Proc. Natl. Acad. Sci., India, Sect. A Phys. Sci. (January–March 2015) 85(1):149–157
DOI 10.1007/s40010-014-0169-x

RESEARCH ARTICLE

Mood Swing Analyser: A Dynamic Sentiment Detection Approach

Kalyani · Ekta Gupta · Geetanjali Rathee · Pardeep Kumar · Durg Singh Chauhan

Received: 21 November 2013 / Revised: 6 May 2014 / Accepted: 10 May 2014 / Published online: 16 December 2014

Abstract This paper presents the mood swing analyzer—a novel dynamic sentiment analysis approach that determines the swings in the mood of its user by following a purely unsupervised machine learning technique. This approach uses an internal model to detect the polarity of the sentiments automatically and classify them into clusters based on K-means algorithm hence eradicating the need for normalization. In reaction to a high deviation in the users mood obtained the concept of appropriate message dropping has been proposed. Detailed algorithmic explanation along with the experimental results is well illustrated in this paper. This paper also discusses an extension of this approach in the real world to stop suicidal attempts due to cyber depression.

Keywords Mood · Swing · Unsupervised · Cyber depression · Polarity · K-means

Kalyani · E. Gupta · G. Rathee · P. Kumar (✉)
Department of CSE & ICT, Jaypee University of Information Technology, Wakanaghat, Solan, H.P., India
e-mail: pardeepkumarkhokhar@gmail.com

Kalyani
e-mail: kalyanisoni001@gmail.com

E. Gupta
e-mail: ekta.gpt@gmail.com

G. Rathee
e-mail: geetanjali.rathee123@gmail.com

D. S. Chauhan
GLA University, Mathura, India
e-mail: pdschauhan@gmail.com

Introduction

Now days, social networking is growing exponentially and has become popular around the world providing the benefit of connecting and allowing people interact around the world. Among all, facebook is the widely used application. It is basically a network of friends where people upload their photos, do comments on their friend's status and write their feelings more naturally. Facebook is the only platform where people express their emotions frequently. Such social networking sites are a boon in the way; that they never let a person feel alone and help him stay connected to his near and dear ones over ages. But the associated bane not to be ignored is increase in cyber bullying which in turn leads to depression of the victim and in severe cases even to suicides. In a research of annual bullying survey 2013, provided by the Cyber Bullying Research Centre an estimation of 7 out of 10 young people are victim of cyber bullying. 54 % of young people using facebook have reported cyber bullying over the network. Contrary to physical bullying where the victim can see the bully physically and can recognize him/her by face, cyber bullying refers to getting bullied by an anonymous or unseen person which is more frightening, discouraging and relentless and has a greater impact on the victim leading to low self esteem. Its persistence may even invoke suicidal thoughts in the mind of the victim.

Sentiment analysis is defined as the computational linguistic of emotions and user opinion in a given domain. It is also referred to a sentiment classification which is generally a judgment of opinion whether it is positive, negative or neutral. Sentiment classification is basically divided into three levels i.e. word based, sentence based, document based [1]. Here we focus on document based level. The method used to classify document based sentiment is again

categorized into two different approaches i.e. lexicon-based approach (LBA), corpus-based approach. LBA is associated with sentiment lexicon and some linguistic features while corpus-based approach is associated with machine learning techniques [2]. Each LBA and corpus-based approach has its own merits and demerits. LBA does not require any labeled training-set during initial classification of text and provide better results for less bounded domain, but the major drawback of LBA is that it is less accurate during consideration of different domains. While corpus-based approach provides better results when domains are different and fit the algorithm to the training datasets with better accuracy. The major drawback of corpus-based approach is that it may suffer over fitting to the training datasets. Researchers have proposed both lexicon-based and corpus-based approaches. Blitzer et al. [3] proposed a corpus-based approach and classified the movie reviews into three classical machine learning techniques i.e. naive bays, maximum entropy and support vector machine (SVM). The major drawback of this approach is that its sentiment classification performance is poor. Various kinds of linguistic features and classification models to improve the performance of sentiment classification have been proposed [4–7]. The weakness of all these prior approaches is that they are based on supervised learning in which each class label knows its goal attributes but suffers a problem that in case of inter-domain analysis, people have to retain the classifier again and again to adjust to the needs of domain. To overcome this drawback several LBAs have been proposed. Such as a method point wise mutual information (PMI) [8] which evaluates sentiment orientation and uses two seed words (either poor or excellent) specified by the developer which classify the documents based upon the hierarchical extraction of entities from the messages. Another approach to extract seed words automatically with a motive to make the approach unsupervised is also proposed [9, 10]. The user's sentiment using the data from facebook and twitter describes its application to adaptive e-learning [11–13] focused upon distant learning. To extract the sentiment of a user they classify the document on the basis of tweets ending. The tweet, ending with positive emotions indicate positive sentiment orientation and the tweets ending with negative emotion indicate negative sentiment orientation. The major drawback of this approach is that, collection of data was done through search queries which may be biased. To overcome this drawback, instead of using twitter, researchers took facebook as a platform for analysis where the users express their sentiments in a more effective and natural way. To extract the user sentiment using face book as a platform a LBA by using a dictionary of words, they define a classifier pipeline in which user sentiment score has been identified by passing through nine phases i.e. text pre-processing, sentence segmentation, tokenization level 1 (only white spaces are considered), emoticon detection and removal, tokenization level 2 (list of stings are considered), interjection detection, POS tagging, chunking parsing and polarity calculation and classify the user message either positive, negative or neutral and visualize different messages on a tool-kit called *sentbuk* [14]. However, it is less accurate, as it follows supervised machine learning and gives worse results during specific domain. A hybrid approach i.e. a combination of lexicon-based and corpus-based (machine learning) approach to overcome with previous drawbacks of both lexicon and corpus based approaches have been proposed [15]. In this paper we proposed an efficient and improved approach i.e. mood swing analyzer (MSA) using the cluster based unsupervised machine learning approach i.e. K-means with its experimental results.

Mood Swing Analyser

The MSA is an application that analyses the swings or changes in the mood of its user and sends a message accordingly to cheer up the user.

Basic Assumptions and Pre-requisites

In any social networking site a user needs the privacy, therefore for any application to access the user's data his permission is required.

The input of the algorithm is given in Table 1. 1. Previous year's weekly MSA result is $P(U, W)$. After the user has allowed access, we run the algorithm for a year and gather the sentiment changes on a weekly basis which is used later in the algorithm to calculate the deviations and similarity. Too large or a short intervals taken for analyzing the swings sometimes lead to false positives. Hence, optimally, we have fixed the period of data analysis as a week. 2. The raw user data that needs to be pre-processed in order to calculate the sentiment polarity. The raw data basically consists of mean of the sentiments showed by the messages published by the user (s), number of messages written (m), number of comments to the messages made (c), number of likes made to the messages on his/her wall (l), and number of likes made to the comments to messages on his/her wall (k). These data-inputs are used to calculate the $P(U, W)$. 3. The initial dictionary (I_Dict) which has been made using the lexical/corpus based methods. Here we consider some seed words both in the negative and positive corpora forming the initial dictionary. As the algorithm proceeds we use a sliding window to automatically extract and fill up the dictionary based on some negative and positive language traits. The seed words taken are generally adjectives

related with the domain which as a developer we think have more tendencies to occur. 4. The message database (MD) which consists of variety of message for various situations to send to the user. The general messages it contain are the greetings (birthdays, festivals, etc.) messages and some positive messages to cheer up the user, when his sentiment analysis results to negative. The domain of such messages is expandable and is an area of concern for future work.

Output of the MSA algorithm is deviation of the user's mood from his/her usual pattern, and a similarity graph showing the degree of similarity to his average sentiment over a time span.

This is stored in the database of MSA for sentiment prediction accuracy in future. Based upon the graphical peak points the analysis report is posted on the user's wall and a message from the application is sent to the user. The posting on the user's wall lets his/her friends know the status of user's mood. Based upon this report the friends can post messages accordingly to easy out or normalize the sentiments of the user. In addition to this, even the application itself sends an appropriate message based upon the graphical report obtained to the user's inbox. This may prove to be an effective approach in curbing the effect of negative sentiments in the long run hence easing the rate of suicidal attempts to some extents.

Table 1 Mood swing analyzer algorithm

Pre - condition: Application must have the user access permission.

Input: The required inputs for the application are:
1. Previous year's Weekly Mood Swing Analyzer result for each user (P).

2. Raw user data for pre-processing (Q).

3. Initial Dictionary for Lexicon/Corpus based analysis (I_Dict).

4. Message Database (MD).

Output: Deviation of user's mood from his/her usual pattern and similarity to the average sentiment of the user.

```
1. FOR EACH (week, W)
2.     FOR EACH (day, DY)
3.         FOR EACH (user, U)
4.             Pre-process the data (D)
5.             Apply point-wise mutual information (PMI) to
               extract tokens in hierarchical levels.
6.               FOR EACH (level, L)
7.                   Classify each token using K-means
                     machine learning approach into
                     classess given by the Internal Model
                     (IM).
8.               END FOR
9.         END FOR
10.        Calculate Cumulative Polarity.
11.    END FOR
12.    Calculate 's'.
13.    Calculate the deviation for each week as, dev_W(P, Q).
14.    Drop appropriate message from the MD.
15.    Calculate Similarity (Sim).
16. END FOR
```

Detailed Algorithmic Explanation

MSA runs for each user who has allowed access to his credentials on a weekly basis analyzing each activity for each day see Table 1. The messages posted by user are always conceived by MSA as a token which we get through tokenization using PMI explained below. Root level (level 0) contains the whole raw data which is further divided into sentences (level 1), phrases (level 2) and finally into words (level 3). The FOR loop at line 6 through line 8 runs first for level 3 and then backtracks to level 0 where we have the whole data designated as positive, negative or neutral. To analyze the mood polarity here we are using an internal model (IM). IM determines the polarity fast and automatically as it follows K-means, which is a purely unsupervised machine learning approach. Now at line 10 we calculate the cumulative polarity for each day for each user. By the end of line 11 we have categorized the whole data as positive/negative/neutral with a measure. Then, at line 12 we calculate the mean of the sentiments (cumulative polarity) for a week which gives us the value of s. At line 13 we calculate the deviation, $d_E(P, Q)$ considering the previous year's/week's vector P and current week's vector Q, given by:

$$d_E(P,Q) = \sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\cdots+(p_n-q_n)^2} = \sqrt{\sum_{i=0}^{n}(pi-qi)^2} \quad (1)$$

where both the previous year's weekly MSA result and present data are a function of the data extraction parameters (*m, c, l, k* and *s*). Deviation is calculated using the Euclidean distance formula. This gives us the fluctuation measure of the user's mood. We can even consider P vector to project just the previous week. At line 13 we are dropping a message from message database (MD) containing messages for various occasions. This message can vary from a simple birthday greeting message to a positive motivating message to lighten the user's mood in case the MSA produces a high deviation on the negative side. Contrary to $d_E(P, Q)$, at line 16 we calculate the similarity (*Sim*) given by:

$$Similarity\,(Sim) = \frac{1}{1 + d_E(P, Q)} \quad (2)$$

Similarity is a measure of the similarity in the sentiments of the user over a time period. We maintain a record of all these

Table 2 Hirarchial Tokenization Table

$Level_i$	Tokens
Level 0: (Document)	<Got my first salary today ☺.There is no way I can justify my salary level, but I'm learning to enjoy with it.>
Level 1: (Sentence)	<Got my first salary today ☺ > <There is no way I can justify my salary level, but I'm learning to enjoy with it>
Level 2: (Phrase)	<Got my first salary today ☺ > <There is no way I can justify my salary level > <but I'm learning to enjoy with it>
Level 3: (Word)	<Got > <my> <first> <salary> <today> <☺ > <There> <is> <no> <way> <I> <can> <justify> <my> <salary> <level> <but> <I> <am> <learning> <to> <enjoy> <with> <it>

Example showing the token extraction hierarchically using Point-wise Mutual Information

information about deviation and similarity so as to produce as a graphical report to the user. This can prevent cyber-depression leading to suicidal attempts, whose occurrence is on a hike now-a-days. If the report shows steep rise and fall in the graph then the user is assessed to have a high mood fluctuation. Steep rise refers to more number of messages and posts indicating that the user is too much interactive and is happy. For such instances analyzing the user's messages polarity and context, a message is dropped to its inbox congratulating him for an achieved success. On the other hand, if the graphical analysis shows a steep fall, this may indicate low interactivity due to the busy schedule of the user or negative interactivity due to cyber bullying. This is the case where user is expected to be a victim of cyber depression and hence is an area of concern. For such cases, MSA drops particular cheering up/consoling messages to the user's inbox based upon the context.

Tokenization

Data collected is first pre-processed to extract the tokens in a hierarchical way using PMI which is basically a LBA. Root level is the least tokenized level. Here, we have considered the indivisible entity at each level shown by enclosed triangular braces as a token see Table 2. At root level whole message/document is considered which is further broken into sentences by taking the full stop (.) as a separator. In any case the sentence contains phrases level 2 categorizes again considering semicolon (;) or comma (,) as the separator. Final level is the word level where space () is the separator. Table 2 shows the tokenization at each level by considering an example message by a user.

Internal Model (IM)

IM to MSA assigns the polarity to each document/message in a fast and efficient way dynamically. Implicitly it follows the principle of K-means, which is unsupervised approach to make polarity clusters by taking the language features into considerations.

As with the case of every unsupervised machine learning approach, K-means algorithm followed by IM also demands the specification of initial centroid data for each cluster taken. For sentiment analysis in MSA, we take an I_Dict containing three dictionaries with positive, negative and neutral seed words forming the cluster centroid. The same dictionary is considered here as a pre-requisite for cluster formation. Input to IM algorithm are the tokens formed hierarchically by following PMI in the step 5 of MSA algorithm. At the end this algorithm provides the user message classified into clusters with a assigned polarity to all of them. Based upon this result we calculate the mean of the sentiments shown by the messages posted by the user.

The algorithm shown in Table 3 begins with setting the clusters with the initial data from the I_Dict. Seed words form the centroid data of the clusters around which points form a constellation of sentiments known as clusters. Positive sentiment cluster is denoted by POS, negative by NEG and neutral by NEU. This algorithm is dynamic in the way that it takes into consideration the language traits. Words forming the initial cluster centroid are classified based upon the traits only. For example if bad falls in NEG the not bad should fall into POS. The dynamic approach uses the concept of sliding window to classify the document efficiently and in a faster way. Some other language traits such as transition suffix detection are also used. Once the document assigned a cluster the characteristic traits add up to the cluster providing new words for further classification hence increasing the circumference of the cluster dynamically. This simplifies the classification of later documents as the dictionary now contains more classifying words. Next, the algorithm runs for each level of token data in a bottom-up hierarchical approach i.e. first the lowest level of tokenization is attended. For each token in a level, the further step IM is given in Fig. 1.

For faster sentiment analysis, we take the language traits/features into consideration. At first we check if it contains any negative language traits then without analyzing further we update NEG cluster and then assign the document a polarity value of -1. But, if the document does not contain any negative traits, we check for if it contains any positive word (say good) or negative word. Detection of a positive word as well as a negative word is followed by tests. First one is the prefix detection test we detect the

Table 3 Internal Model Algorithm

Pre-requisite: Initial Dictionary (I_Dict)
Input: Token Dataset for initial data in clusters
Output: Cluster Classified Documents with assigned polarity.

```
1. Set 3 clusters as Positive (POS), Negative (NEG) and neutral (NEU) with
initial data from I_Dict.
2. REPEAT
3.   FOR EACH (Level)
4.     FOR EACH (Token)
5.       IF (Contain -ve Language Traits)
6.                 Update NEG.
7.       END IF
8.        ELSE IF (Contain +ve/-ve word)
9.                 Detect occurrence of negative prefix.
10.                  Detect Transition Suffix.
11.                  IF (Result is +ve)
12.                    Update POS.
13.                END IF
14.                ELSE IF (Result is -ve)
15.                   Update the NEG.
16.                END ELSEIF
17.                ELSE
18.                   Update the NEU.
19.                END ELSE
20       END ELSEIF
21.      ELSE
22.                Update the NEU.
23.      END ELSE
24.    END FOR
25.  END FOR
26.  Assign Polarity
27. UNTIL (No Change)
```

prefix of the current token for if it is any negative word (say not). For a negative word as a prefix, it toggles the polarity of the current detected token. For example good (positive sentiment) prefixed by not becomes a negative sentiment as a whole. Second, we do a transition suffix test which checks for and transition word as a suffix (say and) cancels the polarity effect of the prefix token with the suffix token and the analyzed result becomes neutral. This procedure is followed until we judge the result to be positive/negative or neutral and then accordingly we update the clusters. After the clusters have been updated a polarity value is assigned to each document as a whole. IM algorithm runs until the clusters stop updating.

The cluster categorized documents for a particular user over a span of experimental time is shown in Fig. 2. Positive cluster, POS lies in the first quadrant. Neutral cluster, NEU lies in the second and the fourth quadrant. Negative document cluster, NEG lies in the third quadrant. As we can see here, most of the dots form NEU cluster showing that usually the tokens imitate a neutral stand from the messages.

Experimental Results

Data Analysis and MSA results

To analyze the fluctuations in the mood of its user MSA calculates the deviation, $dev_W(P, Q)$ for 2 week vectors

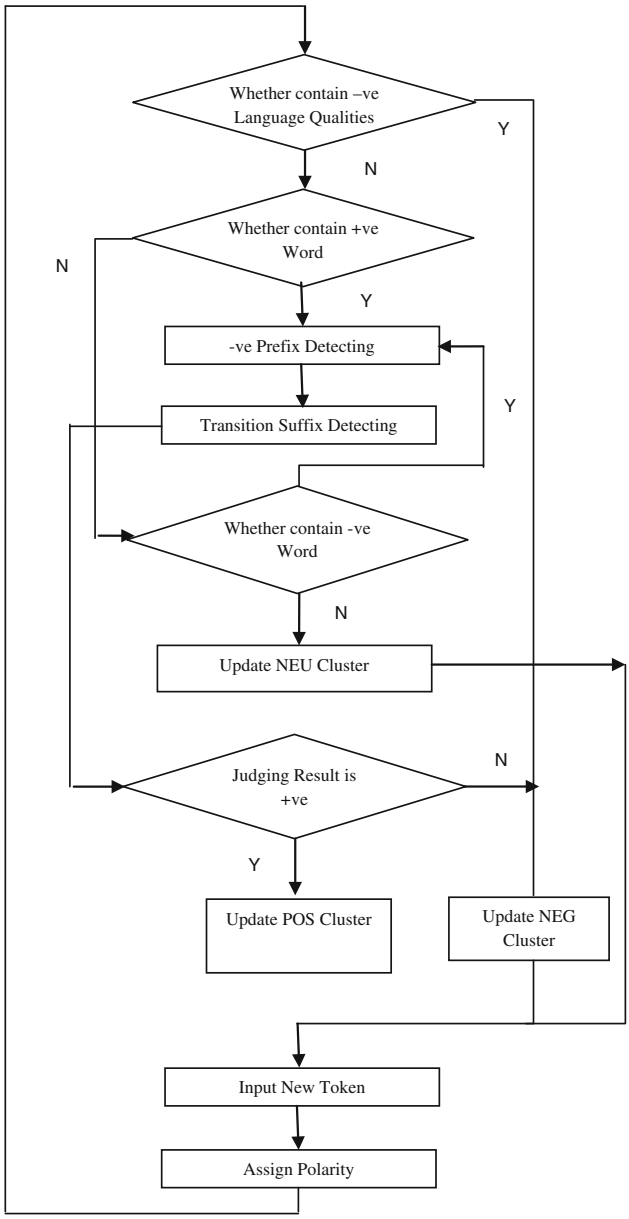

Fig. 1 Flowchart showing the step-by-step procedure of the internal model for mood swing analyzer

P and Q. For a yearly report MSA considers the first week of Q to be compared with the first week of P and so on. The algorithm may be conceived as; the user who has allowed access for MSA to run would not receive any report for a year and a week. But MSA can provide report within a week even. This way we are not constraining the user from having a weekly mood swing report. In case of any high fluctuations appropriate messages to the user is sent and the report is posted on user's wall so that his/her friends can see the deviations and hence can act accordingly.

In our experiment, we have gathered data for 16 weeks for a particular user and have shown the mood deviations and similarity report graphically. Table 4 shows the

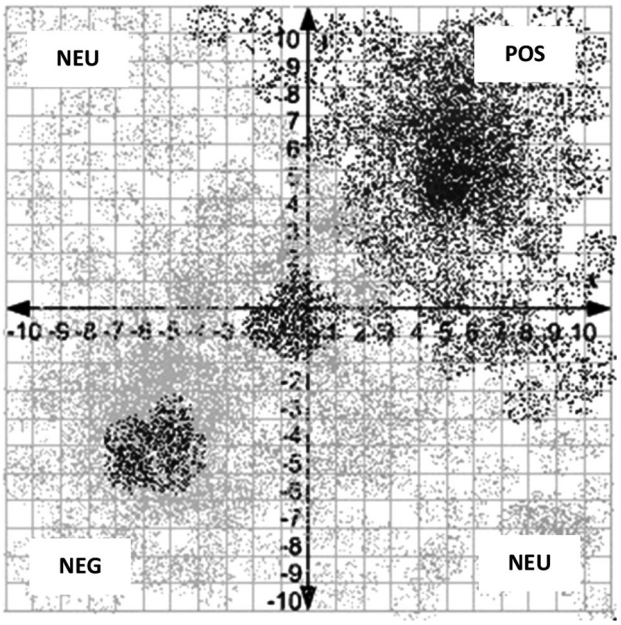

Fig. 2 Graphical representation of the user's documents in the respective clusters according to the sentiment analysis done by the IM

cumulative polarities of each day of the week which we use to compute the deviation later. Cumulative polarity refers to the sum total of all the polarities of the messages for that day by a particular user. *s* refers to the mean of the sentiments showed by the messages published by the user is calculated as the mean of the cumulative polarities of a week.

The week vectors P and Q are functions of the data parameters and are represented as:

$$P(U, W) = (s_P, m_P, c_P, l_P, k_P)$$
$$Q(U, W) = (s_Q, m_Q, c_Q, l_Q, k_Q) \quad (3)$$

Every time we take successive values of Q, the old parameters of Q are augmented with P to get the new value of P and now this P value is used to calculate the deviation for the current week Q. The augmentation refers to the mean of all the previous parameters forming a new augmented entry P_{new}. Table 5 shows the deviations and similarities of sentiments expressed by a user for experimental time span of 16 weeks. For week 1, as it is the first week of analysis, there is neither any $dev_W(P, Q)$ nor any *Sim* value that can be calculated. For successive weeks the values are calculated and shown by using Eqs. (1) and (2) for deviation and similarity respectively.

A steep fall or rise of peak represents the heightened change of emotions of the user, which indicates the occurrence of an unusual happiness in user's life whose effect gradually decreases in the weeks to come. At the end of each week according to the deviation obtained the MSA sends a message to the user. Based upon the deviation obtained, the similarity graph is plotted, using Eq. (2). Similarity (Sim) shows how similar is the user's current emotional state relative to his/her average emotional state predicted over time. Mostly the dots lie closer to each other in the range [0.03, 0.17]. Hence we fix a boundary 0.17 shown in Fig. 3, for if the dot lies above that then it is a matter of concern as user is showing a high fluctuation from his usual emotional behavior.

Table 4 Tabular representation of the day-wise cumulative polarity of each week for 16 weeks of analysis time period

$Week_i$	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
1	−2	3	4	1	0	2	−4
2	−4	2	0	0	3	4	1
3	3	2	5	1	0	0	−1
4	2	1	5	−2	−5	−2	0
5	1	3	2	5	−4	−2	0
6	−2	−3	3	0	0	4	3
7	1	1	4	−4	0	0	0
8	2	3	−3	1	0	0	−2
9	−2	−3	−1	−3	0	0	1
10	2	1	1	1	0	2	1
11	2	3	4	1	5	0	1
12	−1	−1	−1	2	0	0	2
13	4	−3	−4	−6	−2	0	0
14	2	2	1	3	0	−1	−3
15	3	2	1	3	4	−4	−3
16	2	3	−3	2	0	0	−3

Table 5 Calculated deviations and similarities on the mood of a MSA application user based upon the week vector parameters

$Week_i$	S	m	c	l	k	dev_W (P, Q)	Sim
1	.57	26	15	36	2	–	–
2	.85	14	15	17	1	22.49	.04
3	1.42	19	15	11	1	15.56	.06
4	−.14	20	15	16	4	6.07	.141
5	.714	21	17	20	3	2.56	.28
6	.714	12	27	33	2	19.17	.05
7	.286	10	11	17	4	12.06	.08
8	.14	20	13	16	3	6.96	.13
9	−1.14	15	24	2	4	20.69	.05
10	1.14	17	10	12	5	10.06	.09
11	2.71	15	13	23	3	6.88	.12
12	.142	7	16	22	6	11.55	.08
13	−1.57	30	6	14	3	17.74	.05
14	.571	17	18	19	7	5.5	.15
15	.85	25	10	20	4	10.41	.09
16	.142	19	12	23	3	6.99	.13

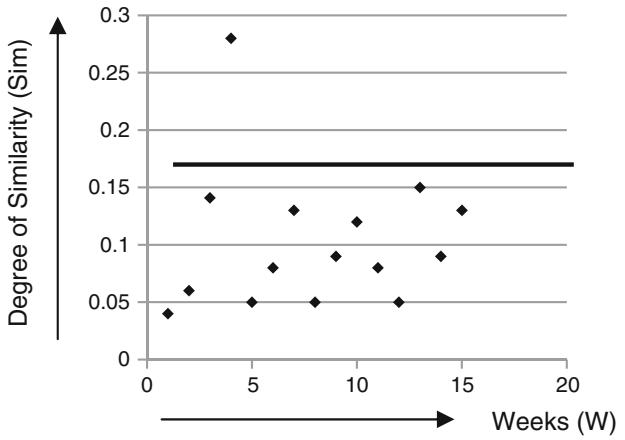

Fig. 3 Graphical representation of the mood similarity for a user over a period of 16 weeks

In this case we have one such point at week 4 and hence we prefer posting this report at user's wall and sending a warming message from the message database (MD).

Comparative Analysis

In order to evaluate MSA for its accuracy of LBA at the initial stage and later the K-means machine learning approach, the decision taken by IM, were compared with those of a human as a judge. For analyzing, we have considered only the status messages for an user, as here the users are expected to write the messages more spontaneously and naturally, unlike the greeting messages sent in the form of comments or messages to other users which at many times are just a form of commitment which are likely to hinder the performance of the MSA accuracy.

Evaluating IM approach of MSA with other Machine Learning Approaches

For a test user we have considered a pre-processed dataset of 600 messages collected over a span of 16 weeks and categorized them according to IM approach with each cluster, (POS, NEG and NEU) having some 200 messages approximately, using the Weka explorer. After classification, we used the Weka filter to generate a word vector (W_v) at level 3 tokenization, containing all the words appearing in a cluster along with the labeled cluster. (Eq. (4)):

$$W_V = [w_1, w_2, w_3, \ldots, w_n, \text{ cluster}] \quad (4)$$

We have used some basic filtering concepts here, such as converting the messages into lower class, adding the repeated words into the cluster accordingly, etc. As the number tokens obtained after filtering were very large, making it impossible to analyze further, we used the concept of correlation-based feature selection to reduce the dimensionality of the obtained token-set. For accuracy test this dataset was classified by a human user afterwards. Then we applied several machine learning approaches reconstructing IM in order to obtain an accurate classifier using the algorithms such as:

(a) J4.8 implementation of C4.5 decision tree.
(b) Naïve-Bayes.
(c) Support vector machines (SVM)
(d) K-means

Table 6 Tabular representation of the accuracy obtained using various machine learning approaches

Algorithm	Parameter	Accuracy (%)
J48	Confidence factor = 0.25	82.76
Naïve-Bayes		84.16
SVM	Kernel: radial exp ($-\gamma$ * |u − v|^2)	84.23
SVM	Kernel: sigmoide tanh($-\gamma$ *u′*v + coeff0)	84.65
K-Means	Clusters: 3, I_Dict	85.03

Table 7 Comparative analysis between the previously proposed approaches to mood swing analyzer (MSA)

P	ML approach	Trng. dataset given	Overfitting/domain Ext.	Normalization	User notification
A					
LBA	S	Y	N	Y	N
CBA	U	N	Y	N	N
PLSA	S/U	Y/N	May Be	Y	N
SWN	U	N	Y	Y	N
SB	S	Y	N	Y	N
MSA	U	N	Y	N	Y

The results we obtained are given by Table 6. Initially we attempted to analyze without pre-processing the dataset, but later on found that pre-processing proved to provide better results in a fast and efficient way, the accuracy obtained using K-means proved to be higher (85.03 %) than those using the other machine learning approaches.

Evaluating MSA approach with previously proposed approaches

Parameter based comparative analysis of the previous approaches is given in Table 7. *P* represents the parameter and *A* denotes approach. We have considered LBA which is supervised machine learning approach where it is required to specify the training dataset. LBA has another disadvantage that it cannot be extended across domains, as we will have to retain the classifiers again and again. CBA, eradicates the problem of initial specification of training dataset and cross domain extension, but may fall for over fitting, which refers to the result obtained on a favorably taken constrained data set instead of the whole data set which could have given a bad result as compared to the one obtained with this constrained/over fitted dataset. The phrase level sentiment analysis (PLSA) considered each phrase as a token to be analyzed in a supervised way. Further two more approaches SentWordNet (SWN) and SentBook (SB) were proposed which also followed supervised machine learning approach where human-intervention is required in the form of knowledge. Hence, in a supervised approach, the results obtained are inferior to the ones obtained by the unsupervised ones where least human intervention is there. Analyzing them all, finally we proposed our approach where we used K-means to detect the mood swings without the need for normalization and a user notification is generated for each noteworthy event.

Conclusion

The work described in this paper focuses on the proposed application-MSA, for detecting the mood swings of its users and generate a report for the deviations and similarity in the emotions of the user. We have used the unsupervised machine learning approach as K-means for its advantage over the supervised ones. The IM, detects the polarity of the messages reducing the need for normalization thereby increasing the efficiency of the algorithm. Algorithmic details of the proposed approach along with the experimental results show the working of the application in an efficient manner and can be useful to tackle the suicidal attempts due to cyber-depression.

References

1. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? sentiment classification using machine learning techniques. In: Proceedings of Conference on EMNLP, Philadelphia. pp 79–86
2. Engstrom C (2004) Topic dependence in sentiment classification. MPhil dissertation, University of Cambridge, United Kingdom
3. Blitzer J, Dredze M, Pereira F (2007) Biographies, bollywood, boom-boxes blenders: domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the

Association of Computational Linguistics, Prague, Czech Republic. pp 440–447
4. Pang B, Lee L (2004) A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. pp 271–278
5. Mullen T, Collier N (2004) Sentiment analysis using support vector machines with diverse information sources. In: Proceedings of EMNLP. pp 412–418
6. Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of HLTIEMNLP. pp 347–354. doi:10.3115/1220575.1220619
7. Read J (2005) Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In: Proceedings of ACLStudent Research Workshop. pp 43–48
8. Turney P (2002) Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: Proceedings of ACL. pp 417–424
9. Zagibalov T, Carroll J (2008) Automatic seed word selection for unsupervised sentiment classification of chinese text. In: Proceedings of COLING. pp 1073–1080
10. Zhen YG, Zeng N, Xu M (2010) Automatic positive sentiment word extraction for chinese text classification. In: Proceddings of Computer Design and Applications. pp V1-250-V1-255. doi: 10.1109/ICCDA.5541454
11. Esuli A, Sebastiani F (2006) SentiWordNet: a publicly available lexical resource for opinion mining. In: Proceedings of Language Resources and Evaluation (LREC). pp 417–422
12. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr 1–2:1–135
13. Pak A, Paroubek P (2009) Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the Seventh Conference on International Language Resources and Evaluation. pp 1320–1326
14. Martín JM, Ortigosa A, Carro RM (2012) SentBuk: sentiment analysis for e-learning environments. In: Proceedings of the international symposium of computer in education, pp 1–6
15. Ortigosa A, Martín JM, Carro RM (2013) Sentiment analysis in facebook and its application to e-learning. Computer in Human Behaviour 31:527–541. doi:10.1016/j.chb.2013.05.024