A
Dissertation
On

Feature Selection Optimization using Evolutionary Algorithms for Sentiment Analysis

Submitted in Partial Fulfillment of the Requirement
For the Award of the Degree of

Master of Technology

in

Software Engineering

by

Renu Khorwal
Roll No. 2K14/SWE/15

Under the Esteemed Guidance of

Dr. Akshi Kumar
Assistant Professor

Computer Science & Engineering Department, DTU

COMPUTER SCIENCE & ENGINEERING DEPARTMENT
DELHI TECHNOLOGICAL UNIVERSITY
DELHI - 110042, INDIA

2014-2016

ABSTRACT

With the growth of web 2.0 the data present online has grown tremendously. People express their views and opinions about various products and policies and these views are very important for gauging the reaction of people towards the products and policies through sentiment analysis. Selecting and extracting feature is a vital step in sentiment analysis and greatly influences the accuracy of sentiment classification. The statistical techniques of feature selection like document frequency thresholding produce sub optimal feature subset due to the Non Polynomial(NP) hard nature of the problem. Evolutionary algorithms are used extensively in optimization problems. Optimization techniques could be applied to feature selection problem to produce Optimum feature subset. They render feature subset selection by reducing feature subset dimensionality and computational complexity thereby increasing the classification accuracy. Firefly algorithm, an evolution based optimization algorithm is used in various optimization problems to produce an optimum solution. Here firefly algorithm is used for producing optimum feature set on four different datasets. Also firefly algorithm optimization results are compared with feature selection using genetic algorithm. Firefly algorithm here increases the performance of the classification considerably in terms of accuracy and training time required to train the classifier. It produces superior results as compared to the baseline model and feature selection using genetic algorithm.

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Dr. Akshi Kumar for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

Secondly, I am grateful to Dr. OP Verma, HOD, Computer Science & Engineering Department, DTU for his immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Renu Khorwal

University Roll no: 2K14/SWE/15

M.Tech (Software Engineering)

Department of Computer Science & Engineering

Delhi Technological University

Delhi – 110042

Computer Science & Engineering Department
Delhi Technological University
Delhi-110042
www.dce.ac.in

CERTIFICATE

This is to certify that the work contained in this dissertation entitle "**Feature Selection Optimization using Evolutionary Algorithms for Sentiment Analysis**" submitted in the partial fulfillment, for the award of degree of M.Tech in Software Engineering, Department of Computer Science Engineering at **Delhi Technological University** by **Renu Khorwal**, Roll No. **2K14/SWE/15**, is carried out by her under my supervision. The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma in any university/institution to the best of my knowledge and belief.

Date: ………………

Dr. Akshi Kumar

(Project Guide)

Assistant Professor

Department of Computer Science Engineering

Delhi Technological University

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SA	Sentiment Analysis
GA	Genetic Algorithm
FA	Firefly Algorithm
OM	Opinion Mining
SVM	Support Vector Machine
NB	Naïve Bayes
EA	Evolutionary Algorithms
EO	Evolutionary Optimization
SI	Swarm Intelligence
NIA	Nature Inspired Algorithms
NP-Complete	Non-Polynomial Complete
NP-Hard	Non-Polynomial Hard
PSO	Particle Swarm Optimization
ACO	Ant Colony Optimization
ABC	Artificial Bee Colony
ATM	Asynchronous Transmission Mode
DFA	Discrete Firefly Algorithm

CHAPTER 1

INTRODUCTION

With the advent of social media, opinion-rich data resources such as online review sites and personal blogs, microblogging sites have proliferated enormously. People express their views or opinions/attitudes on a variety of issues, discuss current issues, complain, and provide feedback and suggestions for the products and policies they use in their daily life or which concerns them. This unstructured social media data is used to mine the overall attitude of the writer towards a specific issue. Sentiment Analysis or Opinion Mining [1], as an intelligent mining technique, helps to capture & determine opinions, emotions and attitudes from text, speech, and database sources, which correspond to how users retort to a particular issue or event. Being a Natural Language Processing task it tries to gauge the opinion of writer about the issue at hand and examine the overall contextual polarity of the data. Sentiment mining from social media content is a tedious task, because it needs in-depth knowledge of the syntactical & semantic, the explicit & implicit, and the regular & irregular language rules. Sentiment analysis is actively used and can further be enhanced to be used in almost every business and social domain because opinions are central to almost all human perceptions of reality and are key influencers of our behaviors. Politicians and governments often use sentiment analysis to understand how the people feel about themselves and their policies. Product manufacturing companies poll microblogging sites to gauge the general sentiment of masses for their product and rightly market and improve their product.

Sentiment Analysis is a multi-step process encompassing various sub-tasks, that are, Sentiment Data collection; Feature Selection; Sentiment Classification and Sentiment Polarity detection. Mainly sentiment analysis task can be carried out in two phases: the first phase includes the selection and/or extraction of features from the textual opinions, and the second step involves the sentiment classification of the opinion data into various polarity classes (positive, neutral, negative, etc.). Feature selection in sentiment analysis has a very significant role in enhancing accuracy of the system as the opinionated documents usually have high dimensions, which can adversely affect the performance of sentiment analysis classifier. Effective feature selection technique recognizes significant and pertinent attributes and improves the classification accuracy thereby reducing the training time required by classifier. Due to the high- dimensional, un-structured nature of the social media content, the problem of classification manifolds, thus fostering the need to look for improved & optimized techniques for feature selection. Features denote properties of textual data which are measured to classify text, like bag-of words, n-grams (unigram, bi-grams, trigrams), header information, word position and ordered word list [7].

The Traditional methods for feature selection that are chi-square, information gain, and mutual information etc. assign certain values to features based on specific statistical equation. These methods are successful in reducing the size of the corpus but with a compromised accuracy. When domain dimensionality increases, the number of features increases and it becomes inflexible to keep track of optimal feature subset. These present methods of feature selection produce sub-optimal feature subsets due to the Non-Polynomial (NP) hard nature of the feature subset selection problem. Thus, for solving the high dimensionality problem and to improve the classification performance, the focus has now shifted to assess techniques that can augment to provide superior results. Optimization techniques could be applied to feature selection problem to produce Optimum feature subset. Evolutionary algorithms have been successful at coming up with good solutions for complex problems, when there is a way to measure quality of solutions [10]. Algorithms such as Nature-Inspired Algorithms [9], Genetic Algorithms [7], Simulated Annealing [15], etc. have been explored much in literature for improved classification and optimization. Nature is a rich source of hypothesis from which many researchers get inspired. Today, in almost every field Nature-Inspired Algorithms are used to get an optimized solution for a problem. Nature-Inspired Algorithms can be classified as Swarm Intelligence Algorithms, Bio-Inspired Algorithms and Physics-Chemistry Algorithms [9].

Swarm Intelligence algorithms are used extensively in optimization problems. Swarm Intelligence is a distributed system whereby self-cooperating global behavior is produced by anonymous social agents interacting locally having local perception of its neighboring agents and the surrounding environment. These algorithms work on the principle of distribution of labor and distributed task allocation producing global patterns, the individual agents such as ants, bees, can do simple task while the cooperative work of whole colony brings out intelligent behavior [5].

Owing to the challenging research problems and extensive array of practical applications, sentiment analysis has been a dynamic area of research in the last decade. Most of the work done till date is based on insight to the problem and its solution is limited to the understanding of natural language. Moreover, processing relies on conventional lexicon-based or machine learning algorithms which are syntactically efficient but lack the "semantic" aspect to produce human understandable results. Incorporating nature-inspired techniques, more specifically swarm intelligence helps to solve the global optimization problem for feature subset selection in the classification of sentiment by extracting and exploiting the collective local & global behavioral patterns thus improving the search capability in the problem space and efficiently finding a minimal feature subset.

1.1 Motivation

The motivation for feature selection in sentiment analysis is to reduce the dimensionality, noise, redundant and irrelevant features from the feature set and produce a minimal feature set which improves the accuracy of classification and training time of classifier. Swarm Intelligence algorithms are widely used in optimization problems where the nature of problem is Non-Polynomial(NP) Hard. Swarm optimization specifically firefly optimization algorithm is used here to render feature subset selection by reducing feature subset dimensionality and computational complexity thus increasing the classification accuracy. Significant enhancement of performances of systems are compelling researchers, marketers, policy makers to invest in this area. Improved sentiment classification results can help almost every domain extending from business Intelligence to policy making capturing the true sentiment of general public. Merchants can acquire valuable feedback (e.g. consumers' satisfaction regarding their products), Public administrations can capture the understanding of e-Government and e-Rulemaking and Reputation management.

1.2 Research Objective

The objective of this work is to present a technique for feature subset selection using swarm intelligence algorithm to select optimum feature subset which improves the classification accuracy and reduces the computational complexity and removes noisy, irrelevant and redundant features. Specifically, firefly algorithm is used for feature selection and its performance is compared with genetic algorithm. The algorithm used will be explained in the forthcoming chapters with the experimental results.

1.3 Scope of work

Here, the method of feature selection with firefly algorithm is studied and also it is compared with Genetic algorithm. Twitter data and movie review data in two languages Hindi and English is used for classification. The technique proposed selects feature subset for a large feature set to improve the classification accuracy and reduced the time taken by the system. Also genetic algorithm is used on same dataset for same purpose with same classifier to compare the result with firefly algorithm.

1.4 Organization of thesis

In this chapter, I have highlighted the concept of sentiment analysis and importance of appropriate feature selection technique for sentiment analysis, motivation to do this thesis, my objective, and scope to do the work in same field. Chapter 2 provides a detailed picture of feature selection in sentiment analysis and the prior work done till date. In chapter 3 I have presented the proposed scheme, along with architecture and algorithm used to carry out research. Chapter 4 includes the implementation detail. Chapter 5 discusses the experimental results. Finally, Chapter 6 concludes the thesis, and discusses the future scope in current work.

CHAPTER 2

LITERATURE REVIEW

Sentiment Analysis is a data mining task that that extracts the user's opinion expressed in reviews, blogs, posts, tweets etc. SA has received a lot of focus from researchers, analysts in recent years. Identifying words or phrases that carry sentiments is a crucial task in sentiment analysis. Different approaches have been applied in the field of sentiment analysis but mainly can be categorized in two fields lexicon approaches and automated approaches. The work based on symbolic approaches (lexicon based) uses lexicon and set of features are generated by manually by humans. In this method the document is represented as collection of words and the sentiment of each word is evaluated using a word dictionary like SentiNet and the overall polarity of document is determined by using individual word polarity of the document. On the other hand, in automated approaches the general statistical measures are used where features are selected based on empirical indication like tf-idf(term frequency-inverse term frequency), term occurrence or term frequency. Machine learning algorithm is then applied for classifying the sentiment. The commonly used machine learning algorithms for this purpose are Support Vector Machine, Naïve Bayes, Maximum Entropy, K nearest Neighbors and Artificial neural network etc.

The workflow defining the sentiment analysis task is shown in figure1. Data on which sentiment analysis is to be performed is firstly fetched & pre-processed to select the features for further use. The selected feature subset is then sent to the classifier which gives us the polarity of the dataset. Based on polarity accuracy, precision and error rate are calculated. Each step is explained in detail in next sections. Sentiment analysis can be performed at several granularity levels, i.e., aspect level, sentence level, and document level.

Document level opinion classification

Document-level sentiment classification targets sentiment classification task where the content to be classified is based on a single entity or aspect and opinion is searched for overall document based on that entity. The polarity of individual sentiment words determines the polarity of the document as a whole. Document level classification does not work with bog posting and various forum content as the nature of content there may be varied. The author may express views and opinions on multiple aspects/entity or multiple products and may compare them with comparative sentences.

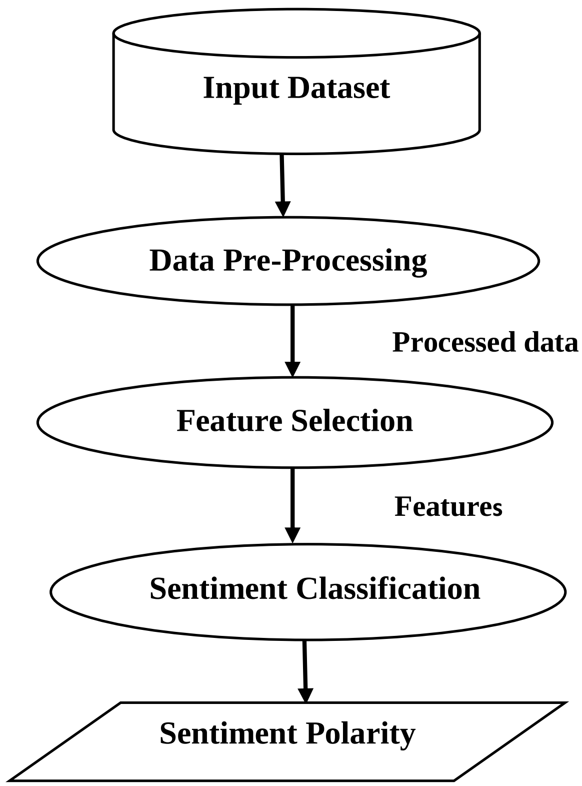

Figure 2.1: Sentiment Analysis basic workflow

Sentence level Opinion Classification

Subjectivity classification which is the process of classifying a sentence as objective or subjective and the corresponding subjective sentences are further classified as being positive, negative or neutral. This classification of subjective sentences into positive, negative and neutral expressions is called sentence-level sentiment classification. This method is similar to document level sentiment analysis except that in this polarity of each sentence is determined rather than the document as whole. The polarity of each sentence is determined in this method and it is assumed that each sentence carries opinion for one entity or aspect. And also there are some sentences which does not carry any sentiment value which are known as objective sentences. The process goes o in two steps first every sentence is classified as being subjective or objective and in the second step the polarity of each subjective sentence assessed in first step is determined. Subjective sentences contain opinion words which determine the sentiment of the whole sentence about particular entity/aspect. There are several scenarios where this form of sentiment analysis is not feasible there may be complex sentences carrying information about various entities in one sentence.

Aspect level Sentiment Classification

Typically, in a review or opinion rich content user usually writes or gives both positive and negative feedback about any entity/aspect or a particular feature of object. This classification undertakes that a document contains opinion on several entities and their features or aspects. Views on different features of a particular object may be expressed by a user in a document. For e.g. in a mobile phone review the user could mention his views on several features of the mobile phone. Sentence and document based sentiment classification does not go for such features. To perform aspect level sentiment classification, we require identification of those features, aspects, and entities and provide sentiment based on those entities.

2.1 Sentiment Analysis Process

As shown in fig 1 sentiment analysis is a multi-step process and in this section all the sub-tasks of sentiment analysis process are discussed briefly.

2.1.1 Data Collection

There are various sources of data social media sites, blogs, audit sites, review sites etc. These sites are very rich source of opinion data. The data can be collected from these platforms in two ways either using the APIs provided by the sites or by web scraping also called crawling. Various Social media sites provides APIs to extract data from their sites for analysis. For example, twitter provide the twitter API which one can gain access to after providing the required credentials. These APIs have a learning curve, but once you understand them, you'll build a real understanding of what is possible and what is not, and you won't be limited by any single data gathering tool. Another option is web scraping for the sites for which no API is there data collection becomes a tedious task so web scraping is used. Scraping is a computer software technique for extracting information from various websites. Such software programs simulate human examination of the World wide web over the Hypertext transfer Protocol.

2.1.2 Data Pre-Processing

The data is cleaned and prepared for classification in this step. This is a very important step in sentiment analysis and affects classification accuracy significantly. The aim of this process is to remove any unwanted content from the data and convert the data in a format which can be accepted by the classifier. Unwanted content is the content whose removal does not affect the classification accuracy and that will not be required by the classifier for assigning class to the classifier. Data cleaning simplifies the task of classification and also improves the classification performance in terms of accuracy and processing time. Data preprocessing involves data cleaning and data transformation which are elaborated in detail in the following section.

Data Cleaning

In this process data is cleaned all redundant data are removed. The following have to be removed to make the data clean:

- Urls
- Punctuation marks (, . ‘)
- Repeated words
- Conversion from upper case to lower case
- Emoticons
- Tags (@tags)
- Extra letters

Data Transformation

Data transformation involves the following things:

- Stop word Removal- Stop words are the words which does not carry any sentiment value and are used more frequently in text. These are functional words and does not carry any meaning they provide language a structure. Some of the commonly used stop words are “a”, “an”, “the” , “ you” , “ I” , “it”, “him” , and “and”. So these words are removed from the sentiment dataset. Also if we want to add specific words to the stop word we can add them to the stop word list.
- Tokenization- This process splits the document into sequence of tokens. Textual data comprises block of characters called tokens. The documents is separated sequence of tokens and these tokens are passed for further processing.
- Stemming-It is the process for reducing the token words to their root from or stem. Through this process relevant tokens can be described in a single type. For e.g. “He is an engineering student” This sentence after cleaning, tokenizing and stemming becomes “Engineer student”.

Words	**Stem**
User, used, using, users	Use
Teaching, teaches, teacher	teach
Engineering, engineered, engineer	engineer

Table 2.1: Different forms of the word and their root

2.1.3 Feature Selection

The next step in sentiment analysis is feature selection. This is the most crucial step of sentiment analysis with direct impact on performance. This is the step on which the work of this thesis is based upon. Many statistical feature selection methods are present and have been used successfully for sentiment analysis. Feature selection methods convert the textual data into numerical feature set and selecting most informative features among them comprises the feature selection process. The most informative features are those features which are rich in sentiment value and improves the classification accuracy. This is explained in more detail in section 2.2. Feature selection and reduction process is shown in figure 2.2

2.1.4 Sentiment Classification/ Polarity Detection & Result Interpretation

After constructing the feature vector, we apply classifier on the processed feature vector to train the classifier and classify documents. The feature vector along with the respective class labels for each tuple is fed to the classifier for training the classifier. After training the classifier the classifier could be used to predict for new data called as test data. There are various classifiers that could be used among them a few popular ones are SVM, Naïve Bayes, and Maximum Entropy. The prediction result is analyzed using some measure the commonly used measures are Accuracy, Precision, Recall and F-measure.

Accuracy – It is defined as the ratio of number of correctly classified documents over total number of documents and is defined as:

$$Accuracy \ = \frac{Number\ of\ correctly\ classified\ tuples}{Total\ number\ of\ tuples}$$

Precision- It is the ratio of retrieved instances that are relevant. Precision is defined as:

$$Precision \ = \frac{TP}{TP + FP}$$

Recall- It is the fraction of relevant instances that are retrieved. It is defined as:

$$Recall \ = \frac{TP}{TP + FN}$$

F –measure- It is a harmonic mean of the precision and recall and is defined as:

$$F - measure \ = \ 2 * \frac{Precision * Recall}{Precision + Recall}$$

Where

TP (True Positive)- number of items correctly belonging to the positive class

TN (True Negative)- number of items correctly belonging to the negative class

FP (False Positive)- number of items incorrectly classified as belonging to positive class

FN (False Negative)- number of items incorrectly classified as belonging to negative class

2.2 Feature Selection

Feature selection methods reduce the original feature set by removing irrelevant, redundant and noisy features of text to select features that are more informative and converges the classification process quickly providing more sentiment value. Effective feature Selection could improve the classification accuracy and decrease the running time or classification time of classifier.

2.2.1 Traditional Feature Selection methods

Several approaches are applied to the problem of feature selection in sentiment classification. The major and frequently used approaches are document(term) frequency (DF), information gain (IG), mutual information (MI), tf-idf, x 2 -test (CHI) and term strength (TS).

Information Gain – It provides term goodness criterion; it measures the number of bits of information obtainned for category prediction by knowing the presence or absence of term in the document. It measures information obtained (in bits) for class prediction of an arbitrary text document by evaluating the presence or absence of a feature in that text document. The expected information needed to classify an tuple for portion D or identify the class label of an instance in D is called as entropy and is given by:

$$Info(D) = -\sum_{i=1}^{n}(P_i)\log_2(P_i)$$

Where n represents the number of classes (n=2 for binary classification) and *Pi* denotes probability that a random instance in partition D belongs to class *Ci*.

Document Frequency Thresholding - It is the number of documents in which a term occurs. In this the term whose frequency is less than a particular threshold frequency is removed thus reducing dimensionality. Selecting frequent features will improve the likelihood that the features will also be comprised by prospective future test cases. The basic assumption is that both rare and common features are either non-informative for sentiment category prediction, or not impactful to improve classification accuracy. This method is simplest, scalable and effective for text classification.

Mutual Information- It is used in statistical language modeling of word association and related applications. (Yang & Pedersen, 1997). It is defined as

$$\mathrm{MI}(t, c_i) = \log\left(\frac{A \times N}{(A+E) \times (A+B)}\right) \quad \text{and}$$
$$\mathrm{MI}_{\max}(t) = \max_i(\mathrm{MI}(t, c_i))$$

where the denotations of A, B, E, D and N are the same as the definitions in CHI Method described.

CHI square method- The x^2 test is used in statistics, among former things, to test the independence of two events. More specifically in feature selection it is used to test whether the occurrednce of a specific term and the occurrence of a specific class are independent. CHI as a common statistical test represents divergence from the distribution expected based on the assumption that the feature occurrence is perfectly independent of the class value. It is defined as:

$$\mathrm{CHI}(t, c_i) = \frac{N \times (AD - BE)^2}{(A+E) \times (B+D) \times (A+B) \times (E+D)}$$
$$\text{and} \quad \mathrm{CHI}_{\max}(t) = \max_i(\mathrm{CHI}(t, c_i))$$

where A specifies the number of times *t* and *ci* co-occur; *B* specifies the number of times *t* occurs Withoutt *ci*, *E* is the number of times ci occurs without t; D is the number of times neither ci nor t occurs and N is the total number of documents. The CHI statistic will be zero if *Ci* and *t* are independent.

The x^2 statistics and information gain produce good results and are more efficient in optimizing the classification results whereas document frequency is efficient in terms of scalability and complexity [3].

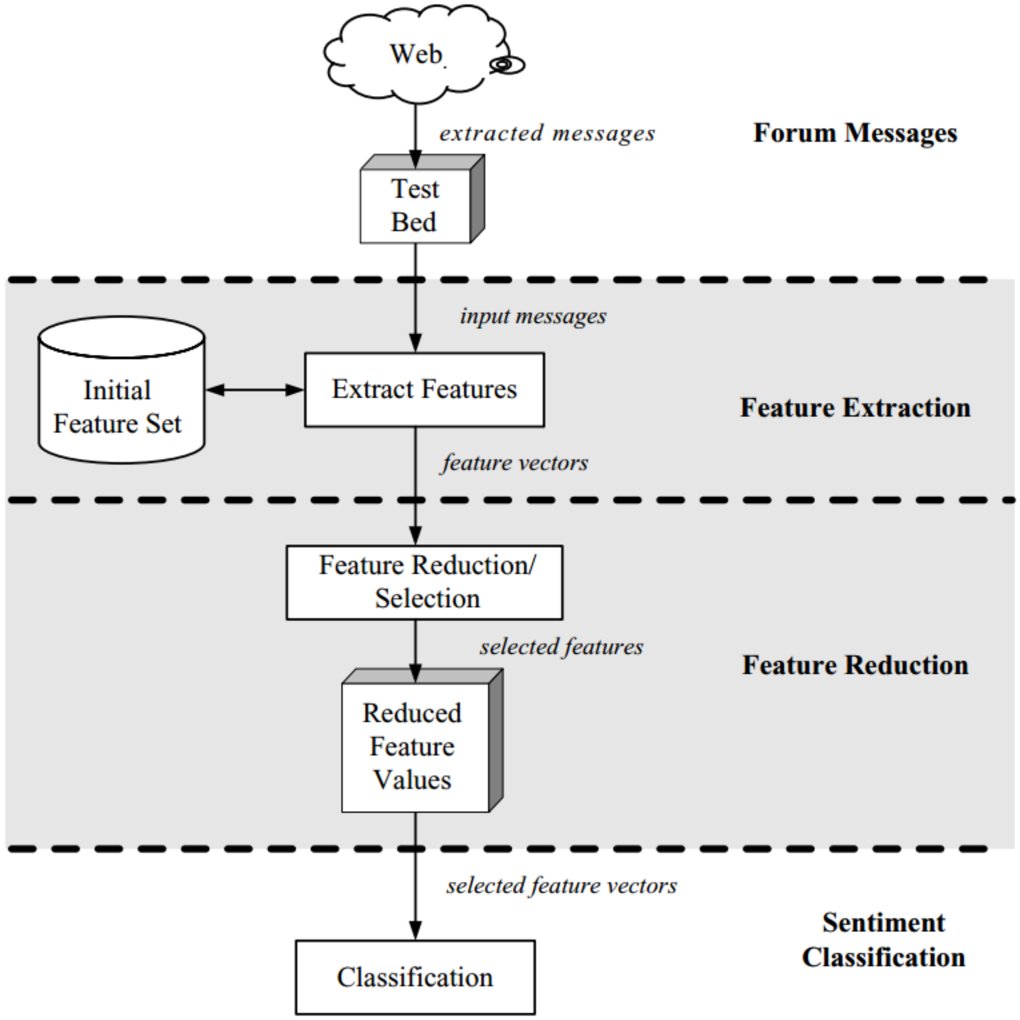

Fig:2.2 Feature reduction process

2.2.2 Need for Optimization

Feature Selection in sentiment analysis is tackling a variety of issues such as large feature space, redundancy[6], noise attributes, context sensitivity, domain dependency[8], and limited work on Lexico-structural features[4], amongst others. Figure 2.2 shows the feature reduction process and its position in sentiment analysis process.

- High Dimensionality: The Social web data usually have high dimensionality. It degrades the performance of classifier and increases the time taken by the classifier.

- Dependency on domain: The classifier trained in one domain often fails to produce satisfactory results when used for classifying the data of another domain. Cross domain independence therefore I need to be tackled.
- POS tagging problem: The accuracy of intelligent feature selection technique depends on the accuracy of POS tagging, so designing an efficient POS tagger is foremost requirement for improving performance. Designing proficient taggers is an issue still to be resolved, especially for languages other than English.
- Lexico-structural features: Most of the work that has been done is for English language. Limited work has been carried out on Lexico-structural features of other languages.
- Redundancy: Many features are redundant and does not contribute any sentiment value so these redundant features have to be found and removed. But finding these redundant features effectively is still an issue to be resolved.

The primary goal of feature selection is to enhance the performance of classifier by selecting only useful and pertinent features and removing redundant, irrelevant and noisy features and thus reducing the feature vector. Further, extracting pertinent and distinct features becomes imperative too when classification algorithms are inept to scale up to the size of feature set in terms of time and space. Absence of proper feature selection technique can cause the classifier to consume more resources and more processing time. The first and foremost challenge in feature extraction is to select the minimal feature subset without any loss of classification accuracy.

In a generic sentiment classification task, a number of words as candidate features are considered, though only a few essentially express sentiments. This set of extra features have to be pruned as they down turn the classification process & tend to reduce the accuracy of the classifier. Thus, feature selection involves searching optimal feature subset using some search strategies. The search could be exhaustive or approximate, exhaustive search produces optimal solution but it is not feasible for large datasets and the social media data usually have huge dimensionality. Exhaustive search in this case becomes impractical as finding optimal feature subset comes in the category of NP-hard problems as for N number of features, the number of possible solutions will be exponential to 2^N. So the focus of researchers has now shifted to meta-heuristic algorithms, which are taken as a subclass of approximate methods.

To produce a more accurate classification and reduce the feature set, study and implementation of several evolutionary optimization techniques have been successfully done in the past and are currently explored too, making it a dynamic area of research. Most common evolutionary optimization techniques used for feature selection are, genetic algorithms, simulated annealing, gene expression programming, swarm algorithms, amongst others.

2.3 Evolutionary Algorithms

Evolutionary algorithm is an umbrella term involving a range of computer based problem solving techniques which uses computational models of generic biological evolutionary practices as core element in their design and implementation. Evolutionary algorithm (EA) is a generic population-based metaheuristic optimization algorithm based on the biological evolution model and natural selection. EA uses operators inspired by natural evolution like selection, reproduction, mutation and recombination. Species mutate over several generations with random mutations of which many are useless but some are advantageous. Candidate solutions to the optimization problem are formed randomly as initial population. The fitness function determines the quality of solutions and good solutions are chosen from the candidate solution set to reproduce. Problem solution using EA is shown in figure 2.3. The candidate solution population evolve iteratively evolved over many generations.

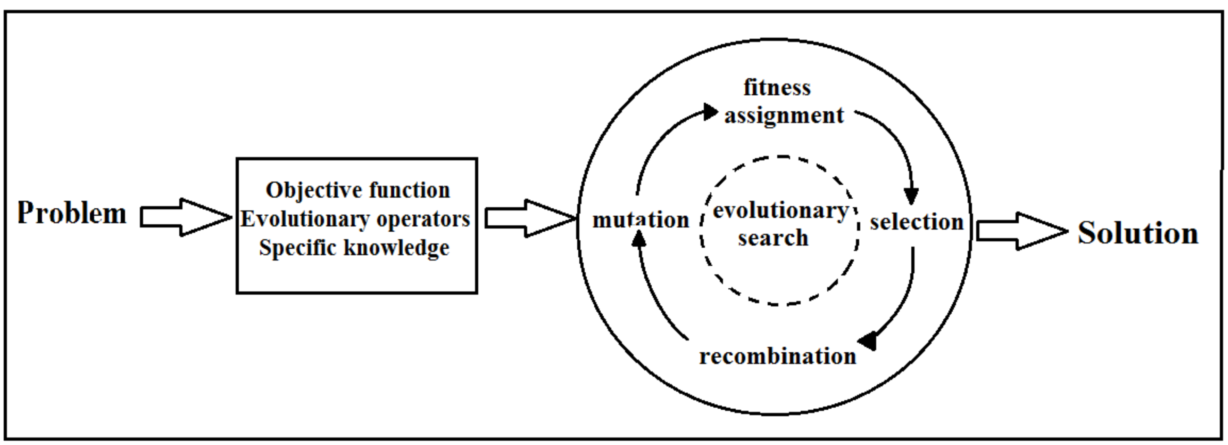

Figure 2.3: Problem solution using Evolutionary Algorithm

The evolution of the population takes place by repeated use of the above mentioned operators. Different evolutionary algorithms have evolved during the last few decades. Swarm Intelligence is basically the collection of interacting agents who follows simple rules to communicate with each other in their local environment without having any central regulator architecture. It is a collective behavior of animals, small insects and other creatures that help each other in either static or a dynamic manner.

Swarm Intelligence based algorithms

Swarm intelligence involves a simple collection of agents interacting locally with each other and with their corresponding environment. The agents interact with each other based on simple behavioral rules that uses local information which is exchanged between agents directly and through the environment. The individual agents are unintelligent, but the overall system leads to intelligent behavior due to the interactions between the agents and their environment. It is the principle of decentralization and self-organization of the group of interacting agents that produces global intelligent behavior. Many algorithms have come into existence which are based on swarm behavior and among them a few popular ones are Ant colony optimization, Ant Bee Colony, Particle Swarm Optimization etc. These algorithms draw inspiration from ants clustering, animal herding, nest building of wasps and termites, bird flocking, bacterial growth, microbial intelligence and fish schooling. Various swarm intelligence algorithms are presented in Table 2.2. Ants make their colonies by collecting small bodies into a single place and they organize their larvae into a single place with younger larvae in the center and the older ones at boundaries. This behavior inspired scientists and researchers to come up with models of this behavior in simulation of problems. Another popular swarm algorithm is the PSO, a population based optimization technique based on social behavior of fish schooling and flock herding. In PSO a collection of agents called particles which search for solution in search space based on its own experience and experience of its neighbors and based on this it decides where to move in the search space it modifies its velocity according to its previous velocity and velocity of neighboring agents.

Reference	Algorithm	Author	Year
[57]	Ant Colony optimization	M Dorigo	1992
[41]	Particle swarm optimization	Dr. Eberhart and Dr. Kennedy	1995
[55]	Bee system	P Lucic and D Teodorovic	2001
[50]	Bacterial foraging	Kevin M Passino	2002
[49]	Fish Algorithm	X.-L. Li, Z.-J. Shao and J.-X. Qian	2002
[48]	Bee Hive	H.F. Wedde, M. Farooq, and Y. Zhang	2004
[58]	Artificial bee colony	Dervis Karaboga	2005
[51]	Bee colony optimization	Dusˇan Teodorovic´ and Mauro Dell'Orco	2005
[56]	Bees swarm optimization	Habiba Drias, Souhila Sadeg, and Safa Yahi	2005
[47]	Glowworm swarm optimization	KN Krishnanand and D Ghose	2005
[43]	Virtual Bees	X.-S. Yang	2005
[54]	Bees algorithms	DT Pham, A Ghanbarzadeh, E Koc, S Otri, S Rahim, and M Zaidi	2006
[45]	Virtual ant algorithm	Xin-She Yang, Janet M Lees, and Chris T Morley	2006
[46]	Good lattice swarm optimization	Shoubao Su, Jiwen Wang, Wangkang Fan, and Xibing Yin	2007
[42]	Monkey Search	Antonio Mucherino and Onur Seref	2007
[52]	Multi-swarm optimization	Ben Niu, Yunlong Zhu, Xiaoxian He, Henry Wu	2007
[53]	Fast bacterial swarming algorithm	Ying Chu, Hua Mi, Huilian Liao, Zhen Ji, and QH Wu	2008
[40]	Firefly Algorithm	X.-S. Yang	2008
	Cuckoo search	Xin-She Yang and Suash Deb	2009
[39]	Bat algorithm	Xin-She Yang	2010
[36]	Consulted Guided Search	Serban Iordache	2010
[59]	Eagle Strategy	X. S. Yang and S. Deb	2010
[44]	Hierarchical swarm model	Hanning Chen, Yunlong Zhu, Kunyuan Hu, and Xiaoxian He	2010
[38]	Weightless Swarm Algorithm	TO Ting, Ka Lok Man, Sheng-Uei Guan, Mohamed Nayel, and Kaiyu Wan	2010
[37]	BumbleBees	Francesc de Paula Comellas Padro´, Jesu´s Mart´ınez Navarro	2011

[35]	Krill Herd Algorithm	Amir Hossein Gandomi, Amir Hossein Alavi	2012
[34]	Wolf search	Rui Tang, S. Fong, Xin-She Yang, and S. Deb	2012

TABLE 2.2 Swarm Intelligence Algorithms

PSO has been successfully use in many optimization problems and have produced very good results. SI-based algorithms are coming up with very good results in problems where the nature of problem is NP-Hard, which otherwise would produce sub-optimum results and consume huge processing power. Due to the scalability and robustness of SI-based algorithms they have become the first choice in finding solutions for optimization problems.

2.3.1 Genetic Algorithm

Genetic algorithm is a search heuristic that derives its inspiration from the process of natural biological evolution, natural selection among species and survival of the fittest. It was first developed by Holland and De Jong. These are probabilistic search methods. GAs are applied in artificial intelligence for finding globally optimum solution from the set of candidate solutions where nature of problem is NP Hard. Genetic algorithm applies natural selection, crossover and mutation operators to evolve from one candidate solution to another an best among the candidate solution is chosen as the global best.

A basic GA consists of five components

1: Population Initialization

2: **repeat**

3: Fitness Evaluation

4: Selection

5: Crossover

6: Mutation

7: **until** requirements are satisfied

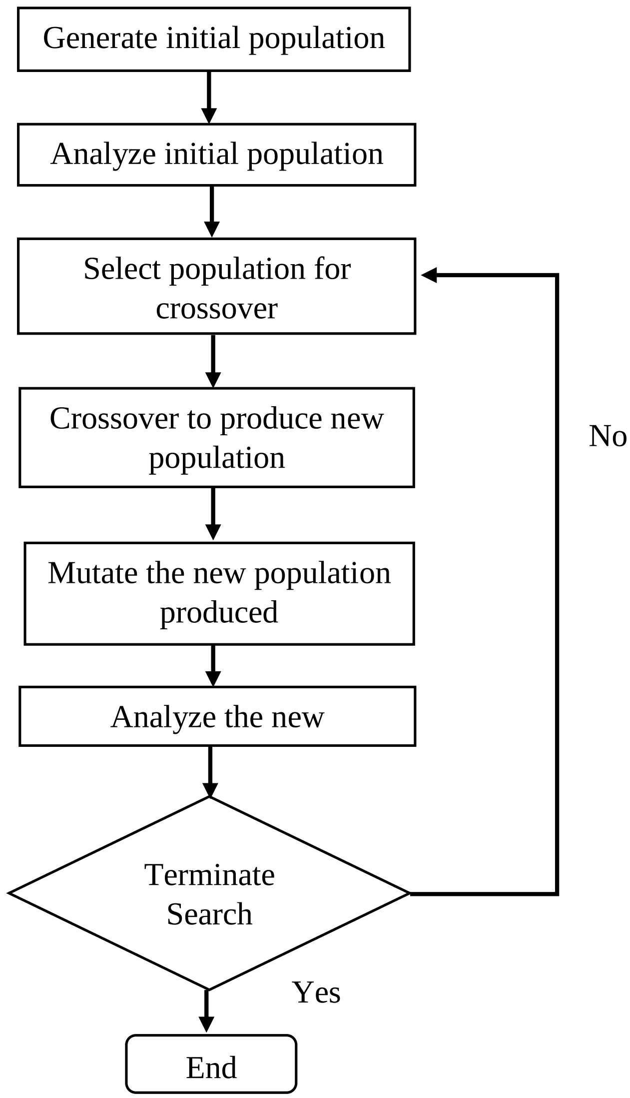

Fig 2.4 Flow chart of Genetic Algorithm

The genetic algorithm is an approximation based algorithm in machine learning which derives its working from principles of natural evolution among species. This is achieved by creating an initial population represented by chromosomes, which is a set of character strings. Each individual in the population represent candidate solution to the problem. Each individual is represented by n-bit binary vector. Therefore, the search space comprises of n-dimensional

Boolean space of candidate solutions. The quality of ach solution is measured by a fitness function. Individual from current generation are selected based on fitness value for reproduction to produce next generation. Genetic operators i.e. crossover and mutation are then applied to the selected individuals for reproduction to from next generation individuals. Mutations and crossover are two genetic operators the modify the individual n-bit binary strings to form new strings. The Mutation and crossover operators are shown in figure 2.4. The crossover operator is applied on two parent strings whereas the mutation operator operates on a single string and modifies a single or a few bits at random in the candidate individual subjected to mutation. Crossover operator is applied on two parent chromosomes and produces two offspring chromosome string based on particular point of crossover.

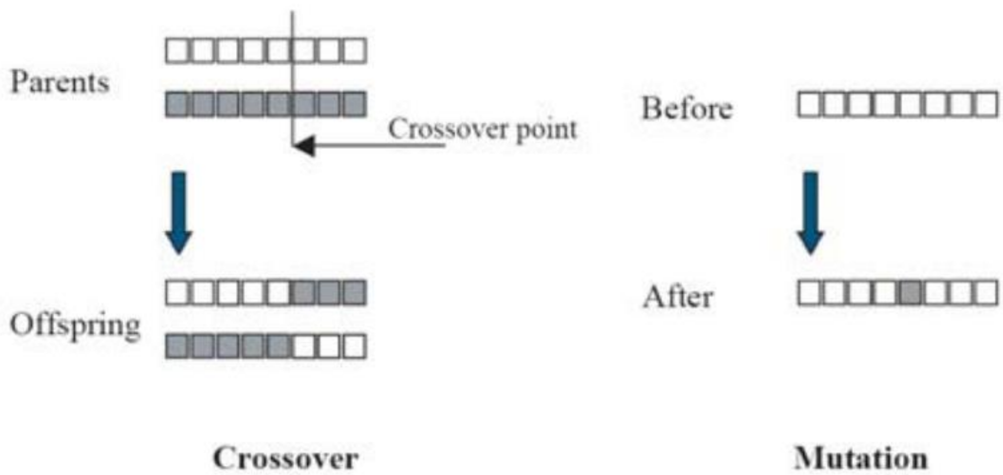

Fig: 2.5 Crossover and Mutation Process

The process of application of genetic operators i.e. selection, crossover, mutation is performed iteratively over many cycles(generations) until an adequate solution is found. After each iteration fitness of new offspring is evaluated of which good solutions are further selection for crossover and less informative solutions are simply discarded based on fitness. The performance of genetic algorithm depends on many aspects such as, the fitness function, the cross point, mutation probability, initial population, and size of population.

2.3.2 Firefly Algorithm

Firefly algorithm (FA) is a biologically inspired algorithm based on flashing behavior of fireflies. FA is a nature inspired global optimization method developed by Xin-She Yang [40] in the year 2008. This is a relatively new algorithm and outperforms other smarm intelligence algorithms. FA have attracted a lot of researchers in the recent years. It is a Population based metaheuristic algorithm based on pattern of fireflies where each firefly represents potential solution in the search space to the problem at hand. FA mimics the behavior of fireflies how they attract partners for mating, how they attract food sources, and they exchange information using their flashing light patterns. The FA, uses three basic rules they are:

1. All fireflies are unisex and can attract other fireflies equally based on brightness.
2. Each firefly is associated with certain brightness and the attractiveness of each firefly is proportional to its brightness. The attractiveness between any two fireflies decreases as the distance between them increases. The less bright firefly will move to the brighter firefly. In case there is no brighter firefly, the fireflies will move randomly.
3. The value of the objective function is used as the brightness of a firefly. For maximization problem, the fitness function value forms the brightness.

There are two important characteristics in the FA that controls the algorithm that are, light intensity variation and attractiveness parameter. Brightness determines the attractiveness of a firefly. More is the distance between two fireflies, weaker is the attractiveness and light intensity between them. The light intensity and attractiveness are formulated as:

$$E(r) = E_0 e^{-\gamma r^2}$$

where r is the distance between any two fireflies, γ is the light absorption coefficient. The light absorption coefficient γ controls the decrease in light intensity and E_0 is initial light intensity. γ is usually set to a constant. The attractiveness parameter is directly proportional to the light intensity parameter which is defined as:

$$\beta(r) = \beta_0 e^{-\gamma r^2}$$

where β_0 represents the attractiveness at distance (r) = 0

The less bright firefly will towards more brighter firefly. The movement of ith firefly towards the jth firefly, based on attraction is formulated by the following equation:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha\left(\text{rand} - \frac{1}{2}\right)$$

where x_i represents the current firefly position and α represent the randomization parameter representing noise in the system. It is usually set to a constant. rand is a random number generated in range [0, 1].

r_{ij} represents the distance between the two fireflies i and j and is defined as:

$$r_{ij} = \sqrt{\sum_{d=1}^{D}(x_{id} - x_{jd})^2}$$

where D represents the size of optimization problem.

Following is the algorithm of firefly optimization. First the initial population is initialized randomly and forms set of candidate solutions and the output is the best firefly position based on the fitness function used.

Initialize a population of n fireflies' positions at random.

Find the best solution based on fitness.

while *stopping criteria not met* **do**

  for every *firefly$_i$* **do**

  for every *firefly$_j$* **do**

   if *firefly j is brighter than firefly i* **then**

    Move firefly *i* towards firefly *j* using equation

   else

  end

 evaluate the positions of individual fireflies

end

Algorithm 1: Pseudocode for firefly algorithm

2.4 Feature Selection Using Evolutionary Algorithms

Present feature selection methodologies, like Information Gain, Document Frequency thresholding and Chi Square assign numerical values to features based on specific statistical equation. Then using some threshold value appropriate features are selected from the sorted feature vector. The selection of threshold value is user dependent and impacts the classification accuracy. This results in selection of sub-optimal feature set and thereby consuming more processing power and more resources.

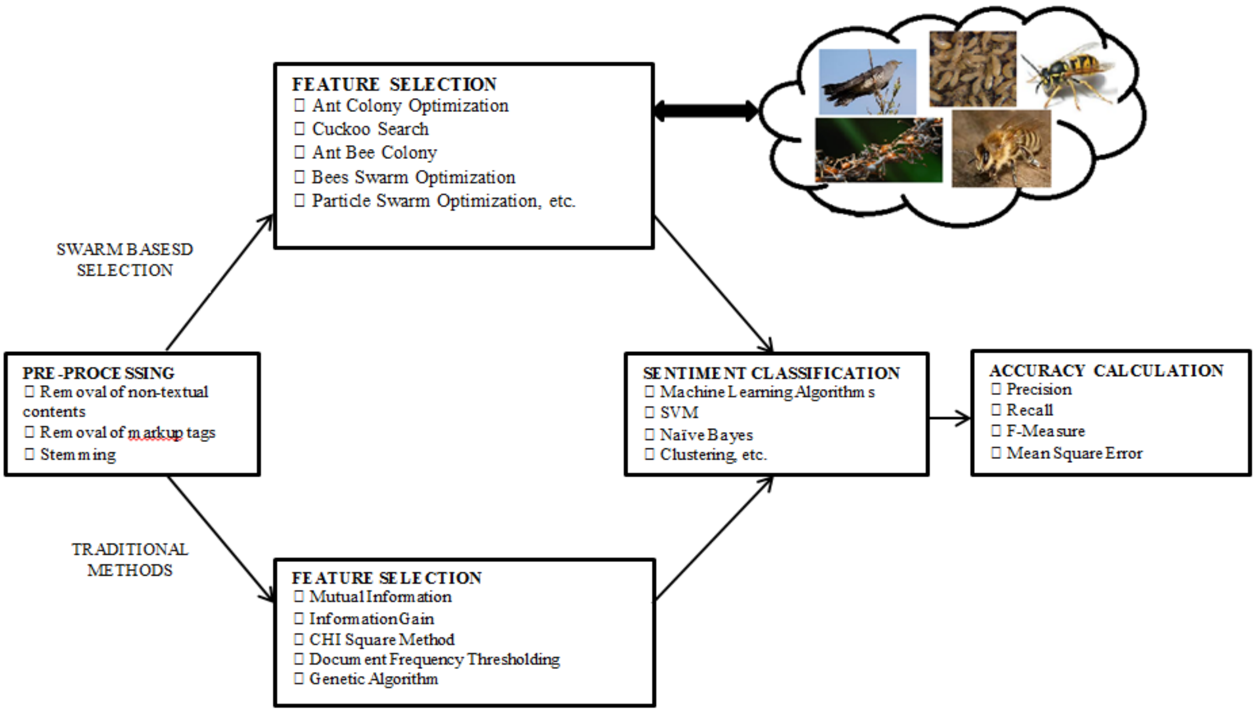

Fig 2.5 Using Swarm Intelligence in Feature Selection

Feature subset selection from high dimensional feature space is a global optimization problem that aims at reducing feature quantity and removing redundant, irrelevant and noisy features thereby increasing the classification accuracy and improving on processing time. Swarm Intelligence algorithms are widely used in such optimization problems where quality of solutions could be measured. These algorithms therefore are used for this optimum feature vector selection problem to render a feature subset that gives improved classification accuracy. Swarm algorithms improves the quality of solution by working in several iterations and application of knowledge of previous iteration on selection of current values.

By reducing number of features, greater classification accuracy can be produced as opposed to using the full set of features. Feature selection basically is a four step process involving feature subset generation, subset evaluation, terminating criterion checking and result validation. Firstly, feature subset is generated in this candidate feature subset is searched based on specific search strategy the candidate subsets are evaluated and compared with a previous best value of the evaluation attribute used. If better subset is produced, it replaces the previous best. This subset generation and evaluation is repeated until a specific stopping criterion is achieved. The swarm optimization goes through several iterations before attaining the global best solution. After each cycle fitness function which is accuracy of classifier is calculated for the candidate subset. The candidate solution generation and fitness function computation continues until the terminating criteria is satisfied. Generally, the stopping criteria is based on two things the error rate and Number of iterations. If the error rate is below a certain threshold, then we stop or if the algorithm exceeds the specified number of cycles.

2.5 Related Work

Recently, many approaches of optimization algorithms based on evolutionary computation, like Artificial bee colony(ABC), Particle swarm optimization(PSO), ant colony optimization(ACO), and genetic algorithm (GA)have been proposed and used successfully to extract features optimally by improving the classification accuracy and reducing the training time. These optimization algorithms can achieve superior results in this domain. Comparisons with baseline systems show that promising accuracies with much reduced feature set can be achieved using swarm intelligence. Experiments revealed it was possible to maintain an 87.15%, state-of- the art classification accuracy when using less than 36% features [12]. The ABC and PSO algorithm being powerful optimization techniques are widely used for solving combinatorial optimization problems. These methods have been used for optimizing the feature subset selection successfully by researchers and have improved the accuracy of classification as stated in table 2.3. The PSO algorithm when amalgamated with the Sentiment classifier enhances the classification accuracy by 4.25% whereas the ABC algorithm produces accuracy increments of 9.94%.

SI technique	Author	Data set	Classifier	Accuracy without Optimization	Accuracy with Optimization	Reference
ABC (2015)	Ruby Dhurve, Megha Seth	Product Reviews	SVM	55	70	[19]
ABC (2014)	T. Sumathi, S.Karthik, M.Marikkannan	Internet Movie Database (IMDb)	Naïve Bayes	85.25	88.5	[23]
			FURIA	76	78.5	
			RIDOR	92.25	93.75	
hybrid PSO/ACO2 (2014)	George Stylios, Christos D. Katsis, Dimitris Christodoulakis	Product Reviews, Governmental decisions data	Decision Tree	83.66	90.59	[21]
PSO (2012)	Abd. Samad Hasan Basaria, BurairahHussina, I. GedePramudyaAnantaa, Junta Zeniarja	Twitter Data	SVM	71.87	77	[27]
PSO (2015)	Deepak Kumar Gupta, Kandula Srikanth Reddy, Shweta , Asif Ekbal	Restaurant Review Data	CRF	77.42	78.48	[32]

Table 2.3. Comparison of Various Swarm Intelligence Techniques on Sentiment Analysis

The complete state-of- art is given in table 2.3. Although we have a variety of nature inspired algorithms for optimization as can be seen from table 1 only few have been explored in domain of opinion mining. The swarm intelligence algorithms produce results that are better in terms of Sensitivity, specificity and accuracy. Using swarm intelligence in opinion mining is an open and emerging area for researchers, significant enhancement of performances of systems are compelling researchers, marketers, policy makers to invest in this area. Improved results can help merchants acquire valuable feedback (e.g. consumers' satisfaction regarding their products) and could facilitate public administrations to capture the understanding of e-Government and e-Rulemaking. This shows that feature optimization is a dynamic area of research and it improve classifier's performance in sentiment analysis and this area has much potential to be discovered further.

2.5 Classification

Classification is the task of assigning data to one of the predefined classes. The class labels may be predefined i.e. supervised learning or the class label may not be present as in case of unsupervised learning. Classification consists of assigning a class label to a set of unclassified cases. There are many classification algorithms used in machine learning to tackle both supervised and unsupervised learning. A few common machine learning classification algorithms are Naïve Bayes, K Nearest neighbor, Support vector machine(SVM), Clustering, Artificial neural networks, etc. Of all the algorithms SVM is the most popular algorithm as it provides good performance and is robust to noise. It is discussed in more detail in the next section.

SVM Classifier

Support Vector Machines (SVM) is a classification model induced from statistical learning. SVM is a supervised machine learning algorithm which is used for classification and regression tasks. SVM has found many applications in the recent years due to its robust nature and high performance. SVMs are used to solve numerous real world problems such as, text categorization, image classification, bio sequences analysis, face recognition, and hand-written character recognition, etc. Support Vector Machines are based on the theory of decision planes that define decision boundaries. A decision hyperplane is basically a boundary that separates a set of objects having different class associations. The SVM separates the classes with a decision surface that maximizes the margin between the classes. The SVM surface against which the decision points push against is called as the optimal hyperplane, and these data points against which the boundary pushes itself are known as support vectors. The support vectors are the critical elements of the training set. The SVM can be adapted to become a nonlinear classifier through the use of nonlinear kernels. While SVM is a binary classifier in its simplest form, it can function as a multiclass classifier by combining several binary SVM classifiers. SVM classifier offers several benefits over the other classifiers. These benefits are:

1. Handling High dimensional Input Space – Since in classification problem involving text we generally have to deal with a large number of features classification becomes difficult. But SVM model uses over fitting protection, which enables handling of large number of features as it does not depend on number of features.

2. Document Vector Space - In spite of the high dimensional representation of the document, each document vectors comprises of only a limited number of non-zero elements.

The characteristics of SVM classifier are:

1. Machine Learning algorithms usually uses a vector-space (attribute-value) type representation of data tuples, where the attribute corresponds to the words. However, word-pairs or the position of a word in the text may have considerable information, and practically infinitely many features can be constructed which can improve accuracy.

2. There may be different representations of words in a piece of text, but the words may have different meaning i.e. having a different meaning based on context. Each word is influenced by the context in which it is referenced. However, the usual computational practice neglects the order in which the words appear. It is not location based. The SVM has to learn to generalize the input-output mapping.

3. The classification categories are normally binary, but generally the documents are not classified so precisely. Sometimes a document partly shows affluence to one class and partly to another but fails to fit well in any of the two classes. Therefore, a third category is often required to keep the documents that does not fit well in any classes to that class. We have to form a multi-class classifier in such cases.

CHAPTER 3

PROPOSED APPROACH

The proposed system selects optimum feature subset from high dimensional feature set for sentiment analysis using firefly algorithm and also the results are compared with genetic algorithm. We used Firefly algorithm and genetic algorithm that incorporates various machine learning algorithms to improve the performance of feature selection. The proposed approach can benefit the sentiment analysis task and improve various decision support systems as it improves the classification accuracy significantly. The proposed system framework is presented in figure 3.1 and work carried out is explained in detail here.

3.1 Feature Selection Using Firefly optimization algorithm

In this section, a discrete FA is proposed to solve the feature selection problem.

Step 1. The first step is to initialize the firefly parameters (size for the firefly population(N), α- the randomness parameter, γ-absorption coefficient, and the maximum number of generations for the termination process(t_{max})).

Step 2. Next initial firefly position is initialized. Initially a random position is allocated to fireflies. $X_i = [x_{i1}; ...; x_{in}]$ where n is the total number of features and m solutions are considered as candidate solutions initially. Position of m fireflies is considered in the multi-dimensional search space, where m denotes the size of firefly population. Each candidate solution X_i is represented as a binary string vector.

Step 3. In the next step the fitness of the population is calculated. For this case the function of fitness is equal to the accuracy of the classifier model i.e. accuracy of the SVM classifier with current solution as feature set.

Step 4. Firefly position modification. The firefly with less brightness will move towards a firefly with more brightness. The new position is based on modification in each dimension of firefly.

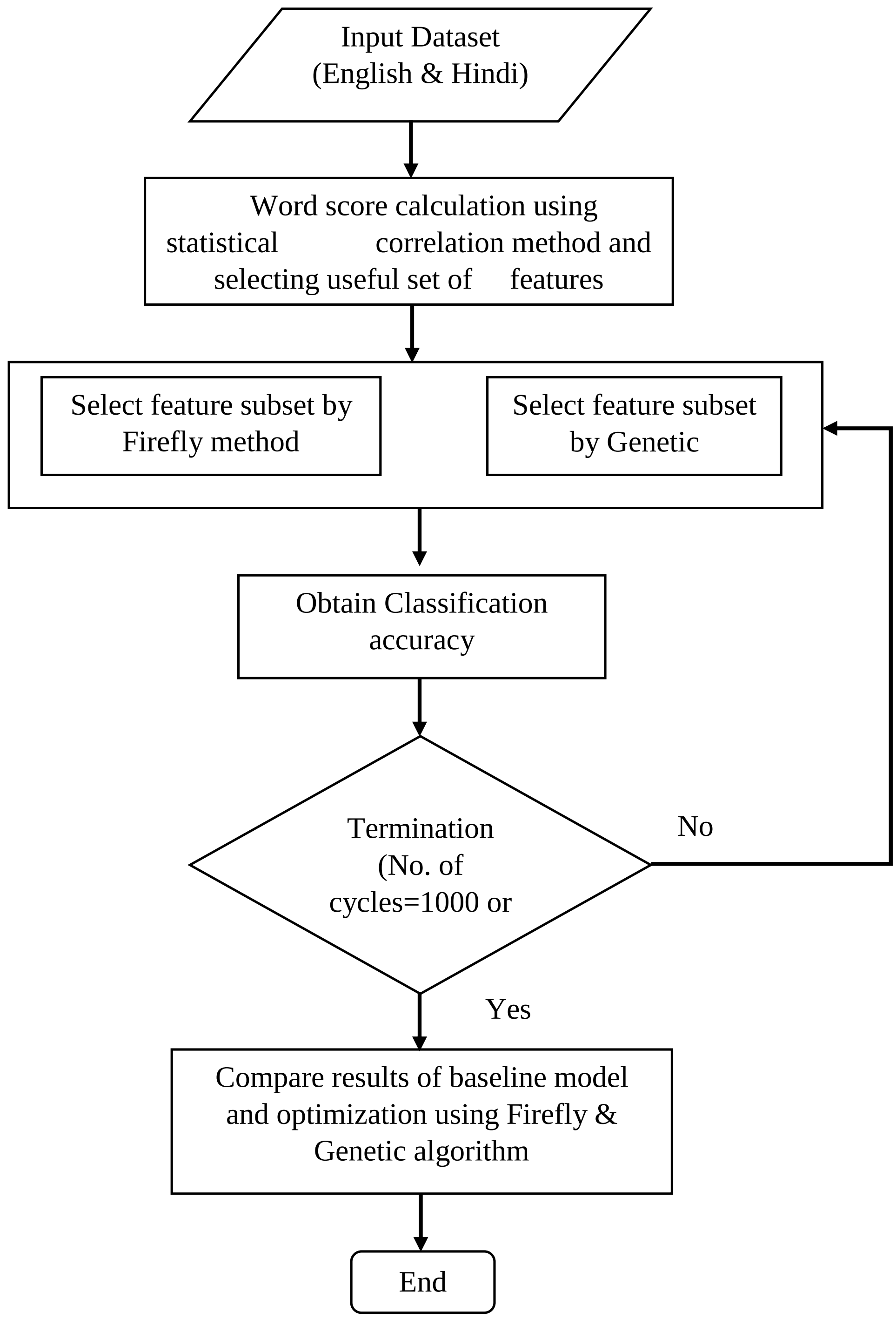

Figure 3.1: Proposed System Framework

Step 5. The new solution produced by modifying firefly position is examined using the fitness value of the SVM classifier. Some solutions with very low accuracy are simply discarded.

Step 6. Store the best solution attained till now and increment the generations counter.

Step 7. If termination criteria are met, then stop the search and display the result else go to step 3. The termination criteria used in this method is maximum number of generations or if classification accuracy is 100%.

The fundamental concept of FA is that each firefly learns from its m.ore attractive neighbors to evalu.ate itself. The search pattern of the standard FA is greatly dependent on its control parameters. If any firefly is not able to find any best matting partner, then the intensity of that particular firefly i is absorbed in the system and it becomes invisible to all other fireflies in the system. The movement of fireflies results in the subset of fireflies which becomes more dependent on each other. Each subset after an iteration then evaluated and checked whether it satisfies the stopping criterion i.e. maximum number of iterations and required fitness value. There are two critical aspects of feature selection problem; and these are the degree of optimality and time required to achieve this optimality. Existing methods achieved success in either of these aspects but not in both. So firerfly algorithm is used here to tackle both the problems simultaneously. The algorithm used is the binary version of the discrete firefly as it deals with candidate solutions in terms of n-bit binary string values.

3.1.1 Initial Population and encoding of fireflies

Initial population is generated randomly as array of binary bits, with length equal to the total number of features. Binary solution vector is used. The potential solutions are encoded as binary strings of fixed length. The potential solution has the form $\vec{X}(i) = (x_{i1}, x_{i2}, ..., x_{in})$ where $x_{ij} \in \{0, 1\}$, i = 1, 2, ..., N, where N specifies the number of feature set i.e. the population size and j = 1, 2, ...n where n is the number of features. The length of the firefly depends upon the total number of features. Suppose we have a set of features F = (f1, f2, ..., fn), then a binary vector is defined to denote a firefly of length n. If any bit of $\vec{X}(i) =$ contains a value of "1" then the respective feature is used for classifier's training, and a value of 0 designates that the feature is not used for classification. We can predefine the number of features we want to use for classification. For example, for a given set of features, F = (f1, f2, f3, f4, f5, f6) and taking N as 4, the firefly population can be represented as follows:

$$\vec{\boldsymbol{X}}(\mathbf{1}) = (\mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1})$$

$$\vec{\boldsymbol{X}}(\mathbf{2}) = (\mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1})$$

$$\vec{\boldsymbol{X}}(\mathbf{3}) = (\mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0})$$

$$\vec{\boldsymbol{X}}(\mathbf{4}) = (\mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \mathbf{1})$$

For the initial population we randomly generate such N solutions. The bit positions are randomly initialized to either 0 or 1. We generate a uniform random number c on the interval (0,1) for every position X_{id} of $X(i)$. Based on the random value every firefly $X(i)$ is generated as follows.

$$\boldsymbol{X_{id}} = \begin{cases} \mathbf{1} & \boldsymbol{if\ c}\ \textbf{is} < \mathbf{0.5} \\ \mathbf{0} & \textbf{otherwise} \end{cases} \tag{1}$$

3.1.1 Updating the Global and Best Position Value

The best particle from the initial population is first selected based on the fitness value of classifier and the firefly having low value moves towards the firefly have more value of the brightness i.e. fitness function. The firefly algorithm was initially designed for continuous optimization. Therefore, the terms in equation (3) are required to be converted into discrete form. So they are converted into discrete form using a sigmoid function given in equation (2).

$$p_{ij} = \frac{1}{1 + \mathrm{e}^{\vartheta_{ij}}} \tag{2}$$

P_{ij} is the probaility that j^{th} bit is set in x_i. The movement of a firefly i towards firefly j based on attraction is computed by equation (3). The Second component in this equation is due to attraction and the third component is used for randomization. And rand is a random number in the range [0, 1].

$$\vartheta_{ij} = \beta_0 e^{-\gamma r_{ij}^2}(x_{kj} - x_{ij}) + \alpha\left(\text{rand} - \frac{1}{2}\right) \quad (3)$$

The i^{th} firefly uses the following update rules given by equation (4), where rand is a uniform random number in [0, 1]. Other value could also be used instead of the rand value used in equation (4).

$$x_{ij}^{t+1} = \begin{cases} 1 & \text{if } p_{ij} \geqslant \text{rand} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

3.1.3 Parameters controlling exploration and exploitation

The two parameter randomness and attractiveness controls the quality of solutions produced and also the convergence rate. Selecting an optimum value of these parameters is an open area for research. Also apart from these two parameters, the convergence of the firefly algorithm is indirectly proportional to number of fireflies The α parameter physically represents the noise existing in the environment and affect the light transmission while in the artificial algorithm it can be selected to allow for solution variation and hence provide for more diversity of solutions $\alpha \in [0, 1]$.

The parameter γ characterizes the variation of the attractiveness, and its value is crucially important in determining the speed of the convergence and how the FA algorithm conducts. The parameter γ also known as the absorption coefficient is important in balancing exploration and exploitation. Generally, γ can be selected in $[0, \infty]$, and it is usually set to a constant. If γ holds $\gamma \rightarrow 0$, the attractiveness and the brightness will be constant. On the contrary, if γ tends to ∞, the attractiveness and the brightness will decrease dramatically, which can result in the fireflies getting lost in the search process. In this case, all the fireflies fly almost randomly. In general, a constant γ may constrain the performance of FA. Thus, fine-tuning the parameter γ can enhance the performance of FA.

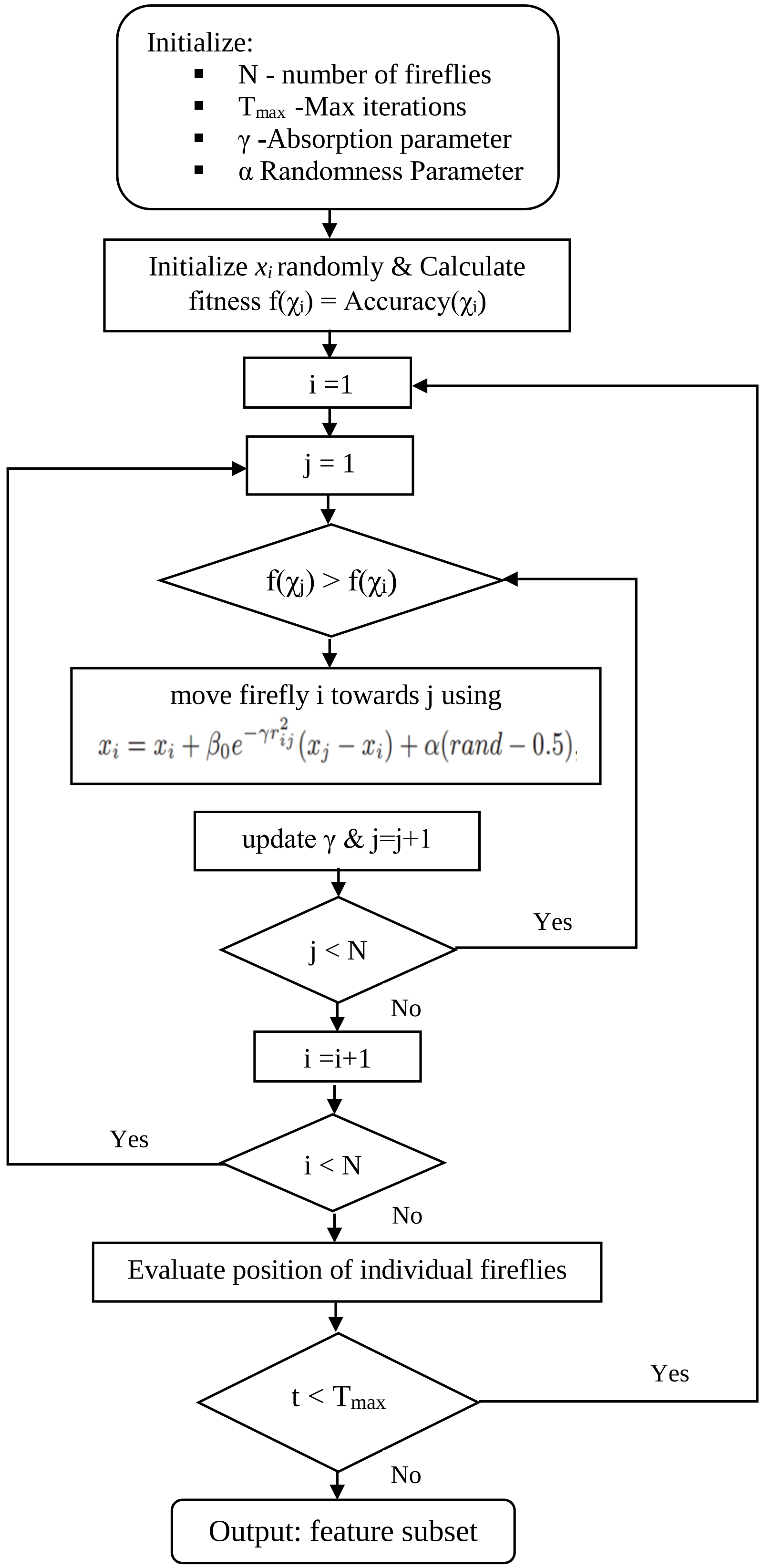

Fig 3.2: Flow chart of firefly algorithm for Feature selection

Algorithm 1 shows the pseudo code of the steps proposed. The fitness function f is used to measure the fitness based on accuracy of particular solution χ_i. The algorithm terminates after fixed number of cycles or if the classification error becomes very low.

Feature Selection Using Firefly Algorithm

Input : N number of fireflies

T_{max} Maximum number of iterations

γ Absorption parameter

α Randomness Parameter(environment noise)

Output : Optimal firefly position and its fitness

1. Initialize parameters N, T_{max}, γ and α
2. Initialize $\chi_i = \phi$, subset of feature selected by i^{th} firefly
3. Initialize x_i, position of each firefly subjecting to $\Sigma\ x_{ij} = s$
4. Calculate fitness $f(\chi_i)$
5. Sort all fireflies according to $f(\chi_i)$
6. While $t < T_{max}$

```
    for i = 1 to N (for each firefly)
      for j = 1 to N (for each firefly)
        if f(χj) > f(χi)
          move firefly i towards firefly j using equation
        end if
       update γ and corresponding attractiveness
     end for
   end for
 determine the features selected & evaluate the position of individual fireflies
 t=t+1
end while
```

7. Output the feature subset

Algorithm 1: Pseudo code of the Firefly algorithm

3.2 Feature Selection Using Genetic algorithm

GA starts with an array of population of candidate solutions. Every candidate solution is encoded with genes that is called individual chromosome. New and improved population is generated with different combinations of previous population. By using an objective function, fitness of each individual is computed. New population is generated from the previous generation selecting good individuals from the previous generation. This process of mixing the parents from previous generations is called as crossover. Another population improvement method is present which is called mutation. The new population individuals are mutated according to the mutation rate of population. Mutation probability is set. The process of feature selection in sentiment analysis using genetic algorithm is given as:

Step 1. Genetic parameter initialization- Firstly parameters population size, maximum generation, and mutation probability are defined.

Step 2. Genetic population initialization. Uniform random population is generated with length equal to the total feature size. The probability of each bit being set is defined and its value used is 0.7. Each bit in the binary individual string represents the features.

Step 3. Fitness Evaluation. The fitness of the initial population is evaluated. For this case the function of fitness is equal to the accuracy value of the SVM classifier model when trained with the selected feature individual. The individual from population having best value is memorized as *bestp.*

Step 4. Select. The feature subsets having good fitness value are selected for reproduction and production of new offsprings.

Step 5. Crossover. Crossover is done between the selected individuals to produce offspring feature subset with mixed features of both the parents. Single point crossover is used in this work.

Step 6. Mutate. The new population is mutated to produce good and advanced feature subset. The current feature subset is mutated with a probability p_{mut}.

Step 7. Repeat. If the maximum number of generations exhausts or the classification error becomes negligible terminate the search otherwise go to step 3 and repeat.

Step 8. Present the value of *bestp* i.e. the best feature subset.

3.2.1. Initial Population

Our model start with creating random initial population. Each chromosome contains number of feature genes and each gene is a binary number. The entries of each chromosome are randomly initialized to either 0 or 1. Figure 3.5 shows the gene representation of the feature vector. The length of the chromosome is equal to the total of features present.

10101000010000101…0111001011001

Fig 3.3: Gene Representation of feature set

In this gene representation if the value of bit is 1 then it means that the corresponding feature is included in the specified subset and used for training purpose, and a value of 0 represents that the corresponding feature is not included in the subset and not used for classifier training. So, if the i^{th} position of a chromosome is 0 then it represents that i^{th} feature does not participate in constructing the classifier. Else if it is 1 then the i^{th} feature participates in constructing the classifier. Initial population of P_{size} is generated with these chromosomes.

3.2.2. Selection

Off- springs are kept in the candidate solution set, only if they are fit better than the least good individual of the population. Selection expresses the principle of 'survival of the fittest'. The mutant fittest chromosomes are selected for reproduction. Sometimes chromosomes having poor or low fitness value are also selected. Each solution has a selection probability of its fitness score divided by the sum of the total solutions scores in the generation. The top n solutions at each generation automatically retained and carried over to the next generation.

3.2.3. Crossover and Mutation

The genetic operators allow Genetic Algorithms to explore and exploit the search space. However, operators typically have constructive well as destructive effects. Features shared by the 2 parents are kept by the 2 offsprings produced by them and the non-shared features are inherited by offsprings. The mutation is an operator which allows diversity in the new population. During the mutation step, an individual chromosome has a probability p_{mut} to mutate. Single-point crossover operator is used in this work. The crossover point i is chosen randomly. The new solutions (offspring) will be created using first i bits of one parent and the remaining bits of the other parent. GAs construct a better solution by mixing the good characteristic of chromosomes together.

3.2.4. Fitness Evaluation

The classifier is constructed with every individual in population size and trained with the features contained in that particular individual. Then the classifier trained is used on testing data to obtain the classification accuracy for every feature subset in the population. So the fitness of the chromosome is accuracy of the classifier.

$$fitness\ (x) = accuracy(x)$$

Fitness of every individual in population is calculated and every time the classifier is trained on training data by using the selected features in the individual chromosome and tested on testing data to evaluate the fitness of each individual. The best value of fitness value is memorized.

Termination Condition - In this approach, the procedures of fitness computation, selection, crossover, and mutation are executed for a maximum number of generations. So, the termination criteria are exhaustion of generations or if the classification error becomes negligible. The best string seen up to the last generation provides the solution to the above feature selection problem.

Algorithm 2 shows the pseudo code of the steps proposed in feature selection using genetic algorithm. The population size and maximum number of generations are input to the algorithm and the output produced is the best feature vector in the population. Other parameters like the crossover point and mutation probability are also defined. Figure 3.5 shows the flowchart of the proposed method of feature selection using genetic algorithm.

Feature Selection Using Genetic Algorithm

Input : P Initial population

P_{size} Population size

g_{max} Maximum number of generations

bestp = P_0

Output : Fittest chromosome *bestp*

1. Initialize $t = 0$
2. Initialize initial population $P(t)$ randomly
3. Compute fitness $f(t) = accuracy(P(t))$
4. While $t < g_{max}$ do

    $t = t + 1$

    for *individual (i) in P* (for each individual)

    $P_i(t) = crossover(P_i(t - 1))$

    $P_i(t) = mutate(P_i(t))$

    $F_i(t) = accuracy(P_i(t))$

    end for

    update *bestp*

    end while
5. Output the feature subset *bestp*

Algorithm 2: Pseudo code of the Genetic algorithm

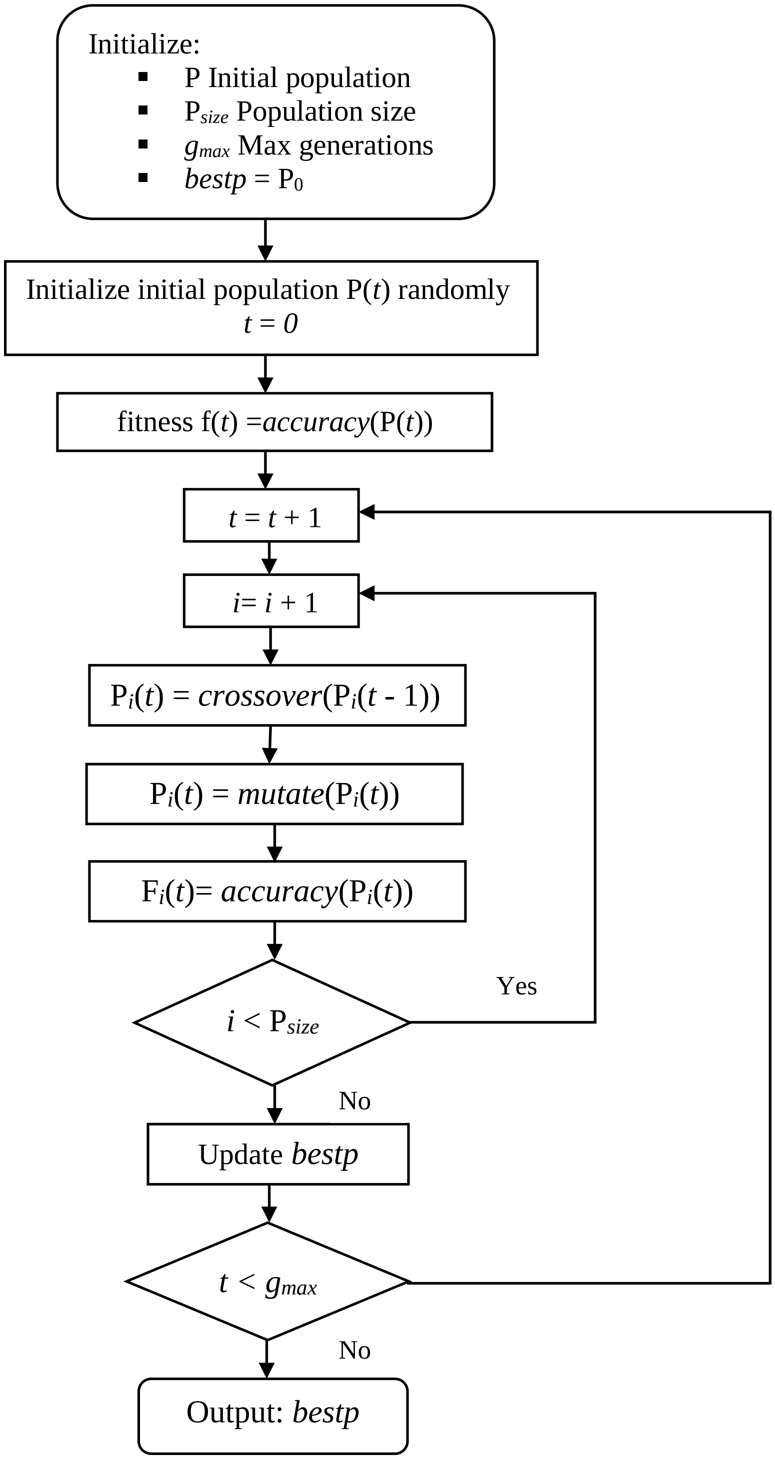

Fig 3.4: Flow chart of genetic algorithm for Feature selection

CHAPTER 4

IMPLEMENTATION

4.1. Environmental Setup

The following configuration has been used while conducting the experiments:

Hardware Configuration

Processor	: Intel core i7
Processor Speed	: 2.50 GHz
Hard Disk Capacity	: 1 TB
Main Storage	: 8.00 GB

Software Configuration

Operating System	: Windows 10
Language used	: Python
IDE used	: PyCharm 2016.1.2
Library used	: NLTK, Libsvm

4.2. Corpora Description

To conduct this work movie review and Twitter data extracted from twitter API are considered here. Dataset in two languages hindi and English is considered. The movie review dataset of hindi language has been taken from Resource Centre for Indian Language Technology Solutions(CFILT), IIT Bombay Sentiment Analysis Group Resources (http://www.cfilt.iitb.ac.in/Sentiment_Analysis_Resources.html).

Dataset	Language	Number of Samples	Positive reviews	Negative reviews
Movie review	English	1000	500	500
Movie review	Hindi	302	127	125
Twitter data Keyword-“Delhi Government”	English	800	400	400
Twitter data Keyword- “दिल्ली सरकार”	Hindi	500	250	250

Table 4.1 Dataset Used

4.2.1 Accessing Twitter data

To access Twitter data we need to get permission access tokens for using twitter API. To do so, the following steps are required.

- Make a twitter account
- Go to https://dev.twitter.com/ and sign into your account.
- Create a new app
- Fill up the data required in the form and agree to the terms and conditions
- This will generate a consumer secret and consumer key, next access tokens are generated from access token tabs.
- Now we have all 4 keys required to fetch twitter data.

Figure 4.1 shows the screenshot of twitter app from where the four access keys are generated. The set of access keys used in this work are:

Consumer key : hSMB5j7ZsbcZP6ehmgRylHWx1

Consumer Secret : n3bJxMnQzpuQkFY42XVT3Nz8yjoM31yt1zCdZsGoGzBxleYG6

Access Token : 2201119075-aRWOZ2LkW0dCbgs1O8ueE3n6tztzkQlWZ9vkSXY

Access Token Secret : sHY2mSDmwpJCRgDPcAH1XNduqF5kZoyD5TYrXzkJbEZSq

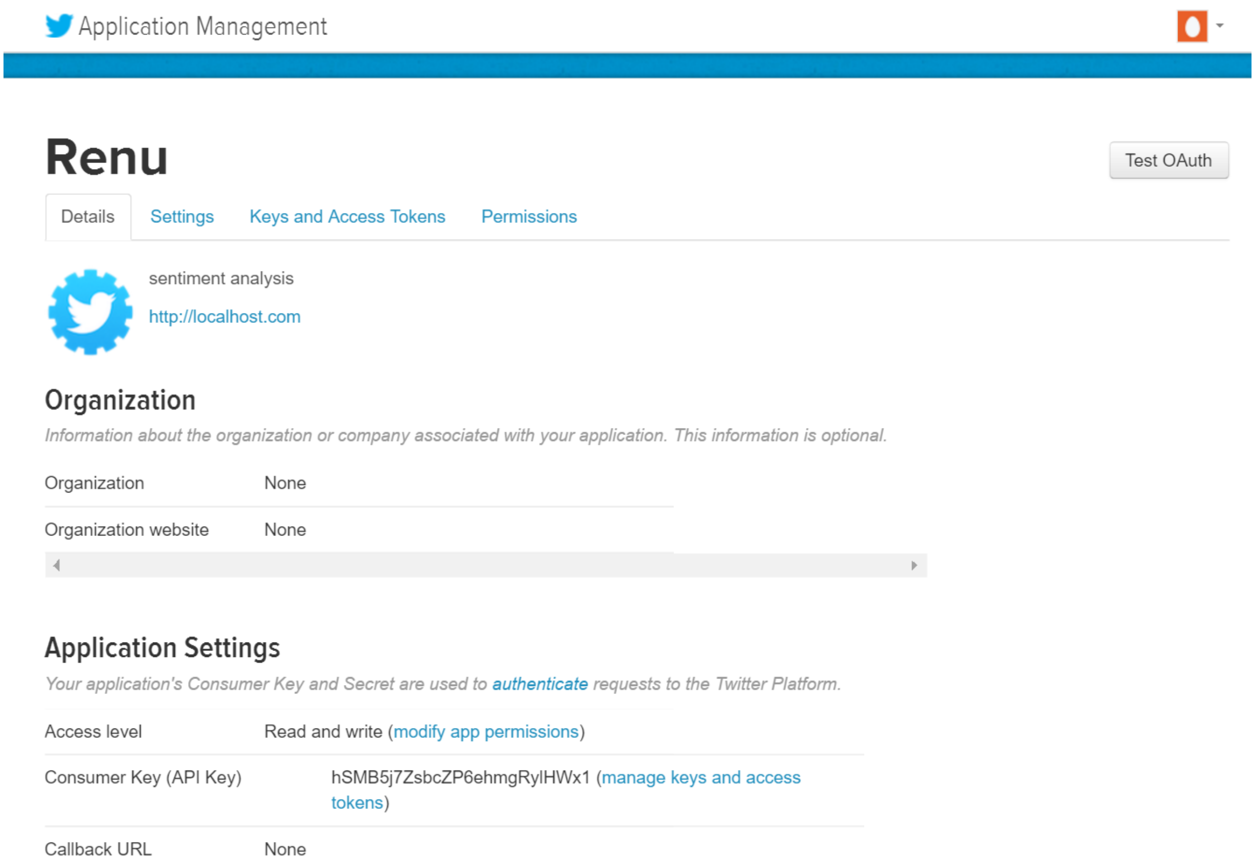

Fig 4.1 Twitter App for extracting Access keys

4.3. Preprocessing Data

Polarity	Raw Sample Tweets
negative	@Shehzad_Ind: RTI reply to @mvadera reveals Modi govt broke laws during 2 year celebration @tehseenp to move court @SiddiquiMaha
negative	@madhukishwar Who advises Modi on such bad choices for Institutions...Shauna NC is the worst choice.
negative	AAM AADMI PARTY : MODI GOVT WAS ELECTED DUE TO RISING PRICES. BUT NOW SILENT
positive	@RajkumarHirani: Saw Udta Punjab. Brave, honest anti drug film. Salutes #AbhishekChaubey @shahidkapoor @aliaa08 @diljitdosanjh Kareena
positive	RT @JantaKaReporter: Raghu ram rajan brought economic stability and growth to the country. He is true Hero
negative	@mkatju: After 2 yrs of its rule, the time has come to now boldly and clearly speak out the ugly truth about Mr. Modi and his govt
positive	Right Raghuram Rajan Is Great Personality He is Real Hero Of India
positive	@RohitRaikwar: What a movie Udta Punjab @Shahidkapoor you won everyone's heart UNSTOPPABLE UDTA PUNJAB @Shahid_online @Shahid
negative	#Udta Punjab not met the expectations,@shahidkapoor @diljitdosanjh @aliaa08. This one is a bad movie with lot of hype

The data after removing the user name removing duplicate words and converting to lowercase becomes:

AT_USER rti reply to AT_USER reveals modi govt broke laws during 2 year celebration AT_USER to move court AT_USER
AT_USER who advises modi on such bad choices for institutions...shauna nc the worst choice
aam aadmi party : modi govt was elected due to rising prices. but now silent
AT_USER saw udta punjab. brave, honest anti drug film. salutes AT_USER AT_USER AT_USER
AT_USER Raghu ram rajan brought economic stability and growth to the country. He is true Hero
AT_USER after 2 yrs of its rule, the time has come to now boldly and clearly speak out the ugly truth about mr. modi and his govt
right raghuram rajan is great personality he is real hero of india
AT_USER what a movie udta punjab AT_USER you won everyone's heart unstoppable udta punjab AT_USER AT_USER
udta punjab not met the expectations, AT_USER AT_USER AT_USER this one is a bad movie with lot of hype

After preprocessing the data becomes as shown below:

'rti', 'reply', 'reveal', 'modi', 'govt', 'broke', 'laws', 'celebration', 'move', 'court
'advise', 'modi', 'bad', 'choice', ‘institution’, 'worst', choice'
'aam',‘aadmi', 'party', 'modi', 'govt', 'elect', 'now', 'rise', 'price', 'silent'
udta', 'punjab', 'brave', 'honest', 'anti', 'drug', 'film', 'salute'
'raghu', 'ram', 'rajan', 'economic', 'stability', 'growth', 'country', 'true', 'hero'
'yrs', 'rule', 'time', 'now', 'bold', 'clear', 'speak', 'ugly', 'true', 'modi', 'govt'
‘raghuram', 'rajan', 'great', 'personality', 'real', 'hero
movie', 'udta', 'punjab', 'won', 'heart', 'unstoppable', 'udta', 'punjab'
'udta', 'punjab', 'not', 'met', ‘expectation’, 'bad', 'movie', 'lot', 'hype'

feature Set ={‘aam’, ‘aadmi’, ‘advise’, ‘anti’, ‘bad’, ‘bold’, ‘brave’, ‘broke’, ‘celebration’, ‘choice’, ‘clear’, ‘country’, ‘court’, ‘drug’, ‘economic’, ‘elect’, ‘expectation’, ‘film’, ‘govt’, ‘great’, ‘growth’, ‘heart’, ‘hero’, ‘honest’, ‘hype’, ‘institution’, ‘lot’, ‘modi’, ‘movie’, ‘not’, ‘now’, ‘party’, ‘personality’, ‘price’, ‘punjab’, ‘raghu’, ‘ram’, ‘rajan’, ‘real’, ‘reply’, ‘rise’, ‘rti’, ‘reveal’, ‘rule’, ‘salute’, ‘silent’, ‘speak’, ‘stability’, ‘time’, ‘true’, ‘udta’, ‘ugly’, ‘unstoppable’, ‘won’, ‘worst’ }

Feature Set length = 55

4.3. Parameter Initialization

4.3.1 Genetic algorithm parameters

Initial Population- Initial Population of N strings is generated where each string is of size P_{size}. Each attribute is switched on with the probability Pi. The population size in this study is set to be 50 and the value of Pi is set to 5/7.

Selection- Accuracy of SVM classifier is used as criteria for selection as fitness function. Solution for the next iteration is selected probabilistically.

Crossover- Crossover is the process of exchange of information between two parents to produce a new offspring. Random single crossover point is used for crossover. Mutation is randomly mutated individual feature characters in a solution string based on a fixed probability P_m. The mutation probability is set to 0.2.

The GA parameters are accustomed are as follows:

1) Population size: 50

2) Number of generation: 50

3) Probability of crossover: 0.7

4) Probability of mutation: 0.2

5) Crossover strategy: Random single point

6) The bits of selected chromosomes that will be mutated: 0.1

4.3.2 Firefly algorithm parameters

Initial Population- Initial Population is initialized similarly as in case of genetic algorithm with same probability of each bit being set. The number of fireflies is initialized to 50. For e.g. in the sample dataset shown above the size of each firefly will be 55 and the firefly population initialized randomly would be

Firefly initial position vector

={[1011101110011011101110111010111101011110111111000110101],
[0111010111100111101101101110111011101011111011011010110],
….

….

[1110111010111101010111110110101110110110111010110110001],
[1110101011011011011110101110111101110110011101101110110]}

The individual firefly positions are used for training the SVM classifier and then used on testing data to compute accuracy of classifier for particular firefly position. This is repeated until the maximum number of generations are over.

The Firefly parameters adapted are as follows:

1) Firefly population size: 50

2) Firefly length= Total number of features

3) Number of generation: 50

3) Absorption coefficient α: 0.5

4) Attractiveness parameter γ: 0.9

CHAPTER 5

RESULTS AND ANALYSIS

The result of classification of SVM classifier, SVM-Genetic classifier and SVM-firefly classifier for the four datasets used in this work is given below.

Result for Movie Review(English) Dataset

Model	**Accuracy**
SVM	79.55
Genetic-SVM	82.15
Firefly- SVM	85.29

Result for Movie Review(Hindi) Dataset

Model	**Accuracy**
SVM	74.12
Genetic-SVM	77.31
Firefly- SVM	79.6

Result for twitter Dataset with keyword- 'Delhi government'

Model	**Accuracy**
SVM	80.3
Genetic-SVM	83
Firefly- SVM	86.71

Result for twitter data set keyword-‘ दिल्ली सरकार’

Model	Accuracy
SVM	73.5
Genetic-SVM	77
Firefly- SVM	78.46

Analysis

Comparisons with baseline SVM system and optimized SVM systems shows that the optimized system produce more accuracy as compared to the baseline model. This shows that promising accuracies with much reduced feature set can be achieved using evolutionary optimization. The firefly algorithm is a powerful optimization algorithm and produces great improvement in accuracy in our model. The genetic algorithm which is also an evolutionary algorithm model is better than the baseline model but less efficient than the firefly model of feature extraction. The system is validated with four different datasets of which two are in hindi language and other two in English language. The twitter dataset is manually labelled as positive or negative. For the English movie review dataset

Method	Average % change
Genetic Algorithm(English)	2.65
Genetic Algorithm(Hindi)	3.34
Firefly Algorithm(English)	6.075
Firefly Algorithm(Hindi)	5.22

Table 5.1: Average percentage improvement in accuracy

The genetic algorithm when amalgamated with the Sentiment classifier enhances the classification accuracy by 3% on average whereas the firefly algorithm produces accuracy increments of 5.64%. The complete accuracy improvements are given in table 5.1. This shows that firefly algorithm is much superior and efficient than the baseline model and also from the genetic model.

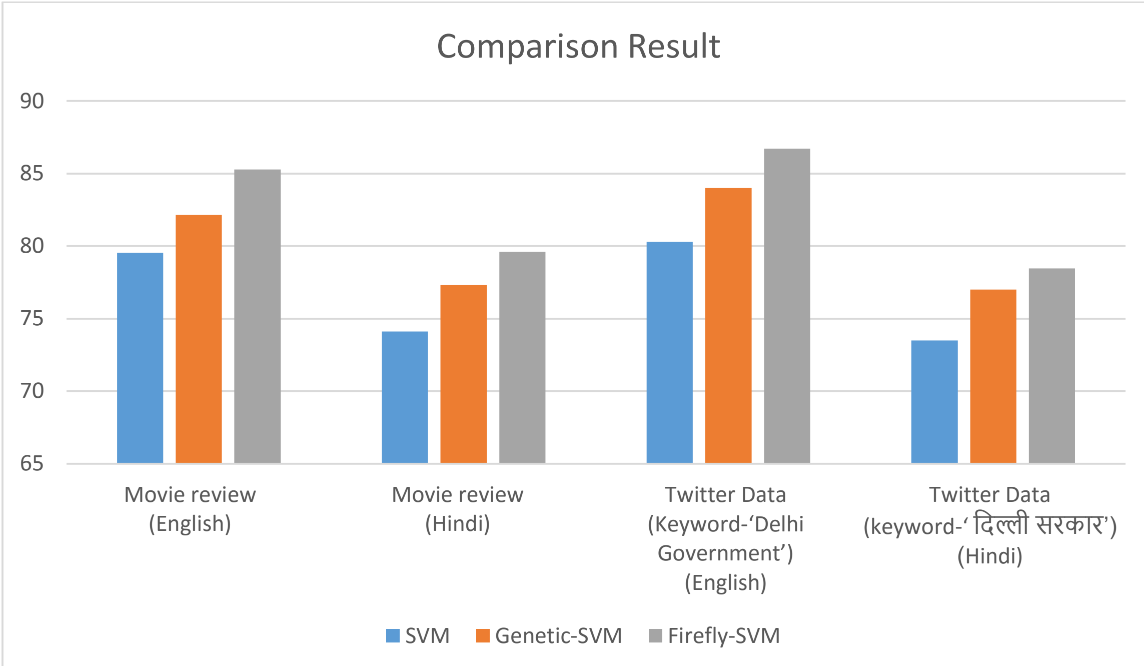

Fig 5.1 Comparison results of Firefly method, Genetic method and baseline SVM

Sentiment analysis systems are applied in almost every business and social domain because opinions are central to almost all human activities and are key influencers of our conducts. Sentiment analysis finds many applications few of them being product Perception: gaining insight into customers' sentiments and evaluating trend change over time, identifying feedback over various products and policies to define new marketing targets and enhance visibility of product, reputetion and brand Management. So accuracy of sentiment analysis systems is very important for these applications. Reduction of vector size significantly leads to improvements in accuracy if the features are noisy or redundant. Firefly algorithm used in this work improves the accuracy of sentiment analysis considerably. The hybrid firefly-SVM model brings an accuracy improvement of 5.64 on average which is significant improvement. Also the hybrid method works very well for languages different than English as shown in this work where hybrid method works well for Hindi language also.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

In this work, a methodology is developed for feature selection in sentiment analysis task that extracts users' opinions from web sources (e.g twitter) and classifies them according to the presence or absence of opinion features. The core contribution of this work is that our method especially suitable for information retrieval and classification area to remove noisy and irrelevant features from input space of dataset while improve the performance of polarity classification.

For reducing feature subset size and to improve the performance by reducing complexity, evolutionary algorithms are used. The firefly algorithm is a powerful optimization algorithm used and is preferred over other optimization algorithms because of its two major advantages: the ability of deal with multimodality and automatic subdivision. Hence firefly algorithm is used for optimizing feature set selection in this work. Four different datasets are used among which two are in Hindi language and other two in English. The datasets are pre-processed, stop words are removed and stemming is performed. Genetic algorithm and Firefly algorithm are used for feature subset selection and SVM classifier is used for classifying the sentiment of the data tuples after selection of features. The Experiment results reveal that the classification accuracy increases on an average by 3% in case of Genetic algorithm and by 5.65% in case of Firefly algorithm.

Also by studying and comparing the performance of swarm inspired subset selection with the other methods of feature selection in opinion mining we found that swarm inspired algorithms converges quickly and produce better results in terms of efficiency and complexity and can efficiently find optimum feature set.

This methodology could support integrated decision support system; Improved results can help merchants acquire valuable feedback (e.g. consumers' satisfaction regarding their products) and could facilitate public administrations to capture the understanding of e-Government and e-Rulemaking.

The Future work includes comprehensive testing in different languages on varied topics keeping the aspect knowledge in frame. Handling sarcasm is a huge challenge which impacts the accuracy essentially. Future work can also focus on the different combinations of classifiers called “ensemble classifiers” in machine learning which can combine features of both component classifier and optimization scheme. Further tests on unstructured and very large datasets acquired from varied sources (Big data) could verify the experimental results.

References

1. Bo Pang., Lilliam Lee.: Opinion Mining and Sentiment Analysis. *Foundations* and Trends in Information Retrieval. Vol. 2, No 1-2 (2008) 1–13 (2008)

2. Akshi Kumar, Teeja Mary Sebastian, Sentiment Analysis: A Perspective on its Past, Present and Future, International Journal of Intelligent Systems and Applications, Vol.4, No.10, 2012

3. Yiming Yang, Jan O. Pederson, A Comparative study on Feature Selection in Text Categorization (1997)

4. Ahmed Abbasi, etl. “Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums” ACM Transactions on Information Systems, Vol. 26, No. 3, 2008.

5. Mehdi Hosseinzadeh Aghdam *, Nasser Ghasem-Aghaee, Mohammad Ehsan Basiri, Text feature selection using ant colonyoptimization, Expert Systems with Applications 36 (2009) 6843–6853

6. A. Abbasi, et al.;Selecting Attributes for Sentiment Classification Using Feature Relation Networks," IEEE Transactions on Knowledge & Data Engineering, vol. 23, 447-462, 2011.

7. Ekbal, A., Saha, S., and Garbe, C. S. “Feature selection using multi objective optimization for named entity recognition” In proceedings of IEEE 20th International Conference on Pattern Recognition, pp. 1937-1940,2010.

8. Y. Lu, C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In Proceedings of the18th international conference on World wide web, pages 131–140, Madrid, Spain, 2009, ACM.

9. Sangita Roy, Samir Biswas, Sheli Sinha Chaudhuri, Nature-Inspired Swarm Intelligence and Its Applications, I.J. Modern Education and Computer Science, 2014, 12, 55-65

10. Carlos M. Fonseca, Peter J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. Spring 1995, Vol. 3, No. 1, Pages 1-16 Massachusetts Institute of Technology, Online December 10, 2007.

11. Azizah Binti Mohamad, Azlan Mohd Zain & Nor Erne Nazira Bazin, Cuckoo Search Algorithm for Optimization Problems—A Literature Review and its Applications, Applied Artificial Intelligence, ISSN: 0883-9514

12. O'Keefe, T., and Koprinska, I. "Feature selection and weighting methods in sentiment analysis", InProceedings of 14th Australasian Document Computing Symposium, pp-67- 74, 2009.

13. Rui Tang, S. Fong, Xin-She Yang, and S. Deb. Wolf search algorithm with ephemeral memory. In Digital Information Management (ICDIM), Seventh International Conference, 165–172, 2012.

14. Amir Hossein Gandomi and Amir Hossein Alavi. Krill herd: a new bio-inspired optimization algorithm. Communications in Nonlinear Science and Numerical Simulation, 2012.

15. William L. Goffe, Gary D. Ferrier, John Rogers, Global optimization of statistical functions with simulated annealing, Journal of EconometricsVolume 60, Issues 1–2, 1994, pp 65-99

16. Kevin M Passino. Biomimicry of bacterial foraging for distributed optimization and control. Control Systems, IEEE, 22(3):52–67, 2002.

17. Xin-She Yang, Janet M Lees, and Chris T Morley. Application of virtual ant algorithms in the optimization of cfrp shear strengthened precracked structures. In Computational Science–ICCS 2006, pages 834–837. Springer, 2006.

18. X.-L. Li, Z.-J. Shao, and J.-X. Qian. Optimizing method based on autonomous animats: Fish-swarm algorithm. Xitong Gongcheng Lilunyu Shijian/System Engineering Theory and Practice, 22(11):32, 2002.

19. Ruby Dhurve, Megha Seth, " Weighted Sentiment Analysis Using Artificial Bee Colony Algorithm", International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064

20. James Kennedy and Russell Eberhart. Particle swarm optimization. InNeural Networks, 1995. Proceedings., IEEE International Conference on, volume 4, pages 942–948. IEEE, 1995.

21. George Stylios, Christos D. Katsis, DimitrisChristodoulakis, " Using Bio-inspired Intelligence for Web Opinion Mining", International Journal of Computer Applications Vol 87 – No.5, 2014

22. DT Pham, A Ghanbarzadeh, E Koc, S Otri, S Rahim, and M Zaidi. The bees algorithm-a novel tool for complex optimisation problems. Proceedings of the 2nd Virtual International Conference on Intelligent Production Machines and Systems, pages 454–459, 2006.

23. T. Sumathi, S.Karthik, M.Marikkannan, "Artificial Bee Colony Optimization for Feature Selection in Opinion Mining", Journal of Theoretical and Applied Information Technology, 10th august 2014. vol. 66 no.1

24. H.F. Wedde, M. Farooq, and Y. Zhang. Beehive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior. Lecture Notes in Computer Science 3172 LNCS:83–94, 2004..

25. M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, 1992.

26. P Lucic and D Teodorovic. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In Preprints of the TRISTAN IV triennial symposium on transportation analysis, pages 441–445, 2001.

27. Abd. Samad Hasan Basari, BurairahHussin, I. GedePramudya Ananta, Junta Zeniarja, "Opinion Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization"

28. Dervis Karaboga, An Idea Based On Honey Bee Swarm for Numerical Optimization, Technical Report-TR06, Erciyes University,Engineering Faculty, Computer Engineering Department 2005.

29. Dusˇan Teodorovic´ and Mauro Dell'Orco. Bee colony optimization–a cooperative learning approach to complex transportation problems. In Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini–EURO Conference and 10th Meeting of EWGT 2005.

30. Habiba Drias, Souhila Sadeg, and Safa Yahi. Cooperative bees swarm for solving the maximum weighted satisfiability problem. In Computational Intelligence and Bioinspired Systems, pages 318–325. Springer, 2005.

31. X.-S. Yang. Engineering optimizations via nature-inspired virtual bee algorithms. volume 3562, pages 317–323, 2005.

32. Deepak Kumar Gupta, Kandula Srikanth Reddy, Shweta, Asif Ekbal, "PSO-ASent: Feature Selection Using Particle Swarm Optimization for Aspect Based Sentiment Analysis", Natural Language Processing and Information Systems,Volume 9103 of the series Lecture Notes in Computer Science pp 220-233

33. Shoubao Su, Jiwen Wang, Wangkang Fan, and Xibing Yin. Good lattice swarm algorithm for constrained engineering design optimization. In Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. pages 6421–6424. IEEE, 2007.

34. KN Krishnanand and D Ghose. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, pages 84–91. IEEE, 2005.

35. Antonio Mucherino and Onur Seref. Monkey search: a novel metaheuristic search for global optimization. In Data Mining, Systems Analysis and Optimization in Biomedicine, volume 953, pages 162–173, 2007.

36. Ying Chu, Hua Mi, Huilian Liao, Zhen Ji, and QH Wu. A fast bacterial swarming algorithm for high-dimensional function optimization. In Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence), pages 3135–3140. IEEE, 2008.

37. Xin-She Yang. Firefly algorithm, stochastic test functions and design optimization. International Journal of Bio-Inspired Computation, 2(2):78–84, 2010.

38. X.S. Yang. A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pages 65–74, 2010.

39. Serban Iordache. Consultant-guided search: a new metaheuristic for combinatorial optimization problems. In Proceedings of the 12th annual conference on Genetic and evolutionary computation, pages 225–232. ACM, 2010.

40. Xin-She Yang and Suash Deb. Eagle strategy using le´vy walk and firefly algorithms for stochastic optimization. In Nature Inspired Cooperative Strategies for Optimization (NICSO2010), pages 101–111.Springer, 2010.

41. TO Ting, Ka Lok Man, Sheng-Uei Guan, Mohamed Nayel, and Kaiyu Wan. Weightless swarm algorithm (wsa) for dynamic optimization problems. In Network and Parallel Computing, pages 508–515. Springer, 2012.

42. Francesc de Paula Comellas Padro´, Jesu´s Mart´ınez Navarro, et al. Bumblebees: a multiagent combinatorial optimization algorithm inspired by social insect behaviour. 2011.

43. Gandomi AH, Yang XS, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Comput Struct 89(23–24):2325–2336