Text Mining Approach for Product Quality Enhancement

(Improving Product Quality through Machine Learning)

Chandrasekhar Rangu
HCL Technologies Ltd
Hyderabad, India

Shuvojit Chatterjee
HCL Technologies Ltd
Hyderabad, India

Srinivasa Rao Valluru
HCL Technologies Ltd
Hyderabad, India

Abstract— **Text mining, also referred to as text data mining, is the process of extracting interesting and non-trivial patterns or knowledge from text documents. It uses algorithms to transform free flow text (unstructured) into data that can be analyzed (structured) by applying Statistical, Machine Learning and Natural Language Processing (NLP) techniques. Text mining is an evolving technology that allows enterprises to understand their customers well, and help them in redefining customer needs. As e-commerce is becoming more and more established, the number of customer reviews and feedback that a product receives has grown rapidly over a period of time. For a popular asset, the number of review comments can be in thousands or even more. This makes it difficult for the manufacturer to read all of them to make an informed decision in improving product quality and support. Again it is difficult for the manufacturer to keep track and to manage all customer opinions. This article attempts to derive some meaningful information from asset reviews which will be used in enhancing asset features from engineering point of view and helps in improving the support quality and customer experience**

Index Terms— Text mining, Natural Language Processing (NLP), crawling, Support Vector Machine (SVM), Associate Rule Mining (ARM), Term Frequency (TF), Document Term Matrix (DTM), Stemming, Lemmatization.

I. INTRODUCTION

Internet users are growing exponentially for the last one decade. With rapid expansion of e-commerce, almost all the products are sold on the web. These days, customers gather complete information about the products (good and bad) of their interest from the web before making a purchase decision. This has enabled many of the customers in saving their time in identifying the right product at a comfortable price point which fulfills their needs along with additional features. In order to improve customer satisfaction and shopping experience, it has become a common practice for online merchants to enable their customers to review or to express opinions on the products that they have purchased. With more and more users becoming comfortable with the Web, an increasing number of people are writing reviews. As a result, the number of reviews that a product receives grows rapidly. Some popular products can get thousands of reviews at some large merchant sites. Furthermore, many reviews are long and have only a few sentences containing opinions on the product. This makes it hard for product manufacturers to keep track of customer opinions of their products.

The objective is to mine and summarize all the customer reviews of a consumer product. This would enable business to identify frequently reported issues and over a period of time the trend of the issues. This knowledge would ensure enhanced product and support quality. In our case the review comments are classified into three major categories, namely 1) Product related 2) Support related and 3) Others. Product related typically covers Design related issues and specific part functionality issues. Support related typically covers product Service-provider issues and other support level issues. All the remaining miscellaneous comments are classified into Others.

We have followed the below steps in deriving the insights:

1) Crawling product reviews from community sites / preferred social media sites
2) Applied text Pre-processing steps to mine the raw text
3) Preparing manually annotated review comments with the help of domain experts to build Text data classifier and then applying the classifier on the remaining comments to predict the target label.
4) Identifying Sub-Level Issues under each main category through Associate Rule mining by deriving key phrases/patters.

978-1-5090-1560-3/17 $31.00 © 2017 IEEE
DOI 10.1109/IACC.2017.92

5) Filtering the review comments by type and timestamp
6) Issues trend visualization w.r.t timestamp (monthly grouped)

In this research we study the processes of identifying the *issue-based summaries* from customer reviews crawled for *Product-X*. Here *issues* refer to sub-level issues of Product related and Support related problems reported from customer point of view. During the implementation process, we 'crawled' a set of review comments from a community website for the period of June-15 to Jan-16. Next, we prepared a training set of reviews manually that contains all three categories of reviews (Product, Support & Other) for training the text classifier. It is hard to get good classification results on textual data. We achieved reasonable classification accuracy in identifying the categories using *Support Vector Machines* technique. This classifier is used to classify the remaining review comments for further processing. For each category of reviews, we have identified few phrases and its associated words which actually describing the sub-issue details under the main category. For Example, Under Product related issues. We have identified sub-issues like *Feature /Component / Functionality* along with phrases like *Not-Working/Only Problem/Issue/doesn't work/error/major issue/ Problem.* There are very high possibilities to get junk words along with these phrases. This problem is reduced by doing proper preprocessing. The next step is choosing the best rule set with suitable support and confidence values. For each valid identified sub issue, we calculated the percentage of reviews that contain this problem and the results are grouped monthly to identify the trend of issues percent over a period of time.

II. OUR APPROACH

In a process to derive the target outcome from the unstructured raw text, the first step is to identify suitable data source. Under items of interest, we have noticed more user comments for *Product-X* in a community website with true opinions on the product performance and difficulties.

DATA EXTRACTION:

In general data extraction or crawling from different web pages or community sites is done through an API support. We have used a freely available API - KIMONO, a chrome extension for data extraction from community sites. It internally allows the user to prepare a generic API convenient to the data structure / fields (s)he is interested in the webpage. For this analysis we have chosen 6 fields from the data to prepare the API and then crawled all 400+ pages review comments with all the selected fields' information for each comment.

Each comment typically contains "*Subject","Username", "No of Likes", "Comment", "User type" and "DateTime"* fields. *Subject* gives the information in brief about the comment, *Username* is the name of the user who commented, *No of Likes* gives how many have actually liked the comment, *Comment* is the actual user review comment available in text, *User Type* has different variants available like *Explorer*, *Member*, *Contributor*, *New*, etc. and *DateTime* is the timestamp value of each comment.

Comment field is mainly used for deriving the insights, all the remaining are used for data summarization and visualization during the analysis.

TEXT PROCESSING:

Text pre-processing is the most important phase for any text mining task. It plays a crucial role in deriving the right patterns from the data. The typical text preprocessing involves Stop words removing, reducing the size of the vocabulary, handling special and unwanted characters. There are set of predefined stop words for English language under *tm* package in R. It contains 170 (approx.) unique list of stop words generally occurs in English language, in addition to this list we added few domain specific stop words which doesn't carry any meaning or value in the analysis. As user comment is a free flowing text, there is very high possibility to have so many special characters, punctuations, symbols, numeric values, white spaces and emoticons. Though Emoticons are the good way to express the user opinion in a short form, they need to be handled separately as they will convert to different encoding pattern during crawling. We have identified very few reviews with emoticons and cleaned them during preprocessing. We applied stemming and lemmatization to reduce the size of the vocabulary to ensure minimal corpus size.

We converted the processed text into structured format by creating TermDocumentMatrix (TDM). Next we calculated term frequencies and inverse document frequencies for each processed review.

After pre-processing was completed, we sorted the word list of each comment based on its term frequency and the top 10 words of each comment were chosen to build the classifier.

TEXT CLASSIFICATION:

In any classification task, it is important to have labels for the data in order to train the classifier. In our case we have taken the pre-annotated reviews covering both Product and Support related comments.

The goal of text classification is to identify the right label for each review comment in the test dataset. We have ensured that the training data representing all levels/categories of the target in order to attain better classification accuracy.

We have identified few cases where the number of words left in the original comment after pre-processing is very low (less than 6 words), which is insufficient to classify. As it is mandatory to have fixed input length to build a classifier, we tested SVM classifier with varying input lengths (from 8-12 words). And finalized at 10 words based on higher accuracy

levels. We have ensured that input length is of minimum 10 words after text processing and the reviews which are less than 10 words after pre-processing are not considered.

Three classification methodologies were tested with the data. (Support Vector Machines, Random Forest & Bayes)

Support Vector Machines has given best classification accuracy on this data. We have used Stratified random sampling to divide the train and test datasets and tried with different SVM kernels (radial/polynomial/linear/sigmoid) with optimal model parameters (gamma, cost, and degree) and achieved a classification accuracy of 69%.

With Bayes and Random forest, we have achieved classification accuracies of 57% and 59% respectively, but the models are more biased towards specific levels in the target variable.

ISSUE IDENTIFICATION:

We have identified few phrases manually after reading various review comments for main categories namely Product Related and Support Related. For each of the phrases we have taken the associated words to identify the sub-issue under the main category. For Example, Under Product Related Issues we have identified sub-issues like *issue1/issue2/issue3* along with phrases like *Not-Working /Only Problem /Issue /error /main problem/ doesn't work.*

There is a very high possibility of getting junk words along with these phrases. It is reduced by doing proper preprocessing in the initial phase and choosing the best rule set with suitable *support* and *confidence* values.

Below are the few review comments with identified phrases and corresponding sub-issues

Example 1: "My Product-X is **not working** most of the time, Unexpectedly the Feature-K goes off and it gets recovered only after restarting the device... "

Example 2: "I bought my Product-X 6 months back, initially it was working fine, no complaints in the initial 4 months but from 5th months onwards the problems are stated coming. The Component-N is **not working**, everything is fine with the physical condition "

Example 3:" I have the Product-X. I find it OK, to be fair we tend to use Feature-K most of the time and I find the upscaling excellent. As for the Feature-N it is **not working** as desired and facing repeated issues".

Example 4:" I'm using this device from last 1 year, screen blinking is the **only problem** I'm facing and rest is working as desired.

There are possibilities to have more number of sub level issues which can't be captured with a defined set of rules. These can be covered by defining more appropriate rules to capture all issues information.

With the above phrases combination, we have identified most commonly reported issues under each category.

III. ANALYTICAL RESULTS

For each identified issue under the two main categories Product and Support, we calculated the percentage of a particular identified issue occurrence using pattern match to know how frequently a particular issue is reported by users. This information is captured for all issues under Product and Support related and then grouped based on the *DateTime* to get the monthly trend.

There is possibility to have two or more issues reported in a single user comment, they will be accounted separately in each issue count in calculating the frequency.

We tabulated the frequency of each identified issues in the respective month and then converted them into a stacked bar plot by taking 'DateTime(Month)' on X-axis and 'Issues %' on Y-axis.

Monthly trend variation of Issues under Product related category in a stacked bar graph is given in Fig-1

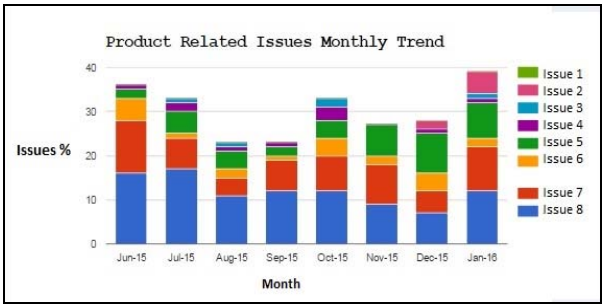

Fig-1

From Fig-1, it is clearly evident that Product-Issue-8 and Product-Issue-7 are reported in every month with little variation in the percentage. There are no Product-Issue-1 related problems reported till Nov-15, reported only in Dec and Jan-16, hence the support teams need to take special interest in coordinating with the customers to reduce or avoid Product-Issue-1 issues further.

Similarly, Monthly trend variation of Issues under Support related category in a stacked bar graph is given in Fig-2

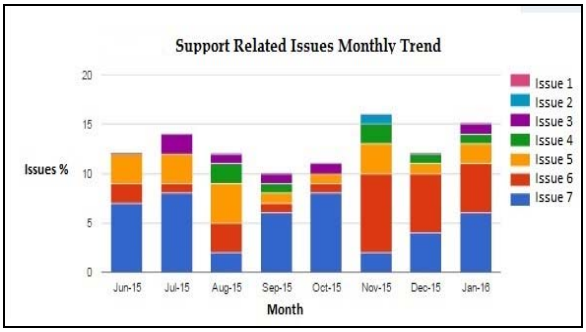

Fig-2

From Fig-2, we can observe that Support-Issue-7 and Support-Issue-6 issues are most commonly reported in every month with little variation in the percentage. Up to Oct-15 Support-Issue-6 related problems are reported very low in % but suddenly from Nov-15 to Jan-16 it has grown very high, here we can raise an alarm to the support team.

IV. CONCLUSIONS

In this paper we showcased different text mining techniques to identify issues of a product.

Related to Text preprocessing, we have used different Natural language processing techniques to reduce the vocabulary size and to build robust classifier. Along with SVM classifier we have tried other classification technique like Bayes, Random Forest and chosen SVM as best classifier for this kind of data.

We believe review summarization is very much important for product manufacturers to know customer opinion on different product features and helps in improving the market strategy as per customer needs. It also helps the support teams in identifying most frequent issues facing, there by some customizations can be made to resolve them in quick time.

The limitations of this approach is that the classification model expects minimum 10–15 words after removing non-usable characters during preprocessing. Hence, smaller size review comments are not taken into consideration. It is difficult to identify all the issues reported in different scenarios other than the defined key phrase list, and it needs minimal manual intervention in filtering out the junk. The key phrase list should be updated and refined well to capture all different problems.

In our future work, we plan to use more advanced text mining techniques, NLP algorithms to improve this mechanism and to make the process more robust and dynamic. So far the work is concentrated on static data sets, to make it real time we are planning a mechanism to crawl real time comments information and mine them on the fly to derive the insights.

Finally, we believe that analyzing user feedback is useful for product makers in understanding their customers well and to stay focused on customer needs in comparison with other market competitors.

V. ACKNOWLEDGEMENTS

We Sincerely thank Mr. Kumar G.N and Mr. Kamesh J.V for their continuous support and timely inputs. We would also like to thank our colleagues who encouraged us during this journey.

REFERENCES

1. Agrawal, R. & Srikant, R. 1994. Fast algorithm for mining association rules. VLDB'94, 1994.
2. Boguraev, B., and Kennedy, C. 1997. Salience-Based Content Characterization of Text Documents. In Proc. of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization
3. Manning, C.D., Sch¨utze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, Massachusetts (1999)
4. Bruce, R., and Wiebe, J. 2000. Recognizing Subjectivity: A Case Study of Manual Tagging. Natural Language Engineering
5. Daille, B. 1996. Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. The Balancing Act: Combining Symbolic and Statistical Approaches to Language. MIT Press, Cambridge
6. Dave, K., Lawrence, S., and Pennock, D., 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. WWW'03.
7. Tait, J. 1983. Automatic Summarizing of English Texts. Ph.D. Dissertation, University of Cambridge.
8. Bhumika , Prof Sukhjit Singh Sehra and Prof Anand Nayyar. A Review Paper On Algorithms Used For Text Classification- (Ijaiem)-2013
9. Mita K. Dalal, Mukesh A. Zaveri- Automatic Text Classification: A Technical Review, International Journal of Computer Applications (0975 – 8887)-Aug-2011

10. Ahmed Faraz- An Elaboration Of Text Categorization And Automatic Text Classification Through Mathematical And Graphical Modelling- An International Journal (CSEIJ), Vol.5, No.2/3, June 2015

11. Thorsten Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features- University at Dortmund Informatik LS8, Baroper Str. 301 44221 Dortmund, Germany

12. Thien Hai Nguyen and Kiyoaki Shirai, Text Classification of Technical Papers Based on Text Segmentation- Japan Advanced Institute of Science and Technology.

13. Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun, Tom Mitchell- Text Classification from Labeled And Unlabelled Documents Using Expectation Maximization,1998-99

14. Menaka S, Radha N- Text Classification using Keyword Extraction Technique, International Journal of Advanced Research in Computer Science and Software Engineering- Dec 2013

15. Timothy P. Jurka, Loren Collingwood, Amber E. Boydstun, Emiliano Grossman, and Wouter van Atteveldt- RTextTools: A Supervised Learning Package for Text Classification