International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 2 | ISSN : 2456-3307

Analysing the Social Data Opinion through Public User Raw Information

T. Sakthisree, Dhivya N, Nithyananthan R, Pavithira T

Computer Science and Engineering, Kathir College of Engineering, Coimbatore, Tamil Nadu, India

ABSTRACT

The social network perspective provides a set of methods for revealing the structure of social networks as well as a variety of hypothesis explaining the patterns discovered in these structures. The study of these structures uses social network discovering to recognizing local and global patterns; locate influential entities, and proficiency network dynamics. Millions of users share their opinions on Social Networks, making it a valuable platform for tracing and analyzing public sentiment. Such tracking and analysis can provide critical information for decision making in various domains. Therefore it has captivated attention in both academia and industry. This approach needs Sentimental data analysis model using Neural Networks. Both positive and negative also comments will be calculated here. To further enhance the readability of the mined reasons, we select the most representative tweets for foreground topics and develop another generative model called Reason Candidate and Background LDA (RCB-LDA) to rank them with respect to their popularity within the variation period. Experimental results show that our methods can effectively find foreground topics and rank reason candidates.

Keywords : LDA, RCB-LDA, KDD, CVS, SVN, ANY, ANN

I. INTRODUCTION

Data mining for software engineering techniques consists of gathering software engineering data, extracting some knowledge from it and, if possible, use this knowledge to improve the software engineering process, in other words “operationalize” the mined knowledge. For instance, researchers have extracted usage patterns from millions of lines of code of the Linux kernel in order to find bug. In essence, data mining for software engineering can be decomposed along three axes: the goal, the input data used, and the mining technique used.

Data engineering at large consists of many tasks from specification, design, development, monitoring at runtime, etc. Each task is itself decomposed in many smaller scale tasks. For example, a programmer constantly switches between tasks, such as navigating code, reading documentation, writing code, debugging, etc. During the last decade, it has been shown that most software engineering tasks can benefit from data mining approaches, the tasks being whether technical or more people oriented. Data mining is to discover structure inside unstructured data, extract meaning from noisy data, discover patterns in apparently random data, and use all this information to better understand trends, patterns, correlations, and ultimately predict customer behavior, market and competition trends, so that the company uses its own data more meaningfully to better position itself on the new waves. The term Knowledge Discovery in Databases (KDD) is generally used to refer to the overall process of

discovering useful knowledge from data, where data mining is a particular step in this process.

The objective of software engineering process in its entirety manipulates all kinds of data. Certainly, one thinks of code, but there are also many written documents (specifications, documentation), design documents (diagrams, formulas), runtime documents (logs), etc. Most of them can be versioned using a Version Control System (e.g., CVS, SVN, Git). Depending on the targeted goal, some artifacts are more or less appropriate, and blended approaches are possible (using different kinds of software engineering artifacts in conjunction). Also, there is usually a fair amount of pre-processing that is specific to the artifacts under consideration: natural language processing for written documents, static analyses for code, etc.

II. METHODS AND MATERIAL

Proposed System

Users of decision support systems often see data in the form of data cubes. The cube is used to represent data along some measure of interest. Therefore called a "cube", it can be two-dimensional, three-dimensional, or higher-dimensional. Each dimension represents some attribute in the database and the cells in the data cube represent the measure of interest. For instance, they could contain a count for the number of times that attribute combination occurs in the database, or the minimum, maximum, sum or average value of some attribute. Queries are performed on the cube to retrieve decision support information.

In case a database that contains transaction information relating company sales of a part to a customer at a store location. The data cube formed from this database is a 3-dimensional representation, with each cell (p, c, s) of the cube representing a combination of values from part, customer and store-location. The contents of each cell are the count of the number of times that specific combination of values occurs together in the database. Cells that is displayed blank in fact have a value of zero. The cube can then be used to retrieve information within the database about, for example, which store should be given a certain part to sell in order to make the greatest sales. A data cube built from m attributes can be stored as an m-dimensional array. Each element of the array contains the measure value, such as count. The array itself can be represented as a 1-dimensional array. For example, a2-dimensional array of size x x y can be stored as a 1-dimensional array of size x*y, where element (i,j) in the 2-D array is stored in location (y*i+j) in the 1-D array. The disadvantage of storing the cube directly as an array is that most data cubes are sparse, so the array will contain many empty elements (zero values). Rollup or summarization of the data cube can be done by traversing upwards through a concept hierarchy. A concept hierarchy maps a set from a low level concepts to higher level concepts, more general concepts, where can be used to summarize information in the data cube. As the values are combined, cardinalities shrink and the cube gets reduced. Explanation can be thought of as computing some of the summary total cells that contain ANYs, and storing those in favor of the original cells.

If your source data is been used in a star or snowflake method, then you already have the elements of a dimensional model:

- Fact tables correspond to cubes.
- Data columns in the fact tables correspond to measures.
- Foreign key principals in the fact tables identify the dimension tables.
- Dimension tables identify the dimensions.
- Also Primary keys in the dimension tables identify the base-level dimension members.
- Such as Parent columns in the dimension tables identify the higher level dimension members.

Columns in the dimension tables containing descriptions and characteristics of the dimension members identify the attributes. Also get insights into the dimensional model by looking at the reports currently being generated from the source data. The reports will analyze the levels of sum that interest the report consumers, as well as the attributes used to qualify the data. While investigating your source data, you may decide to create relational views that more closely match the dimensional model that you plan to create.

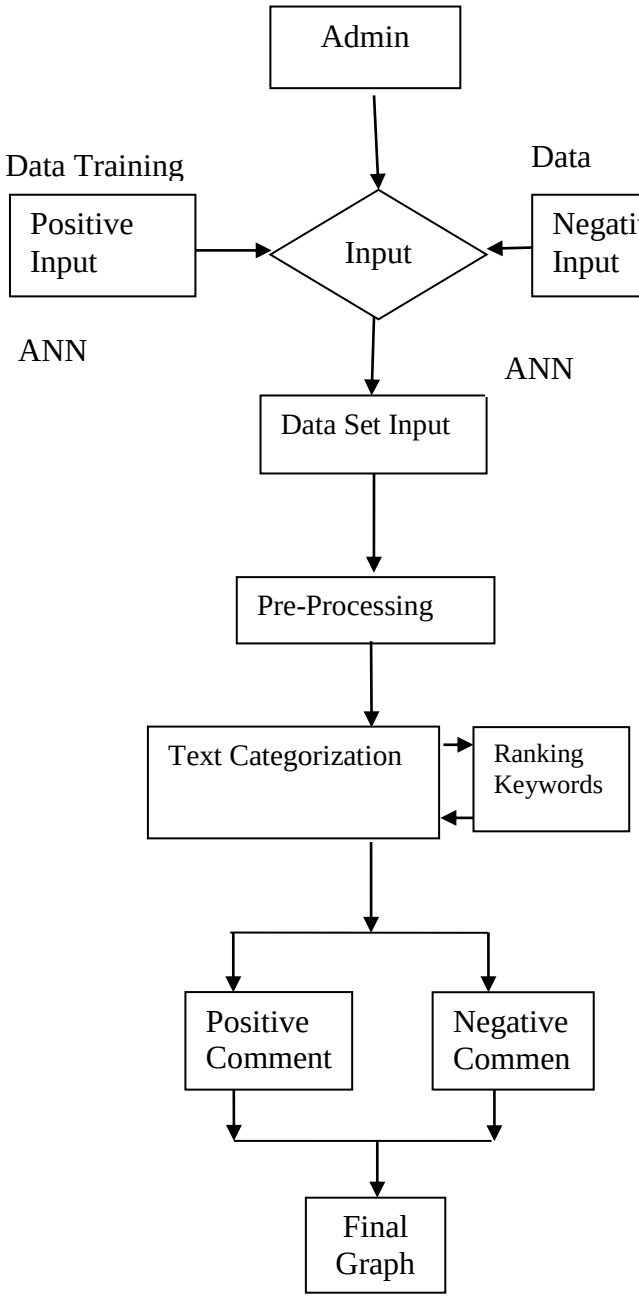

Figure 1. Architecture Diagram

Advantages of Proposed System

- No manual works has been used. The Sentimental Pattern recognition technique has been used for generating various sentimental classifications.
- No repetition allowed due to the implantation of clustering and classification methods and data accuracy verification.
- A well enhanced ranking method has been implemented for improving the data clarity like, establishing the most used keywords and frequent used keywords.
- Even sentiments for the ranking keywords can be generated.
- All results will be shown in graphical formats like graphs and charts.

PROBLEM DEFINITION

Unsupervised Cross-domain Sentiment Classification is the task of adapting sentiment classifier trained on a particular domain (source domain), to a different domain (target domain), without requiring any labelled data for the target domain .By adapting an existing sentiment classifier to previously unseen target domains, we can completely avoid the cost for manual data for the target domain. We model this problem as embedding learning, and construct here objective functions that capture: (a) distributional properties of pivots (example. common features that appear in both source domains and target domains),(b) label constrains in the source domain documents, and,(c) geometric properties in the unlabeled documents in both source domains and target domains. Unlike past proposals that first learn a smaller-dimensional embedding independent of the source domain sentiment labels, and next a sentiment classifier in this embedding, our joint optimization method learns embedding's that are sensitive to sentiment classification. Experimental results on a benchmark datasets how that by jointly optimizing the three objectives we can determine a better performances in comparison to optimizing each objective function separately, there by demon starting the importance of

task-specific embedding learning for cross-domain sentiment analyzes. Among the individual objective functions, the best performance is obtained by (c).Moreover, the proposed method reports cross-domain sentiment classification determines that are statistically comparable to the current state of the art embedding learning techniques for cross-domain sentiment classification.

MODULE DESCRIPTION

- Social Network pattern Creation
- Centralizing the data
- Analysis Model
- Sentimental data analysis model
- Global Patterning report

Social Network pattern creation

A Social Network pattern is a web application that is accessed over a network such as the Internet or an intranet. The term may also mean a computer software application that is hosted in a browser-controlled environment (e.g. Ajax) or coded in a browser-supported language (such as JavaScript, combined with a browser-rendered markup language like HTML) and reliant on a common web browser to provide and to the application executable. Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a client, sometimes called a thin client. The capacity to update and maintain the web applications without distributing and installing software on potentially thousands of client computers is a key reason for their popularity, as is the inherent which is the main support for cross-platform compatibility. Common web applications include web mail, online chatting, image sharing, online retail sales, online auctions and many other functions.

Centralizing the data

Here all data will be uploaded in to a centralized server for data analysis purpose. Centralized data distribution systems defined here as systems which allow distributed end-user applications, databases and data providers to be collabrate with dedicated data sources. Such systems are used in government and commercial organizations with highly distributed structures dealing with on-line information. The analyzed data will be updated in the centralized server, in order to access the data through web server or through ant centralized server. Even thou this project will be implemented using ASP.NET, basically it will satisfies all the web based procedures and applications. In case of the organization may having more than two branches in various locations, (i.e.) in different states. The centralized data will be shared from various locations and the key person, Company account, Existing system and basic details can be shared. This may improve the project quality and code reusability possibilities.

Analysis Model

According to the sentimental data variations, analysis model will be more important. Here in order to implement the foreground and background data verification, neural networks will be used for data training purpose. According to our process two types of word category will be stored in the database. Case 1 will be the happy or fair words and case 2 will be sad, angry words. Case 3 will be the calculated from the exception words from case 1 and 2 which is said to be the moderate words.

Sentimental Data analysis Mode

This is the most important module in this project; here pattern recognition is implemented for sentimental data analysis. The pattern recognition method also hand shacked with clustering and classification methods. These methods will analysis the input data from the data set. And in case of customized social networks, it will analyze online based data also. Each sentence will be analyzed with pattern recognition methods. So that all the sentimental words will be compared accordingly. Repeated comments will be omitted and clustered for fined tuned data report

Global Patterning Report

This module will produce graph patterns with more analyzed data set. All the output data from the previous module will be given as input in this module. So that we can produce various types of charts and graphs. Duplication will be avoided with more information results. A chart will generated for positive, negative and neural comments. Number of comments and number of repeated comments can be generated. Copied comments can be shown. In case of topic based discussion, that topic also can be shown.

III. CONCLUSION

Classification is very essential to organize data, retrieve information correctly and swiftly. Implementing machine learning to classify data is not easy given the huge amount of heterogeneous data that's present in the web. Text categorization algorithm depends entirely on the accuracy of the training data set for building its decision trees. The text categorization algorithm learns by supervision. It has to be shown what instances have what results. Due to this text categorization algorithm, it cannot be successfully classify the documents in the web. The data in the web will be not predictable, volatile and most of it lacks Meta data. The way forward for information retrieval in the web, in the future opinion would be to advocate the creation of a semantic web where algorithms and the techniques which are unsupervised and reinforcement learners are used to classify and retrieve data.

Thus the thesis explains the trends, threads and process of the text categorization algorithm which was implemented for finding the sensitive data analysis.

REFERENCES

[1] H. Becker, M. Naaman, and L. Gravano, "Learning similarity met-rics for event identification in social media," in Proc. 3rd ACM WSDM, Macau, China, 2010.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," J. Mach. Learn. Res., vol. 3, pp. 993–1022, Jan. 2003. TAN ET AL.: INTERPRETING THE PUBLIC SENTIMENT VARIATIONS ON TWITTER 1169

[3] J. Bollen, H. Mao, and A. Pepe, "Modeling public mood and emo-tion: Twitter sentiment and socio-economic phenomena," inProc. 5th Int. AAAI Conf. Weblogs Social Media, Barcelona, Spain, 2011.

[4] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market,"J. Comput. Sci., vol. 2, no. 1, pp. 1–8, Mar. 2011.

[5] D. Chakrabarti and K. Punera, "Event summarization using tweets," in Proc. 5th Int. AAAI Conf. Weblogs Social Media, Barcelona, Spain, 2011.

[6] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," CS224N Project Rep., Stanford: 1–12, 2009.

[7] T. L. Griffiths and M. Steyvers, "Finding scientific topics," inProc. Nat. Acad. Sci. USA, vol. 101, (Suppl. 1), pp. 5228–5235, Apr. 2004.

[8] D. Hall, D. Jurafsky, and C. D. Manning, "Studying the history of ideas using topic models," inProc. Conf. EMNLP,Stroudsburg, PA, USA, 2008, pp. 363–371.

[9] G. Heinrich, "Parameter estimation for text analysis," Fraunhofer IGD, Darmstadt, Germany, Univ. Leipzig, Leipzig, Germany, Tech. Rep., 2009.

[10] Z. Hong, X. Mei, and D. Tao, "Dual-force metric learning for robust distracter-resistant tracker," in Proc. ECCV, Florence, Italy, 2012.

[11] M. Hu and B. Liu, "Mining and summarizing customer reviews," in Proc. 10th ACM SIGKDD, Washington, DC, USA, 2004.

[12] Y. Hu, A. John, F. Wang, and D. D. Seligmann, "Et-lda: Joint topic modeling for aligning events and their twitter feedback," inProc. 26th AAAI Conf. Artif. Intell., Vancouver, BC, Canada, 2012.

[13] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in Proc. 49th HLT, Portland, OR, USA, 2011.

[14] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in Proc. 15th ACM SIGKDD, Paris, France, 2009.

[15] C. X. Lin, B. Zhao, Q. Mei, and J. Han, "Pet: A statistical model for popular events tracking in

social communities," in Proc. 16th ACM SIGKDD, Washington, DC, USA, 2010.
[16] F. Liu, Y. Liu, and F. Weng, "Why is "SXSW" trending? Exploring multiple text sources for twitter topic summarization," inProc. Workshop LSM, Portland, OR, USA, 2011.
[17] T. Minka and J. Lafferty, "Expectation-propagation for the gener-ative aspect model," inProc.18th Conf. UAI, San Francisco, CA, USA, 2002.
[18] G. Mishne and N. Glance, "Predicting movie sales from blogger sentiment," inProc. AAAI-CAAW, Stanford, CA, USA, 2006.
[19] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," inProc. 4th Int. AAAI Conf. Weblogs Social Media, Washington, DC, USA, 2010.
[20] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Found. Trends Inform. Retrieval, vol. 2, no. (1–2), pp. 1–135, 2008