

# Messaging Activity Reconstruction with Sentiment Polarity Identification

Panagiotis Andriotis<sup>(✉)</sup> and George Oikonomou

University of Bristol, Merchant Venturers Building, Woodland Road, Clifton,  
Bristol BS8 1UB, UK

{p.andriotis,g.oikonomou}@bristol.ac.uk

**Abstract.** Sentiment Analysis aims to extract information related to the emotional state of the person that produced a text document and also describe the sentiment polarity of the short or long message. This kind of information might be useful to a forensic analyst because it provides indications about the psychological state of the person under investigation at a given time. In this paper we use machine-learning algorithms to classify short texts (SMS), which could be found in the internal memory of a smartphone and extract the mood of the person that sent them. The basic goal of our method is to achieve low False Positive Rates. Moreover, we present two visualization schemes with the intention to provide the ability to digital forensic analysts to see graphical representations of the messaging activity of their suspects and therefore focus on specific areas of interest reducing their workload.

**Keywords:** Smartphone · Forensics · Text-mining · Short-text messages

## 1 Introduction

This is without a doubt the era of remarkable achievements in the area of mobile communications. Smartphones dominate the market and people have the choice to purchase mobile devices from a wide range of manufacturers. Those devices are primarily used for texting, messaging, chatting and connecting to the Internet. These actions leave traces in the internal memory of the phones, which can be used as evidence in forensic investigations. The evidence acquisition should be performed by a specialist, but it does not require very complicated methods, as stated in [2]. Open source tools can be used to conduct a sound investigation without any cost, when our target phone is running under the Android operating system [2]. The major sources of evidence are text, sound and images stored usually in SQLite databases, in the cache of the device or in emulated and external storage media.

Considering the examination of textual data, a more traditional approach is currently used during an investigation, utilising tools like ‘grep’, ‘strings’ or ‘xxd’ to extract information out of text. However, Text Mining and Natural

Language Processing can offer automated solutions and perform various tasks on our evidence, such as text classification. Sentiment Analysis for example is a research area that became very popular nowadays because of the proliferation of social media and the need to handle ‘big data’ which are available in the Internet. Micro blogging services like Twitter have been used as sources (corpora) of textual information [17]. The similarity between a Twitter post (tweet) and a Short Message Service (SMS) text has been evaluated in [3]. Additionally, Task 2 at the “SemEval-2013: Semantic Evaluation Exercises”<sup>1</sup> competition was dedicated to Sentiment Analysis on Twitter data and a special category included SMS too. The idea was to train classifiers with Twitter data and test their Sentiment Analysis ability on SMS datasets. The concept behind this approach is that, as researchers, we do not have access to large SMS datasets because of privacy and legal issues; hence we can train classifiers using public data which present structural similarities with the short text messages we use in our private communication with other people.

In this paper we propose our methodology to classify SMS (and other short texts such as chatting logs) found in the internal memory of a smartphone considering their emotional polarity. The F-score our classifier achieves approaches the accuracy the current state-of-the-art systems [19] provide, but it needs less computing power. Furthermore, our approach aims to decrease False Positive Rates in order to provide more reliable results during the forensic analysis. Another contribution of this study is the proposal of two different visualization schemes that reconstruct the actual messaging activity of the phone. They focus on the emotional polarity of the exchanged messages and propose novel methods to merge Digital Forensics with Text Mining.

## 2 Related Work

The basic task for a Sentiment Analysis (or Opinion Mining) algorithm is to separate text documents in two (or three) classes; positive, negative (and neutral). Prior work presents how we can use various lexicons and tokenize documents in order to separate words related to sentiment classes [22]. Another approach to solve the problem of opinion mining is the use of Machine-Learning methods [20]. Naïve Bayes, Support Vector Machines (SVM) and Maximum Entropy are among the most popular algorithms used for sentiment categorization. These algorithms can be amplified by other techniques like lexical normalization [14] or distant supervision [21] to make a more robust classification scheme. Sentiment analysis is a concept, which can be applied to various aspects of our lives. At the past, handcrafted lexicons were utilized to create methods to perform opinion mining on market stocks [8]. Additionally, a holistic approach that used multiple opinion words to review various products was presented in [9]. Other systems are able to provide real-time evaluation of the public sentiment for electoral candidates [23].

---

<sup>1</sup> <http://www.cs.york.ac.uk/semEval-2013/> (and its successor on 2014).

Multinomial Naïve Bayes (MNB) and SVM were used in [18] to evaluate the hypothesis that sentiment analysis is easier on short texts than on larger documents. The authors achieved an accuracy of 74.85 % for binary classification and concluded that a unigram feature representation is sufficient for short texts. Sentiment analysis on micro blogs was also the topic of [6]. In [22], Taboada et al. used lexicons and their word valence and concluded that sentiment analysis on blog postings and video games reviews can be robust and accurate. We mentioned previously that Twitter posts are widely used as corpora [17] because they are public. Thus we do not need any special permission to test our algorithms on them. However, automated approaches for Twitter feeds (and micro blogging data in general) might be problematic because the language we are using on such media contains non-standard (elongated or abbreviated) words and unusual vocabulary [15]. Despite these variations, tweets are quite similar to SMS [15] and therefore we can use them to simulate SMS for our research.

Text mining in Digital Forensics was basically used to extract linguistic patterns from emails and perform user-profiling [10]. Other research papers focus on the characteristics that constitute the texting language, which is commonly used in mobile devices and messengers [7]. Despite the fact that we have seen studies aiming to use text mining for various purposes, like text searching optimization [5], there is a limited number of technical papers targeting sentiment analysis on SMS. An interesting study on the public sentiment extracted by SMS content is presented in [24]. A recent sentiment analysis competition (Task 2: SemEval-2013) among academics used Twitter posts to train the contestants' classifiers and a SMS dataset to test the accuracy they could achieve. The best team, which created a complex classifier, reached an accuracy of 0.69. The same concept was applied in [3] but in this context the authors did not train any model for the SMS classification. They were testing a bag-of-words approach and how efficient this simplified method could be to categorize SMS. The algorithm was highly depending on the lexicon that was used and the classification for negative messages was not competitive enough. However, this was the first attempt to use Text Mining in Digital Forensics and the outcome of this work was a system able to produce the 'Sentiment View timeline'.

### 3 Methodology and Evaluation

In order to provide a framework that will decrease some of the problems occurred in previous work (for example the poor classification results for negative messages) we had to decide which is the most efficient classifier for our task. First we trained three classifiers using an open source data mining program called 'Weka' [13]. The software provides a graphical user interface and it can perform numerous machine learning tasks. We trained three different classifiers: Naïve Bayes Multinomial (MNB), the default SVN (Sequential Minimal Optimization: SMO Polykernel) and the Maximum Entropy classifier called 'Logistic'. In order to work with text in Weka we have to utilize the FilteredClassifier module. This module aggregates the classifier functionality but first it uses a filter to

pre-process the documents. Our documents (short texts) were transformed to vectors with the unsupervised filter `StringToWordVector`. In our experiments we filtered the documents using the `LovinsStemmer` (described in [16]) and the default tokenizer. However, we removed the symbols ‘()’ from the delimiter list to capture any emoticon that is present in the text.

We chose popular datasets to train and test our algorithms. The first corpus we used to train the classifiers came from the Sentiment140 (SENT140) dataset [12] that consists of 1.6 million Twitter feeds (classified as positive, negative and neutral). We noticed that the classification scheme used to distribute tweets contained in the SENT140 set was not very accurate and a lot of short texts were distributed in the wrong class. For this reason, we manually classified a random set of tweets (2280) in three classes: neutral, positive and negative. This set was enriched by previous positive and negative lexicons [3] and created the final training data consisted of approximately 5000 documents. The test dataset consisted of 3075 randomly picked SMS from a SMS Corpus [1], initially used for spam filtering. We manually classified these short texts into three classes (neutral, positive, negative).

The basic assumption we made before we train and evaluate our classifiers is that the forensic analysis should be primarily focused on identifying positive and negative mood trends between entities that exist in the smartphone ecosystem. For this reason, during classification and evaluation, we assumed that there exist two main classes. One superclass contained positive and neutral messages and one class contained negative messages.

**Table 1.** Correctly and incorrectly classified instances of Twitter feeds

Classifier	Correctly classified	Incorrectly classified
MNB	78.5471 %	21.4529 %
SVM	76.3458 %	23.6542 %
MaxEnt	70.9826 %	29.0174 %

**Table 2.** Training results using Twitter feeds

Classifier	Class	TP rate	FP rate	Precision	Recall	F-measure	ROC area
MNB	Negative	0.877	0.318	0.756	0.877	0.812	0.849
	Positive	0.682	0.123	0.832	0.682	0.75	0.849
	Weighted avg.	0.785	0.226	0.792	0.785	0.783	0.849
SVM	Negative	0.883	0.371	0.728	0.883	0.798	0.756
	Positive	0.629	0.117	0.827	0.629	0.715	0.756
	Weighted avg.	0.763	0.251	0.775	0.763	0.759	0.756
MaxEnt	Negative	0.83	0.423	0.688	0.83	0.752	0.689
	Positive	0.575	0.164	0.757	0.575	0.653	0.697
	Weighted avg.	0.71	0.301	0.721	0.71	0.706	0.693

We trained the classifiers taken into account the aforementioned assumption. The numbers of correctly and incorrectly classified instances are shown in Table 1 and the results about the classifiers’ accuracy are shown at Table 2. The Naïve Bayes classifier (MNB) seems to achieve better accuracy on the training data since more than 78.5% of the dataset was correctly classified and the weighted ROC area was 0.849. The SVM model also produces satisfying results but the weighted average rates are not very competitive compared to the MNB model. Table 2 suggests that the more appropriate model for our type of documents is the MNB classifier.

Furthermore, we evaluated the classifiers on our SMS dataset. The numbers of correctly and incorrectly classified instances are shown in Table 3 and the results about the classifiers’ accuracy are shown in Table 4. We can see that the MNB classifier was able to correctly classify approximately 74.29% of the dataset. The weighted average of the True Positive Rate (TP) was 0.743 and the False Positive Rate (FP) was 0.358. The other classifiers did not achieve better results on the test set. However, the SVM model seems to perform in a similar manner compared to the MNB model.

A comparison between previous results (presented in [3]) and the outcome of the supervised classifiers shows that the MNB model achieves a better TP Rate on the negative messages. Also the FP Rate is quite low (0.186) suggesting that the NB classifier is better than the bag-of-words approach when we want to classify SMS messages that represent negative emotions. For the SMS with positive emotional fingerprint we had a good TR Rate but the FP Rate was quite

**Table 3.** Correctly and incorrectly classified instances of the SMS dataset

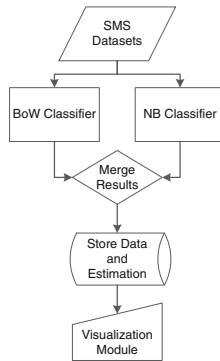
Classifier	Correctly classified	Incorrectly classified
MNB	74.2878 %	25.7152 %
SVM	72.2367 %	27.7633 %
MaxEnt	61.3134 %	38.6866 %

**Table 4.** Evaluation results on the SMS dataset

Classifier	Class	TP rate	FP rate	Precision	Recall	F-measure	ROC area
MNB	Negative	0.57	0.186	0.56	0.57	0.565	0.769
	Positive	0.814	0.43	0.821	0.814	0.817	0.769
	Weighted avg.	0.743	0.358	0.744	0.743	0.744	0.769
SVM	Negative	0.58	0.219	0.524	0.58	0.551	0.681
	Positive	0.781	0.42	0.818	0.781	0.799	0.681
	Weighted avg.	0.722	0.361	0.732	0.722	0.726	0.681
MaxEnt	Negative	0.504	0.341	0.38	0.504	0.433	0.581
	Positive	0.658	0.472	0.771	0.658	0.71	0.611
	Weighted avg.	0.613	0.433	0.657	0.613	0.629	0.602

high. We believe that this feature appeared because we assumed that neutral messages belong to the same superclass with positive messages. For this reason we propose our hybrid classifier that will deal with the problem of classifying neutral and positive messages in a better and more accurate way.

Our hybrid approach aims to propose a methodology that will be able to correctly classify as many SMS as possible efficiently and accurately. This means that we intend to reduce the False Positives the MNB classifier produces in order to make our scheme more robust and less error-prone. As stated in Sect. 2, the team that managed to win the SemEval-2013 competition using a Twitter dataset to train a classifier and a SMS dataset to test it, reached an average F-score of approximately 0.69. Also, recent results from experiments that were testing the sentiment polarity of the same SMS dataset (using a simple bag-of-words approach) [3] showed that the task of identifying negative messages was even more difficult (0.49 TP Rate and 0.29 FP Rate on the negative messages). The MNB classifier in our current experiments achieved a better TP Rate on negative messages (0.57 as shown in Table 2) and a low FP Rate (0.186). However, the MNB classifier cannot distinguish positive and neutral messages because it is trained to classify only two major classes (a ‘positive’ superclass that contains positive and neutral messages and a negative class which contains negative messages).



**Fig. 1.** Hybrid classification methodology

The scheme described in Fig. 1 is a system that merges the advantages of our MNB and the bag-of-words (BoW) classifiers. First, we have to input the SMS database (or any SQLite database that contains information from messaging applications). Android smartphones for example, store this information in the data partition of their internal memory. In more details, the file our system should parse is the mmssms.db SQLite database from the folder/data/com.android.providers.telephony/databases and especially the ‘SMS’ table which consists of attributes describing who sent the message to whom, which is the actual message, when the transaction happened and other relevant information.

The set of short text messages will then be fed to our two classifiers, the bag-of-words (BoW) and the Naïve Bayes (MNB) schemes for classification. BoW will distribute them in three classes (neutral, positive, negative) and NB will classify the messages in two classes (a positive superclass and a negative). After this phase has been completed the results will be passed to the merging algorithm for the final classification of each message. The algorithm classifies as negative those messages that were predicted as negative by the MNB classifier. Those messages that were predicted as positive by the MNB classifier are crosschecked with the output of the second classifier (BoW). If BoW indicates that the message is negative, our system flags the SMS as neutral. If BoW indicates the message as positive then we classify it as positive and, finally, if BoW indicates the message is neutral, we classify it as neutral. Then, the estimations and the messages are stored in a database which will be the source that will feed the visualization module (we will describe it in Sect. 4) to depict the mood trends among the various entities that exist in the SMS database of the smartphone.

Table 5 presents the evaluation of the classification scheme shown in Fig. 1. The basic feature we should underline is that our approach achieves low FP Rates in the negative and the positive messages (0.186 and 0.164 respectively). Also, approximately 60% of the whole set will be successfully classified in the appropriate class providing a fairly clear indication of the emotional trends among people on a given time. Finally, our methodology does not require large training steps and the results approximate those that were achieved by very complex classifiers [19] (for instance the F-score on the positive set is 0.679). These setups were heavily based on very detailed data pre-processing steps. The extracted information from our classifier can be visualized using the methods we present at Sect. 4 in order to reconstruct the emotional fingerprint that the exchange of short texts produces.

**Table 5.** Evaluation results for the hybrid system on the SMS dataset

Class	TP rate	FP rate	F-measure
Positive	0.599	0.164	0.679
Neutral	0.57	0.281	0.615
Negative	0.57	0.186	0.565

## 4 Visualization Module

In this Section we discuss our concept to visualize the extracted information that depict possible mood alterations and relations between the people interacting with the person that owns the smartphone we analyze. The idea is to create a dynamic graph that reconstructs all the activity stored in a database like the mmssms.db which holds the history for short texts messaging. People that exchanged messages with the person under investigation will represent the nodes of the graph and each

message will be the action-edge that connects those entities. In our approach we are using three colours to represent mood trends extracted by the SMS; blue for neutral, green for positive and red for negative messages. For example, if a person A sent a message to a person B (expressing a positive emotion) on a specific time T, then the graph will show two nodes A and B linked with a green arrow starting from A towards B. If another interaction between the two parties takes place again, a new arrow with a new colour will link the two nodes. This time the edge will be shown thicker to underline the fact that these two entities have frequent communication and its colour will imply the emotional fingerprint of the specific interaction.

In addition, the graph will be dynamic, which means that a node will not be shown until the first interaction happens. However, when the interaction takes place, its representative edge can be shown until the final completion of the graph. This attribute will make the graph able to illustrate all interactions that happened in a given time scope. Furthermore, the forensic analyst will be able to see a graphical representation of activities that might affect the mood of persons involved in a case.

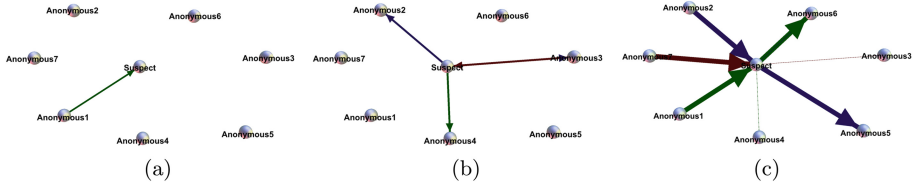
We are using the open source tool ‘Gephi’ [4] to produce the visualizations. This is a platform which accepts various file formats as inputs but we chose to utilize gexf files in our visualization module. We made this decision because gexf format is easy to understand and produce (it is an xml file) and according to the official Gephi documentation provides better functionality. The visualization module we present here is able to produce two types of data graphical representations. The first one is the aforementioned ‘dynamic graph view’ that reconstructs the SMS activity and it is focused on the expressed emotions via the exchanged messages. The second approach we will present is the ‘heat map view’. This type of visualization will provide a convenient and overall view of the predominant mood extracted by exchanged messages between two parties. It is basically a colourful grid which illustrates the emotional fingerprint of the exchanged messages (between two entities) within a month or within a broader period of time.

#### 4.1 Dynamic Graph View

The concept behind the specific visualization scheme is to construct an animated representation of actions between entities that interact in the singular ecosystem defined by the smartphone. Hence, forensic analysts who investigate a case and have seized a smartphone as evidence have the choice to select the time scope of their examination. For this reason the proposed scheme (in order to produce the visualization) requires from the analyst to input the start time (ST) and the end time (ET) of the actions that will be reconstructed. The conceptual design is further discussed at the rest of this subsection.

We assume that our data (and the extracted mood class) are stored in the database described in Fig. 1 and the analyst has set the ST and ET. Thus, each row in our database contains a ‘copy’ of the rows of the ‘SMS’ table (located in the original mmssms.db) and also the extracted emotion polarity;  $-1$  for negative,  $0$  for neutral and  $1$  for positive. Such a row contains attributes like





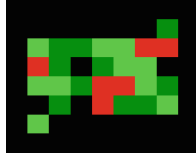
**Fig. 2.** Messaging activity graphical representation

‘address’ (the telephone number interacting with the examined phone), ‘date’ (a timestamp describing when the transaction happened), ‘type’ (1: received and 2: sent), ‘body’ (the short text) and the extracted ‘emotion’. The algorithm which creates the gexf file requires a double pass from this database. During the first pass we will query the database to get rows related to the time scope of our investigation (ST until ET). Furthermore, the first pass will store information (in a temporary storage area) about the entities that interact with the smartphone. These entities will be written in the gexf file as the nodes of the graph. If the algorithm sees a new entity interacting with the phone, the ‘date’ attribute will be written in the gexf file as the ‘start’ attribute of the given node. The ‘end’ attribute will always be the ET. During the second pass the algorithm parses again the database and creates the edges. Each row in the database is a new edge and the timestamp which describes when the action happened will be the ‘start’ attribute of the edge. The ‘end’ attribute again is the ET. If the ‘emotion’ attribute is  $-1$ , the edge will be coloured red. If the ‘emotion’ is  $0$ , the edge will be blue and if it is  $1$  the edge will be shown as green.

Figure 2 demonstrates how the messaging activity is reconstructed in Gephi (after our gexf file has been loaded) and it also highlights the mood fingerprints of the exchanged SMS. For this illustration we replaced messages from an original mmssms.db with random SMS from our testing dataset (Sect. 3) and we used the timeline feature of Gephi. We should also underline that we are using Gephi in this study just to present the concept of the mood and messaging activity reconstruction. Its current version (0.8) is not able to handle multigraphs but, according to the official documentation, the next version (0.9) will be able to present such complex graphs. However, other tools like GraphViz [11] can produce multigraph visualizations and the concept can be easily applied to files supported by the specific open source project. Figure 2 shows three different screenshots from the beginning towards the end of the created timeline from Gephi.

### 4.2 Heat Map View

The second visualization concept we propose in this study is the ‘Sentiment Heat Map view’. This graphical representation of the extracted mood is initially designed to depict interactions between two entities within a timeframe of a month. We assume we want to see the mood fingerprints of exchanged messages existing in the mmssms.db between the person under investigation and someone found in the smartphone’s contact list. Thus, the queries on the database seen



**Fig. 3.** Sentiment heat map view

in Fig. 1 will return tuples only for those two entities within a period of a month (or so). The output is a coloured grid (heat map) formatted as a calendar. Each day of the calendar is shown as a square on the grid; if the exchanged message emits negative mood it will be shown as a red square. The positive mood is depicted with light green, the neutral mood with dark green and if on a given day there is no message, the square will be coloured black. (We use Matlab in this illustration.)

In Fig. 3 we present the ‘heat map view’ produced for a hypothetical scenario which shows the extracted mood from messages that were sent FROM the person under investigation TO a person from the smartphone’s contact list. For clarity we assume that only one message (or none) was sent each day on a specific month. If more than one message we exchanged, the scheme can be further extended either by calculating the overall sentiment valence of all the exchanged messages during a day or by recursively dissecting each square (that represents a day) in smaller coloured squares.

The visualization can be produced either by using open source tools like Octave (utilising the command ‘`imagesc(A)`’) or other commercial tools like Matlab (with the command ‘`HeatMap(A)`’). In both cases  $A$  is a matrix that represents the calendar. If we keep the format of  $A$  intact and if we change the numbers 1, 2, ..., 31 with other numbers that represent the mood we will be able to see a calendar like the one shown in Fig. 3. Of course, this is a simplified illustration of the concept and it does not include cases where on the same day we had more than one ‘sent’ messages. However, we can further extend the idea as discussed in the previous paragraph to include these cases in the future.

## 5 Conclusions and Future Work

In this study we investigated the impact of machine-learning algorithms trained on Twitter posts to classify SMS according to their sentiment polarity. We manually labelled tweets and texts in three categories (neutral, negative, positive) and tested the efficiency of 3 training models (MNB, SVM, MaxEnt). We evaluated the models on the SMS test set and concluded that MNB works better on these short texts and it is faster and more accurate than the other classifiers. Furthermore, we proposed a classification scheme in order to decrease the FP Rate the MNB classifier produced on negative messages. We believe that during a forensic investigation we would be more interested in a method that produces less erroneous estimations. Thus, our scheme is competitive against the current

state-of-the-art systems. These systems use complex feature vectors resulting to a costly consumption of memory, time and processing power.

Additionally, we proposed two visualization approaches to provide the opportunity to show a reconstructed animated representation of the messaging activity illustrating the extracted mood fingerprints. The ‘dynamic graph view’ provides a generic insight to the messaging activity including all the SMS that were sent (and received) from a smartphone. The ‘heat map view’ is a concise solution that focuses on two specific entities and provides an automated calendar-like projection of emotions that occurred during the period of a month. These two modules are designed to reduce the workload of an analyst but they cannot eventually be used as evidence, because they still remain a construction. However, in this study we bring together two research areas (Natural Language Processing and Digital Forensics) aiming to present methods in order to create automated and accurate representations of evidence existing in smartphones, tablets or wearable devices.

Despite the low FP Rates we achieved on positive and negative SMS in our set, there is still space for improvements, especially on the neutral message classification. Future work will focus on the improvement of our feature selection. We also believe that the input of an SMS subset in the training procedure will provide better F-scores and increase the system’s accuracy. Furthermore, the proposed visualization schemata can be extended to cover more than one smartphones which are seized in a specific case. A mapping of the sentiment in a closed environment (for example when forensic analysts examine multiple corporate phones) is also a direction for further work.

## References

1. Almeida, T.A., Maria Gomez, J., Yamakami, A.: Contributions to the study of SMS spam filtering: new collection and results. In: DOCENG 2011: Proceedings of the 2011 ACM Symposium on Document Engineering, pp. 259–262, 1515 BROADWAY, New York, NY 10036–9998, USA (2011)
2. Andriotis, P., Oikonomou, G., Tryfonas, T.: Forensic analysis of wireless networking evidence of android smartphones. In: 2012 IEEE International Workshop on Information Forensics and Security (WIFS) pp. 109–114, 345 E 47th St, New York, NY 10017, USA (2012)
3. Andriotis, P., Takasu, A., Tryfonas, T.: Smartphone message sentiment analysis. In: Peterson, G., Sheno, S. (eds.) *Advances in Digital Forensics X. IFIP Advances in Information and Communication Technology*, vol. 433, pp. 253–265. Springer, Heidelberg (2014). doi:[10.1007/978-3-662-44952-3\\_17](https://doi.org/10.1007/978-3-662-44952-3_17)
4. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. In: ICWSM, pp. 361–362 (2009)
5. Beebe, N.L., Clark, J.G.: Digital forensic text string searching: improving information retrieval effectiveness by thematically clustering search results. *Digit. Invest.* **4**(1), S49–S54 (2007)
6. Birmingham, A., Smeaton, A.F.: Classifying sentiment in microblogs: is brevity an advantage? In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 1833–1836. ACM (2010)

7. Choudhury, M., Saraf, R., Jain, V., Mukherjee, A., Sarkar, S., Basu, A.: Investigation and modeling of the structure of texting language. *Int. J. Doc. Anal. Recogn.* **10**(3–4), 157–174 (2007)
8. Das, S.R., Chen, M.Y.: Yahoo! for Amazon: sentiment extraction from small talk on the web. *Manage. Sci.* **53**(9), 1375–1388 (2007)
9. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 231–240. ACM (2008)
10. Estival, D., Gaustad, T., Pham, S.B., Radford, W., Hutchinson, B.: Author profiling for english emails. In: *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pp. 263–272 (2007)
11. Gansner, E., North, S.: An open graph visualization system and its applications to software engineering. *Softw.-Pract. Experience* **30**(11), 1203–1233 (2000)
12. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. *CS224N Proj. R.* **657**, 1–12 (2009)
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. *ACM SIGKDD Explor. Newsl.* **11**(1), 10–18 (2009)
14. Han, B., Cook, P., Baldwin, T.: Lexical normalization for social media text. *ACM Trans. Intell. Syst. Tech.* **4**(1), 5 (2013)
15. Laboreiro, G., Sarmiento, L., Teixeira, J., Oliveira, E.: Tokenizing micro-blogging messages using a text classification approach. In: *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, pp. 81–88. ACM (2010)
16. Lovins, J.B.: Development of a Stemming Algorithm. MIT Information Processing Group, Electronic Systems Laboratory, Cambridge (1968)
17. Martinez-Camara, E., Teresa Martin-Valdivia, M., Alfonso Urena-Lopez, L., Montejo-Raez, A.: Sentiment analysis in Twitter. *Nat. Lang. Eng.* **20**(1), 1–28 (2014)
18. Melville, P., Gryc, W., Lawrence, R.D.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In: *KDD-09: 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1275–1283, 1515 Broadway, New York, NY 10036–9998, USA (2009)
19. Mohammad, S.M., Kiritchenko, S., Zhu, X.: 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)*, Atlanta, Georgia, USA, June 2013
20. Pang, B., Lee, L.: Opinion mining and sentiment analysis. *Found. Trends Inf. Retrieval* **2**(1–2), 1–135 (2008)
21. Suttles, J., Ide, N.: Distant supervision for emotion classification with discrete binary values. In: Gelbukh, A. (ed.) *CICLing 2013, Part II. LNCS*, vol. 7817, pp. 121–136. Springer, Heidelberg (2013)
22. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **37**(2), 267–307 (2011)
23. Wang, H., Can, D., Kazemzadeh, A., Bar, F., Narayanan, S.: A system for real-time twitter sentiment analysis of 2012 US presidential election cycle. In: *Proceedings of the ACL 2012 System Demonstrations*, pp. 115–120. Association for Computational Linguistics (2012)
24. Wang, Z., Zhai, L., Ma, Y., Li, Y.: Analysis of public sentiment based on SMS content. In: Yuan, Y., Wu, X., Lu, Y. (eds.) *ISCTCS 2012. CCIS*, vol. 320, pp. 637–643. Springer, Heidelberg (2013)