Improved Tweets Polarity Detection using Lexicon-based Features and Caching

Lijo V. P., Hari Seetha

***Abstract:** Tweet Polarity detection the process of observing and identifying the sentiment inclination of text, whether it is positive or negative. In this paper, improved polarity detection on tweets using supervised learning is proposed. This method is using data sets available in public. The pre-processing is improved using proper caching of data items to save the time for processing of duplicate items in data sets. The feature selection strategy ensures reduced dimensionality. The low dimension data improves the classification efficiency. The experiment shows that the method is improving the overall performance in training and testing of polarity detection.*

***Keywords :** Sentiment Analysis, Polarity Detection, Trie, MergedTrie,*

I. INTRODUCTION

Nowadays, social networks are very popular and people use them to share data. Twitter users are enormous and they use twitter to share personal data and get news about political, environmental, recent technical advancements, etc. The richness of sentiment in social network data and other aspects such as political and economic attract researchers. They use the data to find insights to support applications that are using that information as the essence of the business to improve their products or services. An automated system to get and identify the opinion about their product or services is essential as manual processing is not advisable due to a huge volume of data. Opinion mining is dealing many aspects of information in the text such as identifying opinion existence, classification as positive, negative or neutral; categorize data according to the sentiment in it such as joy, sadness, anger, etc.

Polarity detection is the process of classifying the given texts based on its polarity such as positive, negative and neutral. Many methods are using supervised or unsupervised machine learning techniques, some others rely on lexicons to identify polarity and rest is using hybrid approaches. Most of them leave space for improvement. The volume of data provides an opportunity to enhance the accuracy, recall, precision, etc. by utilizing the plenty of instances. The high dimensionality of the data in the case of n-grams and POS features results limited in scalability and efficiency. Many approaches are available with the aim of reducing the dimensionality of data. Lexicon-based approach is one of them. The users post sentiments about services and products on Twitter [36]. Tweets are expressed in one or two sentences and normally give direct messages. So, tweets are an asset for opinion mining.

A sentence-level analysis can be applied to tweets with the assumption that tweets possess sentiments about one single entity. Public Twitter API is useful to retrieve tweets from Twitter. Numerous investigations have carried out to overcome the difficulties of creation of manually-labeled corpus by using emoticons in the tweets [10], [18]. The emoticons are used with the assumption that they might be matched with negative or positive sentiments about opinions expressed by the words in tweets [38]. But, there are situations where this association holds, and there are some cases where the relationship between emoticons and sentiment expressed in tweets is not clear. Hence, emoticons may introduce noise. In this way, Go et al. [18] created a dataset of tweets with around 1.6 million instances. They have used machine learning algorithms and achieved more than 80% accuracy with label prediction. Liu et al. [26] is reported that the language models with human-labeled tweets and emoticon labels outperforming abouve said approaches. Bahri et al. approached to improve the sentiment classification accuracy by emoticon score learning [3]. A multidimensional sentiment analysis is done using emoticons and emojis in [12]. The application of emoticons is investigated in domain of software development in [14].

Supervised classification are used to classify tweets according to their sentiments in [46], [23], [19] and they have used lexical resources for obtaining features.

Sentiment analysis is either a binary classification [7] or a multiclass classification; in case of binary classification group the texts in either positive class or negative class, but in multiclass [6] text is assigned to more than two classes such as fun, happiness, love, neutral, sadness, anger, etc..

The significance of data pre-processing in text analysis attracts researchers and it is not negligible as pre-processing has a very big impact on overall classification performance. In lexicon-based approaches need extensive searches for the words in lexicons for every word in the tweets. This search is a laborious task in case of a large number of lexicons and words. Proper use of data structures such as Trie, HashTable, CDAWG are improving the performance of text analysis. Used prefix tree to improve the performance of preprocessing in [43]. The lexicon-based [7] is using 7 lexicons for obtaining features. Sentiment analysis is moving to multimodal twitter data. In multimodal twitter data include texts, images, audio and video. Kumar [24] proposed a method of opinion mining using multimodal twitter data.

Revised Manuscript Received on December 30, 2019.
***** Correspondence Author

Lijo V. P.*, School of Computer Science and Engineering, Vellore Institute of Technology Vellore, India. Email: lijo.vp@vit.ac.in

Hari Seetha, School of Computer Science and Engineering, VIT-AP, Amaravati, India.. Email: seetha.hari@vitap.ac.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

Sentiment Analysis [25] trusts on text sentiment identification and interpretation and Hearst (1992) proposed a text interpretation method which is rely on direction of the sentiments.

Sentiment classification is aimed to identify the sentiment inclination (opinions) of a text and segregates them in different classes [9]. The lexicon-based approach , machine-learning, and hybrid approach are the main categories of sentiment classification [21]. The lexicon is a precompiled collection of known sentiment terms. The completeness and soundness of lexicons improve the performance of lexicon-based approaches. The corpus-based approach and dictionary-based approach are an example of a lexicon-based approach. The corpus-based is either semantic or statistical. The supervised or unsupervised strategies are used in machine learning. The Naïve Bayes, SVM, Logistic Regression, Rule-based classifiers (RBCs) and Decision Tree are popular supervised algorithms and they are used linguistic features [30].

Above all, the supervised machine learning approach SVM is very popular due to its generalization characteristics and it works well in even unbalanced data [8].

Mamgain [29] shows the situations where SVM beats other methods such as Naïve Bayes, Multilayer perceptron, and so on. The SVM training (collecting a set of Support Vectors (SVs) and other related parameters) is considered as computationally costly. Numerous ways are available in the literature to improve the performance of the SVM by improving its training time. Cauwenberghs and Poggio [11] proposed a steady SVM preparing plan which pursues online recursive training similarly as with one vector at any given moment. In this strategy, they propose a decremental unlearning which pictures the information geometry. Be that as it may, this strategy is computationally high because of its consecutive nature. A distributed and robust SVM is devised in [27]. A hybrid method with SVM and K-means is proposed by Korovkinas et al in [22] for textual sentiment analysis.

In this paper, a caching method is used to save time for pre-processing. The supervised learning is used SVM, Naïve Bayes classifier and Logistic Regression for classification. We investigate the effects of various parameters on the algorithm performance and computation cost, such as the number of instances, number of computing nodes, etc., by performing a substantial experimental evaluation.

The remaining sections are arranged as follows: In Section 2, discuss details of data pre-processing and general classifiers. Section 3 gives insights on the proposed method, Experiments and Results are covered in Section 4 and Section 5 is a conclusion.

II. THE SENTIMENT ANALYSIS

The main components in the system are lexicons, data preprocessing, various classifiers, etc. The subsections are giving details about the components in the system.

A. Data pre-processing for polarity detection

Pre-processing starts with removing the words with less number of characters. It is done based on an assumption that the words with less than three characters contribute nothing to sentiment analysis. So the words with less than 3 characters are removed from the datasets. The second step is that Stop-words removal [42], stop words are words such as prepositions, conjunctions, pronouns, etc. are seldom used to indicate sentiments. So they are removed to reduce the input size/ dimensions. Spell checking is carried out before the Stemming. Stemming and lemmatization [37] are performed as the next step of pre-processing. Handling Negations [15], [9] are important in the sentiment analysis as the negations make a sentiment of sentence opposite to the sentiment words used in the text. NLTK is used to pre-process the data. NLTK is used as following program snippets.

```
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
tokenizer = RegexpTokenizer(r'\w+')
from spellchecker import SpellChecker
filtered_words = [e.lower() for e in tweet.split() if len(e) >= 3]
words_filtered = [i for i in filtered_words if not i in
stopwords.words('english')]
words = [p_stemmer.stem(i) for w in words_filtered]
misspelled = spell.unknown(words)
```

B. Popular Classifiers

There are many classifiers available for linear and probabilistic classification. The SVM [8], Linear Regression [34], Perceptron: artificial neural network (NN) [39], [31] are well known classifiers in linear classifiers.

Some of the probabilistic (Generative) classifiers are Naïve Bayes (NB) [13], Bayesian network (BN) [1] and maximum entropy classifier (ME)..

III. PROPOSED METHOD

In the proposed method, lexicon-based features are used. Those features are derived from various lexicons. The selected 21 features are obtained from 7 popular lexicons. Lexicons are searched for each word in tweets and it returns either sentiment score or positive/ negative label if there is a match. An extensive is search is needed for each word in the tweet to get a score from the lexicon. Fig. 1 shows various components of the tweet polarity detection. Tweets are pre-processed to remove noises and irrelevant information. Sentiment score of each word in tweets are collected using various lexicons. Feature extractors obtain the features based on the word scores and prepare a feature vector for each tweet. A Trie (for an instance) is prepared to cache the word score for handling repeated words to avoid unwanted search in lexicons. This trie can be used in testing time to prepare feature vector of test set. The features vectors are giving as input to any classifier to classify the tweets in to different classes such as positive, negative, etc.

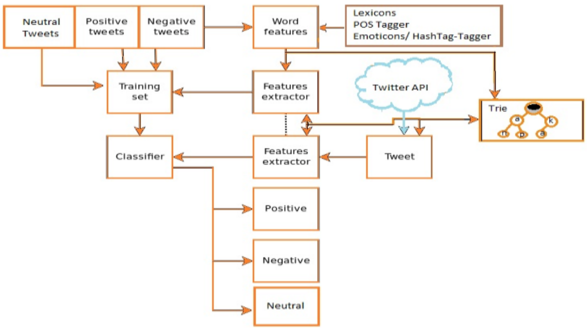

Fig.1. Tweet Polarity Detection

A. Caching

The caching is the method to keep an item in a fast accessible medium to reduce the time for accessing the item repeatedly. The principle behind caching is the well-known locality principle. The temporal locality refers to the probability of reuse the same item with a short duration and spatial locality refers that the reuse of items in adjacent locations.

In our work, the temporal locality principle utilized to handle repeated words in tweets. The word search is carried out on several lexicons to get a score for obtaining features. The time complexity of the search is *O(l)*, where *l* is the length of the search item if it is trie implementation of the lexicon. So, there is a need for a minimum m number of searches for every word in the tweets in case of m lexicons. This will result in a very large time for obtaining the features when there are a large number of tweets. The analysis of the frequency of words in data sets reveals that many words are repeating multiple times in the data set. In this work, this knowledge is used to reduce the number of searches in lexicons by caching the already searched items. The searched items are cached with their scores from various lexicons to give scores for the next search for the same item. This will save time for searching for repeated items. For a lexicons-based system without caching is needed *O(nml)* for obtaining features from m number of lexicons for n number of words with average length *l*. But it is reduced to *O(fml)* where *f* is the number of unique words. In most of the data sets the frequency of words is high and f will be very smaller than n. So this will reduce the time complexity of obtaining features from lexicons. Trie, Hash Tables, Directed Acyclic Word Graph (DAWG), MergedTrie, etc. are some of the important data structures for text indexing.

Table- I: Lexicon-based features

Features	Range	Description	Source
F_1	{0 to n}	# of matching +ve words	Lex1
F_2	{0 to n}	# of matching -ve words	
F_3	{0 to n}	# of matching +ve words	Lex2
F_4	{0 to n}	# of matching -ve words	
F_5	{0 to n}	# of matching +ve words	Lex3
F_6	{0 to n}	# of matching -ve words	
F_7	[0 to ∞[	Score sum of +ve matching words	Lex4
F_8	]-∞ to 0]	Score sum of -ve matching words	
F_9	[0 to ∞[	Score sum of +ve matching words	Lex5
F_{10}	]-∞ to 0]	Score sum of -ve matching words	
F_{11}	[0 to ∞[	Score sum of +ve matching concepts	Lex6
F_{12}	]-∞ to 0]	Score sum of -ve matching concepts	
F_{13}	[0 to ∞[	Score sum of +ve matching words	Lex7
F_{14}	]-∞ to 0]	Score sum of -ve matching words	
F_{15}	{1 to 5}	Strength output for the +ve class	Lex8
F_{16}	{-1 to -5}	Strength output for the -ve class	

A trie data structure is a tree with nodes of letters. The alphabet's letters are stored in nodes and if required the auxiliary information can also be stored in nodes. By traversing through the nodes in a particular pathway of branch helps to retrieve words or strings form trie. For retrieving a word from trie needs a search with the complexity of *O(l)*, where *l* is the length of the searched item. The search is faster when compared to Hash tables and Search trees. The search time of them is a function of a number of words actually stored. But trie's search time is a function of the length of the word. So trie search is faster as most of the lexicons have a larger number of words than the maximum length of the word.

Tries are consumed a large memory when the words have low prefix sharing. Radix Tree, Patricia Trie, and C-Trie have partly solved these drawbacks. The nodes with single child nodes are merged with their parents. In [20], developed a Compact Patricia Trie which resolves the demerit of Patricia that need large memory.

Radix Tree, Patricia Trie [20], and Compact Directed Acyclic Word Graphs (CDAWGs) [5] reduce the memory consumption by compaction for term-level index. But there is an increase intime for insertion, update, and removal operations and the minimization/ compaction process needs some intermediate data structures. The Double Trie (DT) was proposed in [33], [2], and Watson [44] is implemented the DT in C++ and in Java. Java implementation available in a toolkit called as FIRE Engine II [47]. DT balances the spatial and temporal efficiency.

MergedTries [17] are merged two tries, one is prefix trie and another is suffix trie, of Double Trie. This helps to get prefix and suffix overlapping. In addition to this, they reduce the height of the trie by enhancing the DT segmentation; segment the term in exact half. MergedTrie gives better performance in insertion, updating, deletion, etc.

B. Lexicons and Features

The literature gives a large number of lexicons and methods for text polarity analysis. Popular lexicons are OpinionFinder (Lex1) [45], AFINN (Lex5) [35], NRC –{emotion (Lex3) [41], hashtag (Lex7) [32], Bing Liu's Opinion (Lex2) [4] , SentiWordNet (Lex4) [40], Sentiment140 [18], etc.. SenticNet (Lex6) [16], SentiStrength (Lex8) [28], and Sentiment140 (Lex9) [8] are well known methods for sentiment polarity calculation. These lexicons give sentiment scores, sentiment strength score or sentiment label of English words which are common in social network messages. The polarity detection methods are giving a label for the sentiment of the given sentence. The label may be Positive, Negative or Neutral.

The lexicon-based [7] features are used in this works. The features derived from the above-mentioned lexicons are listed in Table-I. Some of them are just a count, the number of words matches with words in the lexicon. But other features are the sum of word scores of the words in each tweet matches with lexicons. The count is either positive words count or negative words count. For, example, OpinionFinder Positive (F_1): count of matching positive words and OpinionFinder Negative (F_2): count of matching negative words and SentiWordnet Positive (F_7) and SentiWordnet Negative (F_8), that are the word score sum of positive and negative words of the tweet that matches the SentiWordnet lexicon, respectively. The classifiers are also trained with 500 unigram features and experiment results are collected for comparison.

C. Twitter sentiment analysis

The supervised learning strategy is used in this work. The features are extracted from each tweet and represent them as vectors. Manually and automatically annotated data sets are used for training and testing. The analysis task is limited to sentiment polarity detection where tweets are classified as positive and negative. For all the tweets in the datasets, the extracted feature vectors are combined with the annotated labels to make input for supervised learning classifiers. For completing this learning task any classifiers like Naïve Bayes, Logistic Regression, SVM, etc. can be used. The learned classifier can be used to infer the polarity of the unseen tweets.

D. Tweet sentiment representation

The lexicon-based features used in [7] are used with changes for polarity detection. Besides, the presence of negation is noticed as a binary value. The feature selection process helps to reduce dimensionality significantly and it overcomes the problem of sparse data in feature vectors. The numbers of features are fixed for all the data sets. The collected features are ranked using Information Gain (IG) and considered only the first 15 features for each data set. IG measures the entropy reduction after getting the best split by a feature. The binary classification is done for positive and negative tweets. Table-II gives the ranked features.

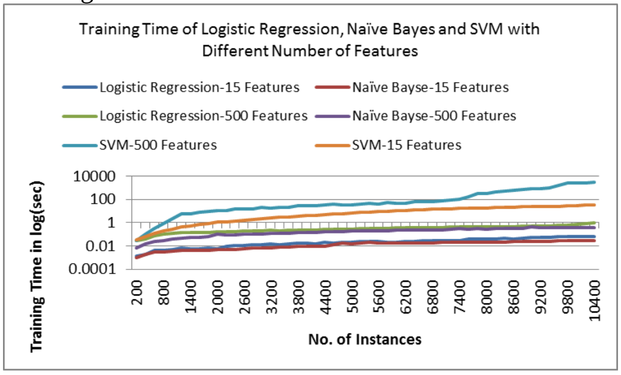

Fig. 2. Comparison of Training Time of various classifiers

Table- II: Ranked Features

Sentiment140		Twitter-Airline	
Features	**Info. Gain**	**Features**	**Info. Gain**
F_4	0.261	F_7	0.283
F_{15}	0.221	F_{14}	0.281
F_{14}	0.215	F_6	0.279
F_{16}	0.208	F_{15}	0.272
F_{10}	0.199	F_{16}	0.262
F_5	0.191	F_5	0.219
F_9	0.173	F_3	0.201
F_3	0.148	F_1	0.198
F_8	0.14	F_9	0.192
F_2	0.137	F_{13}	0.191
F_{11}	0.127	F_2	0.165
F_{13}	0.101	F_{10}	0.151
F_1	0.083	F_{11}	0.141
F_{12}	0.076	F_4	0.103
F_6	0.073	F_{12}	0.091

Table-III: Statistics of Data Sets

Data Sets	**Positive**	**Negative**	**Neutral**
Twitter-Airline	2363	9178	3099
Sentiment140	248576	799999	0

Table-IV: Time in seconds Sentiment140 (19,54,3182 words).

Data Structures	Insertion (sec.)	Success. Search (sec.)	Failure search (sec.)
DAWG	134.62	19.57	12.42
Double Trie	129.42	7.25	6.65
Hash Table	43	39	25.21
MergedTrie	12.15	9.23	6.6
PATRICIA Trie	77.43	69.28	45.81
Trie	163.3	179.51	83.37

IV. EXPERIMENTS AND RESULTS

In this session, we discuss the experiments and results. The training and testing of data sets are done using two popular data sets named Sentiment140 [28] and Twitter- Airline[49]. Each tweet is tagged as positive, negative or neutral. For polarity detection, only positive and negative tweets are considered and discarded neutral.

Six well-known data structures are used to compare their performance in caching. They are used as a pointer-linked Trie implementation, Hash Table with unsorted strings, PATRICIA tree structure, DAWG [50] (dawgdic-0.4.5), double-array Trie and MergeTrie.

These data structures are used to cache the words which are searched in lexicons for feature formation for future use. If the words come for searching next time it can be searched in cached item instead of search in lexicons. That will save time for search in multiple lexicons. The same cached items can be used to speed up while pre-processing the testing data item also. This caching helps in improving overall performance in training and testing.

Table-V: Time in seconds with Twitter-Airline (152044 words).

Data Structures	Insertion	Success. Search(Sec.)	Failure Search(Sec.)
Dawg	1.12	0.29	0.081
Double Trie	1.4	0.08	0.024
Hans Table	1.62	0.92	0.034
Merged Trie	0.95	0.45	0.098
Patricia Trie	1.9	1.6	0.68
Trie	0.29	0.78	0.14

Feature Analysis is performed over the features mentioned in Table-I. We analyze how well each feature split the data sets concerning polarity detection. Sentiment140 method and SentiStrength method are giving positive, negative and neutral labels as output. These methods' features SSPOL and S140 are providing good splits on every data set. The features from AFINN, OpinionFinder, SenticNet, and SentiWordnet are useful for polarity detection. Here, we consider multiple resources and they gave better performance than individual resources. The principle behind this is the required feature may absent in one or more resources but they may be compensated from other resources.

Sentiment Analysis is performed using a methodology that is using a strategy for automatic polarity detection. The tweets and its labels from the three data sets are used for this task. We have used the supervised classifiers for tweet level polarity detection. We compare the performance of NB, SVM, and Logistic Regression classifiers. All the experiments are validated using 10-fold cross-validation.

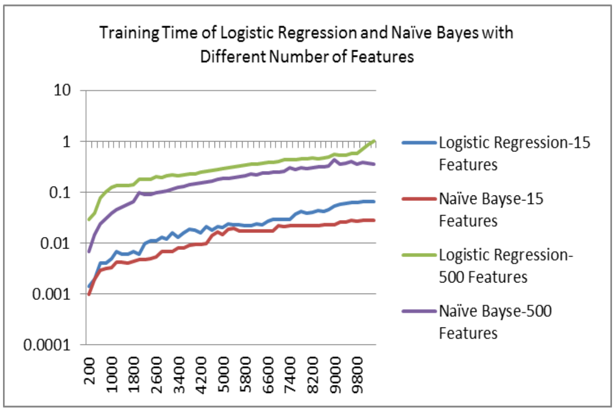

Fig. 3: Training Time of Naïve Bayes and Logistic Regression Classifiers with 15 and 500 Number of Features

The experiments are carried out on Hp computers with Ubuntu 16 OS, RAM- 8 GB, and Processor clock 2.3 GHz. The statistics of data sets Sentiment140, and Twitter- Airline shown in Table-III. Those data sets are publically available, which is ensuring the repeatability of experiments. Fig. 2 shows that the training time of various classifiers with two different numbers of features. One is 500 unigram features and the other is 15 lexicon-based features. It is noticed that there is a significant saving of computation time in case of lexicon-based features. SVM was consumed more than two hours to be trained with 10000 instances of unigram features. The accuracy achieved using the selected 15 features is comparable with the accuracy got with unigram features. The time taken for the same number of instances with selected 15 features is lesser than the time for the 500 features.

The positive impact of dimensionality reduction is supported by the performance of the Naïve Bayes and Logistic Regression. Fig. 3 gives the details of the training time of Naïve Bayes and Logistic Regression with unigram features and lexicon-based features. The scalability test is performed with a different number of instances from the Sentiment140 data set. The test results show that the caching and dimensionality reduction improve efficiency with less time complexity. Fig. 4 shows that the training time of various numbers (14, 15, 16, and 17) of features. Based on information gain (IG) 15 features are used for classification. The highest accuracy of classification is gained with 15 and 16 features, so in favor of computation time, 15 features are considered for further experiments. The classification with more than 16 features does not contribute much to accuracy. Fig. 5 shows the F1 and accuracy measures of 14, 15, 16 and 17 features.

Table-IV and Table-V give the details about the time required for insertion, search, etc. for various data structures with the data sets Twitter-Airline and Sentiment140. The analysis shows that MegedTrie is performed well in both datasets. The Double Trie has given similar performance as MergeTrie for the search operations. But it requires more time for insertion. The overall performance is improved with the use of caching, where all the searched items are stored in a data structure which is supporting fast searching and indexing, to avoid repeat search of the same item in various lexicons.

Table-VI: Classification performance: 10 fold cross-validation with 15 features

Data Sets	Methods	Accuracy	F1
Sentiment 140	NB	76.82	75.89
	SVM	83.95	82.73
	Logistic Regression	79.51	79.32
Twitter-Airline	SVM	88.01	87.31
	NB	83.46	83.26
	Logistic Regression	82.04	82.02

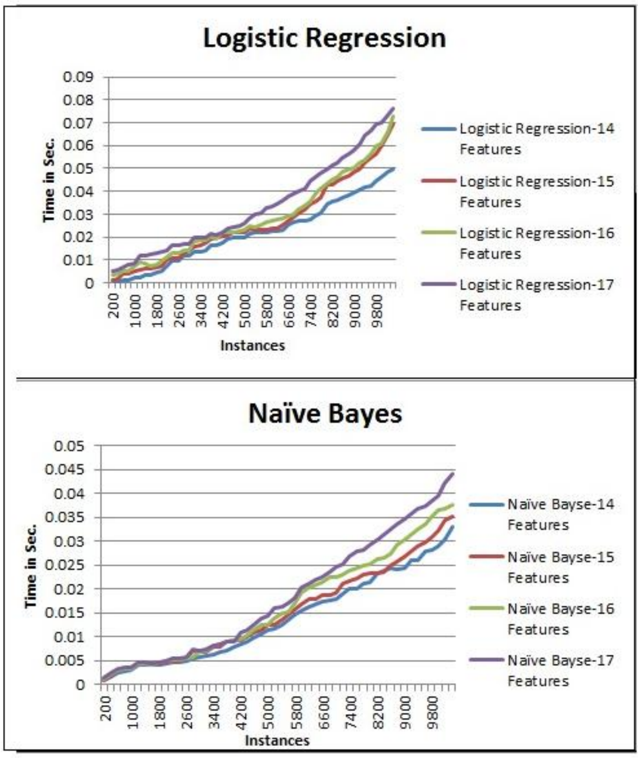

Fig.4. Training Time of Logistic Regression and NB Classifier with 14, 15, 16 and 17 Number of Features

Table-VI is giving details about the performance of the classifier. This is showing the classifiers' accuracy and F1 measure. Our classifier is giving good accuracy with the less computational cost. The proposed method with the reduced dimension of data is showing better results with Naive Bayes (NB), SVM and Logistic Regression.

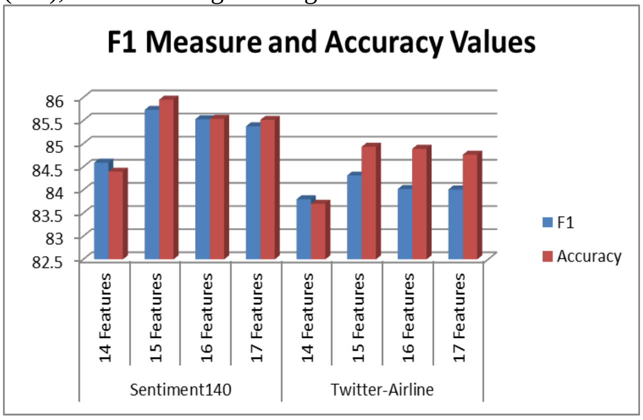

Fig.5. Classification Accuracy with Different number of Features

V. CONCLUSIONS

An efficient polarity detection for tweets is proposed in this paper. The dimensionality reduction and caching help to reduce the computational cost of pre-processing and testing. The extensive experiments show that this method outperforms the existing approaches for polarity detection concerning the computation cost without compromising accuracy. The publically available data sets are used in this method to facilitate repeatability of experiments. The MergeTrie give better performance in indexing of words. Our method is mainly focused to improve the efficiency of the sentiment classification in terms of computation cost; speedups are ensured even in case of large data sets. As a future work, we will take additional measures to improve the accuracy of sentiment classification.

REFERENCES

1. Aggarwal, C.C. “Mining text data”, Data Mining, Springer International Publishing, Switzerland, 2015, pp.429–455.
2. Aoe J.; Morimoto K.; Shishibori M.; Park HK, “A Trie Compaction Algorithm for a Large Set of Keys”. IEEE Transactions on Knowledge & Data Engineering, 8, 1996, pp. 476–491.
3. Bahri, S., Bahri, P., & Lal, S. (2018). A novel approach of sentiment classification using emoticons. Procedia computer science, 132, 669-678.
4. Bing, L. (2012). Sentiment analysis: A fascinating problem. In Sentiment Analysis and Opinion Mining, pages 7–143. Morgan and Claypool Publishers.
5. Blumer A.; Blumer J.; Haussler D.; McConnell R.; Ehrenfeucht A. (1987). Complete inverted files for efficient text retrieval and analysis. Journal of the Association for Computing Machinery, 34 (3), pp. 578–595.
6. Bouazizi, M., & Ohtsuki, T. (2019). Multi-class sentiment analysis on twitter: Classification performance and challenges. Big Data Mining and Analytics, 2(3), 181-194.
7. Bravo-Marquez, F., Mendoza, M., & Poblete, B. (2014). Meta-level sentiment models for big social data analysis. Knowledge-Based Systems, 69, 86-99.
8. Burges, C.J.C. : A tutorial on support vector machines for pattern Recognition. Data Mining and Knowledge Discovery. Vol. 2, No. 2, (1998) 121–167
9. Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (Eds.).: A practical guide to sentiment analysis. Cham, Switzerland: Springer International Publishing (2017).
10. Carvalho, P., Sarmento, L., Silva, M. J., and de Oliveira, E. Clues for detecting irony in user-generated contents: oh...!! it’s ”so easy” ;-). In Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion Hong Kong, China, 2009.
11. Cauwenberghs, G., & Poggio, T.: Incremental and decremental support vector machine learning. In Advances in neural information processing systems (2001) 409-415
12. Chauhan, D., & Sutaria, K. (2018). Multidimensional sentiment analysis on twitter with semiotics. International Journal of Information Technology, 1-6.
13. Chen, J., Huang, H., Tian, S. and Qu, Y. (2009) ‘Feature selection for text classification with Naïve Bayes’, Expert Systems with Applications, Elsevier, Vol. 36, No. 3, pp.5432–5435.
14. Claes, M., Mäntylä, M., & Farooq, U. (2018, October). On the use of emoticons in open source software development. In Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (p. 50). ACM.
15. Councill, I. G., McDonald, R., and Velikovich, L.: What’s great and what’s not: Learning to classify the scope of negation for improved sentiment analysis. In Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, NeSp-NLP ’10, Stroudsburg, PA, USA. Association for Computational Linguistics (2010) 51-59
16. E. Cambria, R. Speer C. Havasi, A. Hussain, SenticNet 2: a semantic and affective resource for opinion mining and sentiment analysis, in: FLAIRS Conference, 2012, pp. 202–207.
17. Ferrández, A., & Peral, J. (2019). MergedTrie: Efficient textual indexing. PloS one, 14(4), e0215288.
18. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Technical report Stanford University (2010).
19. Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. Target-dependent Twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, volume 1, pages 151–160, 2011.
20. Jung M.; Shishibori M.; Tanaka Y.; Aoe J. (2002). A dynamic construction algorithm for the Compact Patricia trie using the hierarchical structure, Information Processing & Management, 38(2), pp. 221– 236.
21. Kolchyna, O., Souza, T. T., Treleaven, P., & Aste, T.: Twitter sentiment analysis: Lexicon method, machine learning method and their combination. arXiv preprint arXiv:1507.00955 (2015).
22. Korovkinas, K., Danėnas, P., & Garšva, G. (2019). SVM and k-Means Hybrid Method for Textual Data Sentiment Analysis. Baltic Journal of Modern Computing, 7(1), 47-60.
23. Kouloumpis, E., Wilson, T., and Moore, J. Twitter Sentiment Analysis: The Good the Bad and the OMG! In Fifth International AAAI Conference on Weblogs and Social Media, 2011.
24. Kumar, A., & Garg, G. (2019). Sentiment analysis of multimodal twitter data. Multimedia Tools and Applications, 1-17.
25. Lijo, V. P., & Seetha, H.: Text-based sentiment analysis: Review. International Journal of Knowledge and Learning, Vol. 12(1) (2017) 1-26.
26. Liu, K., Li,W., and Guo,M. Emoticon smoothed language models for Twitter sentiment analysis. In Proceedings of the 26th AAAI Conference on Artificial Intelligence. Toronto, Ontario, Canada, 2012.
27. Liu, Y., Ding, H., Huang, Z., & Xu, J. (2016). Distributed and robust support vector machine. In 27th International Symposium on Algorithms and Computation (ISAAC 2016). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
28. M. Thelwall, K. Buckley, G. Paltoglou, Sentiment strength detection for the social web, J. Am. Soc. Inform. Sci. Technol. 63 (1) (2012) 163–173.
29. Mamgain, N., Mehta, E., Mittal, A. and Bhatt, G.: Sentiment analysis of top colleges in India using Twitter data, International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), IEEE, New Delhi, India, March, (2016) 525–530.
30. Medhat, W., Hassan, A. and Korashy, H. (2014) ‘Sentiment analysis algorithms and applications: a survey’, Ain Shams Engineering Journal, Vol. 5, No. 4, pp.1093–1113.
31. Minsky, M., & Papert, S. A. (2017). Perceptrons: An introduction to computational geometry. MIT press.
32. Mohammad, S. M., Kiritchenko, S., and Zhu, X. (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).
33. Morimoto K.; Iriguchi H.; Aoe JI. (1995). A dictionary retrieval algorithm using two trie structures. Systems and Computers in Japan 26(2), pp. 85–97.
34. Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems (pp. 841-848).
35. Nielsen, F. . (2011). A new anew: evaluation of a word list for sentiment analysis in microblogs. In Proceedings of the ESWC2011 Workshop on ’Making Sense of Microposts’: Big things come in small packages 718 in CEUR Workshop Proceedings, pages 93–98.
36. Pak, A., and Paroubek, P. Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of the Seventh International Conference on Language Resources and Evaluation. Valletta, Malta, 2010.
37. Porter, M. F.: An algorithm for suffix stripping. In Program, vol. 14 (1980) 130–137.
38. Read, J. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics. Michigan, USA, 2005.
39. Ruiz, M.E. and Srinivasan, P. (1999) ‘Hierarchical neural networks for text categorization (poster abstract)’, Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, August, USA, pp.281, 282.

40. S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining, in: Proceedings of the Seventh International Conference on Language Resources and Evaluation, Valletta, Malta, 2010.
41. S.M. Mohammad, P.D. Turney, Crowdsourcing a word–emotion association lexicon, Comput. Intell. 29 (3) (2013) 436–465.
42. Salton, G. and McGill, M. J.: In Introduction to Modern Information Retrieval. McGraw Hill Book Co. (1983).
43. Su, Y. J., Chen, R. C., Hsiung, C. M., Chen, Y. Q., Yu, S. W., & Huang, H. W. (2016, March). Using Prefix Tree to Improve the Performance of Chinese Sentiment Analysis. In 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA) (pp. 955-959). IEEE.
44. Watson B. W. (1996). Implementing and using finite automata toolkits. Natural Language Engineering 2 (4), pp. 295–302.
45. Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In hltemnlp2005, pages 347–354, Vancouver, Canada.
46. Zirn, C., Niepert M., Stuckenschmidt H., and Strube, M. Fine-grained sentiment analysis with structural features. In Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP), pages 336–344. Chiang Mai, Thailand, 2011.
47. http://www3.cs.stonybrook.edu/~algorith/implement/watson/implement.shtml.
48. http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip.
49. https://www.kaggle.com/crowdflower/twitter-airline-sentiment/downloads/twitter-airline-sentiment.zip/2
50. https://code.google.com/archive/p/dawgdic/downloads

AUTHORS PROFILE

Lijo V. P. obtained his Master’s degree in Computer Science and Engineering from National Institute of Technology (formerly R. E. C.) Calicut and he is doing Ph.D at Vellore Institute of Technology, Vellore. He has research interests in the fields of big data, data mining, text mining and machine learning. He has published a few research papers in national and international journals and conferences. He is a life time member of Computer Society of India (CSI). He is currently working as Assistant Professor in the School of Computer Science and Engineering at Vellore Institute of Technology, Vellore, India.

Hari Seetha obtained her Master’s degree from National Institute of Technology (formerly R. E. C.)Warangal and obtained M.Phil as well as Ph.D from School of Computer Science and Engineering, V.I.T University. Her M.Phil work was based on Application of Data Mining in Short-term electric load prediction. She worked on Large Data Classification during her Ph.D. She has research interests in the fields of pattern recognition, data mining, text mining and machine learning. She received Best paper ward for the paper entitled “On improving the generalization of SVM Classifier” in Fifth International Conference on Information Processing held at Bangalore. She has published a few research papers in national and international journals and conferences. She has been one of the Editor for the Edited Volume on “Modern Technologies for Bigdata Classification and Clustering” published by IGI Global in 2017. She is a member of editorial board for various International Journals. She is presently guiding 6 Ph.D students. She is listed in the 2014 edition of Who’s who in the world published by Marquis Who’s Who, as the biographical reference representing the world’s most accomplished individuals. She had been a co-investigator to a major research project sponsored by the Department of Science and Technology, Government of India. She served as a Division Chair for Software Systems division and Program Chair for B.Tech (CSE) program, in the School of Computer Science and Engineering at VIT University, Vellore, India as well as Assistant Director(Ranking and Accreditation) at VIT University,Vellore. She is currently working as Professor and Dean for the School of Computer Science and Engineering at VIT-AP, Amravati, India.