IJCSN International Journal of Computer Science and Network, Volume 5, Issue 5, October 2016
ISSN (Online): 2277-5420 www.IJCSN.org
Impact Factor: 1.02

An Implementation of New Frequent Pattern Mining Algorithm for Business Intelligence Solution

[1] Ujwala Mhashakhetri, [2] Dr. Rahila Sheikh

[1] Department of Computer Science & Engineering
Rajiv Gandhi College of Engineering, Research and Technology
Chandrapur, Maharashtra, India

[2] Department of Computer Science & Engineering
Rajiv Gandhi College of Engineering, Research and Technology
Chandrapur, Maharashtra, India

Abstract - In the digital world, frequent pattern mining algorithm is used widely for business implementation in the areas like online marketing, sales and advertisement. In order to make better business decision both on the individual and organizational level we search for the others opinion. Social media, discussion forums, review, blogs and micro-blogs are opinion rich resources. Mined information from these resources can be successfully utilized for decision making by customer or organization if opinion orientation are considered carefully. In this paper, we propose a business intelligence solution using frequent pattern mining. Evaluation results show that the proposed algorithm is more accurate, efficient and work better with dense datasets.

Keywords - *Data Mining, Frequent Pattern Mining, Frequent, Opinion Mining.*

1. Introduction

Data mining is the process of searching, collecting, analyzing large amount of data from the database through the different perspective and categorizing, summarizing it with different dimensions into useful information. Data mining software is the analytical tool used to find the patterns or co-relation among the field of relational database.

Generally individual or organization is likely to depend on others opinion for decision making and in order to get required information they search through large amount of data which is available on social networks, blogs and reviews. Consider an example in which an individual want to buy good product in market, firstly he/she will go through review of product and then take the decision depending on the other customer's review. In the same way company can also review the customer's opinion about the product features and can make required improvement in product according to requirement.

Many algorithms and techniques are proposed by the researcher to automatically extract useful information. Opinion mining determines whether the comment is positive, negative or neutral orientation [1]. Product feature extraction is difficult for analyzing opinion as the opinion orientation identification is significantly affected by the target features [2]. In this paper we focus on the feature extraction and orientation detection from customer reviews using the improved frequent pattern mining algorithm [3]. Previously proposed frequent pattern mining algorithms which are used for discovering frequent itemsets from the transaction datasets are **Apriori algorithm** [4], **AprioriTID** algorithm[4], **FP-Growth** algorithm[5]**, Eclat** algorithm[6], **dEclat** algorithm[7], **Relim** algorithm[8], **H-mine and H-mine(Mem)** algorithm[9], **LCMFreq** algorithm[10], **PrePost** and **PrePost+** algorithms[11],

FIN algorithm[12]. Algorithms for performing **targeted and dynamic queries about association rules and frequent itemsets** are **Itemset-Tree** [13], **Memory-Efficient Itemset-Tree** [14].

In this paper we are implementing a new algorithm based on Item-Set tree data structure which extracts opinion orientation from reviews. It has been found that the

performance of algorithm increases for incremental data. Section 2 defines the problem and introduces and proposed approach. Section 3 gives the overall implementation of the algorithm with experimental result and evaluation and section 4 conclude the paper and related references.

2. Problem Definition & Proposed Methodology

2.1 Problem Definition

The basic problem is how to identify frequent and infrequent features from the reviews. Another problem is about review orientation, a word with the neutral or negative orientation for one product can be positive for another product. Again it is necessary that the data structure must handle increasing volume of review data available on social network. Next the classification accuracy of previously used Naïve Bayes algorithm [15] is 79% so, it is necessary to improve the classification accuracy in the proposed algorithm.

2.2 Proposed Approach

For the incremental dataset the itemset tree data structure is the best solution because of its incremental nature as new transaction can be efficiently added in the existing tree. It also has a property of compactness. In the proposed approach we have used itemset tree data structure. The data flow architecture of the proposed system is given in the figure 1 and each component is detailed subsequently. Input to the system is review datasets and is available at http*://www.cs.uic.edu/~liub/FBS/CustomerReviewDat.zip*.

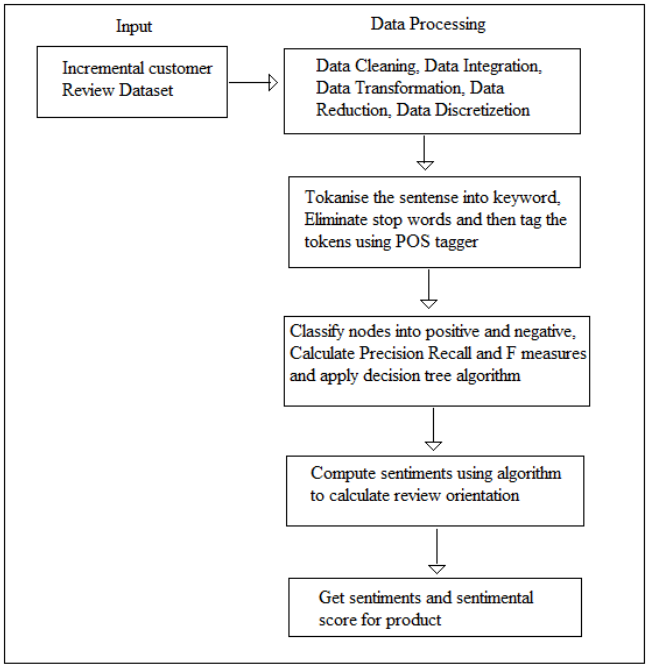

Fig. 1: Data Flow Architecture.

Algorithm for the system is given below.

Algorithm1:
Algorithm for extracting Sentiment of Review Comment
Require: Product Review Document
Ensure: Sentiment of User comment.
1. Fetch the comment.
2. Convert the unstructured comment data to structured document.
3. Tokenize the sentences into keywords.
4. Eliminate Stop words and tag the tokens using POS tagger.
5. If term is not in the dictionary check for the correct word.
6. Classify Node for positive and negative.
7. Calculate Precision Recall and F measure.
8. Apply decision tree algorithm.
9. Compute sentiments using algorithm 2
10. Return sentiment and sentiment score of review

Algorithm2:
Algorithm to calculate the review orientation
1. Procedure Review Sense( )
2. begin
3. for each review sentence si
4. begin
5. sense = 0;
6. For each review word rw in si
7. sense + = Word Sense (rw, si);
8. /* Positive =1 , Negative =-1*/

9. if (sense >0) si's sense = Positive;
10. else if (sense<0) si's sense = Negative
11. endfor;
12. end
1. Procedure Word Sense (word, sentence)
2. begin
3. sense = orientation of word in bag of keywords;
4. If(there is NEGATIVE_WORD appears closely around word in sentence)
5. sense = opposite(sense);
End

3. Implementation

For implementing the proposed algorithm it is necessary to have required dataset as input. Here the dataset is review dataset. The overall implementation includes fallowing modules,

- Data cleaning and processing
- Testing datasets
- Performance analysis

3.1 Data Cleaning and Processing

Data cleansing, data cleaning or data scrubbing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. Used mainly in databases, the term refers to identifying incomplete, incorrect, inaccurate, irrelevant, etc. parts of the data and then replacing, modifying, or deleting this dirty data or coarse data.

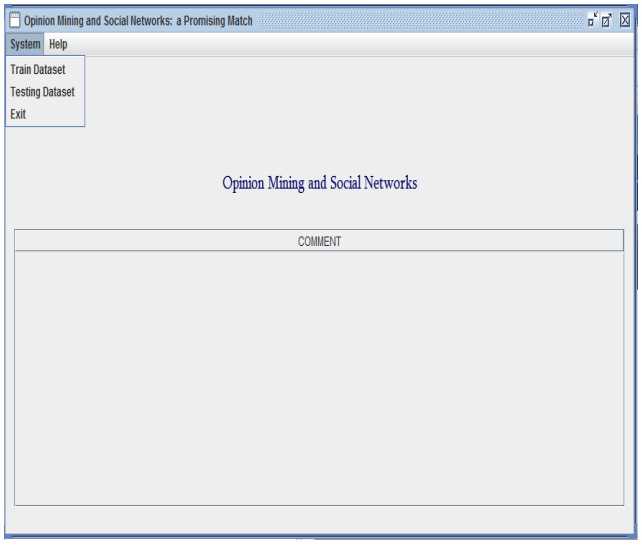

Fig. 2: Main Window.

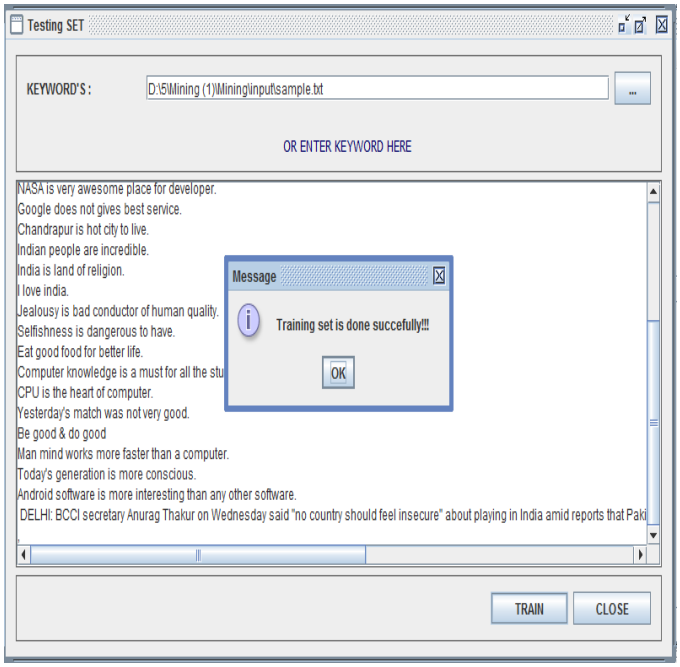

Fig. 3: Window for selecting Dataset and training sets

After cleansing, a data set will be consistent with other similar data sets in the system. The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores. Data cleansing differs from data validation in that validation almost invariably means data is rejected from the system at entry and is performed at entry time, rather than on batches of data.

Figure 2 shows the main window with the option of training and testing datasets below the system. In the machine learning training data is dataset on which machine learning program learn to perform co-relational tasks such as classify, cluster and learn the attributes.

Figure 3 shows the window for training dataset. The selected dataset appears in the panel, which is train using train button.

3.2 Testing Datasets

The main objective of experiments was to test the accuracy of the proposed algorithm for collections of opinions harvested from the social networks sites. Testing data is the data, whose outcome is already known and is used to determine the accuracy of the machine learning algorithm, based on the training data.

As the training is successfully completed the next move is testing dataset. Following figure 4 shows that the window for testing dataset.

The result of testing process is shown in the figure 5. It shows the score for each sentence with the total positive, negative and neutral score for dataset at the bottom.

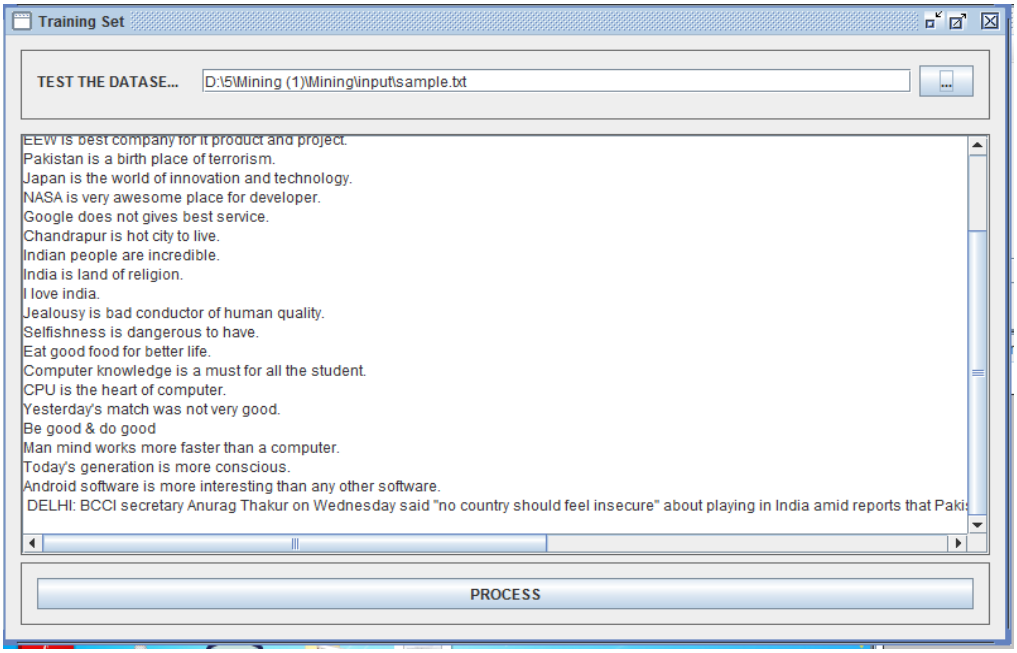

Fig. 4: Window for selecting Dataset to be tested

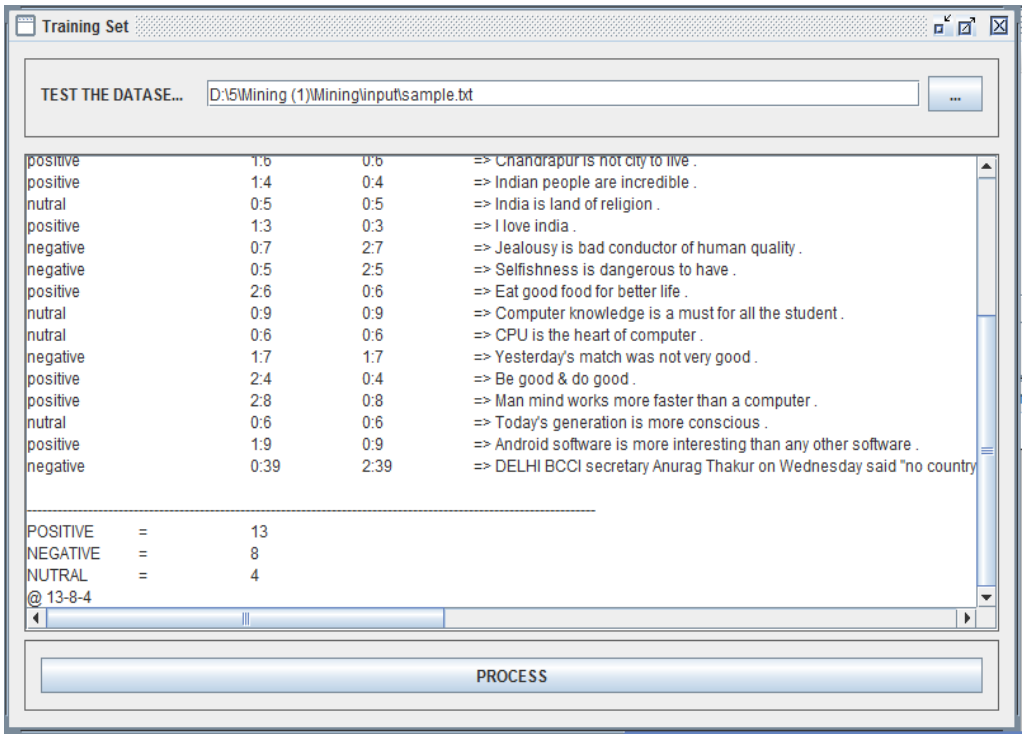

Fig. 5: Test results of Dataset

3.3 Performance Analysis

Datasets used for analysis are

- Movie dataset from IMBD
- Product review dataset from amazons .com
- Twitter dataset of digital India

Cross domain analysis with respect to time for different datasets:

- Movie dataset:

Following figures shows the result of comparison between Naïve Bayes Classifier and Proposed Classifier based on Itemset Tree Data structure for Movie dataset.

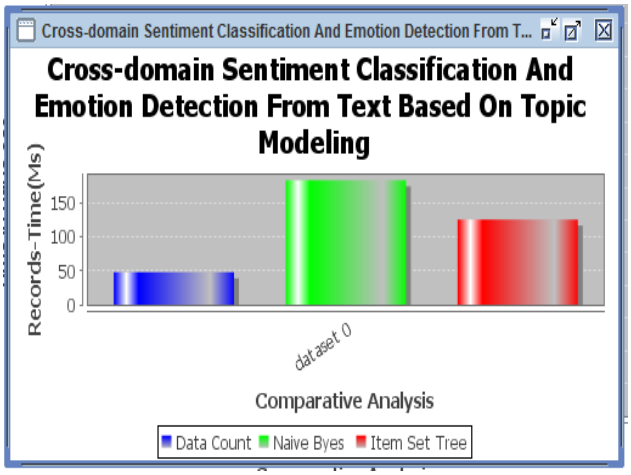

Fig. 6: Time required for movie dataset along with data count

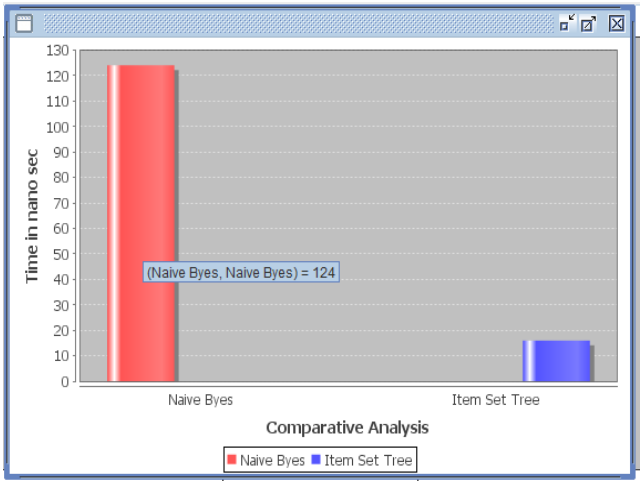

Fig. 7: Time required for movie dataset

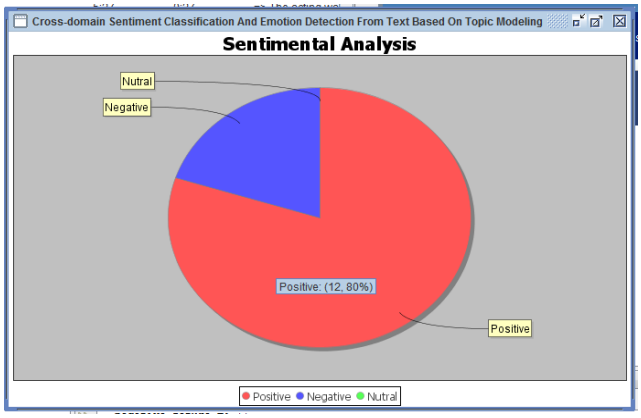

Fig. 8: Sentimental analysis for movie dataset

- Product Review Dataset:

Following figures shows the result of comparison between Naïve Bayes Classifier and Proposed Classifier based on Itemset Tree Data structure for Product Review dataset.

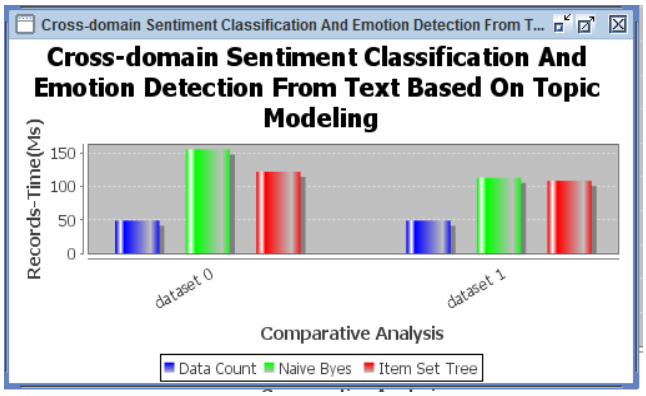

Fig. 9: Time required for movie and product review dataset along with data count

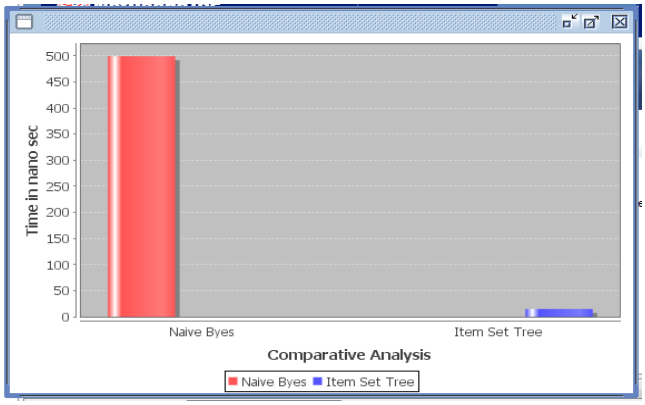

Fig. 10: Time required for product review dataset

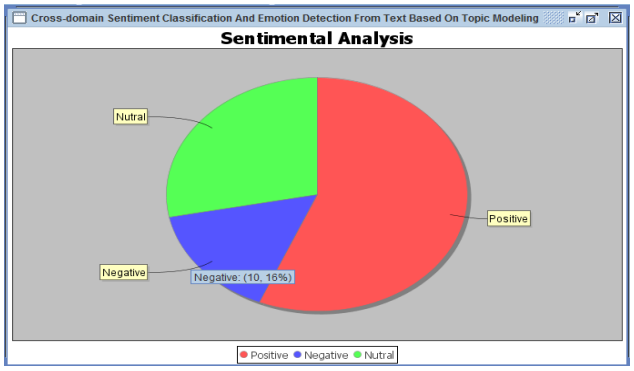

Fig.11: Sentimental analysis for product review dataset

- Twitter Dataset:

Following figures shows the result of comparison between Naïve Bayes Classifier and Proposed Classifier based on Itemset Tree Data structure for Twitter dataset.

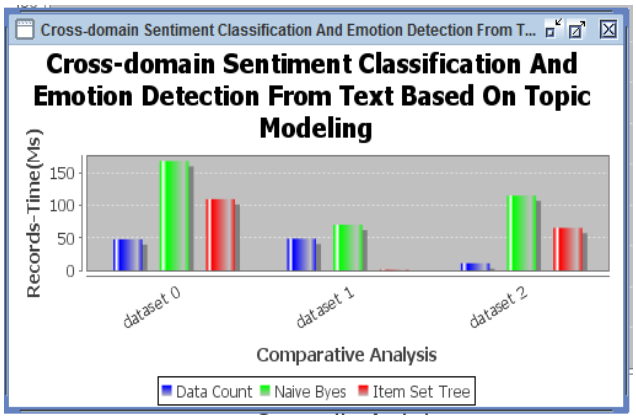

Fig. 12: Time required for movie, product review and twitter dataset along with data count

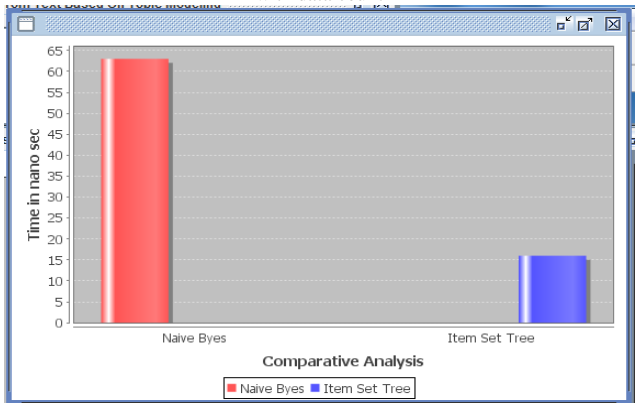

Fig. 13: Time required for Twitter dataset

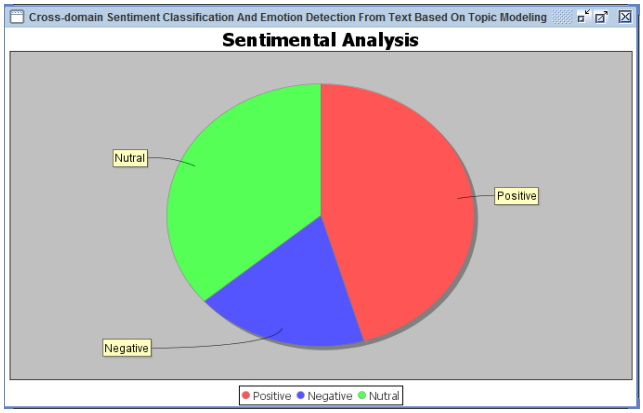

Fig. 14: Sentimental analysis for Twitter dataset

Classification accuracy calculation for all three datasets:

Machine learning:

TP= true positive, TN=true negative, FP=false positive, FN=false negative

Table 1: Machine learning

	Prediction of model: positive	Prediction of model: negative
Truth: positive	TP	FN
Truth: negative	FP	TN

Where, TP and FP are true positives and false positives (i.e., numbers of positive examples from the test set classified correctly and incorrectly), and TN and FN are true negatives and false negatives (i.e., numbers of negative examples from the test set classified correctly and incorrectly).

Accuracy Calculation:
True positive rate= TP/TP+FN
True negative rate= TN/TN+FP
Total accuracy = TP+TN/TP+TN+FP+FN =TP+TN/N
Positive predictive value=TP/TP+FP
Negative predictive value=TN/TN+FN
Sensitivity = true positive rate, Specificity = true negative rate

Following figures 14, 15, 16 gives the classification accuracy for each datasets with TP,FP,TN and FN values

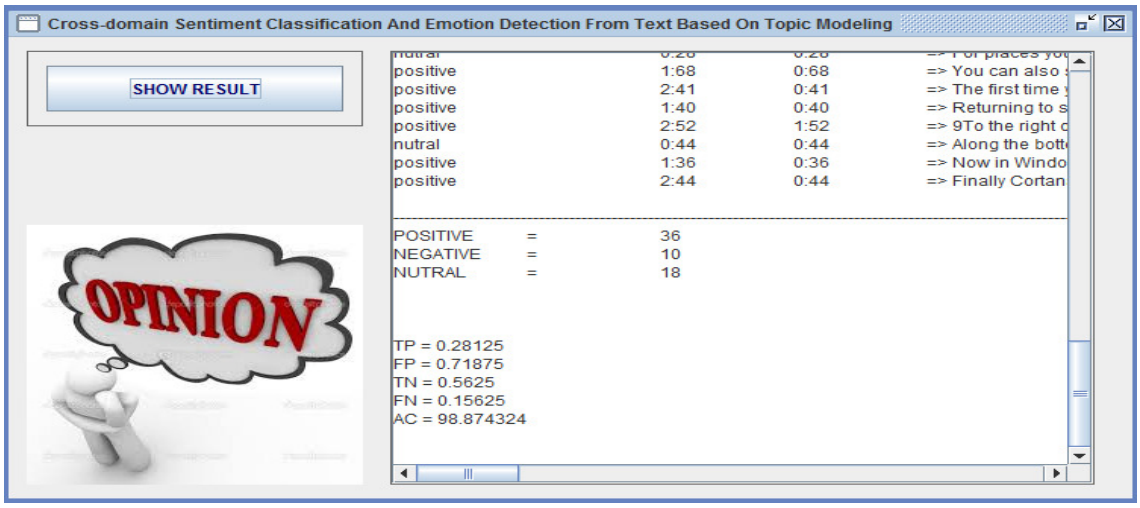

Fig. 14: Classification accuracy for movie dataset

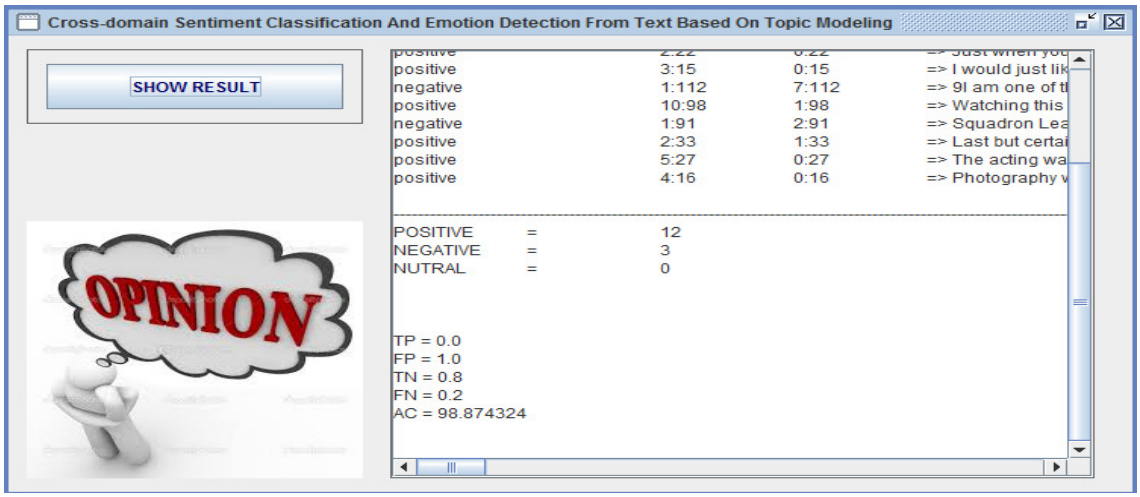

Fig.15: Classification accuracy for Product Review dataset

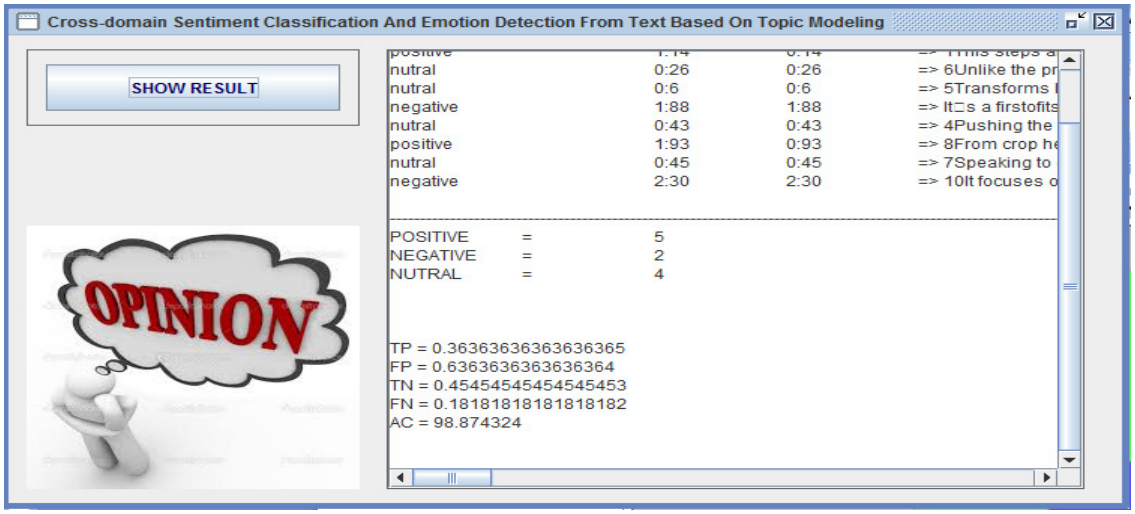

Fig. 16: Classification accuracy for Twitter dataset

Performance with dense and sparse datasets:

- In the frequent pattern mining the database is dense if frequent items in it have high relative support.
- The relative support = absolute support /no of transactions in the database.
- When relative support have high value such as 10% or more the projected database is said to be dense and when it is below 1% or less then the database is said to be sparse.
- Itemset tree structure works better in the dense database as compare to sparse database. Here it is beneficial because as mining progresses the projected database goes smaller making it denser. On another side H-mine works better with sparse dataset and gradually decreases performance as the dataset goes denser. Therefore for incremental datasets Itemset tree structure is better than H-mine.

4. Conclusion

In this paper, we have proposed the new algorithm with itemset tree data structure for determining product popularity in the market from the incremental customer review datasets. Latter the algorithm is implemented on review datasets and performance is calculated. Result shows that it works better than Naïve bayes algorithm. Again the result of comparison shows that proposed algorithm works better than H-mine algorithm in dense datasets while H-mine algorithm works better in sparse datasets. As mining progresses the projected database goes smaller making it denser.

Future Scope

This system can be implemented to work in real time scenario, for that it is necessary to implement the proposed model in cloud environment. We can compare the performance between cloud and non cloud environment on the basis of space required, cost and time.

References

[1] B. Liu, "OPINION MINING," In: Encyclopedia of Database Systems, 2004.

[2] R. Hemalatha, A. Krishnan and R. Hemamathi, "Mining Frequent Item Sets MoreEfficiently Using ITL Mining," in *3rd International CALIBER*, Ahmedabad, 2005.

[3] Ujwala mhashakhetri, Dr. Rahila Sheikh, "Frequent pattern Mining Implementation on Social network for Business Intelligence", in International Journal on Recent and Innovation Trends in Computing and Communication, Volume: 4 Issue: 5, May-2016.

[4] Rakesh Agrawal , Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules", In J.B. Bocca, M. Jarke, and C. Zaniolo, editors, Proceedings 20th International conference on Very Large Data Bases, pages 487-499. Morgan Kaufmann, 1994.

[5] JIAWEI HAN, JIAN PEI, YIWEN YIN, RUNYING MAO, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, 8, 53–87, 2004.

[6] Mohammed J. Zaki, "Scalable Algorithms for Association Mining", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 12, NO. 3, MAY/JUNE 2000.

[7] Mohammed J. Zaki and Karam Gouda, "Fast Vertical Mining Using Diffsets", presented at 9th International Conference on Knowledge Discovery & Data Mining, Washington, DC, 2003.

[8] Christian Borgelt, "Keeping Things Simple: Finding Frequent Item Sets by Recursive Elimination", published in Proceeding of 1st international workshop on open source data mining: Frequent pattern mining implementations, pages 66-70, 2005.

[9] JIAN PEI, JIAWEI HAN, HONGJUN LU, SHOJIRO NISHIO, SHIWEI TANG and DONGQING YANG, "H-Mine: Fast and space-preserving frequent pattern mining in large databases", IIE Transactions (2007) 39, 593–605.

[10] Takeaki Uno , Masashi Kiyomi , Hiroki Arimura, "LCM ver. 3: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets", FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004.

[11] DENG ZhiHong, WANG ZhongHui & JIANG JiaJian, "A new algorithm for fast mining frequent itemsets using N-lists", Sci China Inf Sci, 2012.

[12] Zhi-Hong Deng , Sheng-Long Lv, "Fast mining frequent itemsets using Nodesets", Applied Soft Computing 07/2015

[13] Miroslav Kubat, Alaaeldin Hafez, Vijay V. Raghavan, Jayakrishna R. Lekkala, " Itemset Tree for Targeted Association Querying", published in IEEE Transactions on Knowledge and Data Engineering Volume 15 Issue 6, Page 1522-1534, November 2003.

[14] Philippe Fournier-Viger , Eserance Mwamikazi , Ted Gueniche and Usef Faghihi, "MEIT: Memory Efficient Itemset Tree for Targeted Association Rule Mining", published in the 9th International Conference on Advanced Data Mining and Applications (ADMA 2013), At Hangzhou, China.

[15] Pravesh Kumar Singh, Mohd Shahid Husain, "METHODOLOGICAL STUDY OF OPINION MINING AND SENTIMENT ANALYSIS TECHNIQUES" in International Journal on Soft Computing (IJSC) Vol. 5, No. 1, February 2014