Simultaneous Feature Selection and Parameter Optimization Using Multi-objective Optimization for Sentiment Analysis

Mohammed Arif Khan[1], **Asif Ekbal**[1] **and Eneldo Loza Mencía**[2]
[1]Dept. of Computer Science & Engineering, Indian Institute of Technology Patna, India
[2]KE Group, Dept. of Informatics, Technische Universitat Darmstadt, Germany
[1]{arif.mtmc13, asif}@iitp.ac.in
[2]eneldo@ke.tu-darmstadt.de

Abstract

In this paper, we propose a method of feature selection and parameter optimization for sentiment analysis in Twitter messages. Appropriate features and parameter combinations have significant effect to the performance of any classifier. As base learning algorithms we make use of Random Forest and Support Vector Machines. We perform sentiment analysis at the message level, and use the platform of SemEval-2014 shared task. We achieve substantial performance improvement with our proposed model over the systems that are developed with random feature subsets and default parameter combinations.

1 Introduction

Social media has grown enormously over the last decade. Huge amount of unstructured texts are generated through social media platforms such as Twitter, blogs etc. People's opinions on certain aspects or events are always important. Mining relevant information from such large amounts of text manually is almost impossible, and so there is an obvious need to develop automatic systems that can extract the most important information from these large data sets. Sentiment analysis or opinion mining, a multi-disciplinary area covering natural language processing, data mining and machine learning, aims at extracting emotions from texts. This is an active research area, which has been used in different applications such as financial prediction (Mittal and Goel, 2012), evaluating customer feedbacks (Maks and Vossen, 2013) and understanding users' opinions about products and/or services (Mukherjee and Bhattacharyya, 2012) etc. Literature shows that machine learning (Bo Pang and Vaithyanathan, 2002), (Boiy and Moens, 2009) and lexicon-based (Maite Taboada and Stede, 2011), (Cataldo Musto and Polignano, 2014) approaches have more predominantly been used for sentiment analysis.

One of the earliest studies on sentiment analysis on microblogging websites was provided by (Alec Go and Huang, 2009). The work presents a distant supervised based approach for sentiment classification using hash tags in tweets. (Kevin Gimpel, 2011) propose an approach to sentiment analysis in twitter using Part-of-Speech (PoS) tagged n-gram features and some twitter specific hash tags. The scarcity of labelled data is often a bottleneck for any machine learning based system, and sentiment analysis is not an exception. A technique for automatically creating labelled datasets for sentiment analysis research has been proposed in (Pak and Paroubek, 2010). (Apoorv Agarwal and Passonneau, 2011) used tree kernel decision tree that made use of the features such as kernel decision, Part-of-Speech information, lexicon based features and several other features. Previous studies such as (Dmitry Davidov and Rappoport, 2010) observed that hash tags and smileys work good for sentiment analysis. (Nielsen, 2011) concluded that the AFINN word list performs slightly better than ANEW (Affective Norms for English Words) in twitter sentiment analysis. Most of the supervised approaches have used the features such as N-grams, PoS information, emoticons and opinion words. Among the supervised approaches, Support Vector Machine (SVM) (Vapnik, 1995) and Naive Bayes (Mitchell, 1997) have been popularly used.

The performance of any classification technique depends on the features used to represent training and test patterns. Feature selection (Liu and Yu, 2005; Liu and Motoda, 1998) is the technique of automatically selecting a subset of relevant features for a classifier. It helps to build a robust model and reduces the complexity of the learning

algorithm. This is also termed as attribute selection/ subset selection etc. By removing most irrelevant and redundant features from the data, feature selection helps to improve the performance of a classifier. In a ML approach, appropriate feature selection can be thought of as an optimization problem. Some of the prior works where feature selection has been modelled within the frameworks of evolutionary optimization techniques include (Ekbal and Saha, 2012; Ekbal and Saha, 2013). These works primarily focussed on named entity recognition in multiple languages. In general a classifier has many parameters whose values heavily influence the performance of a classifier. Therefore, like feature selection, determining the appropriate values of parameters for a classifier is another important key issue.

In this paper we propose a method of feature selection and parameter optimization within the framework of multiobjective optimization (MOO) (Deb, 2001). We implement a diverse set of features, consider their various subsets, and optimise various functions such as recall and precision, F-measure and number of features etc. As the base learning algorithms we use Support Vector Machines (Vapnik, 1995) and Random Forest (Breiman, 2001). We have carried out experiments on the benchmark set up of SemEval-2014 shared task [1]. Evaluation shows that our proposed system achieves substantial performance improvement over the model developed with random feature subsets and default parameter settings. The key contributions of the paper are two-fold, *viz.* (i). proposing a joint model of feature selection and parameter optimization, especially for sentiment analysis and (ii) the use of MOO based evolutionary methods in the broad areas of opinion mining.

The rest of the paper is structured as follows. Section 2 describes the features that we implanted for sentiment analysis. In Section 3 we present our proposed method for feature selection and parameter optimization. In Section 4, we report the evaluation results with analysis and discussions. Finally, in Section 5 we conclude the paper.

2 Features for Sentiment Analysis

Feature plays an important role in sentiment classification. We implement 55 features, and we categorise these into the five groups.

1. *Emoticon Features:*

   Positive Smiley (pSmiley): It is a common practice that people represents emoticons through variety of smileys. A smiley present in a tweet directly represents its sentiment. A feature is defined that identifies whether the positive smiley(s) is/are present or not in a tweet. We used a set of positive smiles available at this web page [2].

   Negative Smiley (nSmiley): Similar to positive smiley, we also obtain a set of negative smileys from the same source. The value of this feature is set to "yes" or "no" depending upon whether the tweet contains the negative smiley or not.

   Last Token Smiley (LastTokenSmiley): This feature indicates whether the last token in a tweet is a smiley or not. The presence of this smiley clearly indicates that tweet can't be of neutral type.

2. *Lexicon Features:* We use three automatically built sentiment lexicons, namely NRC Hash tag Sentiment Lexicon (Saif Mohammad and Zhu, 2013) , Sentiment140 Lexicon (Saif Mohammad and Zhu, 2013) and Bing Liu Lexicon (Xiaowen Ding and Yu, 2008).

   NRC Hash tag Sentiment Lexicon: (Saif Mohammad and Zhu, 2013) showed that hashtagged emotion words are good indicators that the tweet as a whole (even without the hashtagged emotion word) is expressing the same emotion. We adapted that idea and obtain the following features from this Lexicon:

   (i) (LexNRC): Scores of all the words are summed up. A feature value is, thereafter, set to based on the overall score of the tweet. The feature values are set to +1 , 0 or -1 depending upon whether the overall score is above 1, varies within the range -1 and +1 or less than -1.

   (ii) (PositiveLexNrcToken): A feature is generated that takes the value equal to the number of words present in a tweet having their scores greater than zero.

[1] http://alt.qcri.org/semeval2014/task9/

[2] http://www.datagenetics.com/blog/october52012/index.html

(iii) (NegativeLexNrcToken): A feature is defined that takes the value equal to the number of words having the negative scores.

(iv) (MaxLexNrc): This feature indicates the maximum positive score among all the tokens in a tweet.

(v) (MinLexNrc): This feature indicates the minimum value (i.e. maximum negative score) of polarity among all the tokens of a tweet.

(v) (LastNonZeroScoreNrc): This feature corresponds to the polarity of the last token of the tweet. This feature has been defined based on the observation that the sentiment as expressed in the last word of the tweet has great influence on the overall sentiment of any tweet.

Sentiment140 Lexicon: In this lexicon, the individual scores of the tokens have been calculated based on the number of tweets in which these tokens co-occur with the positive or the negative emoticons. For every tweet in the data set, following features are defined using the sentiment score, i.e. *score(w)* of each token w in the tweet:

(i) (Lex140): Feature that corresponds to the total score = $\sum_{w \in tweet} score(w)$.

(ii) (PositiveLex140Token): Feature that indicates the number of tokens in the tweet with score(w) > 0

(iii) (NegativeLex140Token): Feature that indicates the number of tokens in the tweet with score(w) < 0

(iv) (MaxLex140): Feature that indicates the maximal score of any token in the tweet = $max_{w \in tweet} score(w)$

(v) (MinLex140): Feature that corresponds to the minimal score of any token in the tweet = $min_{w \in tweet} score(w)$

(vi) (LastNonZeroScore140): The score of the last positive token (score(w) > 0) in the tweet

Bing Liu's Lexicon: We define the following two features based on this lexicon.

(i) (BllPositiveWords): A feature is defined that has the value equal to the number of words of a tweet present in the BLL's (Bing Liu Lexicon) word list of positive lexicons.

(ii) (BllNegativeWords): This feature is defined based on the number of words of a tweet present in the BLL's (Bing Liu Lexicon) word list of negative lexicons.

3. *SentiWordNet Features:* Based on the SentiWordNet dictionary (Andrea Esuli and Sebastiani, 2010) we define the following features that depend on the number of words bearing positive, negative and neutral sentiments.

   SWN Positive words (SwnPositiveTokenCount): This is an integer-valued feature that is set equal to the number of words having more positive sentiment.

   SWN Negative words (SwnNegativeTokenCount): Similar to the feature defined above, this is also denoted by an integer-valued feature that takes the value equal to the number of words that bear more negative sentiments.

   SWN Neutral words (SwnNeutralTokenCount): This feature determines the number of neutral words for a tweet. This is obtained by the following formula:
   SWN Neutral words = (number of words in a tweet) - (number of SWN positive words + number of SWN negative words).

   SWN Polarity (SwnPolarity): For every tweet a polarity score is assigned based upon the scores of the constituent tokens. Let x and y denote the sum of positive and negative sentiment scores, respectively, for all the words of a tweet. Now we assign the polarities of the words as follows. If (x-y) > 0.5 then polarity of the tweet is assigned to be positive; If (x-y) $<$ -0.5 then polarity of the tweet is assigned to be negative; and if (x-y) lies between -0.5 to 0.5 then the polarity of the tweet is assigned to be neutral.

4. *Part-of-Speech (PoS) Features:* (CHRIS NICHOLLS, 2009) found that different PoS categories contribute to sentiment analysis in varying degrees. To extract the PoS of every word present in a tweet,

we use the CMU ARK tagger [3]. We define a feature vector that considers the number of PoS categories present in a tweet. These features are listed in Table 2 from feature number 30 to feature number 54.

5. *Miscellaneous Features:* Beside the above four categories of features, we also implemented the following features:

   Hash Count (HashCount): This feature gives the difference of number of positive hashtags (ex, #excellent, #thankful) and negative hashtags (ex. #bad, #terror). A positive value (negative value) of this feature indicates positive (negative) sentiment.

   Tweet Length (TweetLength): Generally, a long tweet have more number of stop words. This feature counts the number of words present in a particular tweet. More is the number of stop words present in a tweet higher is the chance of it not being of neutral polarity.

   Capital Characters (InitCap): If majority of the words present in a tweet are capitalised then there is more chance that the overall sentiment of the tweet is non-neutral. We compute the ratio of captalised words with respect to the total number of words present in a tweet. If this ratio is above a certain threshold then we set the feature value to 1, otherwise 0.

   All Cap Words (AllUpperWords): This is an integer-valued feature, the value of which is set to the number of words equal to the number of capitalised words. This feature is defined based on the assumption that the words written using only the capitalised letters express sentiments more strongly.

   Negation (NotPresent): The presence of words such as "not", "couldn't", won't", "shouldn't" etc. reverts the polarity of the sentence. We manually create a list of all such words from the training data. A binary-valued feature is then defined that is set to "yes" or "no" depending upon the presence of such word in the tweet.

   Stop Words (StopWords): In the previous study (Hassan Saif and Alani, 2012), it has been shown that the classifiers learned with stop words outperform those learned without stop words. Here we define a feature based on the number of stop words present in a tweet. If the number of stop words is greater than 20 % (in terms of the number of words) then the tweet has more possibility of being neutral.

   Elongated Words (ElongatedWords): To represent strong emotions, people, in general, use to repeat the same character more than twice. Some of the examples are: "happppppy", "coooool" etc. We define this feature in such a way that checks whether there is at least one elongated words present in the tweet.

   Last Token (LastToken): This feature checks whether "?" or "!" is present in the last position of the tweet or not. The presence of such token in the last position denotes that the overall sentiment expressed in the tweet may be neutral. For ex. "U remember her from the 90s?", "Hello Mumbai Gud morning!".

[3] http://www.ark.cs.cmu.edu/TweetNLP/

3 Proposed Approach

In this section, firstly we introduce the concept of multiobjective optimization (MOO), formulate the problem of simultaneous feature selection and parameter optimization and then describe the proposed approach.

3.1 Multi-objective Formulation for feature subset selection

Multiobjective optimization (MOO) deals with the concept of optimising more than one function at a time. Compared to single objective optimization (SOO) that concerns in optimising only one function at a time, it has the ability to produce more than one feasible solutions which are equally important from the algorithmic point of view. This concept has been widely used for solving many decision problems. Mathematically, MOO (Deb, 2001) can be described as follows:
Find the vectors

$$x^* = [x_1^*, x_2^*, ..., x_N^*]^T \quad (1)$$

of decision variables that simultaneously optimize the M objective values

$$f_1(x), f_2(x), ..., f_M(x); N \geq M, N > 1, M > 1 \quad (2)$$

while satisfying the constraints, if any.

3.2 Formulation of the Problem

Performance of any classifier depends greatly on the parameters used. Generally we choose the best parameters of any classifier following a heuristics based method, where various combinations are tested on a held-out dataset and then finally the most promising one selected. This process is time-consuming and computation intensive, and therefore, some automatic techniques are most preferred.

Given a set of features F, appropriate parameters P and two classification quality measures, recall and precision, determine the feature subset F^* and parameter subset P^* such that maximize [recall, precision] where F$^*\subseteq$ F and P$^*\subseteq$ P.

In our case we use precision, recall, accuracy and the number of features as the objective functions. We build the following two frameworks as follows: (i). maximizing recall and precision, and (ii). minimising the number of features and maximising the accuracy. As precision and recall have trade-off (Buckland and Gey, 1994) so this combination will provide non-dominant pareto optimal solutions. In second framework we aim for high accuracy while using least computational cost (i.e. minimizing number of features). Along with these objective functions, we use the following parameters:

Random forest: Number of trees;
LibSVM: Cost and gamma parameters;
LibLinear: Cost parameter.

In Random Forest, more number of trees provides better accuracy but it uses more computational time. Cost parameter 'C' is a regularisation parameter, which controls the trade-off between achieving a low error on the training data and minimising the norm of the weights. It determines the influence of the misclassification on the objective function. As gamma increases, the algorithm tries harder to avoid misclassifying training data, which leads to overfitting.

3.3 Encoding of the Problem

If total number of features is N and the number of parameter to be optimized is M, then the length of the chromosome will be N+M. For an example, we encode a problem in Figure 1. The first 12 bits of the chromosome represent the features and the remaining bits encode the parameters. Each of the first 12 bits represents a feature, and this is randomly initialised to either 0 or 1. If the i^{th} position of a chromosome is 1 then the i^{th} feature participates in constructing the classifier else not. Here out of 12 features, 7 (first, fourth, fifth, sixth, eighth, tenth and eleventh) have been used to construct the classifier. If the population size is P then all the P number of chromosomes of this population are initialized in the same way.

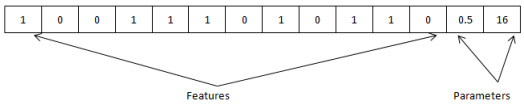

Figure 1: Encoding of the problem

3.4 Fitness Computation

For the fitness computation, the following procedure was executed.

(i) Let there are F number of features present in a particular chromosome (i.e. there are total F number of 1's present in the chromosome).

(ii) Construct a classifier with only these F features.

(iii). Perform 3-fold cross validation to compute the objective function values.

(iii) The aim is to optimize the values of the objective functions using the search capability of NSGA-II .

3.5 Other Operators

After fitness computation, we used binary tournament selection (Deb, 2001) as in NSGA-II, followed by crossover and mutation (Deb, 2001). The most characteristic part of NSGA-II is its elitism operation (Deb, 2001), where the non-dominated solutions among the parent and child populations are propagated to the next generation. The near pareto optimal strings of the last generation provide the different solutions to the feature selection problem. Each of these solutions is

equally important from the algorithmic points of view.

3.6 Selection of Final Solution

MOO provides large number of non-dominated solutions (Deb, 2001) on the final Pareto optimal front. Although each of these solutions are equally important but sometimes the user may require a single solution. Hence, we develop a method for selecting a single solution from the set of solutions. For every solution of the final pareto optimal front, (i) classifier is trained using the features present in that particular solution (ii) evaluated for 3-fold cross validation on the training set (iii) selected the best feature combination that yileds the highest F-measure value and (iv). use this particular feature combination to report the final evaluation results.

3.7 Pre-processing and Experimental Setup

The data has to be pre-processed before being used for actual machine learning training. Each tweet is processed to extract only those relevant parts that are useful for sentiment classification. For example, we removed the tweets that don't have any label information in the training data; symbols and punctuation markers are filtered out; URLs are replaced by the word URL; words starting with digits are removed etc. Each tweet is then passed through the ARK tagger developed by CMU [4] for tokenization and Part-of-Speech (PoS) tagging.

From pre-processed tweets we build the final training set. For each tweet in the training set we extract the vectors based on all the features as defined above. As the base classifiers we make use of two different learning algorithms, namely Support Vector Machine (SVM) (Vapnik, 1995) and Random Forest (Breiman, 2001). For SVM we use two implementations as available in LibSVM (Chang and Lin, 2011) and LibLINEAR (Rong-En Fan and Hsieh, 2008).

4 Experiments and Result Discussion

In this section we describe the datasets we used for our experiments, report the evaluation results and provide necessary analysis.

[4]http://www.ark.cs.cmu.edu/TweetNLP/

4.1 Datasets

We use the data sets from SemEval-2014[5] shared task. The data sets contain 8,223 classified tweets in the training data and 8,987 tweets as part of the test data. Details of the data sets are shown in Table 1. Please note that we make use of our own evaluation script that also considers the accuracies of neutral classes when we perform evaluation. In contrast, the official evaluation of SemEval script simply ignores the neutral classes.

Table 1: Multi-class Data Sets

S.N.	Data Set	Class			Total
		Positive	Negative	Neutral	
1	Training	3071	1210	3942	8223
2	Test	3506	1541	3940	8987

4.2 Feature sets

We generate different models using the various feature combinations as shown in Table 2. Brief descriptions of these feature sets are described as below:

S55: is a set of 55 features as listed in Table 2. The intuition behind selection of the features in S55 is described in Section 2.

R20: is a subset of 20 features randomly selected from S55.

R30: is a subset of 30 features randomly selected from S55.

R40: is a subset of 40 features randomly selected from S55.

OS55 with FS & PO: This corresponds to the feature set on which joint feature selection and parameter optimization are performed.

OS55 with default parameters: This corresponds to the feature set where feature selection is performed but instead of using parameter optimization technique we make use of the default parameter settings.

4.3 Parameters for MOO

We use R20, R30 and R40 feature sets to construct the baseline models. Following parameter values are used for the NSGA-II. Population size = 32 ; Number of generations = 20; Number of objective functions = 2; Number of real variables = 2; Probability of crossover of real variable = 0.98; Probability of mutation of real variable = 0.50; Distribution index for crossover = 14; Distribution index for mutation = 38; Number of binary variables = 55

[5]http://alt.qcri.org/semeval2014/task9/

Table 2: Feature Sets of S55

Feature No	Features	Feature sets									
		R20	R30	R40	S55	OS55 for Exp-PR			OS55 for Exp-ANF		
						Random Forest	LibSVM	LibLINEAR	Random Forest	LibSVM	LibLINEAR
0	HashCount	1	0	1	1	0	0	0	0	0	0
1	TweetLength	1	0	1	1	0	1	0	0	0	0
2	InitCap	0	1	1	1	1	1	1	1	1	1
3	PercentCapital	0	1	0	1	0	1	0	1	1	1
4	AllUpperWword	0	0	1	1	1	0	1	0	0	0
5	NotPresent	0	1	0	1	1	0	0	0	0	0
6	pSmiley	1	0	1	1	0	0	0	1	0	1
7	nSmiley	0	1	1	1	1	1	1	1	1	1
8	LastTokenSmiely	0	1	1	1	1	0	0	1	0	0
9	StopWords	1	0	1	1	0	1	1	0	1	1
10	ElongatedWords	0	1	1	1	1	0	0	0	1	1
11	LastToken	0	0	1	1	1	0	0	0	1	0
12	SwnPositiveTokenCount	1	0	0	1	1	1	0	1	0	0
13	SwnNegativeTokenCount	0	1	1	1	1	1	1	0	1	1
14	SwnNeutralTokenCount	0	0	1	1	1	0	0	1	0	0
15	SwnPolarity	1	1	0	1	1	1	1	1	1	1
16	LexNRC	0	1	1	1	1	1	1	0	1	1
17	PositiveLexNrcToken	1	0	1	1	0	1	0	0	1	0
18	NegativeLexNrcToken	0	1	0	1	0	0	1	1	0	0
19	MaxLexNrc	0	0	1	1	1	1	1	1	1	1
20	MinLexNrc	1	0	1	1	1	1	1	1	1	1
21	LastNonZeroScoreNrc	0	1	1	1	0	0	1	0	0	0
22	Lex140	1	0	1	1	0	0	0	0	0	0
23	PositiveLex140Token	0	1	0	1	1	0	0	0	0	0
24	NegativeLex140Token	0	1	1	1	0	0	0	1	0	1
25	MaxLex140	1	0	1	1	0	0	1	0	0	0
26	MinLex140	0	0	1	1	1	0	0	1	0	0
27	LastNonZeroScore140	0	1	1	1	0	0	0	0	0	0
28	BllPositiveWords	1	0	1	1	1	1	1	1	1	1
29	BllNegativeWords	0	1	0	1	1	1	1	1	1	1
30	CommonNoun	0	1	1	1	1	0	1	1	1	0
31	Pronoun	1	1	0	1	0	1	0	0	1	0
32	NominalPossessive	0	0	1	1	1	0	0	0	1	1
33	ProperNnoun	0	1	1	1	0	0	0	0	0	0
34	ProperNounPossessive	1	1	0	1	1	1	1	1	0	1
35	NominalVerbal	0	0	1	1	1	0	1	1	0	0
36	ProperNounVerbal	0	1	0	1	0	0	1	1	0	1
37	VerbCoupla	1	0	1	1	1	0	0	1	0	0
38	Adjective	0	1	1	1	1	0	1	1	0	1
39	Adverb	0	1	0	1	0	1	0	1	1	0
40	Injection	0	0	1	1	0	1	0	0	1	0
41	Determiner	1	0	1	1	1	0	0	1	0	0
42	Preposition	0	1	1	1	1	0	1	0	0	1
43	ConditionalConjunction	0	0	1	1	0	1	1	0	1	1
44	VerbParticle	1	1	0	1	0	0	0	0	0	0
45	ExistentialPredeterminer	0	1	1	1	0	0	0	0	0	0
46	ExistentialPredeterminerVerbal	0	1	0	1	0	0	0	1	0	0
47	NumberOfHash	1	0	1	1	0	0	0	1	0	1
48	AtMention	1	0	1	1	1	1	1	1	0	1
49	DiscourseMarker	0	1	1	1	1	1	1	1	0	0
50	UrlEmail	1	0	1	1	1	0	0	1	0	0
51	Emoticon	0	1	0	1	1	1	1	1	1	1
52	Numeral	0	1	1	1	1	1	1	1	1	1
53	Punctuaion	0	1	0	1	1	1	0	0	0	1
54	OtherPOS	1	0	1	1	1	1	1	1	1	1
Number of features		20	30	40	55	33	25	26	31	23	26

1 denotes the presence and 0 denotes the absence of a particular feature in the corresponding feature set

4.4 Experimental Results

At first we perform experiments using recall and precision as the two objective functions. Experimental results are shown in Table 3. Here we define the three models as follows: Model-1: Random Forest; Model-2: LibSVM and Model-3: LibLinear. The baseline model developed with random forest classifier shows the highest performance (i.e. 53.60%) for the system that makes use of 40 features. Random forest based baseline model developed with 20 features (i.e. R20) yields the F-measure values of 50.40% and the model developed with 30 features (i..e R30) shows the F-measure value of 52.30%. With LibLinear implementation of SVM we achieve the higher performance with 51.00% F-measure value. While considering both feature selection (FS) and parameter optimization (PO), we obtain the F-measure values of 59.30, 59.10 and 57.70 for the models optimal subset (OS55 with FS and PO) with respect to the first, second and third model, respectively. It is to be noted that when we perform only feature selection we observe the increments of 00.40, 14.80 and 04.90 percentage points for the respective models. However for the model which was developed both with feature selection and parameter optimization, we see the improvements of 04.20, 20.00 and 04.30 percentage points, respectively.

Table 3: Results for Exp-PR

S. N.	Feature Sets	Para-meters	Classifiers		
			Random Forest	LibSVM	Lib-LINEAR
1	R20	P	52.10	60.40	49.90
		R	52.70	56.30	53.20
		F1	52.30	50.60	50.10
2	R30	P	50.50	52.30	52.90
		R	50.80	44.60	52.10
		F1	50.40	31.00	50.40
3	R40	P	53.50	58.70	50.50
		R	54.10	53.50	52.20
		F1	53.60	46.20	51.00
4	S55	P	55.10	57.90	53.50
		R	55.60	49.20	55.10
		F1	55.10	39.10	53.40
5	OS55 with FS & PO	P	62.20	61.00	62.40
		R	61.00	60.60	60.30
		F1	59.30	59.10	57.70
6	OS55 default paameters	P	55.50	59.40	62.30
		R	56.10	57.30	60.60
		F1	55.50	53.90	58.30
% improvement due to FS			00.40	14.80	04.90
% improvement due to FS and PO			04.20	20.00	04.30

P=Precision, R=Recall, F1=F-measure (all in %) FS= Feature Selection, PO= Parameter Optimization

Table 4: Results for Exp-ANF

S. N.	Feature Sets	Para-meters	Classifiers		
			Random Forest	LibSVM	Lib-LINEAR
1	R20	A	52.65	56.33	53.32
		NF	20	20	20
		F1	52.30	50.60	53.10
2	R30	A	50.84	44.64	52.00
		NF	30	30	30
		F1	50.40	31.00	50.30
3	R40	A	54.11	53.49	55.20
		NF	40	40	40
		F1	53.60	46.20	54.10
4	S55	A	55.61	49.2	54.16
		NF	55	55	55
		F1	55.10	39.10	49.90
5	OS55 with FS & PO	A	61.32	60.58	59.06
		NF	31	23	26
		F1	59.70	59.20	55.50
6	OS55 default paameters	A	56.08	58.53	60.24
		NF	31	23	26
		F1	55.50	56.10	58.00
% improvement due to FS			00.40	17.00	08.10
% improvement due to FS and PO			04.60	20.10	05.60

A=Accuracy (in %), NF=Number of Features, F1=F-measure (in %), FS= Feature Selection, PO= Parameter Optimization

Hence it can be concluded that performing feature election and parameter optimisation together is better suited compared to the model that makes use of only the default parameters of the classifiers. However, third model achieves higher performance with the defaults parameters only. The parameters selected through MOO are shown in Table 5.

Results of the experiments when accuracy and number of features are optimised are shown in Table 4. Results show that both feature selection (FS) and parameter optimization (PO) yield better accuracies with F-measure values of 59.70, 59.20 and 55.50 for the respective models. When the selected feature set is used train the classifiers with default parameters, the models show the F-measure values of 55.50%, 56.10% and 58.00% for the three models, respectively. This again shows quite similar behaviours, i.e. first two classifiers perform superior with the joint model framework. However third model yields higher performance when the classifier is trained with the features selected through MOO based approach, but uses the default parameters.

Here we observe that in both the experiments, LibLinear implementation of SVM provides better result with default parameter configurations compared to their respective optimized parameter sets. This may be attributed to the fact that further attention is required to select the parameters of LibLinear model. Our approach is evolutionary in nature, and therefore, better accuracies can be achieved by either increasing the number of generations or size of the population or both.

Table 5: Optimized Parameters

S.N.	Classifier	Parameters	Experiment	
			Exp-PR	Exp-ANF
1.	Random Forest	Trees	862, 853	1990
2.	LibSVM	Cost	2^14	2^14
		Gamma	2^(-10)	2^(-9)
3.	LibLINEAR	Cost	4	2^(-2), 2^(-7), 2^(-8)

4.5 Comparisons

To compare our results with some of existing systems which were developed on the same data sets viz. NRC Canada-B (Xiaodan Zhu and Mohammad, 2014), Coooolll-B (Tang et al., 2014), TeamX-B (Miura et al., 2014), SAIL-B (Nikolaos Malandrakis and Narayanan, 2014), DAEDALUS-B(JulioVillena Roman, 2014) and SU-sentilab-B(Gizem Gezici and Saygin, 2013), we evaluate our best model (OS55 with FS & PO for Exp-ANF) using SemEval-14's scorer. This scorer considers only the F-measures for positive and negative classes. In Table 6 we present the results of comparisons. We observed that classification of neutral class was more challenging compared to the positive and negative classes. Hence in all our experiments we also considered neutral class along with positive and negative classes. It shows the F-measure values of 59.60, 37.30 and 67.50 for positive, negative and neutral class, respectively. When the official evaluation scorer was executed, the system yields the F-measure values of 64.16%, 74.75%, 68.39%, 60.62% and 35.48% for LiveJournal2014, SMS2013, Twitter2013, Twitter2014 and Twitter2014Sarcasm datasets, respectively which is named as 'Our System I'. Further we evaluate considering only positive and negative classes and named as 'Our Sys-

tem II'. Here 'Our System I' has much better F-measure than the average F-measure of our system which ignores the neutral class. This shows that our system provides better F-measure for neutral class.

Table 6: Comparisons with some existing systems

S. N.	System	F-measure					
		Live Journal 2014	SMS 2013	Twitter 2013	Twitter 2014	Twitter 2014 Sarcasm	Average
1.	NRC Canada-B	74.84	70.28	70.75	69.85	58.16	68.78
2.	Coooolll-B	72.90	67.68	70.40	70.14	46.66	65.55
3.	TeamX-B	69.44	57.36	72.12	70.96	56.50	65.27
4.	SAIL-B	69.34	56.98	66.80	67.77	57.26	63.63
5.	Our System I	64.16	74.75	68.39	60.62	35.48	60.68
6.	SU-sentilab-B	55.11	49.60	50.17	49.52	31.49	47.18
7.	Our System II	57.77	45.00	49.40	47.84	25.76	45.15
8.	DAEDALUS-B	40.83	40.86	36.57	33.03	28.96	36.05

5 Conclusion and Future Work

In this work, we have posed the problem of simultaneous feature selection and parameter optimization as a MOO problem, and evaluate this for sentiment analysis. We have implemented significantly diverse set of features for the task. Experiments show the effectiveness of the proposed approach with significant performance improvements over the various baselines developed with random feature subsets. It is also evident from the evaluation results that simultaneous feature selection and parameter optimization is better compared to the only feature selection.

In future, we would like to add more features to increase the baseline performance. More detailed parameter selection, for example, the kernel function of SVM need to be optimised to realise the effects of more systematic parameter optimization.

Acknowledgments

We thank the anonymous reviewers for their comments. This work is, in part, supported by German Academic Exchange Service (DAAD) through DAAD-IIT Master Sandwich Program.

References

[Alec Go and Huang2009] Richa Bhayani Alec Go and Lei Huang. 2009. Twitter sentiment classification using distant supervision. Technical report, Univ. stanford.

[Andrea Esuli and Sebastiani2010] Stefano Baccianella Andrea Esuli and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Seventh conference on International Language Resources and Evaluation LREC'10*, Valletta, Malta, May.

[Apoorv Agarwal and Passonneau2011] Ilia Vovsha Owen Rambow Apoorv Agarwal, Boyi Xie and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, pages 30–38. Association for Computational Linguistics.

[Bo Pang and Vaithyanathan2002] Lillian Lee Bo Pang and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

[Boiy and Moens2009] Erik Boiy and Marie-Francine Moens. 2009. A machine learning approach to sentiment analysis in multilingual web texts. *Information retrieval*, 12(5):526–558.

[Breiman2001] Leo Breiman. 2001. Random forests. *Mach. Learn.*, 45(1):5–32, October.

[Buckland and Gey1994] Michael K. Buckland and Fredric C. Gey. 1994. The relationship between recall and precision. *JASIS*, 45(1):12–19.

[Cataldo Musto and Polignano2014] Giovanni Semeraro Cataldo Musto and Marco Polignano. 2014. A comparison of lexicon-based approaches for sentiment analysis of microblog posts. *Information Filtering and Retrieval*, page 59.

[Chang and Lin2011] Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A library for support vector machines. In *ACM Transactions on Intelligent Systems and Technology*.

[CHRIS NICHOLLS2009] FEI SONG CHRIS NICHOLLS. 2009. Improving sentiment analysis with part-of-speech weighting. In *Eighth International Conference on Machine Learning and Cybernetics*, Baoding, July.

[Deb2001] Kalyanmoy Deb. 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Ltd, England.

[Dmitry Davidov and Rappoport2010] Oren Tsur Dmitry Davidov and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *Coling 2010: Poster Volume, pages 241–249*, Beijing, August.

[Ekbal and Saha2012] Asif Ekbal and Sriparna Saha. 2012. Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition. *IJDAR*, 15(2):143–166.

[Ekbal and Saha2013] Asif Ekbal and Sriparna Saha. 2013. Full length article: Simulated annealing based classifier ensemble techniques: Application to part of speech tagging. *Inf. Fusion*, 14(3):288–300, July.

[Gizem Gezici and Saygin2013] Berrin Yanikoglu Dilek Tapucu Gizem Gezici, Rahim Dehkharghani and Yucel Saygin. 2013. Su-sentilab: A classification system for sentiment analysis in twitter. In *Proceedings of the International Workshop on Semantic Evaluation*, pages 471–477.

[Hassan Saif and Alani2012] Yulan He Hassan Saif and Harith Alani. 2012. Semantic sentiment analysis of twitter. In *11th international conference on The Semantic Web ISWC'12, Volume I*, Heidelberg.

[JulioVillena Roman2014] Gonzalez Cristobal Jose Carlos JulioVillena Roman, Janine Garcia Morera. 2014. Daedalus at semeval-2014 task 9: Comparing approaches for sentiment analysis in twitter.

[Kevin Gimpel2011] Nathan Schneider Kevin Gimpel. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceeding HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2 Pages 42-47*.

[Liu and Motoda1998] Huan Liu and Hiroshi Motoda. 1998. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Norwell, MA, USA.

[Liu and Yu2005] Huan Liu and Lei Yu. 2005. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. on Knowl. and Data Eng.*, 17(4):491–502.

[Maite Taboada and Stede2011] Milan Tofiloski Kimberly Voll Maite Taboada, Julian Brooke and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

[Maks and Vossen2013] Isa Maks and Piek Vossen. 2013. Sentiment analysis of reviews: Should we analyze writer intentions or reader perceptions? In *RANLP*, pages 415–419.

[Mitchell1997] Thomas M. Mitchell. 1997. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

[Mittal and Goel2012] Anshul Mittal and Arpit Goel. 2012. Stock prediction using twitter sentiment analysis. *Standford University*.

[Miura et al.2014] Yasuhide Miura, Shigeyuki Sakaki, Keigo Hattori, and Tomoko Ohkuma. 2014. Teamx: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 628–632, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

[Mukherjee and Bhattacharyya2012] Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. Feature specific sentiment analysis for product reviews. In *Computational Linguistics and Intelligent Text Processing*, pages 475–487. Springer.

[Nielsen2011] F. Å. Nielsen. 2011. Afinn. Technical report, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, March.

[Nikolaos Malandrakis and Narayanan2014] Colin Vaz Jesse Bisogni Alexandros Potamianos Nikolaos Malandrakis, Michael Falcone and Shrikanth Narayanan. 2014. Sail: Sentiment analysis using semantic similarity and contrast. *SemEval 2014*, page 512.

[Pak and Paroubek2010] Alexander Pak and Patrick Paroubek. 2010. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Seventh International Conference on Language Resources and Evaluation, LREC 2010*.

[Rong-En Fan and Hsieh2008] Kai-Wei Chang Rong-En Fan and Cho-Jui Hsieh. 2008. Liblinear: A library for large linear classification. In *Journal of Machine Learning Research*.

[Saif Mohammad and Zhu2013] Svetlana Kiritchenko Saif Mohammad and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June.

[Tang et al.2014] Duyu Tang, Furu Wei, Bing Qin, Ting Liu, and Ming Zhou. 2014. Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 208–212, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

[Vapnik1995] Vladimir N. Vapnik. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

[Xiaodan Zhu and Mohammad2014] Svetlana Kiritchenko Xiaodan Zhu and Saif M Mohammad. 2014. Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets. *SemEval 2014*, 443.

[Xiaowen Ding and Yu2008] Bing Liu Xiaowen Ding and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240. ACM.