification

Rocchio text classification [6] scheme is based on methods of relevance feedback implemented in information retrieval systems based on VSM. This algorithm is based on the general assumption that every user has a clear picture that which document should be denoted as relevant or irrelevant. Rocchio text classification assumes that any category can be denoted by finding appropriate prototype vector of that category. Once the prototype vector is constructed, now, one can test each incoming document against each category and can assign the document to the most relevant category.

Cosine similarity (or other) measure can be used for measuring similarity between documents and category. The document having the highest similarity score can be categorized into the corresponding category.

The prototype vector can be constructed from the following formula.

$$\boldsymbol{c_i} = \frac{\boldsymbol{\alpha}}{|\boldsymbol{D_i}|} \sum\nolimits_{\boldsymbol{d} \in \boldsymbol{D_i}} \frac{\boldsymbol{d}}{||\boldsymbol{d}||} - \frac{\boldsymbol{\beta}}{|\boldsymbol{D} - \boldsymbol{D_i}|} \sum\nolimits_{\boldsymbol{d} \in \boldsymbol{D} - \boldsymbol{D_i}} \frac{\boldsymbol{d}}{||\boldsymbol{d}||} \tag{3.8}$$

Where C_i denotes the prototype vector for class i.

D_i Denotes the document set for category i.

α and β are constants tf-idf scheme.

Now each document d_i will be tested against prototype vector $C_i$for i$\in$ 1,2,3…n with the cosine similarity measure. d_i will be assigned the class having highest relevance score.

3.6 Fuzzy category center

The field of text document classification is inspired from the theory of IR. In case of IR systems we retrieve relevant documents against the user query, while in the text classification task we retrieve the category of text document. Hence in text classification task one needs to calculate the similarity between the category and the document. Now the million dollar question is what could be the best possible way to represent the category? Different approaches have been proposed by information retrieval researchers to represent the category as a feature vector of terms.

$$\boldsymbol{C_j = \{(t_1, 0.34), (t_2, 0.98), (t_3, 0.76), \dots \dots (t_m, 0.21)\}}$$

According to the Rocchio text classification scheme we can represent a particular category as prototype vector which can be entertained with the given formula.

$$\boldsymbol{c_i = \frac{\alpha}{|D_i|} \sum_{d \in D_i} \frac{d}{||d||} - \frac{\beta}{|D - D_i|} \sum_{d \in D - D_i} \frac{d}{||d||}}$$

This relation leads to the appreciable classification results.

In another method, proposed by D.H. Widyantoro et al [10], category vector can be represented as a feature vector of terms extracted from text document corpus with fuzzy weights. Given formula calculates a term-category relation that shows importance of a particular term to a particular category.

$$\boldsymbol{\mu(t_i, C_j) = \frac{dist(t_i, C_j)}{max_{\forall\ i,j}(dist(t_i, C_j))}} \qquad (3.9)$$

Where

$$\boldsymbol{dist(t_i, C_j) = \frac{\sum_{t_i \in C_j} t_i}{\sum_{t_i \in (\forall\, C)} t_i}} \qquad (3.10)$$

An amendment made by Shian-chi Tsai et al [25] over the previous formula gives even better results. Shian-chi et al analyzed the concept proposed by D.H. Widyantoro et al [10] and found that the concept proposed by D.H. widyantoro et al may not give the appropriate result because they did not consider the fact, that, if frequency of a particular term is very high for a few documents over particular category documents then this term may not be a representative term of that particular category irrespective to that if that particular term is frequent overall category documents then this term must be a good representative term of that particular category. Shian-chi Tsai et al tried to normalize this effect by proposing the following amendments in the formula.

$$\boldsymbol{\mu(t_i, C_j) = \frac{dist(t_i,C_j)}{max_{\forall\ i,j}\ (dist(t_i,C_j))} \times \frac{P(t_i,C_j)}{max_{\forall\ i,j}(P(t_i,C_j))}} \tag{3.11}$$

Where $P(t_i, c_j)$ denotes the number of documents having term t_i in category c_j.

Chapter-4

Proposed Work and Experimental Results

As we have discussed earlier that fuzzy logic has wide scope in the field of text classification. In our work we have specifically used fuzzy similarity approach with IR for text classification.

4.1 Objective

There are two main objectives of our work.

a) Propose a framework for classification model based on fuzzy similarity..

b) Performance analysis of proposed model under following targets.

- Compare the performance of standard and fuzzy similarity measures.
- Study and analyze the performance of various term weighting schemes for fuzzy approach based classification.
- Study and analyze the performance of various fuzzy similarity measures for fuzzy approach based classification.

4.2 Proposed Model

Our proposed model is inspired from theory of information retrieval. In an IR system an user query is matched against the indexed documents from the corpus and relevant documents are retrieved according to their relevance to the query. In case of text classification, using IR, corpus of text documents can be replaced by a set of classes having learning text and query by the document to be categorized. Standard text similarity measures like: cosine, Jaccard and dice can be used to calculate the similarity score among indexed classes and documents to be categorized. Here a class can be indexed as bag of words extracted from documents of that class by applying some term-category relation.

In figure-4.1 we provide a stepwise description of text classification task based on the theory of IR using standard text similarity measures.

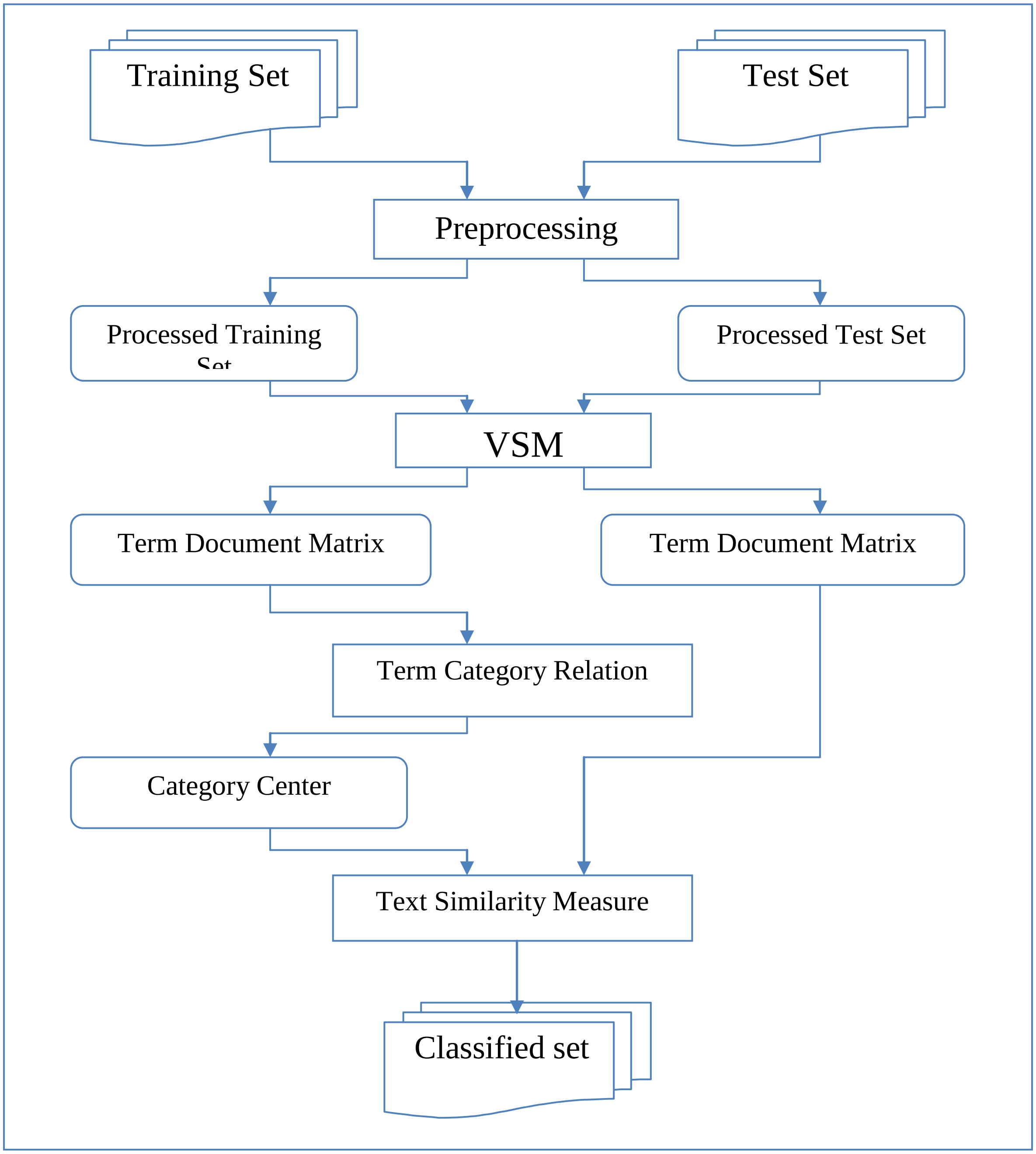


Figure 4.1 : Standard classification approach

In our proposed model for text classification we used fuzzy logics to represent text documents as feature vector of fuzzy terms, fuzzy term category relation to index the classes and finally fuzzy similarity measure to calculate the similarity score among classes and documents. The whole intuitive idea behind applying fuzzy theory to above phases of classification is because of the fact that, the representation of these phases is ambiguous hence it can be easily modeled by fuzzy logics. When the phases of text classification task can be precisely modeled then classification model ought to give appreciable results.

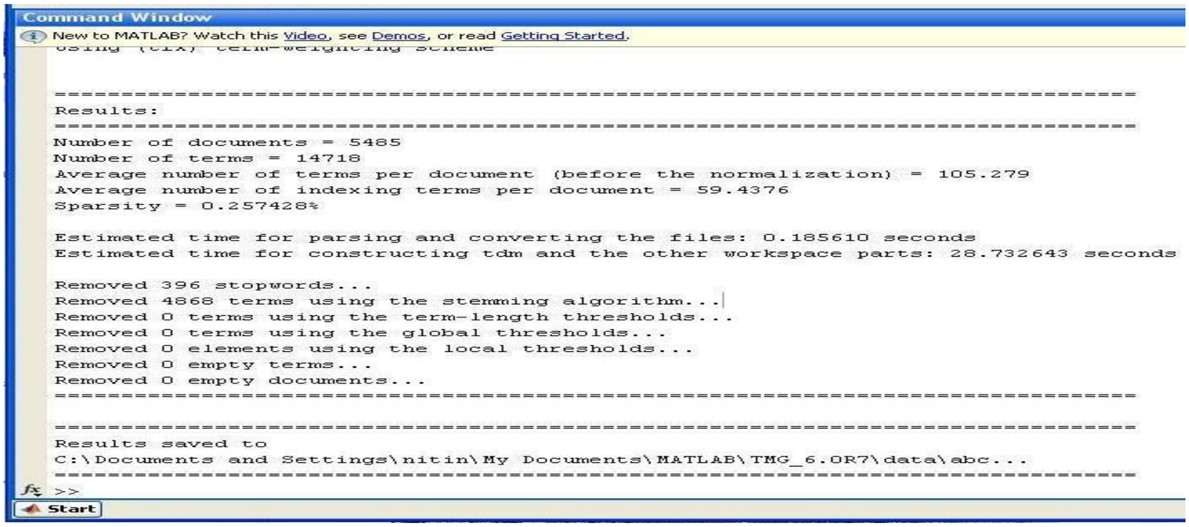
Figure-4.3 describes framework for our proposed model. Stepwise details of the model are as follows.

4.2.1 Preprocessing

Preprocessing is the task of filtering raw text documents in order to get reduced but more accurate text document set that can be easily processed by machine learning algorithms.

Preprocessing involves following two steps.

- Stopwords removal
- Stemming


Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Start

```
using (tfx) term-weighting scheme


================================================================================
Results:
================================================================================
Number of documents = 5485
Number of terms = 14718
Average number of terms per document (before the normalization) = 105.279
Average number of indexing terms per document = 59.4376
Sparsity = 0.257428%

Estimated time for parsing and converting the files: 0.185610 seconds
Estimated time for constructing tdm and the other workspace parts: 28.732643 seconds

Removed 396 stopwords...
Removed 4868 terms using the stemming algorithm...
Removed 0 terms using the term-length thresholds...
Removed 0 terms using the global thresholds...
Removed 0 elements using the local thresholds...
Removed 0 empty terms...
Removed 0 empty documents...
================================================================================

================================================================================
Results saved to
C:\Documents and Settings\nitin\My Documents\MATLAB\TMG_6.0R7\data\abc...
================================================================================
>>
```

Figure 4.2 : Snapshot of preprocessing from TMG

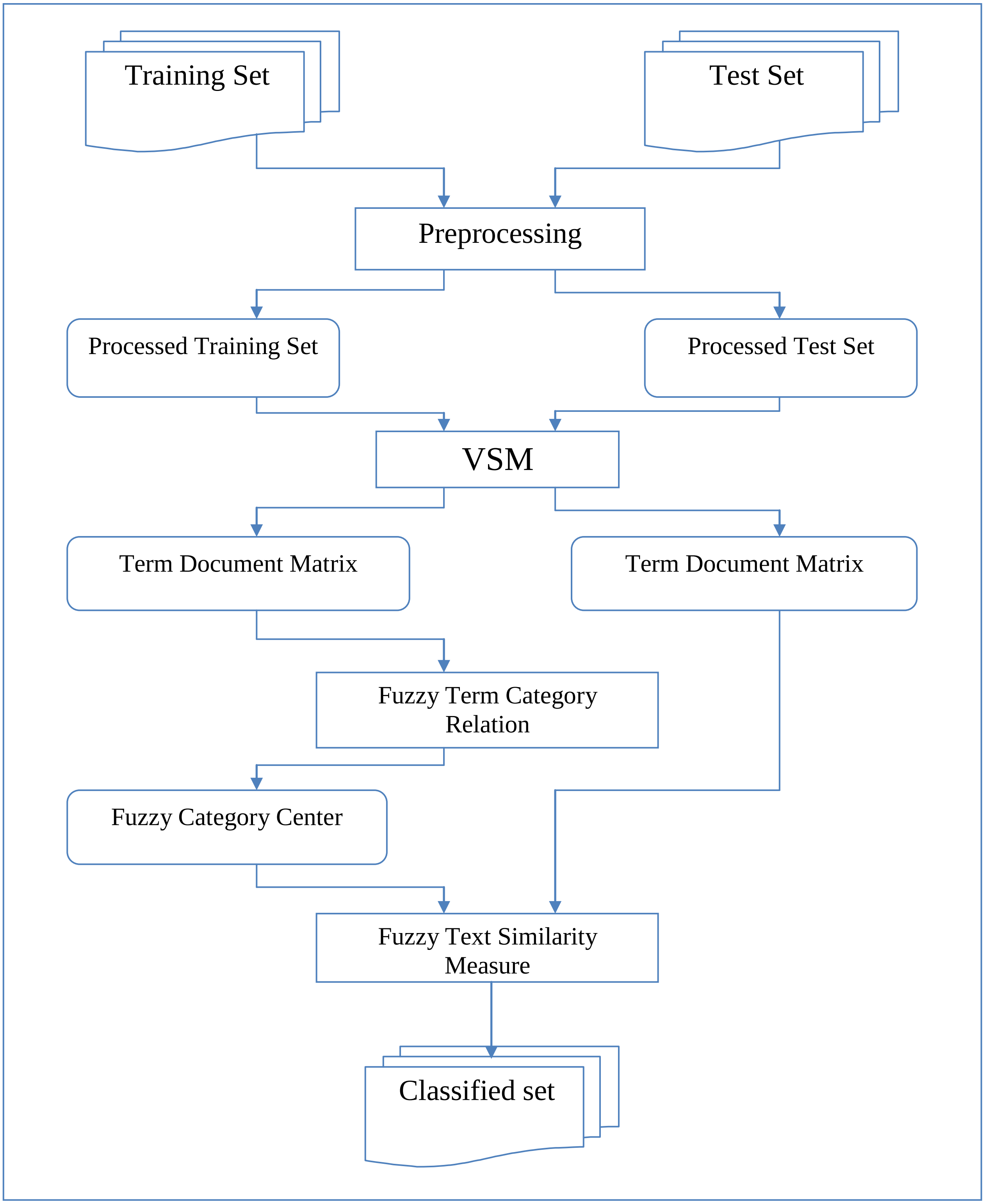
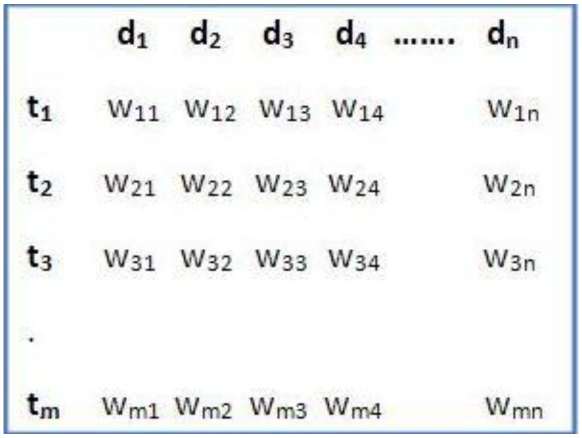


Figure 4.3 : Fuzzy classification approach

4.2.2 Vector Space Model

This model is perhaps the most popular and widely used IR models. It is an algebraic model that creates a vector space in which each document can be represented as a feature vector of terms, where each particular term represents a coordinate value in a particular dimension. For a fixed and preprocessed collection of documents, a m dimensional vector is generated for each document. The outcome of vector space model is a TDM which represents each document as a feature vector of terms.



	d_1	d_2	d_3	d_4	....	d_n
t_1	w_{11}	w_{12}	w_{13}	w_{14}		w_{1n}
t_2	w_{21}	w_{22}	w_{23}	w_{24}		w_{2n}
t_3	w_{31}	w_{32}	w_{33}	w_{34}		w_{3n}
.						
t_m	w_{m1}	w_{m2}	w_{m3}	w_{m4}		w_{mn}

Figure 4.4 : Term document matrix

Mainly the VSM works in two steps.

- Document indexing
- Term weighing

The main purpose of vector space model is an efficient fuzzy document representation.

4.2.2.1 Document indexing

Document indexing is the process of generating index words for documents. Index words are the set of words, extracted from the document set uniquely. Preprocessed document set is passed to the vector space model. VSM module takes the whole dataset as input, tokenize them into set of unique terms in order to form a set of index terms. Now each document is represented as feature vector of these index terms as.

$$\boldsymbol{d_j = \{(t_1, 0.34), (t_2, 0.98), (t_3, 0.76), \ldots \ldots (t_m, 0.21)\}}$$

4.2.2.2 Term Weighting Scheme

Term weighting scheme is used to assign particular weights to index terms. These schemes are based on statistical measurement of terms such as frequency count, probabilistic distribution of frequency, logarithmic or normalized frequency count etc. popular term weighting schemes are:

a) Term frequency

This scheme applies the weight of a particular index term deduced by the frequency count of that particular term t_i in particular document d_j known as tf_{ij}.

$$\boldsymbol{tf_{ij} = \sum_{k=1}^{f} 1 \;\; where\ f\ is\ freq\ count\ of\ t_i\ in\ d_j} \tag{ 4.1}$$

The shortcoming of this scheme is, that, it doesn`t discriminate the importance of a term which appears in many documents of the collection. Such a term may not be discriminating.

b) tf-idf Scheme

This is the most well known weighing scheme where tf stands for the term frequency and idf stands for the inverse document frequency.Idf of i_{th} term can be formulized as.

$$\boldsymbol{idf_i = log\ \frac{N}{df_i}} \tag{ 4.2}$$

Where N is the total number of training documents and df_i is the number of documents in which i_{th} term is present.

This scheme grasps the concept, that, if a term is frequent in all the documents then it can`t be discriminating. Hence this scheme tries to normalize the effect of that term by multiplying idf to the tf, to give appropriate statistical measure. tf-idf is formulized as:

$$(\boldsymbol{tf\text{-}idf})_{ij} = \boldsymbol{tf}_{ij} \times \boldsymbol{idf}_i \tag{4.3}$$

c) tf-inverse idf (A Proposed Technique)

This is the variation of the tf-idf scheme. This measure is mainly useful for text classification task using VSM. Terms which are frequent over documents belonging to particular category, may be better representative of that category. To increase the weight of those terms we need to implement inverse-idf measure. It increases the weight of terms which are frequent over particular category.

$$(\boldsymbol{tf{\sim}inverse{\sim}idf})_{ij} = \boldsymbol{tf}_{ij} \times \frac{\boldsymbol{1}}{\boldsymbol{idf}_i} \tag{4.4}$$

4.2.3 Fuzzy Term-Category relation

This is a binary relation between a term and category of particular document. It describes the importance of a particular term to a particular category. The relevance of 'terms' to 'categories' is expressed by a fuzzy relation: R**:** T×C →[0,1]. This relation is called fuzzy because elements of the relation are having fuzzy membership values. The membership value of this relation is described by using training data set, where training data consists of labeled text documents.

Let us assume that training document set D can represented as:

D={(d1, c(d1)),(d2, c(d2)),(d3, c(d3))…..(dn, c(dn))} where each document has crisp participation with only one class. We know that a document can be represented as fuzzy feature vector of terms.

$$\boldsymbol{d_j = \{(t_1, 0.34), (t_2, 0.98), (t_3, 0.76), \ldots \ldots (t_m, 0.21)\}}$$

The membership value of fuzzy term-category relation is determined as:

$$\boldsymbol{\mu(t_i, c_j) = \frac{\frac{\alpha}{|D_j|} \times \sum_{t_i \in D_j} t_i \; - \; \frac{\beta}{|D-D_j|} \times \sum_{t_i \in (D-D_j)} t_i}{\sum_{t_i \in D} t_i}} \qquad (4.5)$$

Where D is whole training data set.

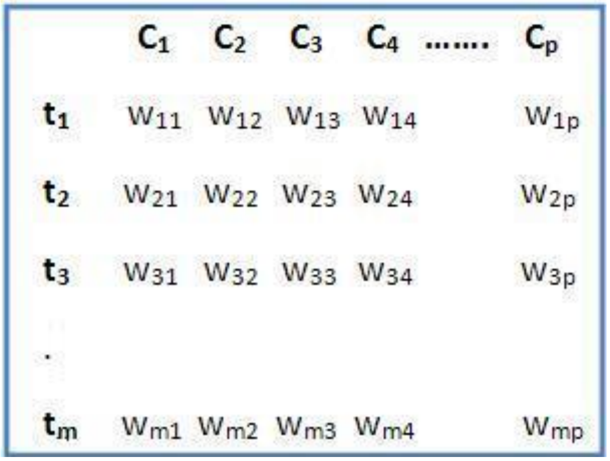
D_j is set of documents that belong to class i

D-D_j is the document set that doesn`t belong to class i.

α and β are constants carry value 16 and 4 respectively.

Only +ve values of μ(t_i, C_j) are considered.

The output of this stage comes out to be a term category relation matrix. Here each category can be represented as a fuzzy feature vector of terms extracted from training data set. This vector consequently works as category centre.



	C_1	C_2	C_3	C_4	....	C_p
t_1	W_{11}	W_{12}	W_{13}	W_{14}		W_{1p}
t_2	W_{21}	W_{22}	W_{23}	W_{24}		W_{2p}
t_3	W_{31}	W_{32}	W_{33}	W_{34}		W_{3p}
.						
t_m	W_{m1}	W_{m2}	W_{m3}	W_{m4}		W_{mp}

Figure 4.5 : Term Category matrix

This phase starts with testing data and model learnt in the training phase. First the testing data needs to be preprocessed and then it is passed to the VSM in order to get its TDM. Once the term document matrix of test data is generated this can be passed to fuzzy similarity measure along with category center learnt but before we do that we need to extend the dimensions of test documents represented by TDM. Such that appropriate dimension of test document is getting compared to the right dimension of category center. This extension is really needed because our classification model is based on syntactical similarity, so to compare two same words we need to compare the weight of two same dimensions of test document and category center. Testing phase mainly involves description of two steps.

4.2.4 Fuzzy Text Similarity Measure

After calculating the appropriate category center we got to evaluate the similarity score between category center and a particular test document. The intuitive idea behind the fuzzy similarity measure is that, we apply fuzzy operators over, category center and fuzzy feature vector of test document. Fuzzy similarity measure formula is given as:

$$\boldsymbol{Sim(d, C_j)} = \frac{\sum_{k=1}^{n}(\mu_r(t_k, C_j) \otimes \mu_d(t_k))}{\sum_{k=1}^{n}(\mu_r(t_k, C_j) \oplus \mu_d(t_k))} \tag{4.6}$$

Where $\mu_r(t_k, C_j)$ is term category score of term tk in category Cj and $\mu_d(t_k)$ is weight of kth term in document. ⊗ is fuzzy conjunction and ⊕ is fuzzy disjunction operator respectively

4.3 Dataset used and Software Requirement Specification

Following are the details of dataset and tools used to perform our task.

4.3.1 Dataset Used

We used R8 dataset [3] for our work. This contains 5485 training documents and 2189 test documents which can be spread over 8 categories. These categories are: ‘acq’, ‘crude’, ‘earn’, ‘grain’, ‘interest’, ‘money-fx’, ‘ship’ and ‘trade’.

This R8 set is taken as a subset from famous reuter-21578 classification dataset. Reuter-21578 is a popular dataset which is widely used for performing text classification experiments. The complete reuter-21578 dataset consist of 21578 documents which are published in reuter`s newswire in 1987. To make the R8 dataset, founder of this dataset, took the most frequent 8 categories and corresponding documents out of 672 total categories and 21578 documents from reuter-21578.

4.3.2 Software Requirement Specification

- Windows 32/64 bit OS.
- TMG version 6.0R7 or above
- MATLAB version R2010a
- Microsoft Excel 2007
- Turbo C++ 3.0

4.4 Experiments and Results

Experiments have been performed using tf, tf-idf and tf-inverse-idf weighting scheme. These techniques are used to signify the importance of terms in documents. Most popularly used weighting scheme is tf-idf. Tf-inverse-idf is a proposed technique used, because it can increase the weight of frequent occurring terms in documents of a particular class, which can further act as category word for that class.

For performing the standard classification cosine [6], jaccard [6], dice [6] and eucledian [6] text similarity measures were used and obtained the following results.

Similarity measure	**TF**	**TF*IDF**	**TF*1/IDF**	**TF in %**	**TF*IDF in %**	**TF*1/IDF in %**
Cosine	2064	2089	2052	94.29	95.43	93.74
Euclidean	1053	1070	148	48.10	48.80	6.76
Jaccard	1849	2065	1931	84.47	94.36	88.21

Dice	2964	2089	2052	94.29	95.43	93.74

Table 4.1 : Results for standard similarity measures

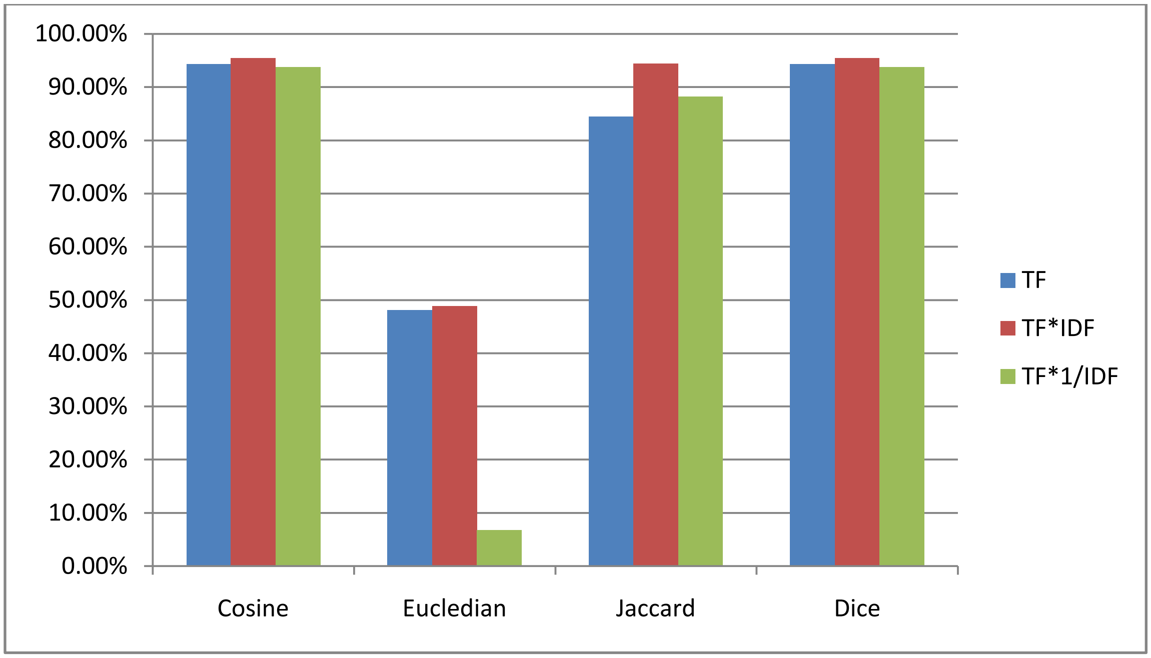


Figure 4.6 : Barchart for standard similarity measures

For incorporating the concept of fuzziness in text similarity following fuzzy text similarity formulas were used.

Fuzzy similarity formula	**x⊗y**	**x⊕y**
Einstein	$\frac{x \times y}{2-(x+y-x \times y)}$	$\frac{x \times y}{1+x \times y}$
Algebric	$x \times y$	$x+y-x \times y$
Hamacher	$\frac{x \times y}{x+y-x \times y}$	$\frac{x+y-2 \times x \times y}{1-x \times y}$
Min-max	Min (x, y)	Max (x, y)

Table 4.2: Fuzzy text similarity formulas [10]

Where ⊗ is a fuzzy conjunction and ⊕ is a fuzzy disjunction operator used in actual fuzzy similarity formula.(see section 4.2.4)

When above fuzzy text similarity measures were taken into consideration with tf, tf-idf and tf-inverse-idf following results were obtained.

Fuzzy Similarity measure	TF	TF*IDF	TF*1/IDF	TF in %	TF*IDF in %	TF*1/IDF in %
Algebraic	1849	2065	1931	84.47	94.36	88.21
Einstein	1469	1503	1404	68.66	67.10	64.13
Hamacher	2075	2096	2055	94.79	95.75	93.87
Min-max	2069	2105	2076	94.52	96.16	94.84

Table 1.3: Results for fuzzy similarity measure

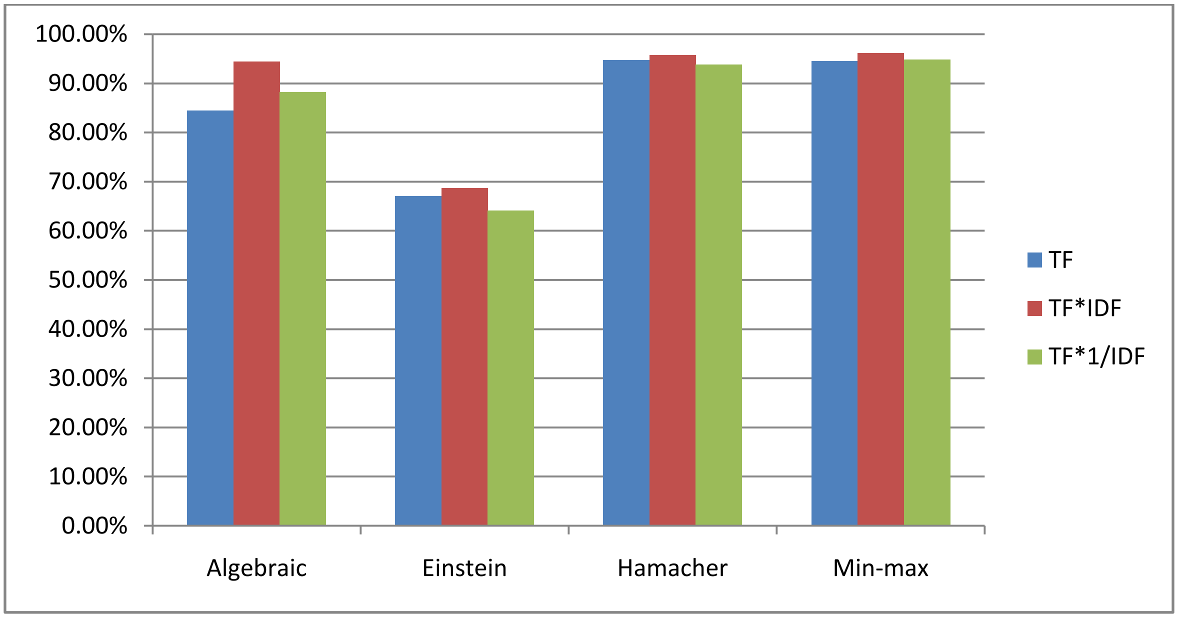


Figure 4.7 : Barchart representation for results of fuzzy measures

Finally the results were compared for standard text similarity measures and fuzzy text similarity measures to achieve the objectives.

Fuzzy Similarity measure	avg	In %	Standard Similarity measure	avg	In %
Hamacher	2075	94.79	Cosine	2068	94.47
Min-max	2083	95.16	Dice	2068	94.47
Algebric	1948	89.00	Jaccard	1948	89.00
Einstein	1459	66.65	Eucledian	757	34.58

Table 4.4 : Mean results

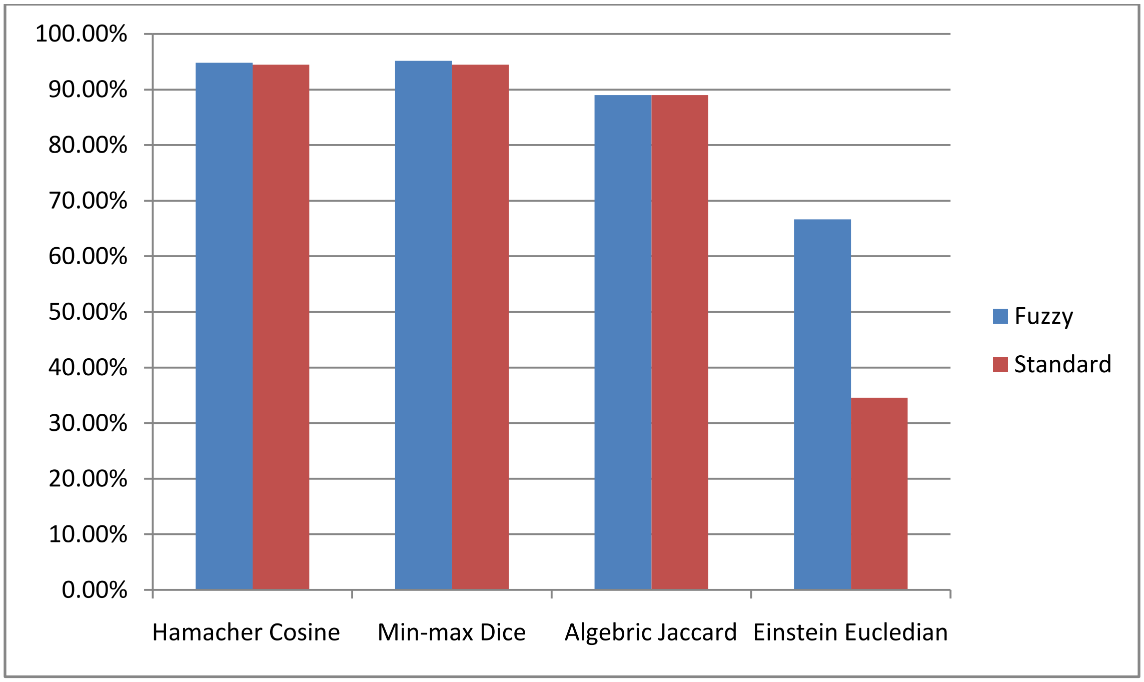


Figure 4.8 : Barchart comparison of standard to fuzzy neasures

On taking the mean of all results obtained for standard and fuzzy measures we obtained the following graph.

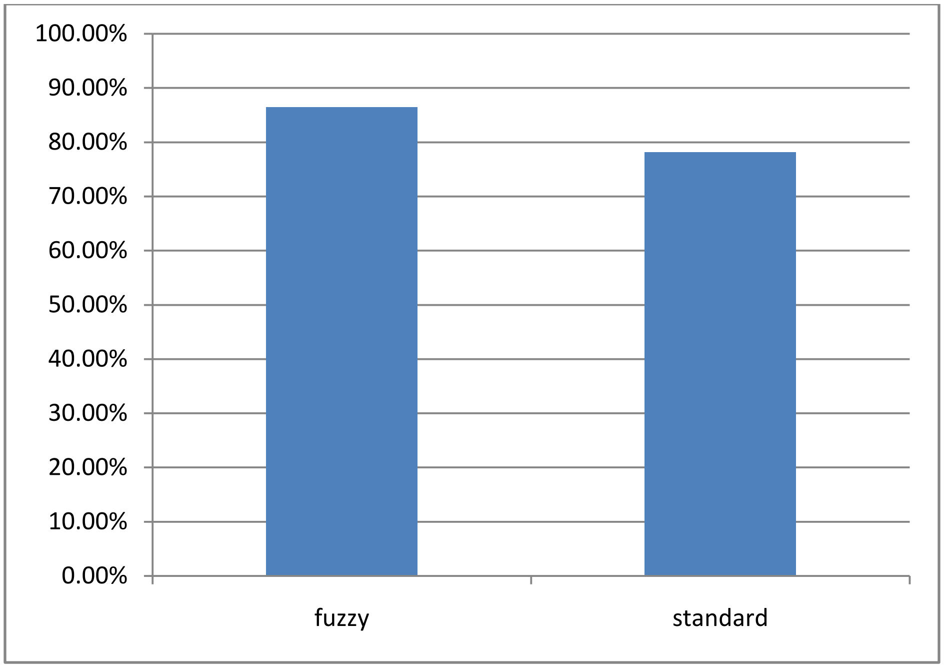


Figure 4.9 : standard measure v/s fuzzy measure

4.5 Analysis

It was seen that for tf, tf-idf and tf-inverse-idf weighing scheme different-2 results were obtained for fuzzy and standard similarity measures. On an average “min-max”, a fuzzy similarity measure, performs well among fuzzy measures while “cosine” and “dice coefficient” predicted the best result for standard classification measures. In an overall aspect fuzzy similarity measure gave 86.39 % accurate results which are better than standard similarity measures which gave only 78.11% accurate results.

Tf-idf scheme has outperformed well in comparison to other weighting scheme used. Tf-idf predicted best results for cosine in case of standard similarity measure and for min-max in case of fuzzy similarity measure.

In case of fuzzy similarity measures min-max gives the best result with tf-idf and tf-inverse-idf weighting scheme. In an overall view this measure outperforms among other fuzzy similarity measures.

Chapter-5

Conclusion and Future Scope

5.1 Conclusion

In this dissertation, I proposed a technique based on fuzzy framework on information retrieval model which is helpful for text classification task. Fuzzy classification technique prosed here enhances the overall classification accuracy in comparison to standard classifiation techniques available based on information retrieval framework. Fuzzy classification approach is mainly based on two fuzzy notions, that are, fuzzy category center and fuzzy similarity measure. Accuracy of classifier depends on that, how well fuzzy category centers have been learned in training phase and how much good similarity mechanism is used by fuzzy similarity measure in testing phase. In our case these two notion seems to perform well in order to produce better results.

5.2 Future Scope

In this proposed work our main focus is on single label text classification. But this work can be extended to multilabel text document classification. In case of text document classification accuracy depends on mainly two fuzzy notions, fuzzy category center and fuzzy similarity measure. Hence these two notions needs to be optimized to produce better results.

References

1) A. Honrado , R. Leon , R. O'Donnel , D. Sinclair, *"A Word Stemming Algorithm for the Spanish Language"*, Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00), p.139, September 27-29, 2000.

2) A. Mehler, U. Waltinger, and A. Wegner. *"A formal text representation model based on lexical chaining."*, In Learning from Non-Vectorial Data – Workshop at the KI 2007, September 10, 2007, University of Osnabr¨uck. 2007.

3) Ana Cardoso, "dataset for single label text classification", http://www.web.ist.utl.pt/~acardoso/dataset/

4) Anna Stavrianou, PeriklisAndritsos, Nicolas Nicoloyannis,(2007), *"Overview and Semantic Issues of Text Mining",* Sigmod Record, Vol. 36, No. 3

5) ArmanKiani and M.R. Akbarzadeh, *"Automatic Text Summarization Using: Hybrid Fuzzy GA-GP"*, In Proceedings of 2006 IEEE International Conference on Fuzzy Systems, Sheraton Vancouver Wall Center Hotel, Vancouver, BC, Canada. Pp.977-983.2006.

6) Bing liu, *Web data mining: exploring hyperlinks, contents, and usage data.* New York: Springer, 2007.

7) C. Haruechaiyasak, M.-L. Shyu, S.-C. Chen, X. Li, *"Web Document Classification Based on Fuzzy Association."*, Proceedings of COMPSAC2002 (26th Annual International Computer Software and Applications Conference), pp.487—492.

8) C. Fox, *"A stop list for general text"*, *ACM SIGIR forum*, vol.24, no.2, pp. 19-35, 1991.

9) D. Arotaritei, S. Mitra, "Web mining: a survey in the fuzzy framework", Journal of Fuzzy Sets and Systems 148, 5-19,2004.

10) D. H. Widyantoro, J. Yen, *"A Fuzzy Similarity Approach in Text Classification Task"*, Proceedings of Ninth IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE 2000), pp. 653-658, San Antonio, Texas, May 2000.

11) "Decision tree in text classification", http://www.cs.cmu.edu/afs/cs/academic/class/15381-f00/.../lec1024.ps

12) "Definition of fuzzy logic", http://www.investopedia.com/terms/f/fuzzy-logic.asp

13) H. Dong, F. K. Hussain, & E. Chang, *"A survey in traditional information retrieval models."* In 2nd IEEE international conference on digital ecosystems and technologies (pp. 397–402).2008.

14) H. Zhang and H. Song, *"Fuzzy Related Classification Approach Based on Semantic Measurement for Web Document",* in the International Conference on Data Mining, Hong Kong, 2006.

15) J. I. Serrano, M.D. del Castillo, I. King (Ed.) *et al.*, *"Text representation by a computational model of reading",* Lecture Notes in Computer Science, vol. 4232(I), Springer, Berlin (2006), pp. 237–246

16) "K-nearest neighbor", http://www.scholarpedia.org/article/K_nearest_neighbor

17) K. Shaban, *"A Semantic Graph Model for Text Representation and Matching in Document Mining",* PhD thesis, Electrical and Computer Engineering, Faculty of Engineering, University of Waterloo, Canada, (2006).

18) Kiani, B.A., Akbarzadeh, T.M.R., Moeinzadeh, M.H., *"Intelligent Extractive Text Summarization uses Fuzzy Interference Systems"*, in IEEE conference on engineering of intelligent systems, Islamabad. Pp. 1-4. 2006.

19) Lotfi A. Zadeh, *"Soft Computing and Fuzzy Logic"*, IEEE Software, v.11 n.6, p.48-56, November 1994.

20) M. Lan, C.L. TAN, J. SU, H.B. LOW, *"Text representations for text categorization: a case study in biomedical domain."*, In IJCNN^a07 International Join Conference on Neural Networks. (2007)

21) N. Liu, B. Zhang, J. Yan, Z. Chen, W. Liu, F. Bai, and L. Chien. *"Text representation: From vector to tensor."*, In ICDM 2005: Proceedings of the 5th IEEE International Conference on Data Mining:725–728. IEEE Computer Society, 2005.

22) "Naïve Bayesian classifier", http://www.en.wikipedia.org/wiki/naive_bayesian_classifier

23) "Noisy text analytics", http://www.en.wikipedia.org/wiki/word_sense_disambiguation

24) Porter M.F. *"An algorithm for suffix stripping".* Program. 1980; vol-14, issue-13, pp-130-137

25) Shian-chi Tsai, Jung-yi Jiang, Chunder Wu, shie-ju Lee, *"A Fuzzy Similarity based Approach for Multilabel Document classification"*, in second international workshop on computer science and engineering,2009, wcse '09,Quingdao. Pp. 59-63.

26) Smirnov, *"Overview of Stemming Algorithms."* DePaul University (2008)

27) "Support vector machine", http://www.en.wikipedia.org/wiki/support_vector_machine

28) Timothy J. Ross, *Fuzzy logic with Engineering applications,* U.K.: willey, 2005

29) V. Cross, *"Fuzzy information retrieval."* Journal of Intelligent Information Systems, 3, 29–56 (1994).

30) "word sense disambiguation", http://www.en.wikipedia.org/wiki/word_sense_disambiguation

31) "WordNet", http://www.springerlink.com/content/n2516j53k5p26x76/

List of Abbreviations

SVM	Support vector machine
KNN	K-nearest neighbor
IR	Information retrieval
POS	Part of speech
WSD	Word sense disambiguation
VSM	Vector space model
TF	Term frequency
IDF	Inverse document frequency
TDM	Term Document matrix
TMG	Term matrix generator
MATLAB	Matrix laboratory
NLP	Natural language processing