Sentiment Analysis on Online Social Network Using Probability Model

Hyeoncheol Lee, Youngsub Han
Department of Computer and Information Sciences
Towson University
Towson, USA
hlee23, yhan3@students.towson.edu

Kwangmi Ko Kim, Ph.D.
Department of Mass Communication
Towson University
Towson, USA
kkim@towson.edu

***Abstract*—Sentiment analysis is to extract people's opinion and knowledge from text messages. Recently, demands on automated sentiment analysis tool for text messages generated from web have dramatically increased and the literature on this topic has been growing. In this paper, we propose a semi-automated sentiment analysis method on online social network using probability model. The proposed method reads sample text messages in a train set and builds a sentiment lexicon that contains the list of words that appeared in the text messages and probability that a text message is positive opinion if it includes those words. Then, it computes the positivity score of text messages in a test set using the list of words in a message and sentiment lexicon. Each message is categorized as either positive or negative, depending on threshold value calculated using a train set. To check the accuracy, we compared the sentiments of the proposed method with sentiments of human coders. This research is unique and novel in that it guarantees high accuracy rates and does not require additional information, such as users' profile and network relationship.**

Keywords-sentiment analysis; social network.

I. INTRODUCTION

In recent years, online social network sites, such as Facebook, Twitter, Blogger, LinkedIn, YouTube and MySpace, have changed the way people communicate with each other. People share information, report news, express opinions and update their real-time status on the online social network sites. With the increasing popularity of the online social network sites, a huge amount of data is being generated from them in real time. Analyzing the data in social media can yield interesting perspectives to understanding individuals and human behavior, detecting hot topics, and identifying influential people, or discovering a group or community [10][11].

Several user-generated text messages contain users' emotional state and mood about topics, such as events, products, and services. Sentiment analysis is to extract the users' opinion and knowledge from the text messages [12][13]. Recently, automatic sentiment analysis on online social network has received a lot of attention from researchers. Most approaches focus on identifying whether a text expresses positive or negative opinion about a topic [13][14]. The high volume of such data has called for automated tools that assign positive or negative for much easier and quicker analysis.

In spite of high demands for automatic sentiment analysis on text messages in online social network data, the development of the automatic sentiment analysis faces some challenges as the text messages in online social networks are unstructured, unlabeled, dynamic and noisy [2][15]. Due to the characteristics of the messages, accuracy of previous automatic sentiment analysis approaches remains around 80%, which should be further improved for more accurate analysis. In addition, some existing approaches require additional information, such as user's tendency or relationship, which is not always available on online social networks. For these reasons, we propose a sentiment analysis algorithm that guarantees higher accuracy than existing approaches and can be used broadly in any social network sites without requiring additional information.

The rest of the sections are organized as follows: In Section 2, the related works on sentiment analysis are summarized. Section 3 outlines main methodology of sentiment analysis we propose, and Section 4 presents experiment results. Section 5 concludes our works and gives direction to future research.

II. RELATED WORKS

There are two main approaches to extracting sentiment from text messages. The first approach is lexicon-based sentiment analysis which is found on pattern matching with pre-built lexicon. Many researches tried to extract sentiment or opinion from text messages using this approach [1][17][18][19]. O'Connor et al. [1] analyzed political opinion using a sentiment analysis algorithm. They collected text messages related to political opinion from Twitter from 2008 to 2009. Also, they built a lexicon where each word was categorized as either positive or negative keywords based on OpinionFinder [3]. The number of positive and negative keywords was counted for every message. A message is defined as positive if it contains any positive word, and negative if it contains any negative word. As a result, the ratio of positive messages versus negative messages was compared with survey results and it showed

 ISBN: 978-1-61208-377-3

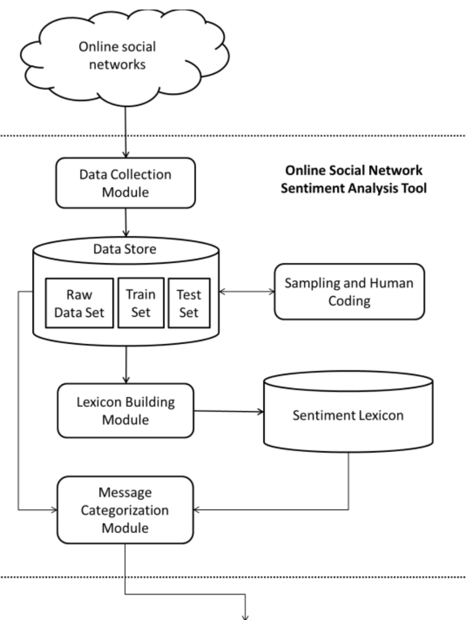
data correlation between results of sentiments analysis and survey is as high as 80%. The results indicate that the method can be used as a supplement for traditional survey. However, this lexicon based approach has weakness in that a message including positive keywords does not necessarily yield positive opinion. For instance, a word *like i*s categorized as a positive word in the lexicon, meaning if a message includes the word *like*, it is categorized as a positive message. Nevertheless, if the message includes the word *don't* right before *like*, the actual opinion of message should be categorized as negative. In this sense, such lexicon-based approach should be improved regarding the nature of language. The second approach is classification-based sentiment analysis, also known as supervised classification. It builds a sentiment classifier using a train set that contains labeled texts or sentences and test new texts using the classifier. Statistical and machine learning techniques can be used in this approach. Bayesian modeling approach has proven to be a capable method for multi-class sentiment classification and multi-dimensional sentiment distribution predictions [5]. Machine learning techniques, such as Naïve Bayes (NB), Support Vector Machines (SVM), Maximum Entropy, Decision Tree and K-Nearest Neighbor Classifier have been shown to be effective methods for sentiment analysis of messages [6][16].

Some of sentiment analysis approaches examine message author's information or behavior. Guerra et al. [2] proposed a sentiment analysis algorithm using bias of social media users toward a topic. They posit users tend to express their opinion multiple times and a user's bias tends to be more consistent over time as a basic property of human behavior. Thus, they measured bias of social media users toward a topic and analyzed sentiment by transferring users biases into textual features. Kucuktunc et al. [7] also proposed a method of analyzing sentiment based on characteristics of users, such as gender, age and education level. However, these methods cannot be broadly used because it requires relationship data among users and previous messages that the users have posted, which are not always provided by social networks due to the privacy laws.

Speriosu et al. [4] applied label propagation (LPROP) approach based on graph representation to analyze sentiment of messages in Twitter. Their assumption is that each tweet written by a user is linked to other tweets written by the same user, and each author is influenced by the tweets written by users whom he or she follows. They represented such a relationship using a graph where the features of the message, such as words, emoticon and authors, are inter-related to each other. Those features affect positivity or negativity of the message in the graph. They tested the accuracy of the LPROP approach with messages in four different topics and compared it with the accuracies of other approaches. The results show that accuracy of the proposed LPROP approach is the highest among other sentiment analysis approaches as it reached 65.7% to 84.7%, depending on the topics. However, there is a room for improving the accuracy of the LPROP because its average accuracy is still 72.08%.

III. METHODOLOGY

In this section, we describe the methodology of sentiment analysis for text messages generated from web. Figure 1 shows the overall process of sentiment analysis on text messages.



Messages	Positivity Score	Sentiment
I love the commercial	0.9	Positive
Well done	0.72	Positive
This is gross	0.40	Negative
...	...	...

Figure 1. Overall Process of Sentiment Analysis Tool

First, data collection module collects text messages from online social networks, such as Twitter and YouTube, which will be saved as a raw data set in data store. Then, it generates *sample* text messages from the data store and human coders categorize the messages into positive or negative opinions. The categorized *sample* messages are saved into a train set in the data store. After that, lexicon building module scans all categorized *sample* messages in the train set and calculates the weighted probability that the message is positive opinion if the word is included in a message. The list of words and the probabilities for each of them are saved in sentiment lexicon. Finally, the message categorization module calculates positivity scores for every message and categorizes whether the messages are positive or negative. To check the accuracy of the proposed method, we generated a test set which is also categorized in the same way the train set is made. Details of methodologies are explained in later on this section.

A. Data Collection Module and Datasets

Several social networks allow us to collect data with Application Programming Interface (API) [8][9]. For example, YouTube provides us with API to collect the data in YouTube. The main purpose of the YouTube API is to integrate the functionalities of YouTube into software applications. In addition to the main functionalities, the API allows developers to collect every kind of YouTube data, such as video information, user profile, and comments. Twitter also provides us with API for data collection.

In this research, we have developed a data collecting tool that automatically collects comments posted on YouTube videos. We have selected 3 commercial videos: *Prom* (for Audi), *Farmer* (for Ram) and *Perfect match* (for Go Daddy) that aired during the Super Bowl Game in 2013 which created a lot of buzz on online social networks. Then, we collected all comments that were posted on the videos using the tool. The comments are saved into a raw data set in data store.

B. Sampling and Human Coding

Among all comments, we randomly selected a total of 3,000 comments, 1,000 comments for each video. The comments were categorized as positive or negative by human coders. Two graduate students were involved in the coding process. We built a data *sample* using the messages that both human coders categorized into the same sentiment. In this process, we excluded messages that have neutral or mixed opinions that have both positive and negative opinions in the sample message. The categorized messages are saved into a train set in data store.

C. Building Sentiment Lexicon

Once sample messages are categorized by human coders and saved into the train set in data store, lexicon building module generates sentiment lexicon. It consists of word, the number of occurrence in positive messages, and the number of occurrence in negative messages and probability that a message is positive opinion if it contains the word, which will be used as base resource to categorize sentiment of messages in message categorization module.

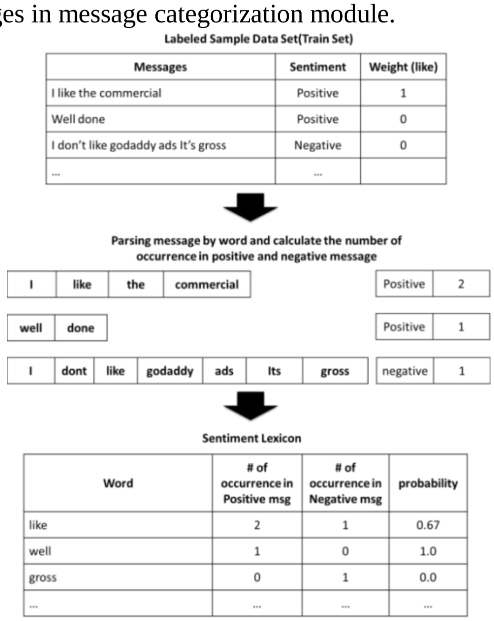


Figure 2. Example of Building Sentiment Lexicon using Labeled Sample Data Set(Train Set)

The process of building sentiment lexicon is as follows. First, it reads a message in the train set. Then it parses the message by word and checks the labeled sentiment and weight. In the comments of YouTube, a user can add a *like* or *dislike* tag, indicating the degree of user's agreement on the message. We use the tags as a weight point. The number of occurrence for every word in positive and negative messages are counted and saved into sentiment lexicon. Finally, the probability that the message is positive opinion if it includes the word is computed for every word and saved into sentiment lexicon.

Figure 2 shows the overall process and example of building sentiment lexicon using labeled sample data set (train set). Assume there are 3 messages in a train set and each message is labeled as shown in figure 2. If the word *like* appears in a message labeled positive opinion, the number of occurrence in positive opinion for the word is increased by one. If the labeled message has a like tag, the number of occurrence in positive opinion for the word is increased by two. If the word *like* appear in positive opinion twice and negative opinion once, the probability that a message is positive will be 0.67 if the message includes the word *like*.

D. Categorize the comments

Once sentiment lexicon is built completely, message categorization module classifies a text message into a positive or negative opinion. The comment sentence is represented with vector space model (VSM), where each word in the message and its probability in sentiment lexicon are shown together. Then the positivity score of a document (comment) is computed as follows.

$$\text{Positivity Score (d)} = \frac{\sum_{i=1}^{n} P(wi)}{n} \qquad (1)$$

In (1), w is each word in a document d and n is the number of words in the document. P is probability of the word which is saved in sentiment lexicon with the word. Example of computing positivity score for a comment is visualized in Figure 3.

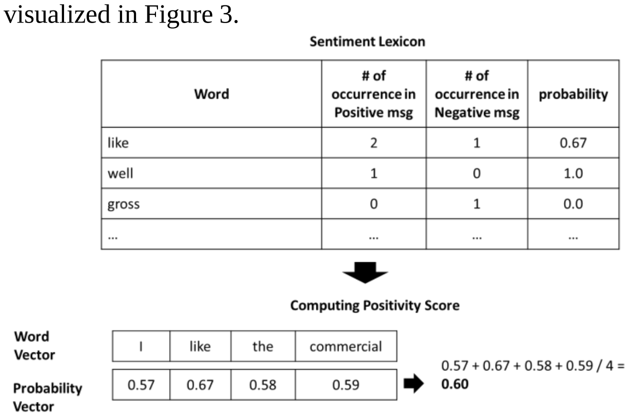


Figure 3. Example of computing positivity score

Once the positivity scores of all comments in train set are computed, message categorization module reads them again and computes the threshold of positivity score to classify the comment as either a positive or negative message. The threshold value is derived by computing mean value of positivity scores for all positive and negative messages in the train set. The example of computing threshold value is visualized in Figure 4.

Positive Messages

message	Positivity score
I like the commercial	0.6
Well done	0.75
This is my favorite commercial	0.68
...	...
Average	0.706

Negative Messages

Message	Positivity score
I hate it	0.23
This is disgusting	0.49
Eww it's gross	0.43
...	...
Average	0.4514

Threshold = (Average of positivity scores in positive messages + Average of positivity scores in negative messages) / 2 = (0.706 + 0.4514) / 2 = **0.5787**

Figure 4. Computing Threshold using positivity scores of positive and negative messages

The last step of sentiment analysis is to categorize messages in the test set using the threshold. The positivity score of each comment in the test set is computed in the same way as the previous step in the message categorization module. Then, it classifies the comment as either a positive or negative message. If the positivity score is greater than the threshold, it is categorized as a positive message. Similarly, if the positivity score is less than the threshold, it is categorized as a negative message. The example of classifying sentiment of comments is visualized in Figure 5.

Test Set

Threshold = 0.58

I	like	the	ads
0.57	0.67	0.58	0.61

0.51 + 0.67 + 0.55 + 0.61 / 4 = 0.61

Positive opinion

Eww	It's	gross
0.09	0.52	0.0

0.09 + 0.52 + 0.0 / 3 = 0.20

Negative opinion

Figure 5. Example of Classifying Sentiment of Comments

Suppose a message "I like the ads" is given as shown in the Figure 5. Each word in the message is represented with VSM and the probabilities are assigned to each word (I:0.57, like:0.67, the:0.58 and ads:0.61). Then, the positivity score is computed according to the (1) and compared with the threshold value. Since the positivity score 0.61 is greater than 0.58, the message is classified as positive opinion. In the similar way, the positivity score of the second message "Eww it's gross" is computed, compared with the threshold, and classified as negative opinion.

IV. EXPERIMENTS

This section presents the experiment results of the sentiment analysis method we proposed.

A. Data Collection

Table 1 shows data collection results. We collected the video information and comments posted under the video on May 26, 2014 using the data collection tool introduced in the previous section. Video ID is an identification key generated by YouTube. We collected a total of 25,003 comments for the videos.

TABLE I. DATA COLLECTION RESULTS

Video Title	Comments Count
Official Ram Trucks Super Bowl Commercial "Farmer"	16683
Audi 2013 Big Game Commercial - "Prom"	2977
Go Daddy Bar Refaeli Kiss Super Bowl Commercial 2013 - FULL	5343

For each video, 1,000 comments are selected and used for building the sentiment lexicon and pre-processing the train data as described in the previous sections.

B. Sentiment Lexicon

Table 2 is part of sentiment lexicon. Every word appeared in the comments is saved in the first column of sentiment lexicon. The number of word occurrence in positive and negative messages is recorded in the second and third column with the words. The probability that a message is positive if it contains the word is computed using the words occurrence in positive and negative messages and is saved in the last column. As a result, sentiment lexicon was built with total of 739 words with the probability.

TABLE II. SENTIMENT LEXICON

Word	The number of occurrence in positive message	The number of occurrence in negative message	Probability
love	41	0	1
great	35	5	0.87
car	23	4	0.85
pretty	9	2	0.81
all	33	8	0.8
good	17	6	0.73
dad	8	3	0.72
prom	13	5	0.72
my	50	34	0.59
make	11	9	0.55
me	25	22	0.53
not	26	29	0.47
stupid	5	6	0.45
never	6	8	0.42
kiss	5	9	0.35
why	4	8	0.33
fuck	2	10	0.16
disgusting	1	13	0.07
awkward	1	13	0.07
gross	0	20	0

C. Sentiment Categorization Results

To show the accuracy of the proposed algorithm, we labeled a test set in the same way as the train set is built.

Then, the sentiments of comments derived by the proposed method are compared with the sentiments labeled by human coders as shown in Table 3. If human coders and the proposed method categorized a message into the same sentiment, the result is classified as *correct*. Otherwise, the result is classified as *incorrect.* The accuracy of the proposed method is computed as shown in the Figure 6. It shows that the accuracy of the proposed method is at 86%. However, the accuracy for the negative messages is relatively lower than the accuracy for positive messages, which needs to be considered and improved in the future research.

TABLE III. CLASSYFYING SENTIMENT OF COMMENTS AND COMPARING THE SENTIMENT BY THE PROPOSED METHOD WITH SENTIMET BY HUMAN CODERS

Text(Comment)	**Positivity Score**	**Sentiment by the proposed method**	**Sentiment by human coders**	**Results**
This was the best commercial! It was so powerful........	0.67	Positive	Positive	Correct
Whoever at Dodge decided to go with this ad is a Goddamn genius!	0.64	Positive	Positive	Correct
love love love!!!!!!	1.00	Positive	Positive	Correct
Just so touching and I loved this.	0.70	Positive	Positive	Correct
Ok so I think that just made me cry a little bit. That was beautiful	0.69	Positive	Positive	Correct
VERY uncomfortable and retarded	0.41	Negative	Negative	Correct
The sound effects though.. oh goshh eww (/.)	0.36	Negative	Negative	Correct
No. I hate it	0.47	Negative	Negative	Correct
AH!! MY EYES	0.59	Positive	Negative	Incorrect
This is DISGUSTING!	0.49	Negative	Negative	Correct

Test Set

Category	**Positive Message**		**Negative Messages**	
Topic	**Correct**	**Incorrect**	**Correct**	**Incorrect**
Audi	84	7	7	2
Dodge	80	6	7	7
Go Daddy	7	3	73	17
Total	171	16	87	26

Correct Message / Total Message = 258 / 300 = **86%**

Figure 6. Sentiment Analysis Results and Accuracy of the Proposed Method

To compare performance of the proposed method with other approaches, we applied F-measure that can be used to compute test's accuracy [13]. F-measure uses two measurement degrees; precision *p* and recall *r*. *p* is the number of correct results divided by the number of all returned results. *R* is the number of correct results divided by the number of results. The F1 score is calculated as shown in (2).

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{2}$$

TABLE IV. COMPARION OF F-SOCRE RESULTS

Method	**F1 score**
PANAS-t	0.737
Emoticons	0.948
SASA	0.754
SenticNet	0.810
SentiWordNet	0.789
SentiStrength	0.894
Happiness Index	0.821
LIWC	0.731
Proposed Approach	0.890

Table 4 shows results of F-measures. F1 score of our approach is 0.890 which is relatively higher than other approaches. However, it is lower than F1 score of Emoticons and SentiStrengh. Improving the accuracy needs to be considered in the future research.

V. CONCLUSION AND FUTURE RESEARCH

This research developed and proposed a sentiment analysis method using probability model that guarantees relatively higher accuracy than existing approaches with broader application. The result shows that it outperforms most existing sentiment analysis approaches in terms of accuracy. In addition, the proposed approach can be implemented only using text information without requiring any additional information. This proposed approach, however, has a limitation that requires preprocessing of sample text messages by human coders. We will investigate a fully automated sentiment analysis method in the next research, and continue to work on improving the accuracy rate of a proposed method.

REFERENCES

[1] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From Tweets to Polls:Linking Text Sentiment to Public Opinion Time Series", Proceedings of the International AAAI Conference on Weblogs and Social Media, May 2010, pp. 122-129.

[2] P. H. Guerra, A. Veloso, W. Meira, and V. Almeida, "From Bias to Opinion: A Transfer-Learning Apporach to Real-Time Sentiment Analysis", Proceddings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining(KDD 11), August 2011, pp. 150-158, ISBN:978-7-4503-0813-7.

[3] T. Winson et al., "OpinionFinder:A System for Subjectivity Analysis", Proceedings of HLT/EMNLP 2005 Interactive Demonstrations, October 2005, pp. 34-35, doi:10.3155/1225733.1225751.

[4] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldridge, "Twitter Polarity Classification with Label Propagation over

Lexical Links and the Follower Graph", Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing, July 2011, pp. 53-64, ISBN: 978-1-937284-13-8.

[5] Y. He, "A Bayesian Modeling Approach to Multi-Demensional Sentiment Distributions Predictions", Proceedings of the Frist International Workshop on Issues of Sentiment Discovery and Opinion Mining, August 2012, Article No.1, ISBN:978-1-4503-1543-2.

[6] A. Sharma and S. Dey, "A Boosted SVM based Sentiment Analysis Approache for Online Opinionated Text", Proceedings of the 2013 Research in Adaptive and Convergent Systems, October 2013, pp. 28-34, ISBN: 978-1-4503-2348-2.

[7] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu, "A Larege Scale Sentiment Analysis for Yahoo! Answers", Proceedings of the fifth ACM international conference on Web search and data mining, February 2012, pp. 633-642, ISBN: 978-1-4503-0747-5.

[8] Google Developers, accessed on May 2014, <https://developers.google.com/youtube/>.

[9] Twitter Developers, accessed on May 2014, <https://dev.twitter,com>.

[10] I. King, J. Li, and K. T. Chan, "A brief survey of computational approaches in social computing". In IJCNN'09: Proceedings of the 2009 international joint conference on Neural Networks, pp. 2699–2706, Piscataway, NJ, USA, 2009. IEEE Press. ISBN:978-1-4244-3549-4.

[11] C. Byun, H. Lee, J. You, and Y. Kim, "Dynamic Seed Analysis in a Social Network for Maximizing Efficiency of Data Collection", Proceedings of the Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference, July 2013, pp. 132-136.

[12] A. Sharma and S. Dey, "A comparative study of feature selection and machine learing techniques for sentiment anlaysis", Proceedings of the 2012 ACM Research in Applied Computation Symposium, October 2012, pp. 1-7, ISBN:978-1-4503-1492-3.

[13] P. Goncalves, M. Araújo, F. Benevenuto, and M. Cha, "Comparing and combining sentiment analysis methods", Proceedings of the first ACM conference on Online social networks, October 2013, pp. 27-38, ISBN: 978-1-4503-2084-9.

[14] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification", Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, June 2009, pp. 1275-1284, ISBN: 978-1-60558-495-9.

[15] C. Byun, Y. Kim, H. Lee, and K. K. Kim, "Automated Twitter data collecting tool and case study with rule-based analysis", Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services, December 2012, pp. 196-204, ISBN: 978-1-4503-1306-3.

[16] A. Sharma and S. Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis", Proceedings of the 2012 Research in Adaptive and Convergent Systems, October 2012, pp. 1-7, ISBN:978-1-4503-1492-3.

[17] R. Feldman, "Techniques and applications for sentiment analysis", Communications of ACM, vol 56, Issue 4, April 2013, pp. 82-89, doi: 10.1145/2436256.2436274

[18] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, "Lexicon-Based Methods for Sentiment Analysis", Computational Linguistics, vol 37, Issue 2, June 2011, pp. 267-307, doi: 10.1162/COLI_a_00049

[19] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis", Foundations and Trends Information Retrieval, vol 2, Issue 1-2, January 2008, pp.1-135, doi: 10.1561/1500000011