Analyzing, Classifying, and Interpreting Emotions in Software Users' Tweets

Grant Williams* and Anas Mahmoud†,
Division of Computer Science and Engineering, Louisiana Sate University
Baton Rouge, LA, 70803
*gwill83@lsu.edu,†mahmoud@csc.lsu.edu

***Abstract*—Twitter enables software developers to track users' reactions to newly released systems. Such information, often expressed in the form of raw emotions, can be leveraged to enable a more informed software release process. However, automatically capturing and interpreting multi-dimensional structures of human emotions expressed in Twitter messages is not a trivial task. Challenges stem from the scale of the data available, its inherently sparse nature, and the high percentage of domain-specific words. Motivated by these observations, in this paper we present a preliminary study aimed at detecting, classifying, and interpreting emotions in software users' tweets. A dataset of 1000 tweets sampled from a broad range of software systems' Twitter feeds is used to conduct our analysis. Our results show that supervised text classifiers (Naive Bayes and Support vector Machines) are more accurate than general-purpose sentiment analysis techniques in detecting general and specific emotions expressed in software-relevant Tweets.**

I. Introduction

In recent years, Twitter has become one of the most popular micro-blogging social-media platforms, providing an outlet for millions of people to share their daily activities through real-time status updates. As of the fourth quarter of 2015, Twitter has averaged around 305 million monthly active users [1]. The sheer volume of user-generated information available through Twitter feeds has revolutionized research in a broad range of human sciences [1], [2], [3]. For instance, researchers have leveraged the aggregate of millions of Twitter messages posted on a daily basis to predict the daily ups and downs of the stock market [1], predict the political affiliation of the masses [2], and uncover and explain temporal variations in social happiness [3].

From a software engineering perspective, Twitter has created an unprecedented opportunity for software providers to monitor the opinions of large populations of end-users of their systems. Using Twitter, software users can publicly express their feelings in the form of micro-blogs, known as *tweets*. In fact, it has become a social media tradition that with the release of each new mobile application, video game, or operating system, people resort to Twitter to describe their experiences and problems and recommend software to their friends, leading these systems to be *trending* worldwide. Such data can be leveraged to understand and rationalize people's emotions toward newly-released software or features, and thus help software developers plan better for future releases of their systems [7].

Emotions in Twitter messages can be detected using sentiment analysis techniques. Sentiment analysis is concerned with determining whether a text conveys positive or negative feelings. In general, sentiment analysis techniques rely on the presence of English opinion lexicons and emotion-evoking words (e.g., *love, hate, like*) to detect feelings in text. However, software relevant tweets often include computer jargon words (e.g., *brick, uninstall, fix, and crash*). These words carry multi-dimensional structures of positive and negative human emotions that are typically overlooked by general-purpose sentiment analysis methods. To address these challenges, in this paper we present a preliminary analysis aimed at detecting and interpreting emotions present in software-relevant tweets. Our analysis is conducted using a dataset of 1000 tweets sampled from the Twitter feeds of a broad range of software systems. Our objectives are to **a)** identify the most effective techniques in detecting emotions and collective mood states in software-relevant tweets, and **b)** investigate how such emotions are correlated with specific software-related events.

The remainder of this paper is organized as follows. Section II describes our research problem and motivates our work. Section III presents our preliminary experimental analysis and discusses our main findings and their potential impact. Finally, Section IV concludes the paper and describes prospects of future work.

II. Background and Motivation

Researchers in behavioral economics, politics, and social studies have reported that emotions play a significant role in human decision-making [4]. Such information is typically collected through face-to-face, email, poll, or survey communication. In recent years, Twitter has emerged as a more instant and a more wide-spread source of public information that can complement traditional data collection methods. Tweets analyzed through sentiment analysis techniques can be used to infer the public's mood toward social, political, and economical issues [5], [6]. For instance, O'Connor et al. [5] aligned public opinions extracted from traditional polls with sentiments measured from Twitter. The authors detected a high correlation rate between sentiment word frequencies in Twitter messages and consumer confidence levels and political opinions as indicated by the polls. Similarly, Bollen et al. [1]

[1]https://about.twitter.com/company

analyzed the textual content of daily Twitter feeds by mood tracking tools. The authors reported that the accuracy of stock market predictions can be significantly improved by the inclusion of specific Twitter mood dimensions. Following this line of research, in this paper we assume that emotions expressed by end users of software in software systems' Twitter feeds represent a valuable source of information for software developers. Such information can be leveraged to understand users' reaction to newly released software. An underlying tenet is that user involvement in the software process is a major contributing factor to software success [7].

In what follows, we demonstrate through the new operating system iOS10's Twitter feed how the public opinion, as measured by Twitter sentiment, fluctuates in correspondence to specific software-related events.

A. *Example: The iOS10 release*

On September 17th, 2016, Apple announced their new operating system for their popular smart phone the iPhone. We collected tweets mentioning the term "ios10" or containing the hashtag "#ios10" over the period from July 29th to September 19th. We analyzed the sentiment in each tweet using *SentiStrength*, a tool that associates common words and phrases with sentiment scores[2]. SentiStrength returns an integer value in the range [-4, 4], representing the overall sentiment of the tweet such that 0 is the neutral state. Figure 1 shows the average sentiment polarity calculated for our tweets over the data collection period, the total number of tweets collected, and important events related to iOS10.

During the public beta-testing process, tweets covered a variety of topics. For instance, the look-and-feel features of the new iOS has generated mainly positive reactions, with tweets such as *"Ios 10 seems pretty smooth and have some beautiful animation. Im waiting . #iphone #iOS10 A whole new ios"*, and *"I ain't gonna lie this #ios10 on my iPhone 6s is pretty aesteticly pleasing"*. The usability of the new OS, however, has generated some negative reactions, mainly due to users struggling with the new interface. For instance, an issue that frequently stood out in the tweets was user complaining about the loss of the swipe-to-unlock feature with tweets such as *"won't upgrade to #iOS10 because the slide to unlock has been removed and I don't care any new features. #Apple #SlideToUnlock"*. In fact, the public negative reaction to this particular feature was so significant that it received news coverage by *CNN*:

> *"to unlock an iPhone, you no longer swipe left but hit the home button. If you use a fingerprint to unlock every time, you wont notice. But others will repeatedly try to log on and instead see a new screen with weather, calendar and other bites of information. It will grow on us, probably."*

[2] http://sentistrength.wlv.ac.uk/

Another controversial issue leading up to iOS10's release was the replacement of the default set of *emojis* with a new one. A number of tweets were posted on this topic, such as *"I don't like ios10 emojis. they look like android emojis"* and *"I hate the emojis on ios10 so much how do I downdate"*. Our analysis also shows a spike in the positive sentiment on September 7th, which is the date of Apple's Fall event (an annual keynote event). During this event the iPhone 7 was announced and a number of Twitter posts exhibited excitement and anticipation, in many cases directly referencing #AppleEvent. Tweets such as *"Time to update my iphone #iOS10 #AppleEvent"*, and *"I just can't wait to upgrade to IOS10 on Tuesday"*, were common.

A noticeable drop in sentiment corresponded with the actual release of iOS10, where the excitement was tempered by some frustration over technical issues. The rate of posting surged to over twenty thousand tweets the day of iOS10's release, and a number of users reported having their phone rendered unbootable (*"bricked"*) in tweets such as *"iOS 10 over-the-air update bricked my 6s. Downloaded, installed, then rebooted to a plug in to iTunes notice. iTunes has to fix. #ios10"* and *"Bricked my iPhone while updating to iOS10, stuck in recovery mode. Restoring in iTunes now"*. We also see a public reaction to changes that were introduced in the beta, but were only experienced by many users on release. Ordinary users reacted more harshly to the new emojis, echoing the sentiment expressed during the early betas: *"Main reason I havent́ updated to iOS10 yet is simply because of the gun emoji...that really just makes me mad"*, and *"i hate how emojis look on ios10 lmao"*.

B. *Motivation*

The iOS10 example shows how public sentiment, as measured by SentiStrength, can drastically change in response to specific software-related events. However, like many sentiment analysis tools, SentiStrength adheres to a uni-dimensional model of mood, making binary distinctions between positive and negative sentiment. Naturally, in our analysis, we assume that a negative sentiment is associated with a bad experience, such as a buggy update or a disappointing beta. A positive sentiment, on the other hand, might indicate a positive experience, such as a new feature being well-received. This binary classification of sentiment, while possibly giving a generalized indication of the public sentiment, may ignore the rich multi-dimensional structure of the human mood. In particular, the human positive and negative moods can be further broken down into specific emotions such as anger, excitement, and frustration. Each of these emotions conveys a different type of information that can be interpreted in various ways. For example, in their analysis of the stock market's movement, Bollen et al. [1] observed that public mood states measured in terms of positive vs. negative did not result in any correlation with stock market movement; rather, the specific emotional dimension *"calm"* was the most predictive of the stock market.

Motivated by these observations, in what follows, we investigate the performance of various sentiment analysis tech-

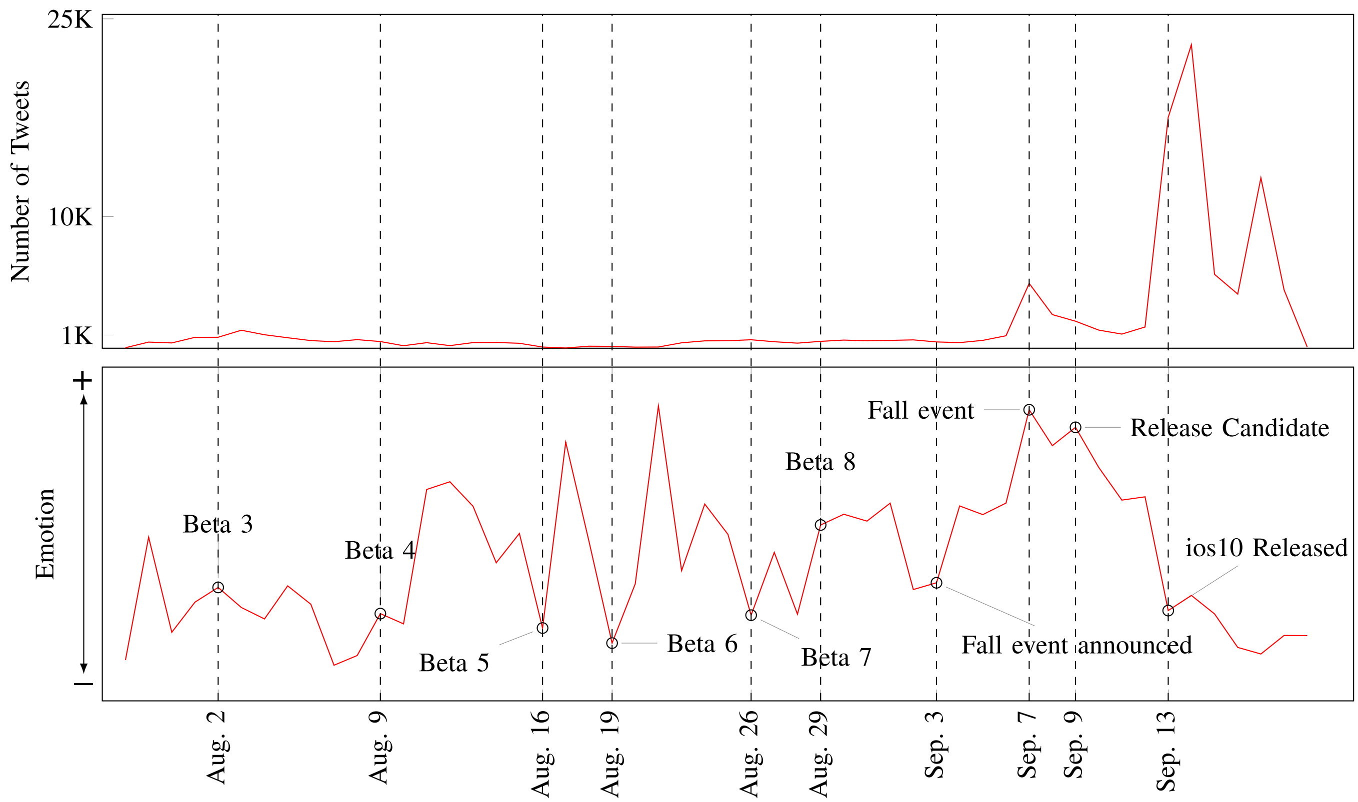

Fig. 1. Number of tweets per day (top), and aggregate sentiment polarity (bottom) with important events marked.

niques in detecting emotion information expressed in software user tweets. Our ultimate goal is to build a sentiment analyzer that is customized to detect and translate these emotions into actionable software engineering requests.

III. Analysis and Approach

This section describes our data collection, qualitative analysis, and automated classification process, and presents and discusses our results.

A. Data Collection

We used the Twitter *Search API* to collect our dataset. This API takes a search query (a strings or a hashtag) related to a certain topic, and returns a set of tweets matching the query (i.e, potentially relevant to the search topic). To conduct our analysis, we collected tweets from the Twitter feeds of 10 software systems, sampled from a broad range of application domains. These systems include: *Windows10, Android, Apple-Support, CallofDuty, Chrome, Instagram, Minecraft, Snapchat, VisualStudio,* and *WhatsApp*. We limited our data collection process to tweets addressed directly to the Twitter account of a given software product (e.g., tweets including *@Windows10*). This strategy ensures that only tweets that are meant to be a direct interaction with the software provider are included. The data collection process was repeated on a daily basis from April 6th to June 4th of 2016, with duplicate tweets being discarded. The resulting dataset contained $360,873$ tweets.

B. Qualitative Analysis

To create our ground-truth dataset, 1000 tweets were randomly sampled from our dataset. These tweets were manually examined by two human annotators, with an average 5 years of experience in programming, to identify subjective expressions. A subjective expression is any word or phrase that is used to express an opinion, emotion, evaluation, stance, or speculation. Each tweet is classified at two levels of abstraction, including its general emotional polarity (positive, negative, and neutral) and the specific emotions it carries (sub-categories of the general negative or positive emotion the tweet conveys). Our qualitative analysis revealed the following types of emotions in our collected tweets:

- *Frustration*: A frustration feeling typically signifies the presence of bugs or unwanted behavior (e.g., *"@ifunny im sick of random ads poping up everywhere"*). For instance, in the domain of video games, *frustration* often signifies excessive difficulty or problems in game control (e.g., *"@CallofDuty @Treyarch Any chance you can make spawns worse? I don't feel like I've had a proper game if Im not spawn killed 8 times a match"*. Frustration is typically associated with terms and phrases such as *sick*, *kill*, and *frustrating*.
- *Anticipation and Excitement*: Tweets with high anticipation inform developers about the features users are looking forward to, for example *"@googlechrome I'm*

TABLE I
SAMPLE SOFTWARE-RELEVANT EMOTIONS

Word/phrase	Polarity	Emotion	Examples
crash, not-working, fix	negative	bug report	*"VisualStudio Visual Studio15 Preview new installer not working"*
listen, waste-of-time	negative	frustrated with update	*"Listen to the people yik yak... fix it :(#yikyak "*
uninstall, bring-the-old, go-back, change-back, ruin	negative	unsatisfied with update	*"Can we bring the old #facebook back"* *"Well PokemonGoApp new tracking ruined the game for those not living in the city. Nice way to kill the game. #uninstall"*
re-download, addicted, obsessed	positive	satisfaction	*"to say that im obsessed with the new #snapchat filters is an understatement"* *"finally!!! I can now redownload #yikyak back"*
cannot-wait, excited-for	positive	excitements and anticipation	*"can the new #callofduty come out already? cant wait ugh"*

starting to regain my respect for Chromebooks. Because you are putting Google Play. Man i'm so hyped.". Excitement commonly appears in video game Twitter feeds, for example, a user bringing attention to their achievement in a game (e.g., *"@Minecraft Check out this @Sway I made! "hero of the city""*). Anticipation and excitement are typically associated with terms and phrases such as *hyped*, *looking forward to*, and *can't wait*.

- *Satisfaction*: Satisfaction and dissatisfaction emotions are typically related to how users feel about software features. For example, when a new feature is well received, users often react with tweets such as *"@googlechrome love the canary build of chrome i thought it be alot buggy but its not its working fine and like the new look its alot better"*. When features are poorly received, users often react with tweets such as *"@instagram I'm sorry but I absolutely HATE the new update"*. Users' level of satisfaction allows developers to plan better for future patches and updates. Satisfaction is associated with phrases such as *"love the new"* and *"great job on"*, while dissatisfaction is typically associated with phrases such as *"change it back"* and *"why did you"*.
- *Bug reports*: A bug report is not an emotion *per se*, however, such tweets often carry a compound negative sentiment. More specifically, tweets reporting problems are often accompanied by frustration and dissatisfaction feelings, expressed through phrases such as *"please fix"* and *"any solutions?"* (e.g., *"@googlechrome Your android upgrade removed all of my open tabs. Going thru history to get them back is onerous can you fix this?"*).

A summary of our detected emotions as well as the list of their evoking expressions and sample tweets is shown in Table I. Table II shows the number of tweets in our dataset that exhibit the different specific emotions[3]. The intensity of the emotion is not considered at this stage of our analysis.

C. Automated Sentiment Analysis

Manually filtering through massive amounts of tweets can be a laborious and error-prone task. Therefore, for any solution to be practical, automated support is needed to facilitate a more effective data filtering process that can capture, with a decent level of accuracy, emotions in software-relevant tweets.

In general, automated sentiment classification methods can be classified into unsupervised and supervised. The unsupervised approach relies on the presence of opinion lexicons, or emotion-indicator words, to estimate the sentiment polarity of the tweet based on the positive-to-negative word ratio, or simply the raw counts of opinion words [5]. The lexical approach focuses on building dictionaries of labeled words, where each word is given a score that indicates its emotional polarity. A common way to classify a text using these scores is by adding the positive values and subtracting the negative values of the terms in the text. If the total score is positive, the text is classified as positive, otherwise it is negative.

While the unsupervised approach can be easily implemented, it can be difficult to collect and maintain a universal sentiment lexicon as different words may reflect different meanings in different contexts [8], [9], [10]. Furthermore, simply relying on the presence of certain emotion words can lead to misleading results. This problem is often observed in Twitter messages due to their limited, and often ambiguous, textual content. For example, the tweet *"@Microsoft I love #Windows10, but not as much as I loved #Windows8.1"* carry both positive and negative feelings toward Windows10. However, a dictionary-based sentiment analyzer will classify this tweet as positive due to the presence of the word *"love"* twice in the tweet.

The supervised approach, on the other hand, attempts to overcome these limitations by training prediction models (e.g., Naive Bayes and Support Vector Machines) on manually labeled tweets to make sentiment predictions for new data [2], [11]. OpinionFinder[4], is an example of a supervised sentiment classifiers that is trained on the Multi-perspective Question

[3]Data is publicly available at http://seel.cse.lsu.edu/data/semotion17.zip

[4]http://mpqa.cs.pitt.edu/opinionfinder/

TABLE II
NUMBER OF TWEETS CONVEYING EACH EMOTION

Emotion	#Present	#Absent
Frustration	209	284
Dissatisfaction	133	360
Bug Report	218	275
Total negative:	493	
Satisfaction	182	177
Anticipation	42	317
Excitement	131	228
Total positive:	359	

TABLE III
POLAR SENTIMENT CLASSIFICATION RESULTS

Classifier/Dataset	Precision	Recall	F
SentiStrength Positive	0.76	0.73	0.74
SentiStrength Negative	0.69	0.65	0.67
NB Positive	0.81	0.81	0.81
NB Negative	0.77	0.77	0.77
SVM Positive	0.78	0.78	0.78
SVM Negative	0.70	0.70	0.70

Answering (MPQA) Opinion Corpus. This corpus contains news articles from various news sources manually annotated for opinions (i.e., beliefs, emotions, sentiments, speculations, etc.). A main limitation of this approach is that a model that is trained using a certain corpus might not be able generalize well for other domains. Furthermore, preparing large enough datasets of manually labeled emotion text is a labor-intensive, extremely subjective, and time-consuming task [12].

To classify our data, we investigate the performance of two text classification algorithms, including Naive Bayes (NB) and Support Vector Machines (SVM). These two algorithms have been heavily used in Twitter sentiment analysis and have shown interchangeably good performance across a broad range of tasks [13], [11]. To implement NB and SVM, we use Weka[5], a data mining software suite that implements a wide variety of machine learning and classification techniques. SVM is invoked through Weka's Sequential Minimal Optimization (SMO) class, which implements John Platt's algorithm for training a support vector classifier [14]. In our preliminary analysis, we find the default linear kernel of SVM to be most effective for Twitter sentiment classification. To train our classifiers, we use 10-fold cross validation. This method of evaluation creates 10 partitions of the dataset such that each partition has 90% of the instances as a training set and 10% as an evaluation set.

D. Results and Discussion

To get a sense of the performance of NB and SVM, we initially classify the data using SentiStrength. In our analysis, we classify a tweet as negative if it gets a score of -2 or less and as positive if it gets a sentiment score of +2 or more. Table III summarizes the results of the different classifiers. The relatively weak performance of SentiStrength in comparison to the supervised methods can be attributed to the lack of software-specific words in its sentiment dictionary. For example, the tweet *"@Windows > @apple The surface is smaller but yet more powerful! #WindowsVsApple @surface"* has a positive polarity. SentiStrength miss-classified this tweet as negative due to the presence of the word *"smaller"*. In English, this word leans toward negative sentiment, but in the context of software, smaller size is typically an indication of positive sentiment. Along these lines, the tweet *"@VisualStudio is VSTS going SUPER slow?"* was miss-classified by SentiStrength as positive due to the word *"super"*.

In comparison, our results show that the supervised classifiers NB and SVM managed to achieve decent levels of accuracy in detecting general (Table III) and specific emotions (Table IV), with NB outperforming SVM. In general, both classifiers recognized many positive and negative emotion-evoking software-specific words that SentiStrength missed. For example, the words *"challenge"* and *"backwards"* are typically associated with negative sentiment in day-to-day English. However, in the context of video games, the word *"challenge"* typically indicates a good mood. The word *"backwards"* often appears in the context of backwards compatibility, which is also a positive attribute of software. Similarly, the supervised methods managed to capture words that are considered neutral in day-to-day English, but which are often associated with negative sentiment in software, such as *"uninstall"* and *"roll-back"*. For example, both NB and SVM detected that the word *"please"* is almost always an indication of negative sentiment. In general, the word *"please"* comes associated with requests to help with software problems, such as *"@googlechrome please i need ur help! i don't have sound with google chrome!! and i have not found a solution yet!! could you help me please"*. Similarly, the tweets with the words *"fix"*, *"bug"*, and *"crash"*, were classified as negative as they are typically associated with requests for bug fixes and systems crashing.

In general, in terms of general emotional polarity, our results show that the supervised classifiers can be effective in detecting software-specific emotion words that are often missed by general-purpose sentiment analyzers. However, getting such techniques to achieve acceptable accuracy requires preparing large-scale datasets of manually annotated Twitter data. This can be a challenging task as the language of Twitter messages evolve at very fast pace and keeping track of such noisy colloquial terminologies can be a time-consuming task.

IV. CONCLUSIONS

The research on mining micro-blogging services for software engineering purposes has focused on the way developers use such platforms to share and exchange software development information. Analysis of sample software-developers' tweets has revealed that they frequently make use of social media as a means to facilitate team coordination, learn about new technologies, and stay in touch with the interests and

[5]http://www.cs.waikato.ac.nz/~ml/weka/

TABLE IV
SPECIFIC EMOTION CLASSIFICATION RESULTS (NB AND SVM)

	NB			SVM		
Emotion	**P**	**R**	**F**	**P**	**R**	**F**
Frustration	0.71	0.71	0.71	0.68	0.68	0.68
Dissatisfaction	0.75	0.77	0.76	0.77	0.77	0.77
Bug Report	0.75	0.75	0.75	0.71	0.71	0.71
Satisfaction	0.75	0.76	0.75	0.64	0.64	0.64
Anticipation	0.86	0.89	0.87	0.85	0.87	0.86
Excitement	0.75	0.76	0.75	0.85	0.85	0.85

opinions of all stakeholders [16], [17]. In our analysis, we shift the attention to software users' tweets rather than developers' tweets. In particular, we show that emotions expressed in software systems' Twitter feeds can be a non-traditional source of feedback that enables software developers to instantly connect to, interpret and rationalize their end-users reactions.

Our analysis is conduced using 1000 tweets sampled from the Twitter feeds of multiple software systems. A manual qualitative analysis was conducted to classify these tweets based on their sentiment polarity and specific emotions they express. The data was then automatically classified using SentiStrength and two general purpose text classifiers, including NB and SVM. The results showed that NB and SVM were more accurate than SentiStrength in detecting the emotional polarity of software tweets. Furthermore, both classifiers were able to capture the fine-grained dimensions of human emotions with decent levels of accuracy.

The work presented in this paper can be described as a preliminary proof-of-concept analysis of the value of emotion information in software-relevant tweets. The proposed work can be extended along two main directions, including:

- Analysis: A major part of our future analysis will be focused on improving the accuracy of the sentiment classification model. For instance, in our future work, non-word sentiment signals in tweets, including emoticons and punctuation, will be considered as classification features. Emoticons can be used to predict both the main sentiment of the tweet as well as the intensity of the emotion [18]. For example, a sad smiley `:(` indicates a negative feeling, while the `:(((` emoticon gives an indication of a more intense sense of unsatisfaction [12]. Basic English punctuation can also be utilized to measure the intensity of the emotion. For instance, multiple question marks often indicate more intense confusion, while multiple exclamation points indicate intensity on both positive and negative side. Furthermore, the analysis in our paper is limited to emotion evoking uni-grams, or words. Related research has shown that more complex emotions are better expressed through expressions and phrases (combinations of using uni-grams and bi-grams) [20], [21]. Therefore, part of our future work will be to examine emotion-evoking phrases and word structures (n-grams) in software-relevant tweets.
- Applications: One of the main challenges in our proposed work is to develop a mechanism for automatically interpreting various types of emotions in different application domains. Our goal is to devise an emotion-aware model that can provide instant recommendations to software developers based on the public's mood.

REFERENCES

[1] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.
[2] M. Conover, B. Goncalves, J. Ratkiewicz, A. Flammini, and F. Mencze, "Predicting political preference of twitter users," in *Advances in Social Networks Analysis and Mining*, 2013, pp. 289–305.
[3] P. Dodds, K. D. Harris, I. Kloumann, C. Bliss, and C. Danforth, "Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter," *PLoS ONE*, vol. 6, no. 12, 2011.
[4] A. Damasio, *Descartes' Error: Emotion, Reason, and the Human Brain*. Penguin Books, 2005.
[5] B. O'Connor, R. Balasubramanyan, B. Routledge, and N. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," in *International AAAI Conference on Weblogs and Social Media*, 2010, pp. 122–129.
[6] S. Asur and B. Huberman, "Predicting the future with social media," in *International Conference on Web Intelligence and Intelligent Agent Technology*, 2010, pp. 492–499.
[7] M. Bano and D. Zowghi, "A systematic review on the relationship between user involvement and system success," *Information and Software Technology*, vol. 58, pp. 148–169, 2015.
[8] G. Qiu, B. Liu, J. Bu, and C. Chen, "Expanding domain sentiment lexicon through double propagation," in *International Jont Conference on Artifical Intelligence*, 2009, pp. 1199–1204.
[9] N. Kaji and M. Kitsuregawa, "Building lexicon for sentiment analysis from massive collection of HTML documents," in *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, pp. 1075–1083.
[10] P. Turney, "Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews," in *Annual Meeting on Association for Computational Linguistics*, 2002, pp. 417–424.
[11] S. Wang and C. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Annual Meeting of the Association for Computational Linguistics*, 2012, pp. 90–94.
[12] K.-L. Liu, W.-J. Li, and M. Guo, "Emoticon smoothed language models for twitter sentiment analysis," in *AAAI Conference on Artificial Intelligence*, 2012, pp. 1678–1684.
[13] A. Bermingham and A. Smeaton, "Classifying sentiment in microblogs: Is brevity an advantage?" in *International Conference on Information and Knowledge Management*, 2010, pp. 1833–1836.
[14] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola, Eds. MIT Press, 1998.
[15] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, 1995, pp. 1137–1145.
[16] L. Singer, F. Filho, and M. Storey, "Software engineering at the speed of light: How developers stay current using Twitter," in *International Conference on Software Engineering*, 2014, pp. 211–221.
[17] Y. Tian, P. Achananuparp, I. Lubis, D. Lo, and E. Lim, "What does software engineering community microblog about?" in *Working Conference on Mining Software Repositories*, 2012, pp. 247–250.
[18] H. Wang and J. Castanon, "Sentiment expression via emoticons on social media," in *IEEE International Conference on Big Data*, 2015, pp. 2404–2408.
[19] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Trans. Inf. Syst.*, vol. 22, no. 2, pp. 179–214, 2004.
[20] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Language Resources and Evaluation Conference*, 2010, pp. 1320–1326.
[21] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," in *Stanford Digital Library Technologies Project*, 2009.