Cognitive Search Technique for Textual Data

Dissertation

Submitted

in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Engineering

By

Kamalpreet Kaur Dhoat

Roll No. 121122005

Under the guidance of

Dr. S.U.Ghumbre

Department of Computer Engineering and Information Technology
College of Engineering, Pune-411005
June-2013

DEPARTMENT OF COMPUTER ENGINEERING AND INFORMATION TECHNOLOGY,
COLLEGE OF ENGINEERING, PUNE

CERTIFICATE

This is to certify that the dissertation titled

Cognitive Search Technique for Textual Data

has been successfully completed

By

Kamalpreet Kaur Dhoat
(121122005)

and is approved for the degree of

Master of Technology in Computer Engineering

Dr. S. U. Ghumbre
Guide,
Department of Computer Engineering
and Information Technology,
College of Engineering, Pune,
Shivaji Nagar, Pune-411005.

Dr. J.V.Aghav
Head,
Department of Computer Engineering
and Information Technology,
College of Engineering, Pune,
Shivaji Nagar, Pune-411005.

June 2013

Abstract

People have their own way and level of understanding of defining and describing an incident and because of this particular difference, they narrate same thing in different ways. In some situations, it becomes mere necessity to get the common information from these different narrations. For instance: getting the theme, topic from these narrations. It can be considered as a partly a part of text mining and text categorization. Broadly speaking text mining is a way to evaluate the information to get the topic out of the given text where as text categorization means to categorize the given text, or data for some labels. In this project system learn different documents and then extract relevant and important underlying meaning from it (topic detection), then categorize the textual data and then finally recognizing the pattern for getting the previous textual data that is related to given textual data or document in some sense under the same category. As its doing mining job (to some extent) ,categorization, and recognition, so instead of naming it any one , it should be called as Cognitive Search Technique, as Cognitive Science related to learning, understanding and recognizing by human or machine, and here we are doing that to some extent**.** In Artificial intelligence some of them used neural network application, some used fuzzy logics, and etc while other used reinforcement learning technique as in reinforcement learning algorithms keep on modifying its approach with every inputs and feedback (rewards it gets). Here, we have used hybrid approach, which uses machine learning SVM algorithm for Classification (SVM is optimized using PSO) and then decomposed MLP (multilayer perceptron) for pattern recognition efficiently. It can has applicability, in various areas, like in Content based search, checking plagiarism, spam filtering, for checking different records where a common conclusion or inference is need to be drawn. Since Natural language comes unstructured data it need text pre-processing before applying categorization and pattern recognition algorithm over it.

Keywords: text categorization, Cognitive Search, SVM, NLP, reinforcement learning

Acknowledgments

I express my sincere gratitude towards my guide **Dr. S.U.Ghumbre** for his constant help, encouragement and inspiration till date the project work. Without his invaluable guidance, this work would never have been a Reached to this level.

I would also like to thank all the **faculty members and staff** of Computer and IT department for providing us ample facility and flexibility and for making my journey of post-graduation successful.

Last, but not the least, I would like to thank my classmates for their valuable suggestions and helpful discussions. I am thankful to them for their unconditional support and help throughout the year.

Kamalpreet Kaur Dhoat

College of Engineering, Pune
17 June, 2013

List of Figures

List of Tables

Content

Chapter 1

1. Introduction

Large Scale of textual data is available in different format, either online, in e-format or sometimes in on paper data. Text Categorization, Text mining, summarization and Pattern Recognition has gained attention in the recent era, with increasing popularity of web, and e - document usage in this field of information technology and linguistics circle. Text categorization for large-scale corpus has become a hot topic in the field of decision support systems. Since the most natural form of storing information is as text, text categorization and mining as well as pattern recognition is believed to have a higher commercial potential than data mining. Automated text categorization has been extensively studied, and a good survey article discusses various techniques for document categorization with particular focus on machine learning approaches. The text categorization for large-scale corpus has become the core and foundation of large-scale data processing applications [1]. One of the machine learning techniques, Support Vector Machines (SVMs) are a set of related supervised learning methods used for classification and regression, prediction tool that uses machine learning theory to maximize prognostic accuracy whereas mechanically avoiding over-fit to the information. On the other hand Pattern Recognition in text can be done in various ways one of them in multilayer perceptron. Multilayer perceptron is one of the fundamental approaches in artificial neural network used with back propagation method in it.

1.1 Text Mining

It's become easy for us to store great amount of textual information. This textual is obtainable everywhere in several formats like web, news, on paper, e-documents. However, the quantity of knowledge out there to us is consistently increasing, our ability to soak up and method this data remains nearly constant. Text Mining is that the discovery by system of latest data that was antecedently unknown to us, by mechanically data extraction from completely different sources.

In text mining, the main objective is to explore and reveal unknown data from the gathering of text documents. Text-mining, additionally referred to as data discovery from text (KDT), refers to the method of extracting fascinating patterns from terribly massive text info for the aim of discovering data[2]. It applies same analytics of data-mining however additionally improvise with novel analytical functions from language and data retrieval (IR) techniques. The distinction between regular data processing and text mining is that in text mining the patterns are extracted from language text instead of from structured databases of facts. However, there is a field called computational linguistics (also known as natural language processing) which is making a lot of progress in this domain. The fundamental limitations of text mining are that the information one needs is often not represented in the similar form in text documents. Text mining comprises of following steps as shown in figure 1.

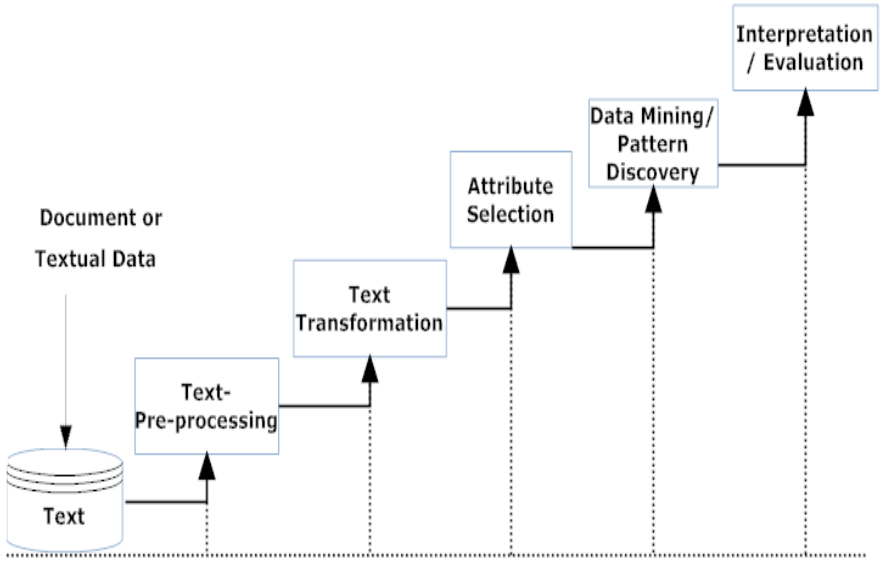


Figure 1.1 Text Mining Steps

1.1.1 Text Preprocessing

Text preprocessing is the first step in text mining where text is prepared for further processing, It is just like compiling stage in any other language processing to prepare a transformed text which will be easy to evaluate. The importance of preprocessing is emphasized by the fact that the quantity of training data grows exponentially with the dimension of the input space.[3] It has already been proven that the time spent on preprocessing can take from 50% up to 80% of the

entire classification process which clearly proves the importance of preprocessing in text classification process. The main objective of preprocessing is to obtain the key features or key terms from text documents and to enhance the relevancy between word and document and the relevancy between word and category.First convert the whole textual data into vector representation which sometimes called as “Bag of Words”, i.e. the input text is divided into individual lexical units i.e. tokens or words. After tokenization, punctuations are removed, afterwards each token is converted to lower case and further processes are applied for stop word removal, stemming, etc, these processes will make the bag of words, redundancy free, stop word free, synonym free, and stemmed words vector. These are the steps in Text Preprocessing as follows:

- **Tokenization** – Splitting up a string of characters into a set of tokens. These tokens become easy to evaluate and process in further stages. Then need to deal with apostrophes, e.g., “John’s sick”, is it 1 or 2 tokens? Hyphens, e.g., database vs. data-base vs. database.
- **Remove stop words**- Most frequently used words in English are many times useless in text classification and text mining. These words are called 'Stop words’. Stop-words, which are language-specific functional words, which represents syntactic of the languages, are part of the arrangements to understand, and are frequent words that carry no information (i.e., pronouns, prepositions, conjunctions). In English language, there are about 400- 500 Stop words [3]. Stop-words also reduce the dimension of the Bag-of-words.
- **Named Entity Recognition and Synonym Identification-** It is part of text mining where name of object, subject, etc are identified, It is required as per the application as some times name of the subject, objects are of great importance. In Language like English, One statement can be written in number of ways, and with different synonyms, (Eg. formal statements, informal words, etc.). In particular in this project no need of different synonym for one word. So reduced all the synonym of the words to its first occurrence in the document. For example (disappear, vanish, if occurs in this order than later one replaced by the former one.
- **Stemming**- Stemming techniques are used to find out the root/stem of a word. Stemming converts words to their stems, which incorporates a great deal of language-dependent

linguistic knowledge. Behind stemming, the hypothesis is that words with the same stem or word root mostly describe same or relatively close concepts in text and so words can be conflated by using stems. More precisely, stemming reduces the different forms of a word that occur because of *inflection* (e.g., plurals, tenses) or *derivation* (e.g., making a verb to a noun by adding the suffix -ation) to a common stem [15].

1.2 Text Categorization

The textual data keeps on growing at quick and however quicker pace. With this increasing rate of textual data availability, it's becoming tougher for the present system of text categorization to categorize bulk textual knowledge at quicker and efficient rate. Text categorization (additionally referred to as text classification) is that the task of sorting a collection of documents into completely different classes (or categories, or below completely different labels) from a predefined set[4]. For data retrieval researchers, this interest has a crucial facet that's towards leverage user knowledge for understanding the inherent judgments of the data retrieval task , i.e. understanding the very fact that it's the user, and solely the user, who will say whether or not a given item of data has relevancy or not. With the help of predefined categories, documents manually classified by the user square measure usually available; as a consequence, this latter knowledge will be exploited for automatically learning the (extensional) that means that the user attributes to the categories, thereby reaching levels of classification accuracy that will be unthinkable if this knowledge were unavailable. Within the last 5 years, this has resulted in more research work in Machine Learning domain and use of text categorization for regression and classification processes. For application developers, this interest is principally owing to the staggeringly exaggerated got to handle larger and bigger quantities of documents, a necessity stressed by exaggerated connectivity and availableness of document bases of every type in any respect levels within the data chain. There would be like a requirement of a system that perceive this need that not solely categorize this knowledge expeditiously instead execute on its own at acting every categorization step. Text Categorizer can categorizes the text in some predefined categories, these categorizes will be ten, hundreds or thousands of thousands. However in this case categorization can decrease economical and longer overwhelming method. Thus before

acting each Text Categorization system use some feature selection algorithms, to limit these range to few thousands.

1.3 Particle Swarm Optimization

Particle swarm optimization (PSO), originally developed by Kennedy and Eberhart[3] , is a method for optimizing hard numerical functions on metaphor of social behaviors of flocks of birds and schools of fishes. It is an evolutionary computation and stochastic optimization technique based on swarm intelligence. A swarm consists of individuals, which are called particles, which change their positions over time. Each particle represents a potential solution to the problem[16]. In PSO system, particles moves in a multidimensional searching space, during its movement each particle adjust its position according to its own experience and experiences of its neighboring particles, this results each particle moves towards the better solution areas, while still having the ability to search a wide area around the better solution areas. Performance is measured with the help of fitness value calculated by fitness function, which is the equation which is derived from the problem statement. It is fast, robust, in many cases such as in solving nonlinear, non-differentiable.

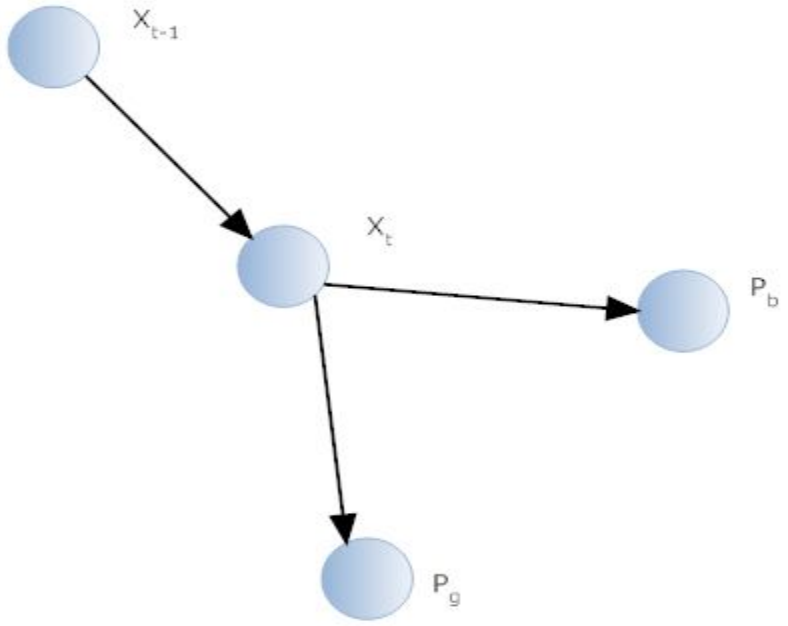


Figure 1.2 Description of particle movement in PSO

PSO has wide spread application not only in computer application, even in other areas such as Power plant, automated generation, optical fiber alignment system etc. Although PSO has some of the drawbacks like early convergences, but still it is used in variety of areas.

1.4 Support Vector Machine for Categorization

Support Vector Machine (SVM) is proposed by Vapnik in 1990's is a one of the conventional method of machine learning for purpose for classification and regression. It is based on structural risk minimization and Vapnik Chervonenks dimensions theory of Statistical Learning Theory. In order to obtain the best generalization ability, it searches for the best compromise between complexity of model and learning ability on the basis of limited sample information [1optmization]. SVM has some advantages such as theoretical foundation is complete, global optimization; training time is short and good generalization performance. It is a function that set each input value to a positive or negative class; we could also say, it assigns a positive or negative label. We can represent the training data as a set

$$X = \{(x_1, y_1), \ldots, (x_l, y_l) : x_i \, 2 \, IR^n, y_i \in \{-1, +1\}\} \quad (1)$$

where x_i are the data points and y_i their label, which can be either −1 or +1. The decision function f_X: IR^n ! {−1, +1} maps the input vectors xi to the negative or positive class. The quality of an SVM will be measured on how well it can classify new data that does not belong to the training set. Ideally, these test data should represent the complete different varieties of data. This ability to achieve a small error rate (also called a small loss) on test data is termed generalization ability. The capacity describes the ability of the machine to learn a given training set without error and measures the richness or flexibility of the function class.

1.5 Pattern Recognition

Pattern recognition is branch extended branch in machine learning (in a very broader sense) means it is the assignment of a label to a given provided input data. Pattern Recognition

Algorithms mainly aims to provide an answer to possible results or could say most likely matching input based results on the basis of statistical variation where as in pattern matching algorithms one look for the exact matching inputs with the existing patterns. Most of the popular pattern recognition algorithms are probabilistic in nature in that they use statistical inference to find the best label for a given instance [5]. There are different applications and usage of pattern recognition in variety of fields like in medical science. It is the basis for computer-aided diagnosis systems which describes a way to support the doctor's interpretations and findings. There are various algorithms in machine learning which are used in pattern recognition process such as multi layer perceptron, fuzzy neural networks, naïve bayes, gray scale arranging pairs [6](GAP basically used in image, and face recognition and related works).

1.6 MLP for Pattern Recognition

Pattern recognition is part in machine learning we can say it is a label assignment strategy to the provided data. Pattern Recognition is used in various area, in various context. In every of these fields of pattern recognition we try to find pattern in a strategic way using machine learning algorithm. It can be both supervised and unsupervised type. There are different machine learning methods, algorithms are used for pattern recognition, and one of them is Multi Layer Perceptron. Pattern Recognition Algorithms mainly aims to provide an answer to possible results. Most of the popular pattern recognition algorithms are probabilistic in nature in that they use statistical inference to find the best label for a given instance.

Multi-Layer Perceptron (MLP) is a machine learning algorithm which comes under the artificial neural network scheme. It is a feed forward ANN model that maps sets of input data onto a set of appropriate outputs. MLP consists of multiple layers of nodes in a directed graph format. In MLP each layer is fully connected to the next one. Except the first layer, which is an input layer? MLP is the simplest model which represent human neuron network to some. These neurons are processing elements which process information via nonlinear activation function. MLP uses a supervised learning technique called error back propagation. Or EBP for training the network and MLP is a modification of simple perceptron that is it can distinguish data that are non-linearly separable. There are different types of activation function used with MLP, the most

common two common activation functions are hyperbolic tangent function or logistic function where both are sigmoidal function.

$$\phi(v_i) = \tanh(v_i) \quad \text{and} \quad \phi(v_i) = (1 + e^{-v_i})^{-1}. \tag{2}$$

Where y_i is the output of the i^{th} node (neuron) and v_i is the weighted sum of the input synapses.

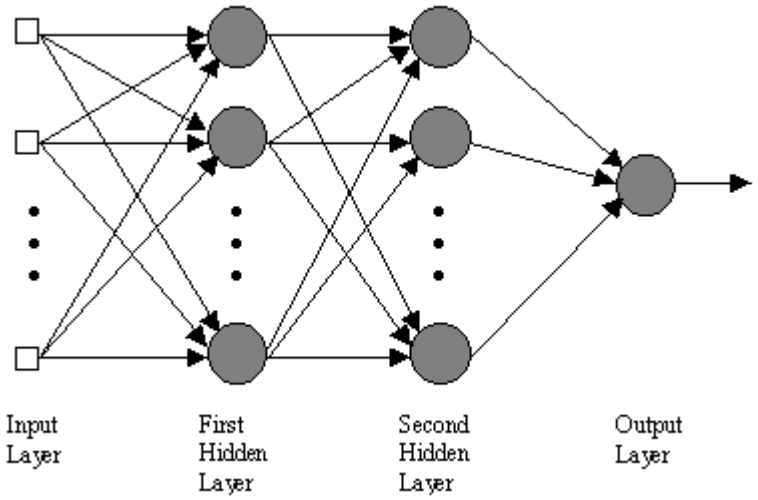


Figure 1.3 Different Layers of Multilayer Perceptron with two hidden layer

1.7 Cognitive Search and Challenges

The dictionary meaning of "Cognitive" is "of or being or relating to or involving cognition". So "Cognition" means something which is known through learning, perception, reasoning or knowledge. Cognitive search means searching with the help of learning, perception, or reasoning, with the use of knowledge. It is novel technique based on the conventional or novel machine learning, artificial intelligence methods which involves learning, knowledge discovery, etc. There is a need for a system or software to understand the kind of information a user is looking for from the provided document. Understanding the content then helping in producing rich efficient results and bringing the user-intent-specific content from the various data sources, thereby satisfying the users' information need. This is a problem in various areas such as

category browsing, genre selection, spam detection, spam filtering and many more. Although there are many different application are there in these areas which uses different novel techniques of machine learning, but none of them taken a step to relate it with cognitive search or in related area. Cognitive search of information may be attempts at something as apparently prosaic as improving search results or may be something more complex, such as attempting to create a system which can be queried with natural language search.

Since the basic need of this system is learning, drawing perception, and then recognizing like in cognitive science and these process of cognitive search are done over textual data which requires text preprocessing. As the arrangement of words in the textual data, hardly matters when it's come to classification, pattern recognition. First challenge is to convert the unstructured textual data into structured one with the help of text processing, with the help of stemming, stop word removal, redundant word problems, etc. Then to find the important words or features which are important by the classification perspective. After getting the final features main challenge begins of cognitive system that is learning for classification and pattern recognition ,where system learn for proper classification of textual data and patterns. Since one could have different data, under different categorizes, so before getting the textual documents which was fairly or nearly related to the provided textual data using pattern recognition,

1.8 Application

- The need to handle and organize documents in which the textual component is either the unique, or dominant, or simplest to interpret, component or to be recognized.
- The need to handle and organize large quantities of such documents, i.e. large enough that their manual organization into classes is either too expensive or no feasible within the time constraints imposed by the application.
- The nature of the documents; i.e. documents may be structured texts (such as e.g. scientific articles), newswire stories, classified ads, image captions, e-mail messages, transcripts of spoken texts, hypertexts, or other.
- The structure of the classification scheme, i.e. whether this is flat or hierarchical. Hierarchical classification schemes may in turn be tree-shaped, or allow for multiple

inheritance (i.e. be DAG-shaped).

- Automatic comparison of two documents. It could be extended and used with other software. Like say in comparing details for any resemblances in two separate Police cases.
- Spam Content Filtering, Which spam the blogs, etc using auto bots. Topic spotting, automatically determining the topic of a text, Language identification, automatically determining the language of a text.
- Genre classification, automatically determining the genre of a text, automatically determining the degree of readability of a text.

Chapter 2

Literature Survey

Our system implementation requires text mining then text categorization and lastly pattern recognition using the machine learning technique (SVM) and feature selection methods. So this chapter will cover the prior work and literature survey in the field of text categorization, particle swarm optimization, and multi layer perceptron.

2.1 Implementing News Article Category Browsing Based on Text Categorization Technique

This paper is basically for Thailand news article and web search for Thai languages. It is using text categorization techniques for categorize the article and data available on web. In short they called it as category browsing. They proposed category browsing allows users to browse and filter search results based on some predefined categories. To implement the category browsing feature, they have applied and compared among several text categorization algorithms including decision tree, Naive Bayes (NB) and Support Vector Machines (SVM)[7]. To further increase the performance of text categorization, they have performed evaluation among many feature selection techniques including document frequency thresholding (DF), information gain (IG) and $\chi 2$ (CHI). Based on all these experiments using a large news corpus, the SVM algorithm with the IG feature selection yielded the best performance with the measure equal to 95.42%. In this paper, they have presented comparative study of text categorization algorithms for Thai texts. They consider three algorithms: decision tree, Naive Bayes (NB), and Support Vector Machines (SVM).

2.2Automatic Text Categorization and Summarization using Rule Reduction

In this paper, they claims that, they have used parsing and rule reduction method for summarizing English language based document. In this work a text analyzer is developed to

derive the structure of the input text using rule reduction technique in three stages namely, Token Creation, Feature Identification and Categorization and Summarization. This analyzer is tested with sample input texts and gives noteworthy results. Extensive experimentation validates the selection of parameters and the efficacy of our approach for text classification [8]. This work can be expanded and used in many practical applications, including indexing for document retrieval, organizing and maintaining large catalogues of Web resources, automatically extracting metadata, and Word sense disambiguation, etc. The proposed Rule Reduction algorithm consists of the following steps.

1) Create Tokens for the given input. In this step, the overall input is split into three major categories of tokens namely alphabets, white spaces and punctuation symbols.

2) Recognize the feature of the created tokens . In this step, the features of alphabet tokens are identified namely as Determiner, Preposition, Noun, Verb, Adjective etc.

This step is done based on the rules we have defined in the text analyzer. Depending upon the rules, the features of the alphabetic words (like whether it is a noun or verb or preposition etc) are identified.

3) Categorize the alpha tokens and summarize it to a sentence. In this step, using the rules, the analyzer categorize the tokens into Noun Phrase, Possessive Pass, Prepositional Phrase or Verb Phrase based on its feature (noun, verb, preposition etc.) and then summarize them to formulate a sentence.

2.3 A Comparison between Keywords and Key-phrases in Text Categorization using Feature Section Technique

Text categorization is the main issue which affects search results. Moreover, most approaches suffer from the high dimensionality of feature space. To overcome this problem, the use of feature selection techniques with statistical text categorization is investigated. The methods were evaluated based on Chi-Square, Information Gain and Gain Ratio. The data used to test the system consisted of 1,510 documents from 2009-2010, word segmentation algorithm to key-phrase 4,408 attributes and single word 2,184 attributes [9]. Classification techniques applied Decision Tree (ID3), Naïve Bayes (NB), Support Vector Machine (SVM) and k-nearest neighbour (KNN). Results showed that the Support Vector Machine was found to be the best technique. This paper uses keywords and key-phrases for text categorization. This research also

uses feature selection methods such as Chi-Square, Information Gain, Gain Ratio to compare the based methods with classification algorithms.

2.4 A Survey of Cross-Domain Text Categorization Techniques

Instead of using conventional text categorization technique they are actually using some novel method for text categorization process. They named it cross domain text categorization technique. As per explained in their work, Cross-domain classification is more challenging problem than single domain classification problem [10]. In this paper survey of cross-domain text categorization techniques have been presented. and in this work they have compared novel methods performance with conventional methods. Some of the novel methods they introduced here are as follows

1. Expectation -Maximization Algorithm
2. Probabilistic Latent Semantic Analysis(PLSA)
3. Latent Dirichlet Allocation

Here, they also presented the strengths and weaknesses of both conventional and novel cross-domain text classification techniques.

2.5 Web Text Categorization for Large-scale Corpus

In this project they have done Categorization of Chinese document and text available on Web. It is mainly intended for large corpus. It was a complex project as Chinese language follow completely different format, as compared to any Latin based languages (English, French, etc). In this study, an approach based on Support Vector Machines (SVMs) for web text mining of large-scale systems on GBODSS is developed to support enterprise decision making. Experimental results show that their approach has good classification accuracy by incremental learning and it shows speed up of computation. The automatic text categorization area has matured and a number of experimental prototypes are available. However, most of these experimental prototypes, for the purpose of evaluating different techniques, have restricted to the heterogeneous, autonomic, dynamic and distributed internet environment [1]. Therefore,

commercial text categorization systems are not widespread. One of the reasons is uncertainty in how to adapt a machine learning approach to a variety of collections with different characteristics on internet. This paper proposes and realizes a kind of multi-task learning algorithm on large-scale corpus for Chinese text categorization. In this study, an approach based on Support Vector Machines (SVMs) for web text mining of large-scale systems on Grid nodes and is developed to support enterprise decision making. Experimental results show that our approach has good classification accuracy and it shows speed up of computation time is almost super linear time is almost super linear. In this project they are using SVM categorization method and different feature selection technique, on grid computing.

2.6 A Friendly Merger of Conceptual Expectations and Linguistic Analysis in a Text Processing System

They named the system as SCISOR system, they said it is a computer program designed to scan naturally occurring texts in constrained domains, extract information, and answer questions about that information. Their system currently reads newspapers stories in the domain of corporate mergers and acquisitions. Their system is completely based on combines full syntactic (bottom- up) parsing and conceptual expectation-driven (top- down) parsing. They have used four knowledge sources, including syntactic and semantic information and domain knowledge, interact in a flexible manner [11]. Their this integration produces a robust semantic analyzer designed to deal gracefully with gaps in lexical and syntactic knowledge, transports easily to new domains, and facilitates the extraction of information from texts . But they are still in the middle of the some of it expects and they are claiming to do in near future.

2.7 Applying particles swarm optimization for support vector machines on predicting company financial crisis

They have claimed that they are predicting company financial crisis with particle swarm optimization (PSO) to obtain optimized parameters setting for support vector machine (SVM) with feature selections. In addition, they also used integrated PSO with SVM approach to

construct the financial crisis prediction model. They hoped their proposed model will become a great analysis tool for the listed companies. This research is divided into three steps: (1) selecting features by principal component analysis (PCA) to reduce unnecessary features (2) applying PSO-SVM data mining techniques for training data, cross-validation and test data to obtain the classification accuracy rate (3) comparing with grid search for SVM and original SVM [12]. Besides, comparing that whether do selecting features will influence the classification accuracy rate or not. Finally, the experimental results showed that their proposed approach was effective in finding for the better parameter settings, and improve the hit ratio on predicting company financial crisis significantly. In fact, it can be found that the average classification accuracy rates are increased when the feature selection is applied. Furthermore, the average classification accuracy rate of the approach is 100% in the training subset, and be 88.98% in the test subset. It is evident that the PSO-SVM approach is as good as the grid search for SVM and original SVM.

2.8 Optimization of SVM Parameters Based on PSO Algorithm

They took the basis of SVM as the base that is The basic idea of SVM learning algorithm can be summarized two steps. Firstly, the input space is transformed to a higher dimensional linear feature space by a nonlinear transform function. Then the optimal linear separating plane can be constructed in this higher dimensional feature space. The nonlinear transformation can be realized by defining proper kernel function [13]. They tried to improvise the learning and generalization ability of support vector machine; they used particle swarm optimization to optimize the rbf-kernel function. The data used is isolated, non-specific vocabulary words. Their experiments results indicate that PSO algorithm can efficiently search a set of values of gamma γ, which improved the speech recognition accuracy. But the time of searching for optimal parameters is a little longer and it is need to be improved.

2.9 Fast Training of Multilayer Perceptrons

In this paper they used a fact that MLP training using back propagation algorithm is slow so a new approach was implemented by them which is faster and somewhat certain then error back propagation. It is a iterative approach which uses inverse transformation for linearization of

nonlinear output activation function, and same and gradient descent, delta rule and other proposed techniques for training the weights of the hidden layers. There experimental results shows that, their approach achieves accuracy as good as or in some cases better than perceptrons trained using error back propagation[14], and the training process is much faster than the error back propagation algorithm.

They even claimed their approach avoids local minima problem. They have also experimented their approach against pattern recognition problem using MLP and get improvised results. They have results which show that their approach is faster and way better than EBP approach for pattern recognition, classification, interpolation, regression, etc.

Chapter 3

3. Proposed System

3.1 Problem Statement

There are numerous solution to text mining, text categorization and pattern recognition, but they are doing either the former or later part efficiently without taking into consideration how much it's related to cognitive science. There is no existing system which performing these tasks together as efficiently as it needed with a cognitive science perspective. This project is to develop self improvising solution to the above problem where system not only mine and classify unstructured text efficiently but also recognize pattern those which are related to the previous records and efficiently use machine learning ways in iterative fashion to make system smarter and self improvising.

3.4 Objective

There are many objectives behind doing this project but to accomplish this project the main objective are as follows

- Text pre-processing to make it apt for further processes.
- Topical search that is finding the hidden meaning, relevant and important data from the given textual data (can be called as text mining).
- System will perform learning and then categorize this relevant important data under different labels.
- Pattern will be recognized which are fairly and nearly related to previous record under same category.

3.3 Proposed Solution

Every system is designed in a pragmatic and stepwise way, this system requires following steps:

a) Text mining requires preprocessing of the input documents to prepare it for hidden

meaning extraction or interpretation. In this step each document is goes under following things

- Document cleanup process.
- Tokenization
- Removal of stop words
- Stemming.
- creating a “bag of words” vector space representation of document features.

b) Reducing the size of the newly created vector space, by using feature selection method –tf-idf, and assigning weights to each feature and then selecting it.

c) PSO based SVM iteratively to classify these document on the basis of feature extracted and the mined data. PSO will improvise the result iteratively. PSO is specifically used to improvise the output of SVM.

d) After classification the features are used to recognize the pattern using decomposed MLP which are related to previously stored records.

3.4 System Requirement Specifications

The cognitive search system’s different modules requires the following
Software Minimum Requirement

- Java sdk
- WordNet Version 2.0
- Weka Version 3.6
- MySQL
- Windows XP operating System

Software Used

- Java sdk
- Net Beans 6.9.1

- WordNet Version 2.1
- Weka Version 3.6
- MySQL
- Windows 7 operating System

While all the implementation and experiments for system were performed on Intel i3 Core processor with 2.60GHz frequency and 4.00 GB RAM. The Weka tool (version 3.6) was used for text categorization and maximum heap size of Weka was set to 3.5 GB with Weka cache size 25007.

Chapter 4.

4. Design and Implementation

4.1 System Design

Cognitive search is used in such a way that system learns then categorizes data then draws some perception out of it and finally recognizes the patterns required by the user. So it's a novel cognitive search technique which partly used cognitive science concept with machine learning algorithms. The overview of the system can be taken from the figure 4.1. The two most basic objective of the proposed idea is to let the system learn first for topical categorization then draw a perception to recognize the pattern required as results intended by the user only. Although the main objectives are just two but to accomplish it, several other important task are needed to be done.

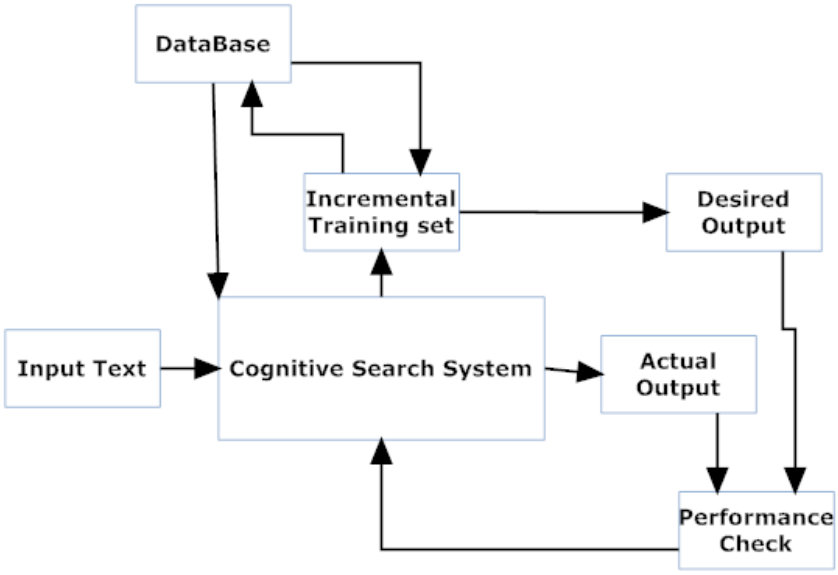


Figure 4.1 Brief Cognitive Search System Layout.

Even though the model is self explanatory, we would like to tell the main function that are performed by the system. Cognitive System is the Heart of the System. All the Categorization, Classification, Pattern Recognition Processes are performed by this module it learn through repeated training provided to the system. Actual output is the output produced by the system

while the desired output is part of the training module, both of the outputs are used in performance check if the result produced by the cognitive module is not correct then performance check module send back to the cognitive system for error correction and learning. While in case the actual output is the correct output as per the training, the new feature set is added to the increment training set model to train the system for the next input.

4.1.1 Detailed Layout of the System

The system consists of five main modules such as cognitive search system, database module, input module, Output and Training Module. Every module in the system has their specific task to accomplish. Refer Figure 4.2. for the detailed layout of all the Modules.

Input Module: This module taken in the input data, then convert the textual data into a vector of words and then send it into text pre-processing method, which remove all the stops, replace synonym and perform stemming on the input text and finally create a “bag-of-words” which is then send to Cognitive Search System.

Cognitive Search Module: First function to be performed by Cognitive Search module is to find the important feature for the given document, after that evaluate the class of the provided document for which we have done research in particle swarm optimization (PSO) based SVM classification for textual data. Lastly finding the textual document which is fairly related to the provided document for which pattern recognition is used which uses a different type of multilayer perceptron that is instead of creating one multilayer perceptron which provide m outputs for n inputs for which decomposition is done, it reduces the number of hidden nodes. In our case we used MLP with input nodes equal to the N input nodes for and M output nodes while N number of hidden nodes. But after decomposing it becomes N input nodes, 2 hidden nodes and 1 output nodes for each DMLP (decomposed MLP).

Output Module: When the output is produced by the cognitive search system it sent to the Output module where it is first get sent to the Training Set, in the training phase, if error occurred the feedback sent to the system to learn more without providing output. Once the SVM got

trained and there is no difference in the desired and actual output, the result of the test set i.e. given textual data is shown by the output module.

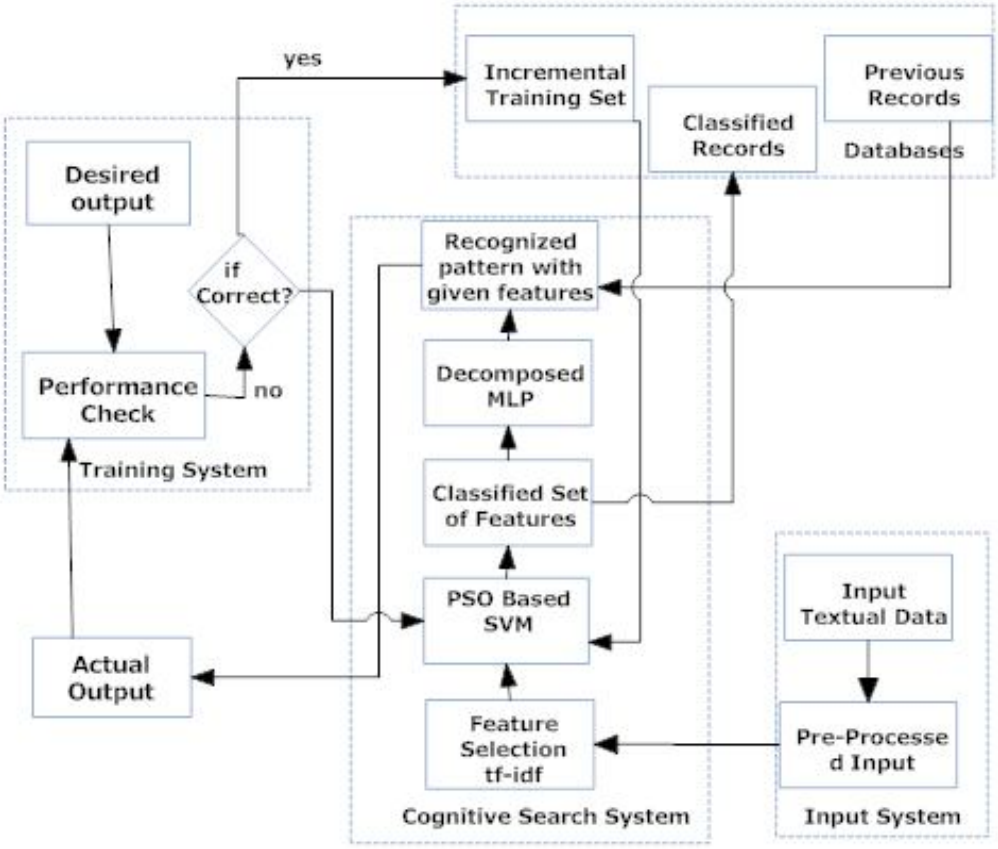


Figure 4.2 Detailed Layout of the Cognitive System

Database Module: This Module stores all the datasets for training purpose, older records and classified records, required by the system. It also has a weight record set for the decomposed MLP initialization and activation.

Training Module: The training module used is different in this system then the usual training module, it is a incremental training system, where once the results obtained by the classification are correct and satisfactory it updates the training set. This is important and different because, a conventional training set, are non-updatable static training set used, which are manually updated once in a while. Rather, this incremental training set increment every times a new text is classified in this way more and more instances will be updated to the training set which in turn increase the efficiency of the SVM and reduce the over-fitting problem in it.

4.2 Implementation

4.2.1 Text Pre-processing.

A single document could have more than hundred thousand of words and nearly half of the word don't provide even little bit of information these are stop words. In English like languages a word is used in different ways in different spelling because change in tense, voice, direct and indirect speech, etc [3]. There are many synonyms used for expressing the same bit of information. These are all important thing that need to be dealt with before using these data as input for the cognitive search. Since it is a prior process to the actual algorithm implementation, thus it is called as text pre-processing refer figure 4.3. Text pre-processing is a stage prior to any actual processing of textual input, since the input data is all unstructured, we convert it into structured format using few simple step of Text pre -processing , such as stop words removal, stemming, synonym identifying, etc.

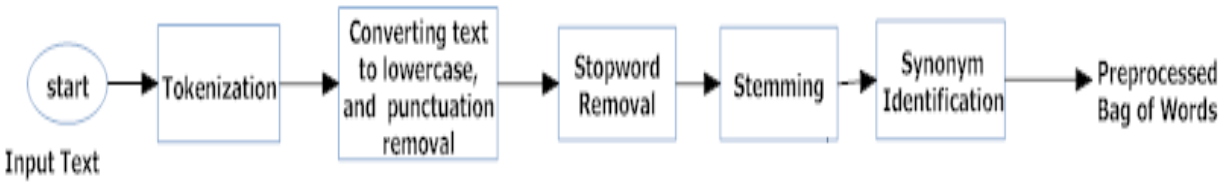


Figure 4.3 Pre-processing Step

I. Stop Word Removal Algorithm

It removes all the stopwords such as (an, the, at, of,etc) from the Array of Words called as BagofWords, and contains which provide some useful and meaningful words. In general it removes pronouns, prepositions, conjunctions from the text.

Algorithm 1. Stopword Removal

1. Input Data-Words , size N. //Array of words
2. Output Data- BagofWords. // Array of words without stopwords
3. A look up is used which contain all the stop words-Stopwords.

4. Let i=0.
5. Repeat 6 to 8 until i≠N.
6. If Words[i] matches with any word in stopwords.
7. Remove Words[i].
8. Increment i.
9. End.

II. Stemming Algorithm

It reduce all the words to their base words, i.e. consideration will reduce to consider, played will reduce to play and soon. It uses Wordnet Dictionary for words references.

Algorithm 2. Stemming Algortihm

1. Input : BagofWords //Array of words without stopwords.
2. Output: StemBagofWords // Stemmed Array of words.
3. Lookup : dict //Wordnet Dictionary.
4. Length of BagofWords : M
5. I=0
6. Repeat 7 to 16 until I ≠ M
7. Lemma = Find the all the words in Dict which are derivation or base of BagofWords[I].
8. J=0
9. Check If BagofWords[I] is a Stem => if False
10. Reduce the BagofWords[I] into Stem of BagofWords[i]
11. Else
12. Repeat 13 to 15 until J≠M
13. Check if BagofWords contain any words which matches with Lemma => if Yes
14. Reduce BagofWords[J] into BagofWords[I]
15. Increment J
16. Increment I
17. End.

III. Synonym Identification

Here also we are using WordNet for dictionary purpose and algorithm is almost similar to the stemming. After synonym identification, the size of the bagofwords and variety of words in the bagofwords reduce considerably without losing important words from the document. Now we use this bagofwords for further process like feature selection, etc.

4.2.2 Feature Selection Methods - Term Frequency – Inverse Document Frequency

One of the main problems with text categorization is the high dimensionality of feature space, i.e. variety of words. The feature set for a text document is a set of unique terms or words that occur in all documents. Feature selection is a method which reduces the number of attributes. The advantage of reducing the attribute list is the processing speed, which in turn gains higher performance. Some of them are tf-idf, CHI-square, mutual information, etc, some of them use probability occurrence of the term, some use normalization, and so on. The most important is the frequency of the feature in the document and frequency of the feature in other document, it tell how much that term is important and how much weight is to be assigned in some cases[16].

The tf-idf measures (term frequency inverse document frequency) are a numerical statistic which reflects how important a word is to a document in a collection or corpus. It is often used as a weighting factor in text mining and information retrieval whereas here we are using it as feature selection method. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to control for the fact that some words are generally more common than others [16]. After checking the threshold for each feature on the basis of weight assigned to it to the average threshold weight, then feature is selected or eliminated. The tf-idf is calculated as :

$$tf \times idf(t,d,D) = tf(t,d) \times idf(t,D) \quad (3)$$

A high weight in tf*idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents; the weights hence tend to

filter out common terms. Since the ratio inside the idf's log function is always greater than 1, the value of idf (and tf-idf) is greater than 0. As a term appears in more documents then ratio inside the log approaches 1 and making idf and tf-idf approaching 0. If a 1 is added to the denominator, a term that appears in all documents will have negative idf, and a term that occurs in all but one document will have an idf equal to zero.

Term frequency: The term frequency (TF) in the given document is simply the number of times a given term appears in that document. This frequency can be normalized to prevent the bias. TF is the measure of the importance of a particular term t within the particular d.

$$tf\,(t; d) = no: of\ occurrences\ of\ t\ in\ d$$

$$Normalized\ tf\,(t; d) = \frac{No.of\ occurences\ of\ t\ in\ d}{Total\ no.\ of\ terms\ in\ d} \tag{4}$$

Inverse Document Frequency: Inverse document frequency (IDF) is a measure of whether a particular term is common or rare across the corpus. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$idf(t,D) = log\ \frac{|D|}{|\{d \varepsilon \mathcal{D} : t \varepsilon d\}|} \tag{5}$$

I. Tf-idf Algorithm

It will find the important features to the document by using weight measure and comparisons with other documents. Finally create a List of Feature which provides important for the document.

Algorithm 3. tf_idf algorithm

1. Input : BagofWords //array of words
2. Output: Features // array of features selected from the document
3. Database : corpus // Document set which contain all the documents

4. Size of BagofWords =M
5. Repeat 6 to 15 until reach M
6. I=0,J=0
7. If BagofWords[I] previously checked =False
8. t = no of occurrences of a BagofWords[I] in BagofWords
9. Tf = t / total no of words in BagofWords // term frequency calculation
10. Other = no of times in which BagofWords[I] occurs
11. Idf =log |total documents / Other+1|
12. Tfidf[I]= Tf*Idf
13. Increment J
14. If Ends
15. Increment I
16. Mean // Mean of all the Tfidf values in the array
17. Repeat 18 to 20 until I reaches
18. K=0
19. If Tfidf[K] > =mean
20. Store corresponding word from BagofWords to Features
21. Ends

4.2.3 Particle Swarm Optimization

In a swarm of size N, every i^{th} particle, at any instant of time t, is located at a position x^t. The directional distance covered by a particle, at time t, i.e. is located at a position x^t. The directional distance covered by a particle, at time t, i.e. $x^t - x_i^{t-1}$, is known as its velocity v_i^t.
Following the description presented here, it is can be seen that if the velocity v_i^{t+1}, at time 't+1', is computed by some means, the particle's new location, x_i^{t+1}, can be estimated (as $x_i^t + v_i^t$). In PSO, the velocity of the population members surfing the search space is calculated by assigning stochastic weights to v_i^t and the attractions from a particle's personal-best or 'pbest' ($p_{b,i}$) and swarm's best or 'gbest' ($p_{g,i}$), and computing their resultant vector. The personal-best and swarm's best or global best are assigned as follows:

'Pbest'– Particle Best is the particles' own best location so far.

'Gbest'–The global best ('gbest') i.e. the best location known in the entire swarm

Once the 'pbest' and 'gbest' are found, following equations are utilized to compute velocity and position for i^{th} particle at iteration t + 1:

$$V_i^{t+1} = wV_i^t + r_1c_1(P_{b,i}^t - X_i^t) + r_2c_2(P_g^t - X_i^t) \quad (6)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (7)$$

Here, r1 and r2 are random numbers in [0, 1], and w, c1 and c2 are pre-specified constants. In each iteration, particle is updated serially according to above position and velocity rules.

I. PSO Algorithm

Algorithm 4. PSO

1. Input : particle values //randomly generated particles
2. Output : Optimum value // as per fitness function
3. T=0
4. Initialize population of particles P_t :
5. For I =1 to N
6. Initialize X_i^t, V_i^t =with random velocity and set $Pbest_i^t$={ X_i^t}
7. End
8. Evaluate Fitness function for all { X_i^t }
9. Initialize Gbest with X_i^t which satisfy gives maximum value for fitness function
10. Evaluate $V_i^{t+1} = wV_i^t + r_1c_1(P_{b,i}^t - X_i^t) + r_2c_2(P_g^t - X_i^t)$ for all { X_i^t }
11. Evaluate $X_i^{t+1} = X_i^t + V_i^{t+1}$
12. Check boundary condition for X and V
13. Evaluate Fitness function for all { X_i^{t+1} }
14. Update $Gbest_{t+1}$ if if fitness value for any X_i^{t+1} is better than $Gbest_t$
15. T=T+1

16. Repeat 10 to 14 for until the termination criterion is met
17. Final Gbest is the Optimum solution
18. End

4.2.4 Support Vector Machine

Many real world data cannot be separated linearly in a reasonable way, not even by using soft margins. In most cases, the process by which the data were generated simply cannot be approximated by a linear function. A loophole is to employ a function φ, the feature map, which maps the data points x_i of the data space L to the feature space H where a linear separation is possible.

$$\begin{aligned} \Phi : \mathbb{R}^n &\rightarrow \mathcal{H} \\ x_i \in \mathcal{L} &\rightarrow \Phi(x_i) \in \mathcal{H} \end{aligned} \tag{8}$$

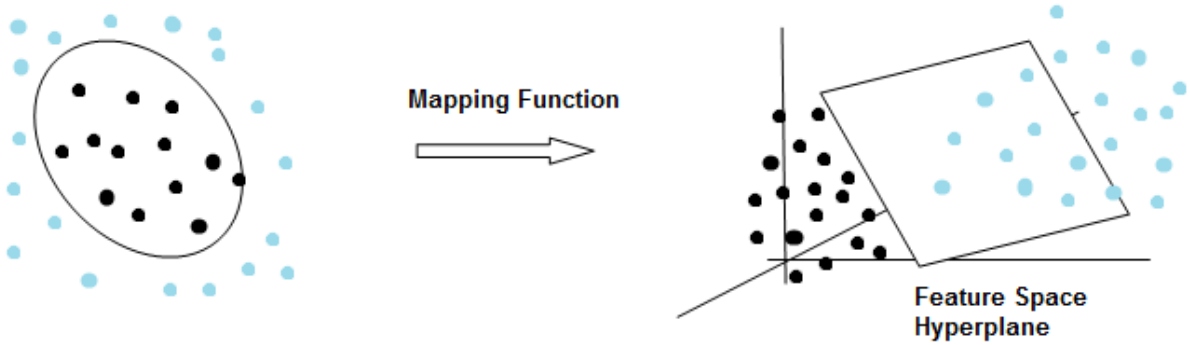


Figure 4.4: Non Linearly Separable Class in actual space to Linearly Separable Feature Space.

The feature space H must be a Hilbert space, which is a vector space in which a dot product (scalar product) is defined and has notions of distance and of angle. As shown in figure 4.2[18] in the first part it is the actual data, after employing kernel function φ it converted into feature space which is linearly separable.

I. SVM Algorithm

Algorithm 5. SVM algorithm

1. Input : Features set // features of the input text
2. Output : Class // category of the input text

3. Features are converted into test set **classify1** that is a set of binary values as per 92 attributes in training set
4. Load the training set **classify** from the database
5. SMO scheme is used with cache 25007 and RBF kernel with variable gamma called from PSO.
6. Weka API is used for creating 10 folds cross-validation for training the SVM.
7. After Training same SVM scheme object is run over test set
8. It provide the class under which Given text is classified.
9. End

4.2.5 MultiLayer Perceptron

Consider Figure two layer MLP for understanding the Forward pass and backward pass here.
In the forward pass we calculate following values for activation function and MLP output:

$$u_j(t) = \sum_i v_{ji}(t) x_i(t) \quad (9)$$

$$z_j = g(u_j(t)) \quad (10)$$

Where v_{ji} are the weights from all the inputs to the hidden nodes. u_j is the weighted sum while g() is the activation function over weighted sum of all the inputs. And the z_j represents the values of the hidden nodes.

Multi-Layer Perceptron (MLP) is a machine learning algorithm which comes under the artificial neural network scheme. It is a feed forward ANN model that maps sets of input data onto a set of appropriate outputs. MLP consists of multiple layers of nodes in a directed graph format. In MLP each layer is fully connected to the next one.

Except the first layer, which is an input layer? MLP is the simplest model which represent human neuron network to some. These neurons are processing elements which process information via nonlinear activation function

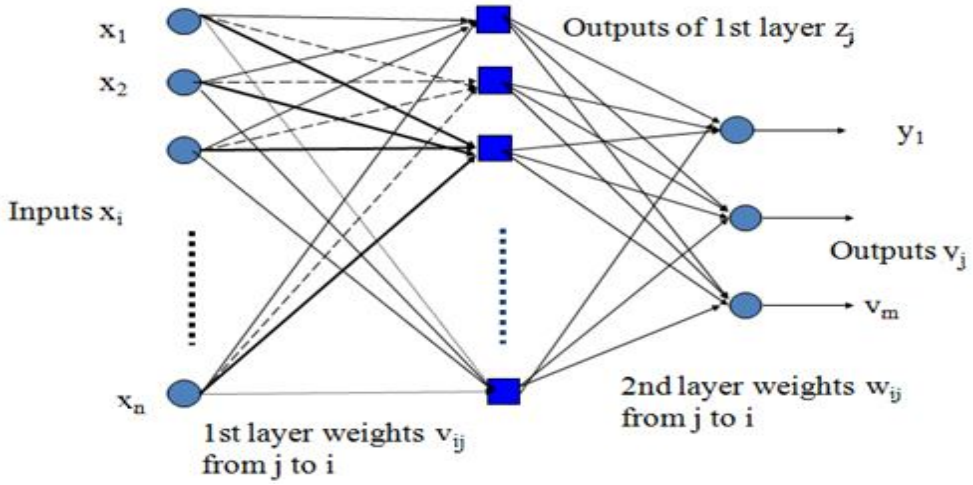


Figure 4.5 MLP with three layer network.

Where w_{kj} are the weights from all the hidden nodes to the output nodes. a_k is the weighted sum while g() is the activation function over weighted sum of all the hidden layer values and the y_k represents the outputs.

$$a_k(t) = \sum_j w_{kj}(t) z_j \qquad (11i)$$

$$y_k = g(a_k(t)) \qquad (11ii)$$

Error Back Propogation (EBP) is a gradient decent based a scheme to use the error produced in the system to correct the weight and indirectly correcting the results. The gradient descent method of single layer perceptron used on the error function to find the correct weights.[21] We see that errors/updates are local to the node i.e. the change in the weight from node i to output j (w_{ji}) is controlled by the input that travels along the connection and the error signal from output j. We therefore want to find out how weight w_{ij} affects the error i.e. we want $\frac{\partial E(t)}{\partial w_{ij}(t)}$

which is equivalent to the δ and Δ for the two network layer as in the following figure. Here we are focusing on three layers, but could easily generalize for more layers,

$$\Delta_i(t) = (d_i(t) - y_i(t)) g'(a_i(t)) \qquad (12)$$

$$\delta_i(t) = g'(u_i(t)) \sum_k \Delta_k(t) w_{ki} \qquad (13)$$

where g'() is again activation function, while di is the required outputs while yi is the actual

outputs. Now changes are done in weights with the help of these Δ and δ values as follows:

$$v_{ij}(t+1) = v_{ij}(t) + \eta\delta_i(t)x_j(t) \quad (14)$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta\Delta_i(t)z_j(t) \quad (15)$$

where η is the learning rate of the perceptron, it should not be too small and too large because too small learning rate will slow down the whole process, while too large while add errors and over fitting issues. Once weights are computer the new z_i and y_k are computed and again the same procedure are followed for the given number of iterations or until MLP trained and provide corrected results.

4.2.6 Decomposed MLP

Multi Layer perceptron is a indeed one of the best technique for pattern recognition process, but the it's quite time taking and cumbersome process, especially the EBP part, as the number of hidden layer, also depends on the output layer so, instead of taking all the output together we create MLP for each output. By doing so we can run those with different threads as they are independent and the training time will also be saved as some of extra hidden nodes are eliminated. It is very beneficial for the textual pattern recognition, as there is more number of patterns to be recognized here. Decomposed MLP will converge more quickly towards the correct result and hence learn faster as compared to simple MLP [20].

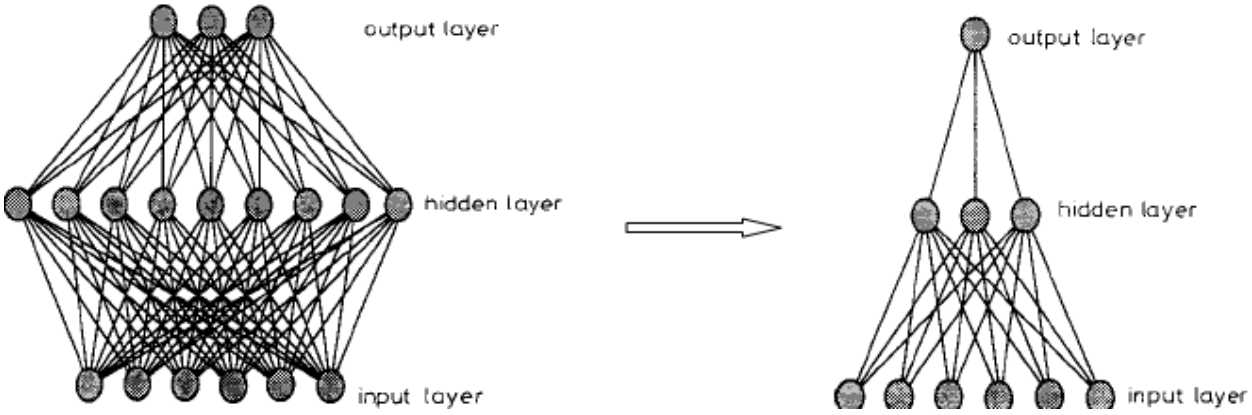


Figure 4.6 MLP decomposition as per Output.

Here, while decomposing the no of hidden nodes are decreased as H/M where H is the number of hidden node in MLP and M is the number of output nodes in the MLP. It may or may not work in all the cases, so instead of rushing to a specific number in DMLP. We find the number by constant training and trying smaller number of nodes and by keep the results accurate.

I. DMLP algorithm

Algorithm 7. DMLP_Algorithm

1. Inputs : Feature //Features from the input Text
2. Outputs: Given Patterns Present or Not
3. Training Set : Dataset which have given patterns
4. Forward Pass
5. No of input nodes N, No of Hidden node per MLP 2 , No of output per MLP 1
6. Compute u_i and z_j with the help of activation function and inputs
7. Compute a_j an y_k with help of of z_j and u_i
8. Check the y_k with d_k where d_k is the required results while y_k are the actual results if not same follow 9 to step otherwise break the loop
9. Back-Propagation
10. Compute δ and Δ
11. Use η =0.1 as learning rate
12. Compute new v_{ij} and w_{jk}
13. Goto 6 and Use new v_{ij} and w_{jk} in the next iteration
14. Stop training
15. Now repeat 5 to 7 to find the patterns
16. End.

Chapter 5

Experiment and Results

In this chapter we briefly present the result of Cognitive Search System, where we presented comparative result for PSO based SVM for different data. We also presented the System final results. Here instead of taking random documents, we selected a domain for the experimentation of the whole process, and the domain is police and crime records.

5.1 Datasets used for experimentation

Since we have created a cognitive system for which data set we needed is not available so we have created our own training and test set of textual data. In the training set for classification, there are 94 attributes and over 150 instances which keep on incrementing after every correct categorization. While there are 30 to 40 test cases which are tested in this system. We have also tried to get police complaint and Fir from police station but because of the confidentiality issues they didn't provided us with the data.

Fields	Type	Null	Default
Sno	int(5)	NO	Null
ransom	boolean	YES	0
missing	boolean	YES	0
taken	boolean	YES	0
abduct	boolean	YES	0
disappear	boolean	YES	0
vanish	boolean	YES	0
find	boolean	YES	0
dead	boolean	YES	0
body	boolean	YES	0
kidnapper	boolean	YES	0
fire	boolean	YES	0
burn	boolean	YES	0
kidnap	boolean	YES	0
hanging	boolean	YES	0

Fields	Type	Null	Default
poisoning	boolean	YES	0
pesticides	boolean	YES	0
rob	boolean	YES	0
money	boolean	YES	0
robber	boolean	YES	0
property	boolean	YES	0
loot	boolean	YES	0
force	boolean	YES	0
theft	boolean	YES	0
steal	boolean	YES	0
mug	boolean	YES	0
extortion	boolean	YES	0
accident	boolean	YES	0
bleeding	boolean	YES	0
serious	boolean	YES	0

Table 5.1 Attribute Set for the Training Set part 1

Fields	Type	Null	Default
damage	boolean	YES	0
break	boolean	YES	0
unconscious	boolean	YES	0
vehicle	boolean	YES	0
robbery	boolean	YES	0
company	boolean	YES	0
employer	boolean	YES	0
boss	boolean	YES	0
colleague	boolean	YES	0
stares	boolean	YES	0
underage	boolean	YES	0
work	boolean	YES	0
run	boolean	YES	0
labor	boolean	YES	0
factory	boolean	YES	0
industry	boolean	YES	0
fraud	boolean	YES	0
illegal	boolean	YES	0
papers	boolean	YES	0
cheat	boolean	YES	0
drug	boolean	YES	0
smuggle	boolean	YES	0
Cocaine	boolean	YES	0
Heroin	boolean	YES	0
bar	boolean	YES	0
callgirl	boolean	YES	0
arm	boolean	YES	0
alchohol	boolean	YES	0
gold	boolean	YES	0
mafia	boolean	YES	0
underworld	boolean	YES	0

Fields	Type	Null	Default
plan	boolean	YES	0
crime	boolean	YES	0
gangster	boolean	YES	0
rape	boolean	YES	0
assault	boolean	YES	0
sexual	boolean	YES	0
call	boolean	YES	0
killer	boolean	YES	0
murder	boolean	YES	0
weapon	boolean	YES	0
killed	boolean	YES	0
death	boolean	YES	0
female	boolean	YES	0
girl	boolean	YES	0
lady	boolean	YES	0
gang	boolean	YES	0
child	boolean	YES	0
abuse	boolean	YES	0
beaten	boolean	YES	0
harrasment	boolean	YES	0
suicide	boolean	YES	0
depression	boolean	YES	0
mental	boolean	YES	0
disorder	boolean	YES	0
Marijuana	boolean	YES	0
pimp	boolean	YES	0
sex	boolean	YES	0
worker	boolean	YES	0
prostitute	boolean	YES	0
tax	boolean	YES	0
Class	varchar(40)	YES	Null

Table 5.1 Attributes Set for the training of the System part 2

5.2 Parameters Used for Comparison

For classification tasks, the terms true positives, true negatives, false positives, and false negatives compare the results of the classifier under test with trusted external judgments. This is illustrated as in Figure 5.1:

	actual class (expectation)	
predicted class (observation)	**tp** (true positive) Correct result	**fp** (false positive) Unexpected result
	fn (false negative) Missing result	**tn** (true negative) Correct absence of result

Figure 5.1 Classification Model

Precision: The fraction of documents correctly classified in class c out of all the documents classified in class c.

$$Precision = \frac{tp}{tp + fp} \tag{16}$$

Recall: The fraction of documents classified in class c from all the documents actually belonging to class c.

$$Recall = \frac{tp}{tp + fn} \tag{17}$$

F-measure: A measure that combines precision and recall is the harmonic mean of precision and recall, the traditional F-measure or balanced F-score.

$$F\ measure = 2 * \frac{precision * recall}{precision + recall} \tag{18}$$

Kappa Metrics: Kappa Statistic is interesting in the sense that it actually tries to compare the accuracy of the system to the accuracy of a random system.

$$kappa = \frac{totalAccuracy - randomAccuracy}{1 - randomAccuracy} \quad \textbf{(19)}$$

Total accuracy is simply the sum of true positive and true negatives, divided by the total number of items, that is:

$$totalAccuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (20)$$

Random Accuracy is defined as the sum of the products of reference likelihood and result likelihood for each class. That is,

$$randomAccuracy = \frac{ActualFalse * PredictedFalse + ActualTrue * PredictedTrue}{Total * Total} \quad (21)$$

5.3 Results of SVM parameter for different gamma value.

We have used Radial Basis Kernel Function which provide better results than poly kernel function with 10 fold cross validation for textual data. Kernel Function has different parameter in itself one of them is Gamma given by K (xi, xj) = exp {-γ * | xi- xj|^2} .On varying this gamma value we have got differ results for different values in data set as follows:

S.no.	no of Attributes	no of instances	gamma	Kappa	Correctly Classified	Avg -F-Measure
1	10	286	0.01	0	0.7027	0.58
2	10	280	0.01	0	0.7027	0.58
3	5	286	0.01	0	0.7027	0.58
4	10	112	0.01	0.3	0.7533	0.65
5	5	112	0.01	0.3	0.7533	0.65
6	10	286	0.02	0.328	0.7098	0.596
7	10	286	0.05	0.229	0.7307	0.581
8	10	286	0.07	0.2742	0.7483	0.71
9	10	286	0.18	0.2885	0.7552	0.715

Table 5.2 Comparative Results of Different Gamma values

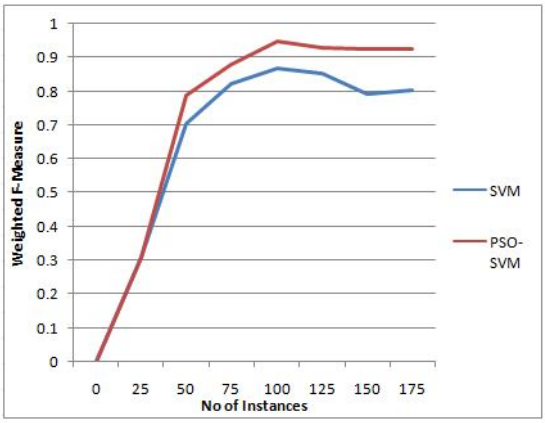
As we can see here with varying gamma and no of instances the kappa statistics, F measure and correctly classified result vary to some extent, while no of attributes put no effect. After studying different result we came to conclusion that we got most of the optimum results for kappa, correctly classified and F- Measures in the range from 0.01 to 0.2 range of gamma, So with the help of PSO we try to optimize these value for gamma range in between 0.01 to 0.2. Also we can conclude that small variation in no. of instance doesn't put any big change in the above used parameters.

5.4 Comparative Results of SVM with PSO Based SVM Scheme

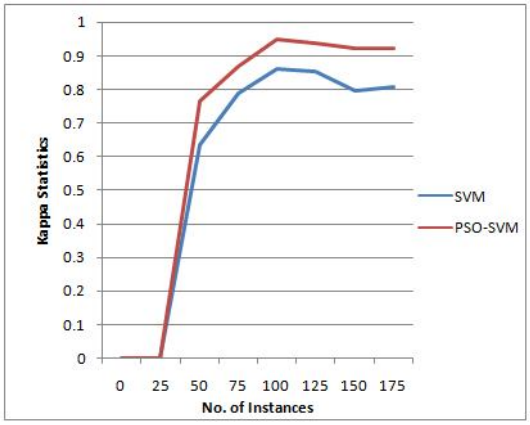
	Fmeausre		kappa Satistics		Correctly Classified	
Total No. of instance	SVM	PSO-SVM	SVM	PSO-SVM	SVM	PSO-SVM
0	0	0	0	0	0	0
25	0.307	0.307	0	0	47.62	47.62
50	0.703	0.789	0.6353	0.7675	75.41	83.61
75	0.823	0.879	0.7908	0.87	86.14	91.09
100	0.867	0.948	0.862	0.9513	89.23	96.15
125	0.852	0.928	0.854	0.938	87.73	93.15
150	0.791	0.924	0.7959	0.924	89.32	93.37
175	0.801	0.925	0.8101	0.925	83.87	93.41

Table 5.3 Comparative Results of SVM and PSO-SVM

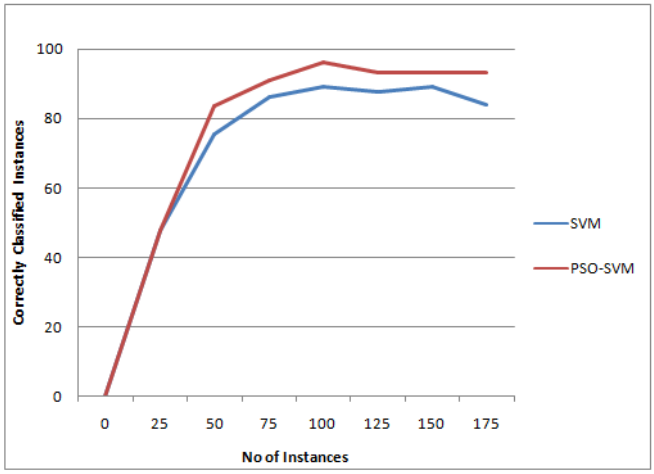
Although the results are not overwhelming for PSO-SVM, but still it's quite better than Simple SVM performance for the test set. We have also created graph where we can clearly visualize the improvement in classification values for our textual incremental training set records. They are as follows:



A



B



C

Figure 5.2 SVM v/s PSO-SVM Graphs where graph A shows Weighted F-Measures results, B for Kappa Statistics, where as the C shows the percentage of instances which are correctly classified.

5.5 Result of the Cognitive System- CRPIS

Here we have created an application for showing the implementation and the usefulness of the Cognitive Search Technique proposed by us. It takes in Police or crime records; it will first classify the record under the 14 to 15 crime categories that we have considered for this system. After classification it recognizes the patterns and finds out which of the previous records have the same pattern and provide a list of it. The first GUI that will be visible to user is as follows,

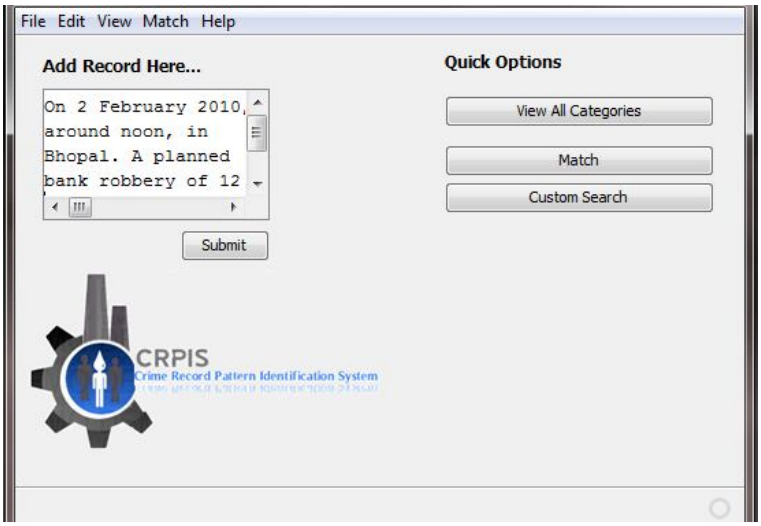


Figure 5.3 GUI of the system

After Entering text in the given text area, on pressing the Submit Button, You will get an option to save the record.

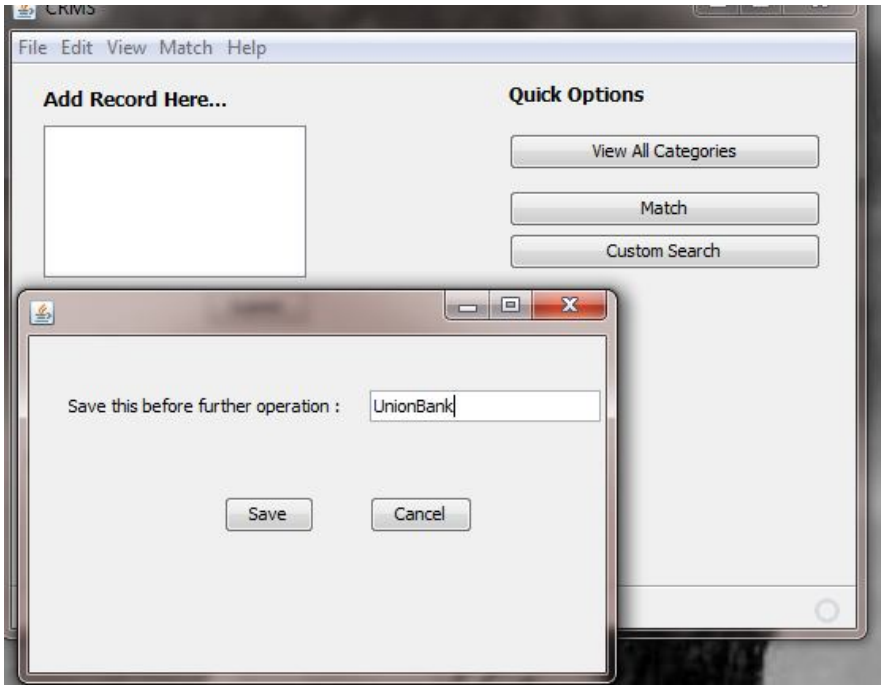


Figure 5.4 Saving the Record submitted.

Feature that are extracted from the given document is as follows.

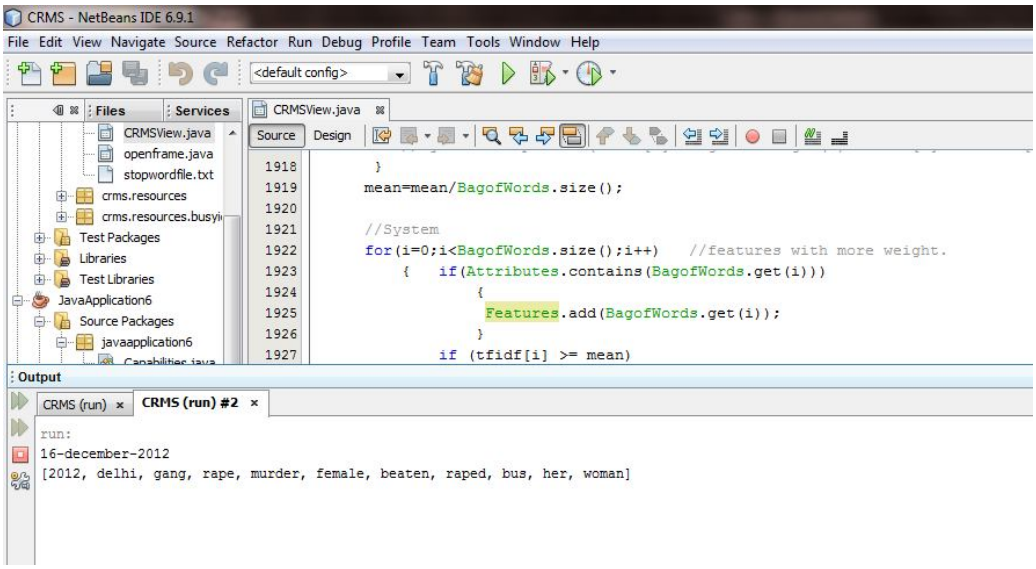


Figure 5.4 Results for Feature Selection.

This is the output that will get after entering the record for the categorization purpose. It classify all the crime record using PSO based SVM where PSO is used after every 50th increment in the training set to get the optimum parameter values for the SVM parameters.

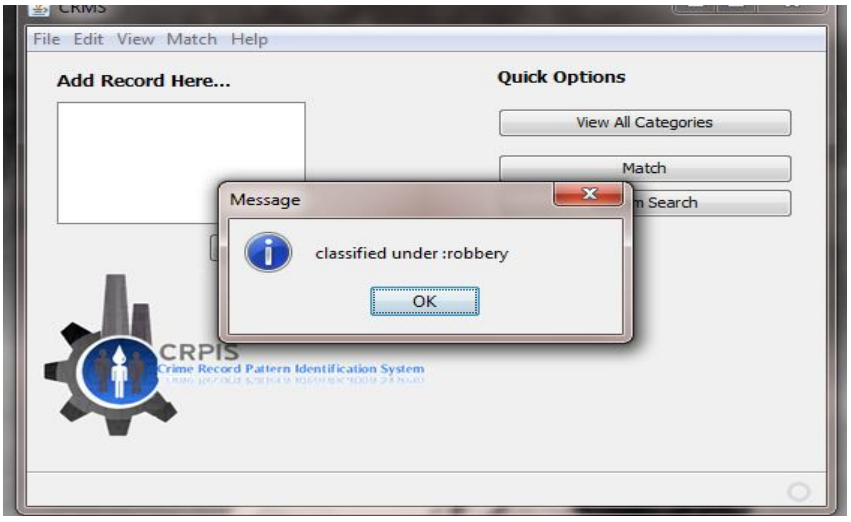


Figure 5.5 GUI for Classification Result

On after closing the Message Box by pressing ok , We have to press the Match button to get all the relevant records that are somewhat related to the given textual record and We will get the following results.

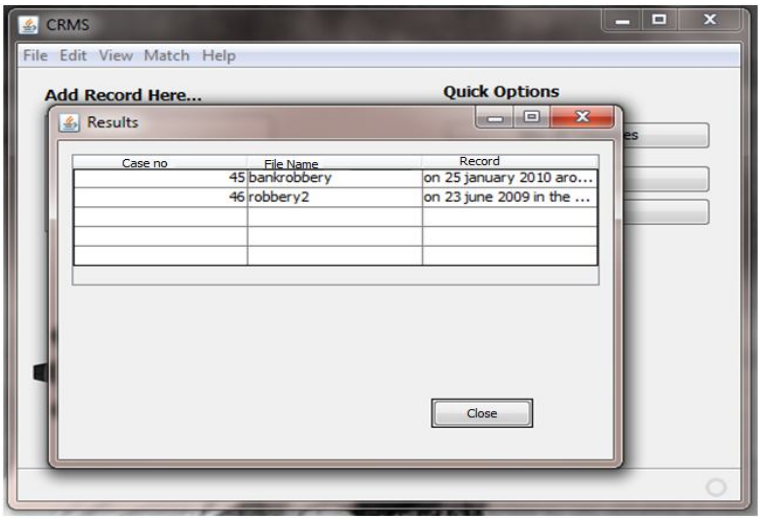


Figure 5.6 Results after Pattern Recognition in previous Record

These are the results we get after performing decomposed MLP for Pattern recognition, over the selected features for submitted textual data. It will be considered as the test set, while it performs training over the already saved training set.

Chapter 6

6.1 Conclusion

The proposed solution and implementation of the system justify its name i.e. it's not only learn for classification and but also successfully recognize different pattern in the textual data. It also shows satisfactory performance in categorization and decomposed MLP for pattern recognition, just PSO still taking large amount of time to give optimize result. PSO and Decomposed MLP are the important module of the system. The algorithm proposed and used for same are working well for the majority of the textual data.

There are many area where this proposed system can be used, such genre classification and recognition, etc. or anywhere where classification, recognition are need to be done in optimized fashion or rather say in cognitive way. Cognitive Search technique can be apply wide spread area, even though it could be improvise in many ways, but still it provide satisfactory result. Since the system doesn't use any online application it can be used where privacy are concerned. Performance of the system doesn't depend on the single method, instead it depends on the all the parts. Cognitive System Performance can also be improved if in future some novel technique (may be like hybrid GA in PSO) with better efficiency then PSO based SVM will be used. Since it's an incremental training set system that's makes it different from other systems.

6.2 Future Scope

As the digital data is kept on increasing, these research areas will keep on getting boon in it. Since the data is increasing lots of work can be done and should be done in this area. Since our focus is not domain related textual data it is wide spread. Our research based application is just a prototype it could be expanded to larger extent. Everything we did is basically serial we can enhance and make it into parallel over the distributed or grid computing system.

Here the classification is done into single class, in future the provision need to provide for sub categories. That is the categories are organized into taxonomy by tree structure.

There are many modifications that can be done at this system level, which can improvise system and overall performance, such as instead of using PSO, a hybrid approach which is GA and PSO based can be used to improvise the kernel function.

Instead of using simple error back propagation, we can use inverse transformation for linearization of nonlinear output activation functions [Fast MLP], Since EBP gives problem of convergence to local maxima and local minima. Pattern recognition and Categorization are the one of the widely studied areas in NLP.

Appendix A

Paper Publication

Title	Conference Name	Published Status
A study of Various Text Categorization for Textual data	International Conference on Computer Science and Information Technology on 17th February 2013 held in Coimbatore	Published

References

[1] Zhijuan lia, lianbo Mu, "*Web Text Categorization for Large-scale Corpus*, 2010 International Conference" on Computer Application and System Modeling (ICCASM 2010).

[2] Moty Ben-Dov, "*Ronen Feldman,Text Mining and Information Extraction*" , The Data Mining and Knowledge Discovery Handbook 2005, 801-831, Springer.

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

[4] ALFIO GLIOZZO, IDO DAGAN, "*Improving Text Categorization Bootstrapping via Unsupervised Learning*" , ACM Transactions on Speech and Language Processing, Vol. 6, No. 1, Article 1, Publication date: October 2009

[5] [Online] Available at : http://en.wikipedia.org/wiki/Pattern_recognition

[6] XinyueZhao , ZaixingHe , ShuyouZhang , ShunichiKaneko , YutakaSato, "*Robust face recognition using the GAP feature*", by ScienceDirect -PatternRecognition46(2013)2647–2657.

[7] Choochart Haruechaiyasak , Wittawat Jitkrittum and Chaianun Damrongrat "*Implementing News Article Category Browsing Based on Text Categorization Technique" in* 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

[8] C. Lakshmi Devasenal and M. Hemalatha, "*Automatic Text Categorization and Summarization using Rule Reduction*", IEEE- International Conference On Advances In Engineering, Science And Management (ICAESM -2012) March 30, 31, 2012.

[9] Vatinee Nuipian and Phayung Meesad "*A Comparison between Keywords and Key-phrases in Text Categorization uses Feature Section Technique*", 2011 Ninth International Conference on

ICT and Knowledge Engineering.

[10] M. Ramakrishna Murty, J.V.R Murthy, Prasad Reddy and S.C.Satapathy, *"A Survey of Cross-Domain Text Categorization Techniques"* , I" Int'I Cont'. on Recent Advances in Information Technology I RAIT-2012 I.

[11] Paul S. Jacobs and Lisa F. Rau, "*A Friendly Merger of Conceptual Expectations and Linguistic Analysis in a Text Processing System*", IEEE- International Conference On Advances In Engineering, Science, 2010

[12] Bo-Tsuen Chen, Mu-Yen Chen *"Applying particles swarm optimization for support vector machines on predicting company financial crisis",* 2010 International Conference on Business and Economics Research vol.1 (2011) © (2011) IACSIT Press, Kuala Lumpur, Malaysia.

[13] Xueying Zhang, Yueling Guo *" Optimization of SVM Parameters Based on PSO Algorithm",* 2009 Fifth International Conference on Natural Computation.

[14] Brijesh Verma, "*Fast Training of Multilayer Perceptrons",* Published in: Neural Networks, IEEE Transactions on Volume:8 , Issue: 6 ,Nov 1997.

[15] [Online] Available at: http://en.wikipedia.org/wiki/Stemming

[16] Wen Yue, Zhiping Chen ,Xinguo Lu ,Feng Lin ,Juan Liu, "*Using Query Expansion and Classication for Information Retrieval"*, 2006 Published in IEEE.

[17] Mohammad Javad Abdi,SeyedMohammad Hosseini, andMansoor Rezghi, "*A NovelWeighted Support VectorMachine Based on Particle Swarm Optimization for Gene Selection and Tumor Classification",* Hindawi Publishing Corporation Computational and Mathematical Methods in Medicine Volume 2012, Article ID 320698, 7 pages doi:10.1155/2012/320698

[18] C. Burges. “A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):121-167, 1998. Algorithms in Bioinformatics II, SoSe’07, ZBIT, D. Huson, June 27, 2007.

[19]Kamalpreet Kaur Dhoat, Prof. S.U.Ghumbre *“A Study of Various Categorization Techniques for Textual data”,* International Conference on Computer Science and Information Technology on 17th February 2013 held in Coimbatore.[online] Avalilable at: http://irnetexplore.ac.in/IRNetExplore_Proceedings/Coimbatore/ICCSIT_17thFebruary2013/ICCSIT_Coimbatore17thFebruary2013.html

[20] *S.* Lucas, Z. Zhao, G. Cawley and P. Noakes, “*Pattern Recognition With The Decomposed Multilayer Perceptron*”, ELECTRONlCS LETTERS 4tn March 1993 Vol. 29 No. 5.

[21] Simon Haykin , “Multilayer Percetrons” , Prentice-Hall, 2nd edition Slides do curso por Elena Marchiori, Vrije Unviersity.