Multi-class sentiment analysis using a hierarchical logistic model tree approach

Masun Nabhan Homsi

Department of Computer Science and Technology Information, Simon Bolivar University,Valle Sartenejas, Baruta, Edo. Miranda - Apartado 89000, Venezuela.

Autor para correspondencia: mnabhan@usb.ve

Fecha de recepción: 21 de septiembre de 2014 - Fecha de aceptación: 17 de octubre de 2014

ABSTRACT

This paper proposes a new hybrid system for multi-class sentiment analysis based on General Inquirer (GI) dictionary and a hierarchical Logistic Model Tree (LMT) approach. This new system consists of three layers, the Bipolar Layer (BL) is of one LMT (LMT-1) for classifying sentiment polarity, while the Intensity Layer (IL) comprises two LTMs (LMT-2 and LMT3) for detecting separately three positive and three negative sentiment intensities. Only in construction phase, the Grouping Layer (GL) is used to cluster positive and negative instances by employing 2 k-means respectively. In Pre-processing phase, the raw text data is subjected to a tokenizer, a tagger, a stemmer and finally to GI dictionary to count and label only verbs, nouns, adjectives and adverbs with 24 markers that are used later to compute feature vectors. In Sentiments Classification phase, feature vectors are first introduced to LMT-1, then they are grouped in GL according to class label, afterward these groups of instances are labeled manually, and finally positive instances are introduced to LMT-2 and negative instances to LMT-3. The three trees are trained and tested on Movie Review and SenTube datasets utilizing 10-folds stratified cross validation. LMT-1 yields a tree of 48 leaves and 95 of size with 90.88% of accuracy, while both LMT-2 and LMT-3 provide two trees of 1 leaf and 1 of size with 99.28% and 99.37% of accuracy respectively. Experiments show that the proposed hierarchical classification methodology gives a better performance compared to other prevailing approaches.

Keywords: Multi-class sentiments analysis, hybrid approach, logistic model tree, general inquirer dictionary (GI).

RESUMEN

En este trabajo se propone un nuevo sistema híbrido para el análisis de sentimientos en clase múltiple basado en el uso del diccionario General Inquirer (GI) y un enfoque jerárquico del clasificador Logistic Model Tree (LMT). Este nuevo sistema se compone de tres capas, la capa bipolar (BL) que consta de un LMT (LMT-1) para la clasificación de la polaridad de sentimientos, mientras que la segunda capa es la capa de la Intensidad (IL) y comprende dos LMTs (LMT-2 y LMT3) para detectar por separado tres intensidades de sentimientos positivos y tres intensidades de sentimientos negativos. Sólo en la fase de construcción, la capa de Agrupación (GL) se utiliza para agrupar las instancias positivas y negativas mediante el empleo de 2 k-means, respectivamente. En la fase de Pre-procesamiento, los textos son segmentados por palabras que son etiquetadas, reducidas a sus raíces y sometidas finalmente al diccionario GI con el objetivo de contar y etiquetar sólo los verbos, los sustantivos, los adjetivos y los adverbios con 24 marcadores que se utilizan luego para calcular los vectores de características. En la fase de Clasificación de Sentimientos, los vectores de características se introducen primero al LMT-1, a continuación, se agrupan en GL según la etiqueta de clase, después se etiquetan estos grupos de forma manual, y finalmente las instancias positivas son introducidas a LMT-2 y las instancias negativas a LMT-3. Los tres árboles están entrenados y evaluados usando las bases de datos Movie Review y SenTube con validación cruzada estratificada de 10-pliegues. LMT-1 produce un árbol de 48 hojas y 95 de tamaño, con 90,88% de exactitud, mientras que tanto LMT-2 y LMT-3 proporcionan dos árboles de una hoja y uno de tamaño, con 99,28% y 99,37% de exactitud, respectivamente. Los experimentos muestran que la metodología de clasificación jerárquica propuesta da un mejor rendimiento en comparación con otros enfoques prevalecientes.

Palabras clave: Análisis de sentimientos en clase múltiple, enfoque híbrido, logistic model tree, diccionario general inquirer (GI).

1. INTRODUCTION

Sentiment Analysis (SA) or Opinion Mining (OM) systems are employed to help institutions, organizations or companies to track automatically their clients' sentiments from a massive raw of data found on their websites and social media platforms. Two main tasks are involved in sentiment analysis, classifying a polarity of a given text, or determining if it is subjective or objective. These tasks could be carried out on three levels; document level, sentence level, and aspect level (Vinodhini & Chandrasekaran, 2012). There exist two main approaches to the problem of extracting sentiment automatically: The lexicon-based approach and the Machine Learning-based (ML) approach. The former utilizes predefined dictionaries of words annotated with their semantic orientation, that are used to calculate sentiment polarity of a text, the latter approach, first extracts feature vectors from a set of data labeled as positive or negative, then these vectors are classified employing one of the supervised ML algorithms, finally the new classifier is used to predict a class for unseen data. A hybrid approach can take advantage of both approaches in order to improve the overall SA system (Pang & Lee, 2008). N-grams and their frequency, Part Of Speech (POS) information, negation and opinion words are considered the most extensively used features in ML sentiment classification. Many ML techniques like Naive Bayes (NB), Support Vector Machines (SVM), K-Nearest Neighbor (KNN), Maximum Entropy (ME), Decision Tree and rule learner are used vastly to build sentiment classifiers (Pang & Lee, 2008).

Most of approaches in SA field deal with polarity classification, ignoring the degree of users' satisfaction or dissatisfaction about any resource found on social media sites. This fact leads sometimes other users to take inaccurate decisions on the same resources, e.g. when they buy a product, watch a video or take a course. Detecting several sentiment intensities in a text will help those users to access valuable information, improving their choice and the quality of their decisions. In this context, a Multi-Class Sentiment Analysis System (MCSAS) is built to extract automatically sentiments, expressed in six dimensions, out from English text. These dimensions are: High Positive (HP), Positive (P), Low Positive (LP), Low Negative (LN), Negative (N) and High Negative (HN).

This paper has three objectives. The first objective is to describe the steps taken to build MCSAS for document classification using General Inquirer (GI) Dictionary and Logistic Model Tree (LMT) algorithm. The second objective is to measure the effectiveness of feature selection on the overall performance of sentiment classification. Finally, the third objective is to compare the new system experimental results with other research results.

GI is a computational lexicon compiled from several sources, including Harvard IV-4 dictionary and Lasswell value dictionary. It contains information about English word senses, including tags that label them in 182 categories Inkpen *et al.* (2005). There are labels for positive and negative words; labels for words of pleasure, pain, virtue, and vice; labels for words indicating overstatement and understatement; labels for words of negation and interjections; etc. Inkpen *et al.* (2005).

LMT is a supervised machine learning algorithm that combines a standard decision tree with Logistic Regression (LR) functions at the leaves. LogitBoost is employed to produce a LR at every node in the tree; the node is then split using an attribute value test. Each LogitBoost invocation is warm-started from its results in the parent node. Cross validation is used to determine the appropriate number of iterations to run. Once the tree has been built it is pruned using CART-based pruning Landwehr *et al.* (2003).

In this research, two datasets are combined to be used together as a single dataset, the Movie Review Dataset and the SenTube Dataset. Movie Review Dataset contains movie reviews along with their associated binary sentiment polarity labels split into 1000 positive and 1000 negative documents

Pang *et al.* (2002). SenTube is a dataset of user-generated comments on YouTube videos annotated for information content and sentiment polarity Uryupina *et al.* (2014). The dataset covers English and Italian videos on the same products (automobiles, tablets). Sentiment is divided towards the product and towards the video. 117 English videos are labeled as positive, 61 as negative and 39 as neutral. Each text document in the Combined Dataset (CD) is pre-processed to extract 24 features using GI dictionary. These features are: Positiv, Negativ, Pstv, Affil, Negtv, Hostile, Strong, Weak, Submit, Active, Passive, Pleasur, Pain, Feel, Arousal, Emot, Virtue, Vice, Ovrst, Undrst, Yes, No, Negate, and Intrj (Landwehr *et al.,* 2003). First of all, the first LMT (LMT-1) is trained and tested to model sentiment polarity.

Then, positive and negative instances are grouped separately by 2-Kmeans, each one yields 3 clusters to represent three different sentiment intensities. Finally, two LMTs, LMT-2 and LMT-3, are also trained and tested to model 3 positive sentiment intensities (HP, P, LP) and other 3 negative sentiment intensities (HN, N, LN) respectively. The new system consists of three layers; Bipolar Layer (BL), Grouping Layer (GL) and Intensity Layer (IL). LMT-1 is located in IL, k-means in GL and LMT-2 and LMT-3 in IL.

The rest of the paper is organized as follows: Section 2 presents related works in sentiment analysis. The methodology and the proposed approach architecture are described in section 3. Section 4 discusses the experiments' results and compares them with other studies related to hybrid sentiment analysis approach. Finally the conclusions and future work are given in section 5.

2. RELATED WORK

Early works of sentiment classification mainly focus on polarity detection of English product reviews or movie reviews. Pang *et al.* (2002) examined the effectiveness of NB, ME, SVM for the sentiment classification of movie reviews. Unigrams (with negation tagging) and bigrams were employed as features. SVM yielded the best results, with 82.9% of accuracy, using unigrams with binary weighting indicating the presence or absence of a feature.

This accuracy was further increased in their later work (Pang & Lee, 2004) to 87.2% by detecting subjectivity before classification step and removing objective text.

Kennedy & Inkpen (2006) present two methods for determining the sentiment expressed by a movie review. The first method uses GI to classify customers' reviews based on the number of positive and negative terms they contain, as well as negations, overstatements and understatements. Negations are used to reverse the semantic polarity of a particular term, while intensifiers and diminishers are used to increase and decrease, respectively, the degree to which a term is positive or negative. The second method uses a ML algorithm, SVM. Authors start with unigram features and then add bigrams that consist of a valence shifter and another word. The accuracy of classification is very high, and the valence shifter bigrams slightly improve it. They also demonstrate that combining the two methods achieves better results than either method alone.

The work in Prabowo & Thelwall (2009) combines rule-based classification, supervised learning and machine learning into a new method. This method is tested on movie reviews, product reviews and MySpace comments. The procedure is that if one classifier fails to classify a document, the classifier will pass the document onto the next classifier, until the document is classified or no other classifier exists. A number of approaches that focus on acquiring and defining a set of rules are used along with SVM learning: General Inquirer Based Classifier (GIBC), Rule-Based Classifier (RBC), Statistics Based Classifier (SBC) and Induction Rule Based Classifier (IRBC). Experiments were carried out by applying different sequences of previous classifiers. Results showed that the use of multiple classifiers in a hybrid manner can improve the effectiveness of sentiment analysis. The sequence RBC->SBC-> GIBC->SVM of classifiers yielded the highest accuracy in comparison to other classifiers' sequences.

Cassinelli & Chen (2009) address the problem of categorizing documents by overall sentiment into two classes (positive or negative) and into multiple classes (one to five stars). They use three sets of vocabulary. The first set consists of the positive and negative words from the GI. The second set of words includes the most frequent words in the whole Movie Review dataset and the third set consists of words learned from applying a boosting classifier to every word in the dataset. For polarity classification and using the first set of vocabulary with SVM, they achieved an accuracy of 73.8%, but when they run the second and the third sets of features together with SVM and Boosting algorithms, they got higher accuracies of 83.55 and 82.95 respectively. For multi-class sentiment classification, they used multi-class decision tree and they got low accuracy.

There are few studies on classifying documents into multi-class sentiment in a hybrid manner and using GI dictionary on Movie Review dataset, therefore, this is one of the contribution of this paper. However, the work in Wilson *et al.* (2004) is somewhat related to this paper, where the authors present experiments to classify the strength of the opinions and emotions being expressed in individual clauses, using boosting, rule learning and support vector regression with a wide range of features, including new syntactic features. Results show that boosting algorithm achieves improvements in accuracy ranging from 23% to 79% and for vector regression an improvements ranging from 57% to 64% over baseline.

An additive model, which uses weighted polarity lexicon, is proposed in another similar study (Pandey, 2011). It works well for binary opinion classification and fails to produce impressive results for multi-class opinion classification. The authors also use SVM for the same task; the results obtained by SVM for multi-class classification are also very low on accuracy as compared to SVM's result on binary classification.

The Machine Learning and NLP group at the University of Trento presents a systematic study on SA from SenTube dataset Uryupina *et al.* (2014) by training a set of supervised multi-class classifiers distinguishing between video and product related opinions Severyn *et al.* (2014). They use standard feature vectors augmented by shallow syntactic trees enriched with additional conceptual information.

3. METHODOLOGY

MCSAS is built to exploit sentiment information in English text. Fig. 1 illustrates a general block diagram of the work flow employed to construct the new multi-class sentiment classifier. The methodology consists of two phases: Pre-Processing phase and Classification phase.

3.1. Pre-processing phase

The primary objective of this phase is to extract feature vectors (FVs) out of documents found in the CD={d_1, d_2, d_3,…,d_n}, where n is the total labeled documents in the dataset. Each FV_i covers the occurrence or the total of different word senses in d_i (where i=1,..,n). In order to accomplish this task, the following algorithm is applied:

1) Translate slang words, acronyms, abbreviations and emoticons found in the raw text di, into their original meanings. Internet Slang Dictionary (ISD)[1] is employed to perform this task and it contains 5380 terms originated from various sources including Bulletin Board, AIM, Yahoo, Twitter, YouTube, Chat Room and others.
2) Search contracted words in d_i and replace them with their original form.
3) Tag each word using Part-Of-Speech Tagger (POS-Tagger). Words tagged as IN (prepositions), DT (determiners), PRP (Personal pronoun), MD (Modal), WRV (Wh-adverb), NNP (Proper noun), EX (Existential there) or WP (Wh-pronoun) are ignored and considered as stop words. A tagged word is represented by (w_j, t_j), where t_j is the lexical category of the word w_j and j=1,.., k.

[1] http://www.internetslang.com

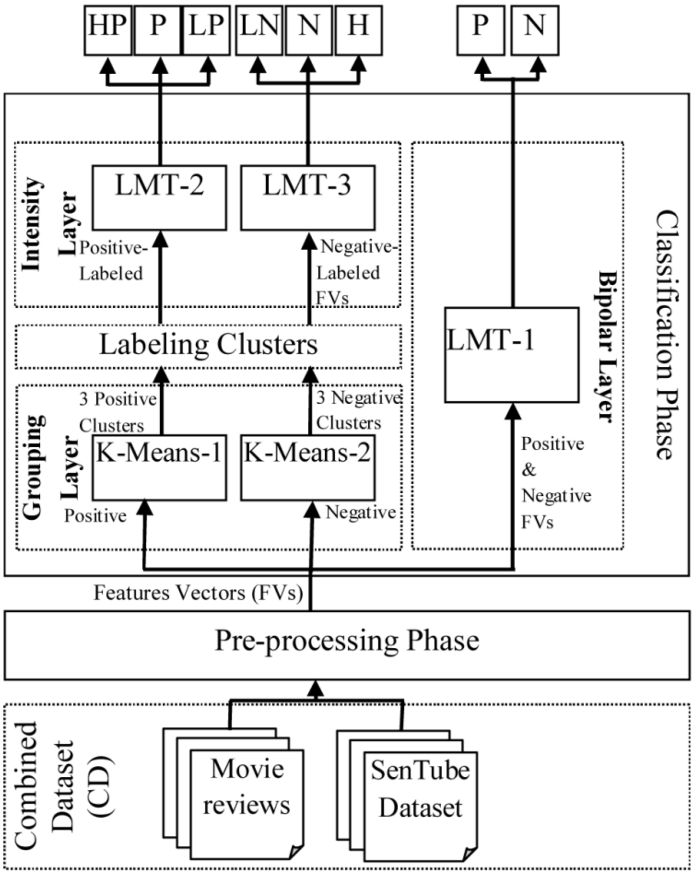

Figure 1. Architecture of MCSAS.

4) Repeat steps from 6 to 16 for each tagged word (w_j, t_j).

5) Tokenize the text by breaking it up into words. d_i={w_1, w_2, w_3, … , w_k}, where k is the number of token in the document.

6) Search the word w_j in GI dictionary to find its different senses.

7) If w_j is found, go to step 9; otherwise stem it to find its root. This is done using Porter Stemming algorithm (Porter, 1980).

8) Search the stemmed word w_j in GI dictionary with its corresponding tag, to avoid ambiguity in word senses.

9) If the stemmed word is found in GI, create the word senses vector W_jS={$Positiv_j$, $Negativ_j$, $Pstv_j$, $Affil_j$, $Negtv_j$, $Hostile_j$, $Strong_j$, $Weak_j$, $Submit_j$, $Active_j$, $Passive_j$, $Pleasur_j$, $Pain_j$, $Feel_j$, $Arousal_j$, $Emot_j$, $Virtue_j$, $Vice_j$, $Ovrst_j$, $Undrst_j$, Yes_j, No_j, $Negate_j$, $Intr_j$, NW_j} and go to step 10; otherwise search its synonym in Wordnik Dictionary[2] and then go to 6. NW indicates if w_j is an explicit Negation Word. The list of negation words used in this research is:”*no*, *not*, *none*, *no one*, *nobody*, *nothing*, *neither*, *nowhere* and *never*” and it is represented by N. W_jS is a binary vector to indicate the presence or the absence of a sense. At this point, words not found in GI Dictionary are considered noise and they are ignored. Additionally, non-English words are also discarded; therefore, language detection procedure is not needed.

10) Calculate Positive Polarity (PP_i) by summing up the number of occurrence of the following senses: Positiv, Pstv, Affil and Yes in d_i. This is reflected from (1) to (5).

$$PP_i=Positiv_i+Pstv_i+Affil_i+Yes_i \tag{1}$$

[2] http://www.wordnik.com

where:

$$\mathrm{Positiv_i}=\begin{cases}\mathrm{Positiv_i}+1, & \text{if } (\mathrm{Positiv_j}=\text{true})\\ \mathrm{Positiv_i}, & \text{otherwise}\end{cases} \tag{2}$$

$$\mathrm{Pstv_i}=\begin{cases}\mathrm{Pstv_i}+1, & \text{if } (\mathrm{Pstv_j}=\text{true and } \mathrm{Positiv_j}=\text{false}\\ & \text{and } \mathrm{Affil_j}=\text{false and } \mathrm{Yes_j}=\text{false})\\ \mathrm{Pstv_i}, & \text{otherwise}\end{cases} \tag{3}$$

$$\mathrm{Affil_i}=\begin{cases}\mathrm{Affil_i}+1, & \text{if } (\mathrm{Affil_j}=\text{true and } \mathrm{Positiv_j}=\text{false}\\ & \text{and } \mathrm{Pstv_j}=\text{false and } \mathrm{Yes_j}=\text{false})\\ \mathrm{Affil_i}, & \text{otherwise}\end{cases} \tag{4}$$

$$\mathrm{Yes_i}=\begin{cases}\mathrm{Yes_i}+1, & \text{if } (\mathrm{Yes_j}=\text{true and } \mathrm{Positiv_j}=\text{false}\\ & \text{and } \mathrm{Pstv_j}=\text{false and } \mathrm{Affil_j}=\text{false})\\ \mathrm{Yes_i}, & \text{otherwise}\end{cases} \tag{5}$$

Compute Negative Polarity ($\mathrm{NP_i}$) by summing up the number of occurrence of the senses: Negativ, Negtv, Hostile, No and Negate in d_i. This is reflected by (6):

$$\mathrm{NP_i}=\mathrm{Negativ_i}+\mathrm{Negtv_i}+\mathrm{Hostile_i}+\mathrm{No_i}+\mathrm{Negate_i} \tag{6}$$

where:

$$\mathrm{Negativ_i}=\begin{cases}\mathrm{Negativ_i}+1, & \text{if } (\mathrm{Negativ_j}=\text{true})\\ \mathrm{Negativ_i}, & \text{otherwise}\end{cases} \tag{7}$$

$$\mathrm{Negtv_i}=\begin{cases}\mathrm{Negtv_i}+1, & \text{if } (\mathrm{Negtv_j}=\text{true and } \mathrm{Negativ_j}=\text{false}\\ & \text{and } \mathrm{Hostile_j}=\text{false and } \mathrm{No_j}=\text{false}\\ & \text{and } \mathrm{NW_j}\notin \mathrm{N} \text{ and } \mathrm{Negate_j}=\text{false})\\ \mathrm{Negtv_i}, & \text{otherwise}\end{cases} \tag{8}$$

$$\mathrm{Hostile_i}=\begin{cases}\mathrm{Hostile_i}+1, & \text{if } (\mathrm{Hostile_j}=\text{true and } \mathrm{Negativ_j}=\text{false}\\ & \text{and } \mathrm{Negtv_j}=\text{false and } \mathrm{No_j}=\text{false}\\ & \text{and } \mathrm{NW_j}\notin \mathrm{N} \text{ and } \mathrm{Negate_j}=\text{false})\\ \mathrm{Hostile_i}, & \text{otherwise}\end{cases} \tag{9}$$

$$\mathrm{No_i}=\begin{cases}\mathrm{No_i}+1, & \text{if } (\mathrm{No_j}=\text{true and } \mathrm{Negativ_j}=\text{false}\\ & \text{and } \mathrm{Negtv_j}=\text{false and } \mathrm{Hostile_j}=\text{false}\\ & \text{and } \mathrm{NW_j}\notin \mathrm{N} \text{ and } \mathrm{Negate_j}=\text{false})\\ \mathrm{No_i}, & \text{otherwise}\end{cases} \tag{10}$$

$$\mathrm{Negate_i}=\begin{cases}\mathrm{Negate_i}+1, & \text{if } (\mathrm{Negate_j}=\text{true and } \mathrm{Negativ_j}=\text{false}\\ & \text{and } \mathrm{Negtv_j}=\text{false and } \mathrm{Hostile_j}=\text{false}\\ & \text{and } \mathrm{NW_j}\notin \mathrm{N} \text{ and } \mathrm{No_j}=\text{false})\\ \mathrm{Negate_i}, & \text{otherwise}\end{cases} \tag{11}$$

11) Count the frequency of explicit negation words ($\mathrm{NW_i}$) as in (12):

$$\mathrm{NW_i}=\begin{cases}\mathrm{NW_i}+1, & \text{if } (\mathrm{NW_j}\in \mathrm{N})\\ \mathrm{NW_i}, & \text{otherwise}\end{cases} \tag{12}$$

12) Search in a window of 2 words to detect Positive Polarity Shifter (PPS_i) and the Negative Polarity Shifter (NPS_i) as follows:

$$PPS_i=\begin{cases} PPS_i+1, & \text{if } \left(NW_{j\text{-}1} \text{ and } Negativ_j\right) \text{ OR } \left(NW_{j\text{-}1} \text{ and } Negtv_j\right) \\ PPS_i, & \text{otherwise} \end{cases} \tag{13}$$

$$NPS_i=\begin{cases} NPS_i+1, & \text{if } \left(NW_{j\text{-}1} \text{ and } Positiv_j\right) \text{ OR } \left(NW_{j\text{-}1} \text{ and } Pstv_j\right) \\ NPS_i, & \text{otherwise} \end{cases} \tag{14}$$

13) Calculate the Total Positive (TP_i) and the Total Negative (TN_i) of d_i by applying (15) and (16):

$$TP_i=PP_i+PPS_i \tag{15}$$

$$TN_i=NP_i+NPS_i \tag{16}$$

14) Calculate potency senses as follows:

$$Strong_i=\begin{cases} Strong_i+1, & \text{if } (Strong_j=\text{true or } Pow_j=\text{true }) \\ Strong_i, & \text{otherwise} \end{cases} \tag{17}$$

$$Weak_i=\begin{cases} Weak_i+1, & \text{if } (Weak_j=\text{true}) \\ Weak_i, & \text{otherwise} \end{cases} \tag{18}$$

$$Submit_i=\begin{cases} Submit_i+1, & \text{if } (Submit_i=\text{true and } Weak_j=\text{false }) \\ Submit_i, & \text{otherwise} \end{cases} \tag{19}$$

15) Count the number of occurrence for the reminder senses using the following generic equation:

$$Sense_i=\begin{cases} Sense_i+1, \ \& & \text{if } (Sense_j=\text{true}) \\ Sense_i, & \text{otherwise} \end{cases} \tag{20}$$

Where $Sense_i$ could be: $Ovrst_i$, $Undrst_i$, $Active_i$, $Passive_i$, $Pleasur_i$, $Pain_i$, $Feel_i$, $Arousal_i$, $Emot_i$, $Virtue_i$, $Vice_i$ or $Intrj_i$.

16) Calculate the percentage of each feature by dividing it by the number of words found in GI. The dimension of the final feature vector is 19 of length to represent one document d_i in CD: FV_i={TP_i, TN_i, $Strong_i$, $Weak_i$, $Submit_i$, $Active_i$, $Passive_i$, $Pleasur_i$, $Pain_i$, $Feel_i$, $Arousal_i$, $Emot_i$, $Virtue_i$, $Vice_i$, $Ovrst_i$, $Undrst_i$, $Intrj_i$, NW_i, $Class_i$}. $Class_i$ represents binary sentiment polarity labels: Positive or Negative.

17) Repeat steps from 1 to 17, if there are more documents to pre-process in CD.

18) CD contains 2178 documents split into 1117 positive and 1061 negative. Due to this imbalanced class distribution, three re-sampling techniques were applied, respectively: Re-sampling without replacement, SMOT and Under-Sampling (Witten *et al.*, 2005). The new dataset CD1 contains 2114 vectors, where each 1107 belong to a sentiment class.

3.2. Sentiment classification phase

In this phase a hierarchical LMT (HLMT) is built to discover multi-class sentiment in English text. It consists of three layers; the first layer is called Bipolar layer (BL) and it uses the first LMT (LMT-1) for detecting positive and negative sentiments. The second layer is called Grouping Layer (GL), where positive and negative instances are grouped separately by 2 k-means, 3 clusters for positive instances and other 3 for negative instances. Instances in each cluster are then labeled manually. GL is only used in construction phase of MCSAS. The third layer is called Intensity Layer (IL) and it utilizes two LMTs (LMT-2, LMT-3) to determine sentiment strength expressed in six levels (three for each LMT).

The three LMTs are trained and tested utilizing 10-folds stratified cross validation. The following algorithm summarizes the steps followed to construct the HLMT.

1) Present the dataset CD_1.
2) LMT-1 is trained and tested to model positive and negative instances in dataset CD_1.
3) CD_1 is partitioned into two sub-datasets CD_{11} and CD_{12} for positive and negative instances respectively.
4) Instances in CD_{11} are grouped in three clusters employing K-means algorithm.
5) The 3 generated clusters for positive dimensions (HP, P and LP) are labeled manually and then instances in CD_{11} are saved in a new sub-dataset CD_P.
6) Instances in CD_{12} are also grouped in three clusters employing K-means algorithm.
7) The 3 generated clusters for negative dimensions (HN, N and LN) are labeled manually and then instances in CD_{12} are saved in a new sub-dataset CD_N.
8) LMT-2 is trained and tested to model positive intensities in CD_P.
9) LMT-3 is trained and tested to model negative intensities in CD_N.

4. RESULTS AND DISCUSSIONS

Several experiments are conducted to evaluate the effectiveness of the new approach MCSAS. They are performed in two different sets. In the first line of experiments, the performance of the proposed feature extraction algorithm is evaluated and compared with TF*IDF algorithm employing six different ML algorithms to detect sentiment polarity in BL. In the second line of experiments, the accuracy of the LMT-2 and LMT-3 in IL is measured to detect sentiment intensities reflected in six classes. Experimental evaluation results are presented in the following subsections. Furthermore, this section compares the new proposed approach with the existing studies in SA.

4.1. Feature extraction and bipolar sentiments

Six ML algorithms: NB, SVM, KNN, Jrip, J48 and LMT are trained and tested, with the objective to get the best bipolar sentiment classification. Results are compared with another method of feature vector extraction, that is TF*IDF with unigram. TF*IDF method represents the importance of a term for a document in a specific corpus (Witten *et al*., 2005). The accuracy of these algorithms using GI or TF*IDF is shown in Table 1.

Table 1. Comparison of ML algorithms accuracy for sentiment polarity classification in BL.

Feature extraction	NB	SVM	KNN	JRip	J48	LMT-1
GI	66.80 %	72.22 %	<u>91.86%</u>	83.74 %	88.66 %	<u>90.88 %</u>
TF*IDF	76.45 %	76.75 %	52.80 %	65.93 %	64.10 %	65.70%

As can be observed, the proposed feature method using GI yields, in general, the best results when compared with all of other methods. NB with TF*IDF gives a higher accuracy than when it runs with GI. Although KNN yields higher accuracy of 91.86 %, LMT is considered, in this research, the best choice for bipolar sentiment classification with 90.88% of accuracy, this is due to that LMT uses linear functions for predicting new instances, while, KNN makes prediction based on the entire training dataset, so its space complexity is represented as O(p*n), where p is the number of features and n is the number of training examples. The generated tree of LMT-1 is of size 95 with 48 leaves (rules). All ML algorithms are performed with 10-folds stratified cross validation.

The confusion matrix of the classification accuracy for LMT-1 is given in Table 2, where 2012 instances (996 positive and 1016 negative) are correctly classified and only 202 instances (111 positive and 91 negative) are misclassified.

Table 2. Confusion matrix of LMT-1 for bipolar layer.

	Positive	Negative
Positive	996	111
Negative	91	1016

4.2. Sentiment intensities

To detect sentiment intensities, the dataset CD_1 is divided into two sub-datasets of 1107 (CD_{11} and CD_{12}) instances based on class labels. Instances in both sub-datasets are partitioned separately in three clusters using 2 K-means algorithms. The distribution of instances over clusters is illustrated in Table 3. 19% of them are classified as HP, while 51% and 30% as P and LP intensities. Negative instances are also distributed as follows: 21%, 47% and 32% to denote HN, N and LN intensities respectively. Labels are assigned manually to each cluster. Thus the new labeled sub-datasets are CD_P and CD_N.

Table 3. Distribution of positive and negative instances over clusters in GL.

Dataset	Cluster 0	Cluster 1	Cluster 2
CD_{11} (Positive)	208 (19%)	565 (51%)	334 (30%)
CD_{12} (Negative)	228 (21%)	521 (47%)	358 (32%)

Both sub-datasets (CD_P, CD_N) are introduced to NB, SVM, KNN, Jrip, J48 and LMT algorithms separately, which are also trained and tested using 10-folds stratified cross validation. Table 4 shows that LMT-2 and LMT-3 exhibit the highest accuracies with 99.28 % and 99.37 % in comparison to other models. They yield 2 trees of size 1 with 1 leave, that is, in prediction phase; only six linear functions are used to classify multi-class sentiments expressed in English text, where every three functions belong to a sentiment pole.

The confusion matrix of both trees is reflected in Tables 5 and 6. For P class, 560 out of 565 are classified correctly. The rest are classified incorrectly as 3 HP and 2 LP, and so for HN class, where 225 out of 228 are classified correctly and the rest are misclassified as N and LN. Thus the per class accuracies are : 99.03% (206/208) for HP, 99.11% (560/565) for P, 99.70% (333/334) for LP, 99.11% (225/228) for LN, 98.88% (354/358) for N and 100% (521/521) for HN. What the confusion matrix and the accuracies demonstrate is that the two LMTs (LMT-2 and LMT-3) perform very high in classifying positive and negative intensities respectively.

Table 4. Comparison of ML algorithms accuracy for sentiment intensity classification.

Dataset	NB	SVM	KNN	JRip	J48	LMT-2/ LMT-3
CD_P (Positive)	93.32 %	97.56%	93.50%	95.48 %	95.03%	99.28 %
CD_N (Negative)	92.14 %	98.28 %	92.77%	93.95%	94.40 %	99.37 %

Table 5. Confusion matrix of LMT-2 for positive intensities.

	HP	P	LP
HP	206 (99.03%)	2	0
P	3	560 (99.11%)	2
LP	1	0	333 (99.70%)

Table 6. Confusion matrix of LMT-3 for negative intensities.

	HP	P	LP
HP	225 (99.11%)	1	2
P	1	354(98.88%)	3
LP	0	0	521(100%)

4.3. Comparison and analysis

The combination of GI dictionary with KNN or LMT for sentiment polarity classification, achieves a high accuracy that reaches to 91.86 and 90.88% respectively. These results are considered one of the best in comparison with other sentiment classifiers. This is shown in Table 7, which compares the proposed algorithm with the studies discussed earlier in section 2. The three studies, Kennedy & Inkpen (2006), Prabowo & Thelwall (2009) and Cassinelli & Chen (2009) share many characteristics with the present one for polarity classification. They are hybrid classifiers for document level sentiment analysis, which use GI Dictionary with supervised machine learning algorithms and Movie Review Dataset.

The study Kennedy & Inkpen (2006) and the present paper take negation words, intensifiers and diminishers into account, but the use of a wider range of features with LMT performs better than the approach based on SVM with less features.

The work most closely related to the present one is Severyn *et al.* (2014), where sentiment polarity is determined by using sentiment lexicon and a supervised ML algorithm,. Both studies use SenTube dataset, but the present work shows a higher accuracy that reaches to 90.88% and it also uses a hierarchical architecture of LMT to detect sentiment intensities in English text.

The proposed hierarchical architecture of LMT has helped to get a high accuracy in multi-class sentiment classification, this is due to the fact that the classification process is done in two stages, polarity text is first determined in BL and then its intensity in IL. Unlike the studies presented in Wilson *et al.* (2004) and Pandy (2011), where classification is carried out directly after feature extraction phase. Table 8 exhibits a comparison among the studies presented in Section 2 for multi-class sentiment classification.

Table 7. Comparison of hybrid sentiment classifiers.

Ref.	Dataset	Feature Selection	Technique	Accuracy %
BL	Movie Reviews+ SenTube	24 word senses in GI Dictionary + Handling Negation	LMT	90.88
			KNN	91.86
Prabowo *et al.* (2009)	Movie Reviews+ MySpace	RBC+SBC+GIBC	SVM	90.45
Kennedy & Inkpen (2006)	Movie review	4 word senses in GI and CTRW Dictionaries + Unigram + Bigrams +Handling Negation	SVM	85.9
Cassinelli & Chen (2009)	Movie Reviews	Frequency of positive and negative word senses from GI	SVM	73.8
Severyn *et al.* (2014)	SenTub.	STRUCT Model: a shallow syntactic tree	SVM	70.5

Table 8. Comparison of multi-class sentiment classifiers.

Ref.	Dataset	Classes	Technique	Accuracy Average %
IL	Movie Reviews	6	2 LTM	99.30
Pandy (2011)	Multi-Domain Sentiment Dataset	4	SVM	71.73
Wilson *et al.* (2004)	MPQA (Multi-perspective Question Answering) Opinion Corpus	4	BoosTexter	57.52
			Ripper	55.55
			Support Vector Regression	36.53

4. CONCLUSIONS

The new proposed approach aims at creating automatic multi-class sentiment analysis system which uses GI dictionary with a hierarchical architecture of LMT. Experiments have demonstrated that the combination of a wide list of features with two-stage classification method has got high accuracies reaching 90.88% for polarity sentiment, 99.28% for positive sentiment intensities and 99.37% for negative sentiment intensities. Having such reliable automatic sentiment analysis system will allow organizations to track users' opinions on their social media sites and provide them insightful business intelligence using which they can take impactful decisions that would leverage their business.

There are several directions for future work. The first direction is to study the performance of the proposed system on other datasets described in the sentiment analysis literature and on multilingual data from social media. The second direction is to investigate the influence of incorporating subjective and objective features to improve the overall classification accuracy. The third direction is to use the new system for building a sentiment – based search engine for YouTube videos. Finally, the fourth direction is to find the relationship between users' personality and their sentiments on social media to recommend them appropriate resources adapted to their needs and interests.

ACKNOWLEDGMENT

This work has been supported by the Decanato de Investigación y Desarrollo (DID) from Simon Bolivar University (USB).

REFERENCES

Cassinelli, A., C.W. Chen, 2009. *CS224N Final Project Boost up! Sentiment categorization with machine learning techniques*. Available at http://nlp.stanford.edu/courses/cs224n/2009/fp/16.pdf, 12 pp.

Inkpen, D.Z., O. Feiguina, G. Hirst, 2005. *Generating more-positive and more-negative text*. In: Shanahan, J., Y. Qu, J. Wiebe (Eds.). Computing Attitude and Affect in Text: Theory and Applications. The Information Retrieval Series, 20, 187-196.

Kennedy, A., D. Inkpen, 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2), 110-125.

Landwehr, N., M. Hall, F. Eibe, 2003. Logistic model trees. *Proc. 14th European Conf. on Machine Learning*, Cavtat-Dubrovnik, Croatia, 241-252.

Pandey, S.J., 2011. *Opinion analysis through constraint optimization*. Master Thesis, Department of Computer Science, University of York, 144 pp.

Pang, B., L. Lee, 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. *Proc. of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL'04, Stroudsburg, PA, USA, 271-278.

Pang, B., L. Lee, 2008. Opinion mining and sentiment analysis, *Foundations and Trends in Information Retrieval*, 2(1-2), 1-135.

Pang, B., L. Lee, S. Vaithyanathan, 2002. Thumbs up? Sentiment classification using machine-learning techniques. *Proc. of the ACL-02 Conf. on Empirical Methods in Natural Language Processing*, Philadelphia, PA, 79-86.

Porter, M.F., 1980. An algorithm for suffix stripping. *Program*, 14(3), 130-137.

Prabowo, R., M. Thelwall, 2009. Sentiment analysis: A combined approach. *Journal of Informatics*, 3(2), 143-157.

Severyn, A., A. Moschitti, O. Uryupina, B. Plank, K. Filippova, 2014. Opinion mining on YouTube. *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, 10 pp.

Uryupina, O., B. Plank, A. Severyn, A. Rotondi, A. Moschitti, 2014. SenTube: A corpus for sentiment analysis on YouTube social media. *Proc. of the 9th Conf. of Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, 4244-4249.

Vinodhini, G., R.M. Chandrasekaran, 2012. Sentiment analysis and opinion mining: A survey. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(6), 282-292.

Wilson, T., J. Wiebe, R. Hwa, 2004. Just how mad are you? Finding strong and weak opinion clauses, *Proc. of 19th National Conf. on Artificial Intelligence*, 761-769.

Witten, I.H, E. Frank, M. Hall, 2005. Data mining: Practical machine learning tools and techniques, *The Morgan Kaufmann Series in Data Management Systems*, 629 pp.