Chapter 1

FORENSIC INVESTIGATION OF SMARTPHONES USING LEXICON-BASED MOOD ANALYSIS AND TEXT MINING METHODS

Panagiotis Andriotis, Atsuhiro Takasu and Theo Tryfonas

Abstract Humans like to express their feelings via their communications with other people. They tend to use specific words to stress their emotional state either in written or in oral language. In our everyday life we share our thoughts and emotions with literally thousands of people being connected to the Internet. Scientists from the area of Text mining and Natural Language Processing have studied our sentiment fingerprints left on text in the past to extract the emotional polarity of customers for a product or to evaluate the popularity of politicians. The methods they use are diverse including simple lexicon-based categorization or more sophisticated Support Vector Machine classifiers. Current research is more focused on the micro blogging world and especially on Twitter. Scholars also mention that there is a notable similarity between a Twitter feed and an SMS. In this paper we investigate the common characteristics of both formats for sentiment analysis purposes and prove the correctness of the former assumption. We therefore employ a lexicon-based approach to extract and calculate the sentiment score of SMS found on smartphones and we present a forensic tool that creates a timeline view depicting their emotional fingerprint. This form of visualization and analysis enriches the forensic investigation procedure with elements showing potential psychological patterns drawn from sent or received text messages. We optimize the performance of our algorithm and introduce the use of indexing in conjunction with our tool to conduct faster keyword searches on the text messages.

Keywords: Indexing, SMS, emotion, timeline, emoticons, forensic analysis, behavioural analysis.

1. Introduction

Since the '90s when we first witnessed a remarkable technological bloom of mobile communications, things have changed drastically considering the mobile networks, the devices and the users themselves. Our needs for processing power and usability have grown exponentially along with the capabilities of various devices. Mobile phones upgraded from simple telephone devices to sophisticated mini computers. The means of communication between users also have changed. However, short text messages (SMS) are still one of the most popular services mobile phones provide because of their simplicity. In addition, modern smartphones can also send emails or provide the functionality to the user to chat with friends through instant messengers. Therefore, we store in these devices valuable textual information that expresses our thoughts and emotions for other people.

A forensic investigation usually includes the examination of all electronic devices the suspect uses such as computers, hard drives, tablets or smartphones. The analysts extract as much information as they can for the person under investigation revealing any artifacts left from the use of these devices. In digital forensics we are interested in retrieving relevant information, recovering deleted files and presenting the extracted data in an efficient way. Text messages are an integral part of our communications and therefore could potentially be valuable pieces of evidence during a forensic investigation.

Experts from the Text Mining area have employed methods to extract information from various types of text. Researchers coming from the fields of Natural Language Processing and Information Retrieval have been studying recently web sites like Twitter because of their openness to the public and the plethora of information one can derive from such micro blogging services. The extraction of the emotional polarity of such messages is a popular subject among them. Others, stress the similarity between a tweet (Twitter post) and an SMS claiming that we can migrate techniques we use for analysis on Twitter feeds in order to analyze SMS. There are indeed common characteristics between the two formats. To name a few, they both consist of a limited number of characters (140 for tweets 160 for SMS). In addition, the '@' feature in Twitter posts can be seen as a direct message to a specific person, just like an SMS we send to a friend through a mobile phone. Finally, special symbols like emoticons or other abbreviations are used in both Twitter posts and SMS.

Despite the variation of resources that deal with the problem of the classification of such messages into two major categories (positive or negative emotional polarity), we noticed a gap in the literature considering

the use of text-mining techniques for forensic analysis purposes. In this paper we present a study on the efficiency of basic text mining methods when they are employed for SMS mood analysis and we also discuss their potential use in Forensics. The main contributions are listed below.

- We compare the efficiency of a lexicon-based Sentiment Analysis algorithm on various datasets (Twitter feeds and SMS) and evaluate the claim that Mood Analysis methods used for Twitter feeds can be also used on SMS.
- We evaluate the importance of word preprocessing methods calculating their contribution to the final outcome of the sentiment scoring. We also study the significance of the lexicon and discuss how a lexicon-based method can be optimized for improving the SMS mood score.
- We propose a methodology to use textual data from smartphones and add indexing to make these data accesible to the forensic examiner and perform faster queries. We also introduce the 'Sentiment Timeline View' in order to accelerate the investigation process and present the data in a novel way.

2. Background

Sentiment Analysis is the process of identifying positive or negative opinions about a subject or topic and it is mainly focused on the automatic extraction of the sentiment-polarity that can be expressed in a piece of text. Prior work in this field uses lexical knowledge and the emotional valence of the words included in vocabularies [15]. Recently, researchers deal with this classification problem as a task to be solved using Machine Learning techniques. Pang and Lee, 2008, for example, present a survey that covers methods to solve opinion related problems [19].

Knowledge-based methods use linguistic models to categorize the sentiment of text passages. These methods are focused on the construction and use of dictionaries in order to capture the sentiment of words. Literature demonstrates approaches where manually crafted lexicons are used to perform mood analysis on stock message boards [7]. Other studies present semi-automated approaches to classify text content and identify the polarity of opinion sentences [26]. Pang et al., 2002, introduced their work on sentiment analysis using a variety of Machine Learning techniques [20]. They are implementing three classifiers (Nave Bayes, Maximum Entropy and Support Vector Machines) to evaluate their efficiency in categorizing movie reviews as negative or positive con-

cluding that Support Vector Machines work better in most of the cases. They also stress that the use of more complicated linguistic models like n-grams do not dramatically improve their findings, suggesting that a unigram approach seems sufficient.

Sentiment Analysis has been applied to solve diverse problems at the past, ranging from online forums hotspot detection [14] to sentiment classification in microblogs [5]. Ding et al., 2008, present their holistic lexicon-based approach to perform opinion mining on reviews expressed for product features combining multiple opinion words found in the same sentence of a review [8]. Melville et al., 2009, demonstrate their framework for sentiment analysis at blogs combining lexical knowledge with supervised learning for text categorization [15]. They conclude that it is preferable to incorporate both methods to achieve better results while conducting blog analysis. Taboada et al., 2011, use dictionaries of words with predefined characteristics (polarity and strength) [24] and highlight that lexicon-based methods for sentiment analysis are robust and can be used in various domains without any training on specific data. Their lexicon-based implementation performs well on diverse tasks like video games reviews and blog postings classification.

Twitter is an open micro blogging service built on the logic that all posts will be available to the public. This feature provided the opportunity to researchers to use Twitter as a corpus [18]. Bermingham and Smeaton, 2010, suggest that it is easier to infer the sentiment polarity of micro blogging posts when compared to blogs, where normally exists a richer textual content [5]. However, the automatic processing of micro blogging posts might be problematic because of the use of non-standard words and unusual punctuations, as Laboreiro et al., 2010, discuss [12]. They also mention the similarity between a micro blogging message and an SMS. In addition, Leong et al., 2012, use sentiment mining to analyze SMS during a teaching evaluation procedure [13].

In the area of Digital Forensics, text analysis has been used in cases where we were interested in extracting patterns from emails and constructing user-profiling methods from text [9]. Furthermore, there exist studies presenting methods to optimize text searching [4] and others that are focused on the investigation and modeling of texting language [6], which is mainly used on mobile phones. We are already familiar with methods that utilize open source tools to perform forensic analysis on Android smartphones [2]. Despite the numerous resources dealing with mood analysis problems and the extended experimentation on the micro blogging world, we could not find any particular study in the literature combining the principles of opinion and text mining with forensics on mobile devices. For this reason, we present in the next sections our work

on extracting emotional characteristics from SMS found on smartphones and creating a special timeline view to depict their sentiment fingerprints merging intelligence with forensics [3].

3. Experimental Setup and Datasets

In this section we demonstrate the datasets we used, the algorithm and the experimental setup we employed for our tests. We gathered 6,566 tweets from different Twitter accounts on a specific day (5th August 2013). We will refer to this dataset as TWT. An Apache Tomcat server requested (utilizing the Twitter API v1.1) the most recent statuses of 33 popular accounts (musicians, actors, athletes, managers) through a PHP script. Since the rate limit of requests is 200 per account we managed to collect 6566 different tweets. The data format the Twitter API returns is JSON, thus we had to convert those tweets in txt format. We also used a dataset consisted of automatically classified tweets as positive and negative [10]. Consequently among the positive messages there will exist negatives and vice versa. We will refer to that set as SENT140 (PoSENT140 for positives and NegSENT140 for negatives).

The SMS dataset we employed was initially used for spam filtering [1]. It contained 5574 messages but some of them were duplicates. We therefore sanitized the dataset resulting 4827 unique messages and we manually classified these messages into two (sentiment) categories. Negative messages were considered those expressing anger, fear, sadness, disgust and boredom (919 messages). We classified as positive the messages that expressed joy and happiness (1867 messages). There were also numerous messages we could not classify into those two emotional categories and they are considered neutral in this study.

The method we used for our experiments is a simple bag-of-words approach. For this reason we employed three different vocabularies, used in the past for Sentiment Analysis on Twitter, that contained different words linked to positive or negative emotions. AFINN [11] is a lexicon containing various words and their valence (from -5 to 5). It also contains bigrams and trigrams but we excluded them from our research. Another well-known lexicon is the Wordnet-Affect [22], [23]. It consists of synsets linked to affective labels-words and we refer to it as WRDNT. Finally, the last dataset we employed (NRC) was used in the 'SemEval-2013 Task 2' competition (http://www.cs.york.ac.uk/semeval-2013/task2/) for Sentiment Analysis on Twitter feeds [16]. However, for the purposes of our experiments we sanitized the set eliminating hashtag symbols (#) and other words that looked irrelevant to our cause, for example '5yo' or plain letters like 't'. The sanitized vocabularies consisted

of 25,675 words and abbreviations for positive emotions and 20,636 for negative.

For this study we used Java and the newest version of Apache Lucene Library: version 4.4 (http://lucene.apache.org). Apache Lucene is an open source project providing text indexing and search capabilities, in conjunction with spellchecking, analysis, text pre-processing and tokenization of text streams. We also employed JDBC drivers for SQLite and MySQL databases in order to provide links between our programs and the databases we utilise.

The methodology we followed to extract the mood score of a tweet or SMS is explained in details. Let $L_p = \{l_{pi}\}$, where i = 1, 2, ..., m, be the set of our positive textual markers (the positive lexicon) and $L_n = \{l_{nj}\}$, where j = 1, 2, ..., n, be the set of our negative textual markers (the negative lexicon). The corpus is denoted as C and it is the total of single tweets or SMS, $C = \{t_k\}$, where k = 1, 2, ..., q. If a positive marker l_{pi}, appears in a tweet or SMS (t_k) in the corpus, we set

$$l_{pi}(t_k) = 1. \tag{1}$$

Else, we set $l_{pi}(t_k) = 0$. We also perform the same calculations for negative markers l_{nj}. This is a simplified 'Boolean' logic that takes into consideration that our documents are of limited size, thus we will probably have a limited range of markers in each tweet contributing to the tweet sentiment score. The tweet sentiment score $s(t_k)$ is equal to the total of positive markers found in a tweet minus the total of negative markers found in the tweet:

$$s(t_k) = \sum_i l_{pi}(t_k) - \sum_j l_{nj}(t_k) \tag{2}$$

4. Experimental Results and Discussion

Text mining techniques usually include text pre-processing before the analysis phase. Stemming is a popular text pre-processing method to obtain the root of a word. For example, the words 'connect', 'connected', 'connection' have the same root that can be represented with the lemma: 'connect' [21]. Our first intention here is to evaluate if stemming is an efficient technique that increases the possibility to achieve better results with equation (2). The experiments for this task were performed using AFINN as our lexicon. We picked one tweet at a time from the collection of tweets and using the Apache Lucene Library we removed stop words. When the test case included stemming, the Porter's stemming algorithm was applied. For each of these words we calculated the mood score (2) using the appropriate lexicon.

Table 1. Calculating textual markers without using stemming and with stemming.

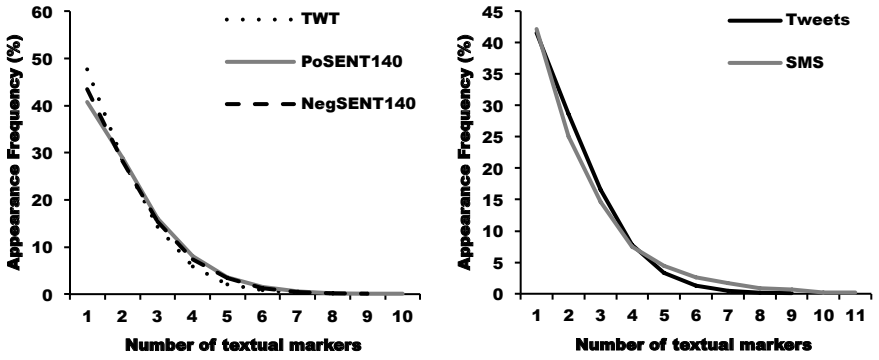
	Dataset	None	Total	Neutral $\mathbf{s(t_k)}$
No Stem	TWT	39.1%	60.9%	48%
	PoSENT140	28.4%	71.6%	41.5%
	NegSent140	25.3%	74.7%	39.3%
Stem	TWT	31.8%	68.2%	42.3%
	PoSENT140	23.3%	76.7%	36.6%
	NegSent140	18.7%	81.3%	34.7%

Table 2. Distribution of textual markers within the datasets without using stemming and with stemming.

	Dataset	1	2	3	4	5	6	7	8/9+
No Stem	TWT	47.7%	28.7%	14.3%	5.9%	2.1%	0.8%	0.3%	0.2%
	PoSENT140	40.8%	29%	16.1%	8%	3.6%	1.5%	0.6%	0.4%
	NegSent140	43.4%	28.3%	15.4%	7.4%	3.4%	1.3%	0.5%	0.3%
Stem	TWT	41.6%	28.6%	16.6%	7.8%	3.3%	1.3%	0.5%	0.3%
	PoSENT140	36.6%	28%	17.7%	9.5%	4.7%	2.1%	0.9%	0.5%
	NegSent140	36.9%	28.1%	17.4%	9.4%	4.7%	2.1%	0.9%	0.5%

First, we calculated the tweet sentiment scores in the TWT corpus. We did the same for the set of positive tweets from the SENT140 dataset and for its negative tweets. Then, we performed the same tests on the same datasets using stemming. The final experiment we did was focused on the SMS dataset using the same principles described above. Table 1 describes the results for the tests we conducted on our datasets without using stemming and with stemming. Column 'None' denotes the percentage of tweets that did not contain any matching words from the lexicon. 'Total' is the percentage of tweets that contained at least one matching words (positive or negative) for each tweet from the lexicon. At the last column we show the percentage of tweets that were classified as neutral (meaning that their Sentiment score was calculated by equation (2) as zero). Table 2 shows exactly the number of textual markers found in a single tweet.

For example, in our TWT set (No Stem), 39.1% of the messages did not contain any textual marker. 60.9% of the tweets contained at least 1 marker (up to 9) and 48% of the tweets were not classified as positive or negative using the equation (2). 47.7% of the Twitter feeds contained only one marker from the lexicon, 28.7% contained 2, etc. The first observation we can make from Table 1 is that on the already classified



(a) (b)

Figure 1. Distribution of lexicon words found in: a) Tweets without stemming and b) Stemmed tweets (black line) and SMS.

dataset we managed to distinguish more tweets using our mathematical equation (2), which is quite what someone could expect. Figure 1a) concatenates the distributions of textual markers shown in Table 2 (No Stem). In other words, it depicts the probability to find matching words in a tweet corpus and our AFINN lexicon. The round dot line indicates the percentage of lexicon words found in the TWT dataset; the dashed line describes the distributions on the negative SENT140 dataset and the solid line on the positive SENT140.

Another characteristic we can infer from Table 1 is that our 'Boolean' logic described in (1) is responsible for creating some odd results like the one that we see in the 'Neutral $s(t_k)$' cell for our TWT set (No Stem). The algorithm sorted 48% of tweets as neutral, but the percentage of the messages that do not contain any matching words was 39.1%. This difference comes from the fact that there exists a possibility the final score to be calculated as zero when a message contains even number of markers. For example, if we have 1 positive and 1 negative word in a tweet, the final score will be 0. We are aware of this problematic approach and we are dealing with it later in this section including the word's emotional valence to calculate the final score.

We should underline that at the case of stemmed tweets the results for the SENT140 positive and negative dataset are almost identical considering the number of matching words in a tweet (Table 2). This finding is an indication that we are using a balanced lexicon (AFINN) in our bag-of-words approach. Table 1 shows that stemming provides better results considering the total tweets classified according to their senti-

Table 3. Calculating textual markers within the SMS dataset.

SMS	**None**	**Total**	**Neutral s(t_k)**
No Stem	40.6%	59.4%	50.2%
Stem	31.9%	68.1%	42.8%

Table 4. Distribution of textual markers within the SMS dataset.

SMS	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8/9**	**10+**
No Stem	45.1%	25.2%	14.5%	6.8%	3.7%	2.5%	0.8%	1%	0.4%
Stem	42.2%	25%	14.6%	7.5%	4.4%	2.6%	1.7%	1.6%	0.4%

ment and also minimizes the percentage of neutral tweets. We therefore conclude that more tweets will be classified as positive or negative using stemming.

Below we present our findings when we performed the same experiments on the SMS dataset and calculated the emotional score as explained above using equation (2). The lexicon again was the AFINN vocabulary and the SMS dataset was first analysed without the use of stemming and then after applying Porter's stem. The goal of these tests was two-fold. First, we wanted to corroborate the conclusion that stemming will help us to classify a greater number of SMS during a forensic analysis and second, we aimed to confirm the claim that methods used for Sentiment Analysis in Twitter will suffice and can be migrated in forensic examinations on smartphones in order to extract users' emotional trends. Tables 3 and 4 present the results.

Table 3 shows that the use of stemming is vital to enhance the efficiency of Sentiment Analysis in SMS since the classified messages increased from 59.4% to 68.1%. In addition, Figure 1b) demonstrates the similarities in the distribution of the number of matching words found in the stemmed tweet set and the stemmed SMS dataset. These similarities indicate that we can employ techniques designed for Sentiment Analysis on Twitter feeds to perform SMS Mood Analysis.

Next, we evaluated the classification ability of (2) using TWT, stemming and three different vocabularies. We already discussed in extend the results AFINN lexicon provided in the previous paragraphs (Tables 1 - 4). The WorldNet-Affect lexicon version we tested consisted of words that could be characterized as formal and not widely used in the micro blogging world. However, in our written communication with people through SMS or Instant Messengers, we use more informal vocab-

ulary. Our initial thoughts about the efficiency of a lexicon like WRDNT, that it will not produce better results than AFINN confirmed after the experiments completion. We noticed that using WRDNT, roughly 1 - 3 matching words could be found in each tweet and the tweets with 'Neutral' $s(t_k)$ were 68.5%.

We also employed NRC for our tests, which is quite different from WRDNT. It consists of a plethora of words and hashtags (#) that were connected with positive or negative emotions. We sanitized the lexicon to fit it to our 'bag-of-words' approach but a lot of lemmas like "okayyy" were present in both categories, positive and negative, making us sceptical about its use. The lexicon also included symbols and 'internet slung'. The percentage of tweets that gave a neutral emotional score was 20.6% while 96.9% of the tweets contained at least one word from the lexicon. However, the distributions of matching words were very different compared to the previous results. For example, in a single tweet the matching 'words' could be more that 20. This happened because a lot of textual markers were common in the positive and negative lexicons. Hence, we concluded that parameters that might work well for methods like SVM classification would not be efficient enough to produce reliable results with a simple algorithm like ours. The fact that the use of such an extensive and diverse lexicon could not drastically decrease the percentage of tweets rated with $s(t_k) = 0$, led us to the decision to use AFINN for the rest of this study.

The rest of this section demonstrates the hit rates and the false positives the lexicon-based approach we employ produces. In order to measure these types of errors we performed the following experiments. The SMS dataset was manually classified into positive, negative and neutral messages. We evaluated the efficiency of equation (2) in distinguishing moods in SMS conducting a series of experiments on the manually classified datasets using the same logic we described earlier. Results are illustrated at Table 5.

Table 5 demonstrates that our approach in the problem of SMS mood analysis tends to produce relatively low false positive rate but the correctly classified messages will be about the half of the whole set (first two rows of Table 5). For the positive messages, the False Positive Rate (FPR) is the percentage of positive messages that were identified as negative messages, while the False Negative Rate (FNR) is the percentage of positive messages that were not identified as positive messages.

Having Table 5 under consideration, we tried to strengthen our algorithm to produce better results. To do so, we studied the characteristics of our SMS dataset. A lot of people (and especially young smartphone users) include in their vocabulary symbols and special words to abbrevi-

Table 5. Success rates and failures of lexicon-based mood analysis.

	Datasets	**Total**	**FPR**	**FNR**	**Hit Rate**
AFINN	Positive	1867	12.1%	32.6%	55.3%
	Negative	919	22.7%	32.4%	44.8%
Emoticons	Positive	1867	11.5%	31.1%	57.4%
	Negative	919	22%	31.7%	46.3%
Emoticons & Valence	Positive	1867	7.3%	23.9%	68.8%
	Negative	919	29.2%	25%	45.8%

ate their messages and produce a more concise text. The most popular symbols are called 'emoticons' and, nowadays, they play a major role in the electronic vocabulary. An emoticon is a symbol that expresses an emotion; happiness for example is represented with a smiling face ":)".

We therefore added extra functionality to our mood-scoring calculator by attaching in our lexicon some of the most popular emoticons found in the web[1]. We repeated the tests on the same SMS dataset but using the new lexicons for classification this time. From a comparison between the first and the middle rows of Table 5 we can understand that the use of emoticons in the lexicon will be helpful for increasing the success rates and decreasing the false positives and false negatives, but indeed the differences are not very wide. However, this is an addition that makes the method stronger and the lexicons more targeted.

We also tried to optimize our method by changing the way the mood score is calculated. In lexicons like NRC or AFINN, the words are presented with their emotional valence. In AFINN for example, the valence of the textual marker 'amazing' is 4 and the word 'approval' is marked with valence 2. The maximum valence is 5 and minimum is -5 denoting the most powerful negative emotion. We rated our emoticons with the higher valence assuming that users add a symbol like that in their texts to express clearly their feelings. Hence, emoticons should have the highest weight when added in the mood score. In this case, equation (2) does not change and the difference lies on the way we calculate the contribution of each marker to the final score. In (1) instead of $\mathrm{l}_{pi}(\mathrm{t}_k) = 1$, we are now counting $\mathrm{l}_{pi}(\mathrm{t}_k) = \mathrm{v}$, where v is the valence of each textual marker, $\mathrm{v} = \{-5, -4, ..., 4, 5\}$. We conducted tests to calculate the mood scores in our SMS dataset using the new AFINN lexicon (including emoticons and the valence of the textual markers) and the results are also presented in the last rows of Table 5.

The results show that the use of valence and emoticons decreased the false negative rates and in general optimized the performance of the

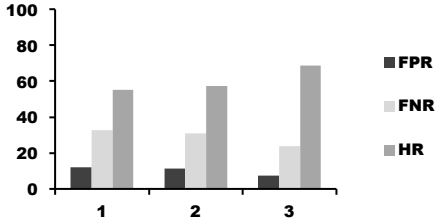


Figure 2. Optimizing the lexicon. (1: Without emoticons, 2: With emoticons, 3: Using word valence).

classifier. There is, however, a problematic response on the statistics in the negative set where we can see a rise of the false positive rate. An explanation for this outcome is that the initial set was used to detect spam messages. Hence, there existed a number of spam messages that consisted of various emoticons and words with strong valence, which probably caused this problem. A countermeasure to bypass such cases would be to avoid take into consideration SMS that are longer than 160 characters. Figure 2 shows the optimization progress on the positive SMS set.

To conclude, the use of valence and emoticons optimized the attribution of the method to distinguish and classify SMS messages as positive or negative. Furthermore, as discussed in the literature review section, there exist more efficient methods in the area of Natural Language Processing than the simple 'bag-of-words' approach we presented here. These machine-learning algorithms are able to produce better results on sets collected from Twitter. We infer that these techniques can be used in SMS during a forensic analysis to produce more accurate results. However, our main goal in this paper is to illustrate how we can add text mining functionality to tools used for forensic investigations. In the next section we will present a tool we developed to demonstrate this concept.

5. Combining Mood Analysis with Forensics

It is possible to perform reliable forensic analysis of smartphones running the Android Operating System (e.g. [2]), utilizing open source tools and basic Linux commands like 'dd' in order to gain a physical image of the internal memory and its partitions (data, system, etc.). Once we mount the image to a Unix-like system (read-only), we will be able to see the data that applications store internally in SQLite databases. The appropriate folder where the SMS data are stored in the data partition

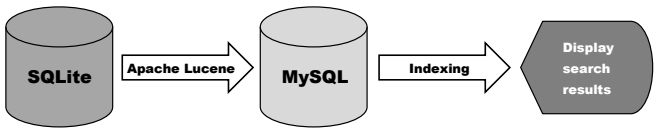


Figure 3. The design concept of our forensic tool.

is the: /data/com.android.providers.telephony/databases/mmssms.db. Thus, when the forensic examiner mounts the data partition on a computer, the sent and received SMS can be viewed using a database browser (looking at the SMS table of the database). The approach we propose here uses the aforementioned text-mining method to present the sentiment state of entities exchanging messages through the phone.

The common practice during a forensic analysis to view and elaborate text would be to use Linux commands like the 'strings' or the 'grep'. Furthermore, the analyst could use open source tools like the SQLite Database Browser[2] or SQLiteman[3] to view the content of databases. The tool we developed uses the Apache Lucene Library to utilize the text mining method we presented in this paper for mood analysis on SMS and, in addition, brings the ability to the analyst to search fast and efficient for keywords in the database using indexing. Indexing is the method that web search companies employ to perform fast and accurate searches and deliver reliable results. In other words, our tool works like a search machine for the database under examination in order to deliver faster results than an SQL query. The logic behind the implementation of the tool is depicted in Figure 3.

We first migrate all the requested content from the SQLite database (which is the database under investigation) to a MySQL database running in our developing machine. Apache Lucene will be responsible to produce a set of keywords of the individual SMS (without using stemming) and also to provide the stemmed words for the SMS mood analysis, as explained in previous sections. When all data have been processed, we utilize again Lucene to create an index of the MySQL database into the developing machine, which will be responsible to answer any text query the analyst might have. The advantages of this approach are mainly that the analyst will be able to indirectly query the SQLite database (through the index) without having any particular knowledge of the SQL language. One more advantage of this method is that the search is faster than querying the database itself.

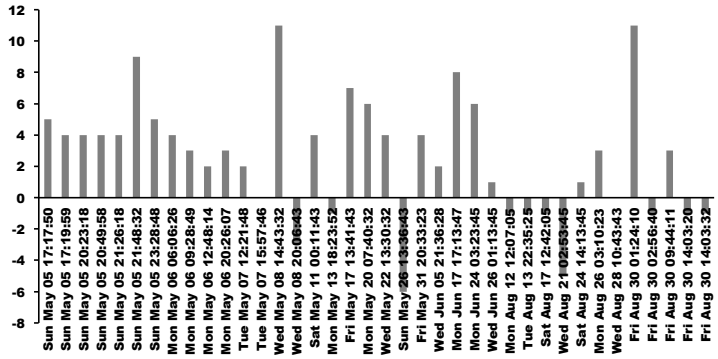
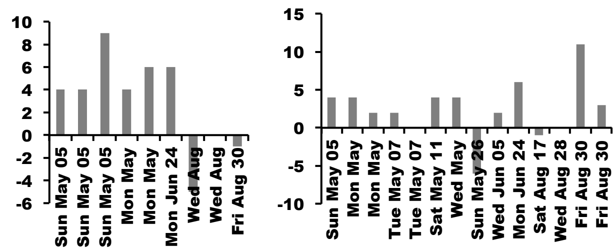


Figure 4. Timeline Emotional Analysis based on SMS found on the device featuring all SMS.

The tool was tested on an original image we took from a Samsung Galaxy Y S5360 smartphone running the Android 2.3.5 OS owned by one of the authors of this paper. Unfortunately, the database contained messages in a language different than English, so our lexicon could not produce any mood score for them. For the needs of this paper and for presentation issues we substituted SMS messages in mmssms.db with random tweets from our TWT database. After running the tool we were in possession of a MySQL database containing the data from mmssms.db and also the keywords extracted from each 'message' along with the calculated sentiment score.

The goal of the proposed framework is to provide to the examiner a timeline analysis interface that depicts the emotional inference of each SMS that was found in the smartphone. This could be an indication of the user's emotional state during a specific time interval. The information we could extract by such a viewing might be helpful to the analysts because they focus quickly on specific periods of time that could be of interest. For instance, when they are dealing with a case where the person under investigation is always in good mood and there exists a period of time the suspect seems to have negative emotions, it might be a good idea to start the examination by analyzing the specific timeframe.

Figure 4 illustrates the mood timeline view of the SMS extracted from the SQLite database including sent and received messages from all contacts. We observe that most of the time the smartphone owner exchanges messages with positive emotional fingerprint. There are although some periods (e.g. from 12th August until 17th August) when the sentiment score is negative and maybe this is a period that a forensic analyst would like to search more carefully. The proposed timeline view



(a) (b)

Figure 5. Timeline Emotional Analysis based on SMS found on the device: a) SMS exchanged with one entity, b) Sent SMS messages.

```
Found 4 hits.
1. It is a boy! So happy for my cousin Kate and the future King of England!
---- Database id: 11 on Mon May 06 20:26:07 JST 2013 received from: +3[redacted]48
2. Happy Halloween from me &amp; the dummy  http://t.co/jQxPvHqi
---- Database id: 32 on Mon Aug 26 03:10:23 JST 2013 received from: +4[redacted]94
3. Happy 4th of July! In 1776, Betsy Ross signed the Gettysburg Address and celebrated at the Boston Tea Party. It was a magical day.
---- Database id: 19 on Mon May 20 07:40:32 JST 2013 received from: +3[redacted]8
4. Congratulations to my dear friend @JimmyKimmel and his beautiful new wife! I am so happy for him. I had no idea he was straight.
---- Database id: 14 on Wed May 08 14:43:32 JST 2013 received from: +3[redacted]40
```

Figure 6. Searching the index.

provides the opportunity to the forensic investigator to act faster and also have a generic view of the user's behavior.

The presentation of data in the timeline view can be more consistent presenting only the communication between two parts, or indicating only the received or sent messages. Figure 5a) depicts the communication between the person under investigation and a single entity (+3xxxxxxxxxx8), based on the telephone number stored in the database and Figure 5b) illustrates the messages that where sent by the person under investigation to all its contacts. Finally, the index of the MySQL database our tool creates is helpful, because we provide the ability for searching the database in a way similar to a simple web search. In addition indexing is more efficient because it is faster than an SQL query. In large datasets (maybe with more interconnected databases) this feature could be critical for faster and accurate searches. Figure 6 demonstrates the results our tool provided after searching the index for the word 'happy'. As we can see there were 4 hits containing the keyword and the tool also returns details like the message id in the original SQLite database, the date the message was sent or received and the contact that performed the activity.

6. Conclusions and Future Work

In this study we proposed a tool that brings together three different areas: Information Retrieval with Text Mining, Smartphone Forensics and Intelligence demonstrating our concept to enhance the forensic analysis with Mood Analysis on SMS found at a smartphone. The aim is to diminish the investigation workload providing solutions that the forensic analyst will judge as valuable or not, depending on the particular case. The tool does not substitute the forensic analyst or the traditional and well-accepted methods but is auxiliary to the standardized forensic procedures and is designed aiming to visualize the emotional content of each SMS. We evaluated the efficiency of a bag-of-words approach to the problem of SMS mood analysis and showed that methods used for Twitter Sentiment Analysis can be migrated for the cause of depicting the emotional polarity of SMS found on a mobile phone.

The results we achieved using emoticons and the words sentiment valence, illustrate the potentiality of the proposed concept to attain better results. The optimization can be achieved by producing a more dynamic lexicon using methods that work better than the 'bag of words' approach we used here. The proof of concept we present here also includes indexing as a fast, tested and accurate technique for string searching. We, thus, merged traditional forensic examination techniques with text mining methods in order to expedite the retrieval of preliminary emotional indications from SMS. Taking into account that humans tend to follow specific behavioral patterns we demonstrate that these can be depicted through timelines ('Chronologies') like the one we present here. Our contribution to the area of Forensics is that with such tools and methods like the one we propose here, we can model the behavior of the person under investigation and extract useful conjectures related to their habits that are worth to be further investigated. Our intentions for the future are to employ the proposed Mood Analysis method to produce a Timeline View for the entire context that can be found in a smartphone (emails, instant messengers, notes, social network activity). Also, we will investigate if the use of SVM classifiers will enhance the efficiency of the tool.

Acknowledgements

This work has been supported by the European Union's Prevention of and Fight against Crime Programme "Illegal Use of Internet" - ISEC 2010 Action Grants, grant ref. HOME/2010/ISEC/AG/INT-002, the National Institute of Informatics, the Systems Centre of the University

of Bristol and also by the project NIFTy (HOME/2012/ISEC/AG/INT/4000003892).

Notes

1. http://en.wikipedia.org/wiki/List_of_emoticons
2. http://sourceforge.net/projects/sqlitebrowser/
3. http://sqliteman.com

References

[1] T.A. Almeida, J.M.G. Hidalgo and A. Yamakami, Contributions to the study of SMS spam filtering: new collection and results, *Proceedings of the 11th ACM symposium on Document engineering*, pp. 259–262, 2011.

[2] P. Andriotis, G. Oikonomou and T. Tryfonas, Forensic analysis of wireless networking evidence of Android smartphones, *Proceedings of Information Forensics and Security (WIFS) IEEE International Workshop*, pp. 109–114, 2012.

[3] N. Beebe, Digital Forensic Research: The Good, the Bad and the Unaddressed, *Advances in Digital Forensics V*, G. Peterson and S. Shenoi (Eds.), Springer, Berlin Heidelberg, p.p. 17–36, 2009.

[4] N.L. Beebe and J.G. Clark, Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results, *Digital Investigation*, vol. 4, pp. 49–54, 2007.

[5] A. Bermingham and A.F. Smeaton, Classifying sentiment in microblogs: is brevity an advantage?, *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1833–1836, 2010.

[6] M. Choudhury, R. Saraf, V. Jain, A. Mukherjee, S. Sarkar and A. Basu, Investigation and modeling of the structure of texting language. *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 10, no. 3–4, pp. 157–174, 2007.

[7] S.R. Das and M.Y. Chen, Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science*, vol. 53, no. 9, pp. 1375–1388, 2007.

[8] X. Ding, B. Liu and P.S. Yu, A holistic lexicon-based approach to opinion mining, *Proceedings of the international conference on Web search and web data mining*, pp. 231–240, 2008.

[9] D. Estival, T. Gaustad, S.B. Pham, W. Radford and B. Hutchinson, Author profiling for English emails, *Proceedings of the 10th Confer-*

ence of the Pacific Association for Computational Linguistics, pp. 263–272, 2007.

[10] A. Go, R. Bhayani and L. Huang, Twitter sentiment classification using distant supervision, *CS224N Project Report, Stanford*, pp. 1–12, 2009.

[11] L.K. Hansen, A. Arvidsson, F.Å. Nielsen, E. Colleoni and M. Etter, Good friends, bad news-affect and virality in twitter, *Future Information Technology*, J.J. Park, L.T. Yang, C. Lee (Eds), Springer Berlin Heidelberg, pp. 34–43, 2011.

[12] G. Laboreiro, L. Sarmento, J. Teixeira and E. Oliveira, Tokenizing micro-blogging messages using a text classification approach. *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pp. 81–88, 2010.

[13] C.K. Leong, Y.H. Lee and W.K. Mak, Mining sentiments in SMS texts for teaching evaluation, *Expert Systems with Applications*, vol. 39, no. 3, pp. 2584–2589, 2012.

[14] N. Li and D.D. Wu, Using text mining and sentiment analysis for online forums hotspot detection and forecast. *Decision Support Systems*, vol. 48, no. 2, pp. 354–368, 2010.

[15] 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*, pp. 1275–1284, 2009.

[16] S.M. Mohammad, S. Kiritchenko and X. Zhu, NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets, *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, 2013.

[17] A. Orebaugh, An Instant Messaging Intrusion Detection System Framework: Using character frequency analysis for authorship identification and validation. *Carnahan Conferences Security Technology, Proceedings 2006 40th Annual IEEE International*, pp. 160–172, 2006.

[18] A. Pak and P. Paroubek, Twitter as a Corpus for Sentiment Analysis and Opinion Mining, *LREC*, 2010.

[19] B. Pang and L. Lee, Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[20] B. Pang, L. Lee and S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of*

the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pp. 79–86, 2002.

[21] M.F. Porter, An algorithm for suffix stripping, *Program: electronic library and information systems*, vol, 14, no. 3, pp. 130–137, 1980.

[22] C. Strapparava and R. Mihalcea, Semeval-2007 task 14: Affective text. *Proceedings of the 4th International Workshop on Semantic Evaluations*, pp. 70–74, 2007.

[23] C. Strapparava and A. Valitutti, WordNet Affect: an Affective Extension of WordNet, *LREC* vol. 4, pp. 1083–1086, 2004.

[24] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, Lexicon-based methods for sentiment analysis, *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[25] Z. Wang, L. Zhai,Y. Ma and Y. Li, Analysis of Public Sentiment Based on SMS Content, *Trustworthy Computing and Services*, Y. Yuan, X. Wu and Y. Lu (Eds.), Springer Berlin Heidelberg, vol. 320, pp. 637–643, 2013.

[26] H. Yu and V. Hatzivassiloglou, Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 129–136, 2013.