Emotion Mining and Sentiment Analysis in Software Engineering Domain

Arvinder Kaur, Amrit Pal Singh, Guneet Singh Dhillon, Divesh Bisht
University School of Information & Communication Technology
Guru Gobind Singh Indraprastha University
New Delhi, India
arvinder70@gmail.com, amritpal1986@gmail.com, gsgsdhillon@gmail.com, doubledeep007@gmail.com

***Abstract*— In last few years, researchers' interest in sentiment analysis in the software engineering domain has risen significantly. Most of the studies show the use of standard sentiment analysis tools such as NLTK API and Sentistrength to find sentiments in issue reports extracted from some version maintenance software. Since these tools are not trained on software engineering texts and comments, their results may not be as accurate as one might hope. In this study we find to what extent do our human evaluators who are familiar with the software engineering terms agree with sentiment analysis tools on presence or absence of emotions. We perform this study on our own supervised dataset (Apache JIRA). In addition to this, we evaluate the performance of the metrics when a new tool is introduced and concluded that disagreement between tools leads to disagreement among results.**

Keywords—Data collections; emotion mining; sentiment analysis.

I. INTRODUCTION

Data and knowledge are of prime importance in today's world. Data is available all across the internet, knowledge is what gives you lead [14]. A better insight of industry, your employees and workers can be gained if one knows what people think. "What other people think" has always been an important piece of information for most of us during the decision-making process" [1]. Sentiments are the best source to gather such kind of information. Study of sentiments in various domains around us gives us useful information as to what kind of emotions people are experiencing while working. For example, consider this comment by "Markus Jelsma" on 06/13/17 on the LUCENE-7695 issue report:

Hello Mikhail Khludnev, your patch works nicely!

This comment tells us that the author is happy with the work of a colleague and is appreciating her/his efforts for getting the job done. There is a sense of joy in the comment and it tells us that the atmosphere of the forum is positive. Positive atmosphere not only has a significant impact on the capabilities and work ethics of an individual, it also enhances the collective productivity of the team and gives an equal chance for the callow beginners to learn and blend in an industrial environment [3]. Similarly here is another comment from LUCENE-7685

What on earth did this comment mean? This makes rewritten query equal the original, so that user does not have to .rewrite() their query before searching. Why would a user have to (in the past) manually rewrite their query before searching.

This one indicates that the author of this comments is little angry about an arte-fact or a fellow programmer, which creates a sense negativity in the working environment. Hence an effective study about these aspects of Sentiment Analysis in Software engineering domain might help us understand the need to have a positive and buoyant working atmosphere in the industry [3].

Sentiment analysis is "the computational study of opinions, sentiments, and emotions expressed in text" [2]. In the field of Natural Language Processing and Sentiment Analysis collecting manually labelled supervised data is a laborious work and consumes a lot of time and resources. Hence, this makes Sentiment Analysis problem more complex and difficult to tackle.

This paper introduces an approach to manually categorize texts on the basis of their sentiments [3]. We followed the procedure performed by Murgia, Tourani, Adams, & Ortu. 2014 with some adaptations which provide the evaluators a more comprehensive knowledge of the context. In addition to this, we also calculated a couple of inter-rater agreement metrics rather than calculating accuracy because there is no benchmark for calculating the accuracy of manually labelled comments [4]. Inter-rater agreements show the level of agreement achieved between human raters and standard polarity analysis tools [4].

For our study of sentiment analysis in software engineering domain, the data was collected from the issue reports of a version maintenance system (Apache JIRA), using python scripts. We clean the data so that it suited our needs, performed the mining of emotions from those comments and followed by computation of metrics.

II. RELATED WORK

Before effective analysis of polarities, supervised data is required. As described by Pang and Lee 2008 [1], "Emotion mining tries to identify the presence of human emotions like

joy or fear from text, voice and video artefacts produced by humans". Humans can show numerous amounts of emotions like *sadness, happiness, anger, fear* etc in their comments. Murgia, Tourani, Adams, & Ortu. 2014 standardized these emotions using an emotional framework. We use a similar paradigm as the base for the identification of emotions in our study.

Sentiment analysis and emotion mining has been recently applied by Jongeling, Sarkar, Datta, & Serebrenik, 2017 to study the agreement between different tools and human evaluators [4]. For labelling the comments, they use the dataset provided by Murgia, Tourani, Adams, & Ortu. 2014 and employed 4 evaluators. Later they used standard sentiment analysis tools on the dataset to find the polarity of those texts. Tools used by Jongeling, Sarkar, Datta, & Serebrenik, 2017 were NLTK, SentiStrength, Alchemy and Stanford NLP [4]. We apply NLTK, SentiStrength, WatsonNLU (an update of Stanford NLP) and Microsoft Azure's TextAnalytics to our dataset and adapt the interpretations of first two from Jongeling, Sarkar, Datta, & Serebrenik, 2017 [4].

We used Adjusted Rand Index (ARI) (Hubert and Arabie 1985; Santos and Embrechts 2009) [5] and weighted Kappa (*K)* as suggested by Bakeman and Gottman (Bakeman and Gottman 1997, p. 66) as the evaluation metrics [6]. These metrics were also used by Jongeling, Sarkar, Datta, & Serebrenik, 2017 [4] and hence are a good pick for the analysis. Weighted kappa is an inter-rater agreement metric whose value is 1 for perfect agreement and 0 for no agreement. For the interpretation of the kappa we chose the schema provided by Viera and Garrett 2005 [7].

III. Methodology

A. Data Collection

We perform emotion mining and sentiment analysis on the data which we collected from the issue reports of MTOMCAT, RAT and LUCENE which are open source Apache JIRA softwares. The data is comments posted by various developers on the forum to ask, solve, up vote or negate an opinion, suggestion or query. Python's BeautifulSoup, urllib and openpyxl libraries were used to scan the whole HTML pages and extract the comments along with their authors, date, time and comment ID. BeautifulSoup is used to get the data associated with an HTML tag and that information can be directly loaded into a spreadsheet using functions of openpyxl library. We parsed the JIRA issue report repository in September 2017.

For the purpose of analysis, 500 comments were selected out of a total of 1117. Unlike Murgia, Tourani, Adams, & Ortu. 2014 full study, we choose the comments in groups of 10 to maintain the flow of context so that the evaluators can get accustomed to the ambience of the forum and identify the polarity of texts more precisely. All the code snippets and stack traces were removed and cleaned from the comments before beginning analysis.

For the further research purposes, we provide our supervised dataset[1] along with a Python script which was used to extract comments and their various parameters and store them simultaneously in an excel file. The dataset has a total nine attributes. Four attributes are comment specific i.e. ID, Date, Author and Comment. Rest five attributes include 4 labels which were identified by human evaluators using Murgia, Tourani, Adams, & Ortu. 2014 procedures, and their final mapping value that specifies whether the comment is being treated as positive, negative or neutral.

B. Emotion Mining

For rating the comments, we followed the 4 evaluator approach used by Jongeling, Sarkar, Datta, & Serebrenik, 2017. Four evaluators including ourselves rated the comments as having one of the following emotions *joy* or *love* indicating positive emotion and *sad*, *fear* or *anger* indicating negative emotion. *Surprise* emotion was eliminated as surprise can take both positive and negative forms [4].

The process of emotion mining was based on the 6 primary emotions of Parrott's framework [13]. Common understanding of this framework was very important for the evaluators and to ensure that raters have a unified perspective towards developers' comments [3]. To achieve this, a list of comments was made which highlighted the occurrence of all emotions with examples from Murgia, Tourani, Adams, & Ortu. 2014 study.

We calculate a comment's mapping value as an integral value which ranges from [-1, 1], -1 for negative, 0 for neutral and 1 for positive. The comment is identified as positive (+1) if 3 or more evaluators have labelled it with positive emotion and no evaluator has labelled it with negative emotion [4]. Likewise, if 3 or more evaluators are inclined to say that a comment has negative emotion and no other has indicated positive emotion, we consider that comment as negative(-1). At last, the comment is marked neutral if 3 or more evaluators have shown neither positive nor negative sentiment [4].

There were 2 comments for which 3 or more evaluators had given *surprise* emotion. These were removed from the final dataset. 112 comments were found to have been labelled contradictory [4], i.e. *joy* by one rater and *fear* by another and they were also removed from the dataset leaving 500-2-112 = 386 comments in the end.

C. Sentiment Analysis

Next we use standard sentiment analysis tools to find the sentiments in these comments. Interpretation of SentiStrength's analysis were adapted from Thelwall at al. 2012 approach. SentiStrength, calculates polarity by assigning a positive integer p between 1 and 5 and a negative integer n between -1 and -5. A text was rated as positive if $p + n > 0$, negative when $p + n < 0$ and neutral when $p + n = 0$ and $p < 4$ [8].

[1] https://www.dropbox.com/sh/l4x6njr4qfy2cos/AABQjl1C7cBnRpSfaFNPyOkFa?dl=0

For NLTK, API at text-processing.com[2] was called using the textual comments as parameters. The API call returned the probabilities of a sentence being negative, positive and neutral. If the probability of comment being neutral is more than 0.5, then it is marked neutral. Else, whichever other probability is greater, comment was marked having that sentiment [9]. To obtain the mapping values for NLTK, we use Pletea, Vasilescu, & Serebrenik 2014 interpretation [8].

We used Watson Natural Language Understanding[3] (WNLU) in this study, which is IBM's sentiment analysis tool. This tools is an API which returns a score that is between [-1, 1]. A negative score signifies negative sentiment, a positive score indicates positive sentiment and 0 for neutral.

Microsoft Azure's Text Analytics API[4] was also used for classification of the texts which works on a machine learning classification algorithm. There were a couple of problems with it. Text Analytics API returns a score (decimal value) between 1 and 0 for every polarity request. Since there is no standardization as to how to classify texts based on their decimal score, comments with a score less than 0.26 were identified as negative, those having score greater than 0.74 were considered positive and the remaining 50% bracket was left for comments showing no sentiments.

D. Evaluation Metrics

After mapping values of all the tools and manual labelling were derived, we calculate the values of metrics. Since our aim is to find whether the standard sentiment analysis tools agree with the human raters over the presence or absence of emotions in issue reports, we choose weighted kappa as our first metric as it counts disagreement as well [4,10]. Kappa is an agreement measure which also take agreement by chance into consideration. We choose Viera and Garrett 2005 interpretation of kappa which says that agreement is almost perfect for $0.81 <= k <= 1$, substantial if $0.61 <= k <= 0.80$, moderate if $0.41 <= k <=0.60$, fair if $0.21 <= k <=0.40$, slight if $0.01 <= k <= 0.20$ and less than chance if $k <= 0$ [7].

We also use Adjusted Rand Index (ARI), which evaluates whether 2 comments are tagged with same sentiments or not [5]. It measures the correlation between two partitions of the same data [4]. The value of ARI ranges from 0 to 1, where 0 indicates independent partitions and 1 denotes identical partitions [5].

IV. RESULTS

Results of our experiments are shown in Table 1 and Table 2. Clearly, the tools don't agree sufficiently with each other and with manually labelled ratings. When identifying emotions, out of 500 comments, only in 29.2% (146 out of 500) comments all the four raters had the same rating. From these 146 comments, 117 showed absence of emotion and were marked nil by all the four raters. From the rest 29 (146-117), 17 were of joy, 7 of sadness, 2 of surprise and 1 of love, anger and fear. When identifying sentiments, the two comments marked '*surprise*' by all the four evaluators were removed as surprise can be treated as both negative and positive.

TABLE I AGREEMENT OF TOOLS WITH MANUAL LABELLING AND WITH EACH OTHER.

Tools	Kappa	Adjusted Rand Index
NLTK vs Manual	0.3	0.06
SentiStrength vs Manual	0.39	0.139
WNLU vs Manual	0.43	0.177
TextAnalytics vs Manual	0.4	0.112
NLTK vs SentiStregth	0.3	0.069
NLTK vs WNLU	0.49	0.221
NLTK vs TextAnalytics	0.38	0.119
SentiStrength vs WNLU	0.49	0.202
SentiStrength vs TextAnalysitcs	0.31	0.07
WNLU vs TextAnalysitcs	0.36	0.114

TABLE II AGREEMENT OF GROUPS OF TOOLS WITH MANUAL LABELLING.

Tools	n	Kappa	Adjusted Rand Index
NLTK, SentiStrength	188	0.6	0.279
NLTK, WNLU	242	0.51	0.24
NLTK, TextAnalytics	208	0.51	0.184
SentiStrength, WNLU	238	0.55	0.283
SentiStrength, TextAnalytics	182	0.64	0.377
WNLU, TextAnalytics	195	0.65	0.381
NLTK, SentiStrength, WNLU	159	0.66	0.405
NLTK, SentiStrength, TextAnalytics	121	0.77	0.554
NLTK, WNLU, TextAnalytics	147	0.73	0.487
SentiStrength, WNLU, TextAnalytics	139	0.73	0.519
NLTK, SS, WNLU, TextAnalytics	109	0.81	0.661

Results clearly indicate that the sentiment analysis tools do not agree with the manual labelling as no tool could achieve substantial or better agreement value of Kappa (0.6 or more). Only moderate and fair agreement was obtained between tools and manual labelling. WatsonNLU scores best with Kappa=0.43 and ARI=0.177, followed by Text Analytics and SentiStrength.

Table 2 shows values for both the metrics after grouping 2 and more tools together and finding their intersection (n = number of comments that tools agreed upon). Interestingly, when comments with same polarity from all the tools were taken into consideration, only 109 comments were gathered which delivered a kappa value of 0.81. This value shows almost perfect agreement.

[2] NLTK API: http://text-processing.com/docs/sentiment.html
[3] WNLU API : http://text-processing.com/docs/sentiment.html
[4] Text Analytics API : https://www.ibm.com/watson/services/natural-language-understanding/

V. CONCLUSION

Kappa values obtained between different tools is slightly better compared to manual labelling but still not substantially enough to say that the tools agree with each other. Kappa values are identical when WNLU is paired with either NLTK or SentiStrength. Taking into account the ARI value we can say that NLTK and WNLU agree with each other the most.

Hence it is safe to say that out of four tools, WNLU has not only showed best results with manually labeled data but has also boosted the value of both metrics when combined with other tools. TextAnalytics is almost equally good when it comes to agreement with humans, but kappa value falls by 0.04 during the agreement of these two tools.

Most of the comments were marked as 'nil' by human evaluators which indicates that either software developers rarely show emotions in issue reports or even human evaluators are not substantially sufficient to find sentiments in software developers' comments. The latter points toward the fact that if humans evaluators are incapable on finding emotions and cannot agree to greater extent on the presence or absence of emotions, machine learning models will have no basis for training, hence an effective machine learning model may be quite difficult to create solely for the software engineering domain. However, a regular use emoticons by software developers might help future researchers to mine emotions more accurately and precisely and make appropriate machine learning models that will in turn automate the process of emotion mining with a significant accuracy.

REFERENCES

[1] B. Pang, & L. Lee (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1-135.

[2] B. Liu (2010). Sentiment Analysis and Subjectivity. *Handbook of natural language processing*, *2*, 627-666.

[3] A. Murgia, P. Tourani, B. Adams, & M Ortu. (2014, May). Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *Proceedings of the 11th working conference on mining software repositories* (pp. 262-271). ACM.

[4] R. Jongeling, P. Sarkar, S. Datta, & A. Serebrenik, (2017). On negative results when using sentiment analysis tools for software engineering research. *Empirical Software Engineering*, 1-42.

[5] L. Hubert & P. Arabie (1985). Comparing partitions. *Journal of classification*, *2*(1), 193-218.

[6] R. Bakeman, & J. M. Gottman, (1997). *Observing interaction: An introduction to sequential analysis*. Cambridge university press.

[7] A. J. Viera, & J. M. Garrett, (2005). Understanding interobserver agreement: the kappa statistic. *Fam Med*, *37*(5), 360-363.

[8] M. Thelwall, K. Buckley, & G. Paltoglou (2012). Sentiment strength detection for the social web. *Journal of the Association for Information Science and Technology*, *63*(1), 163-173.

[9] D. Pletea, B. Vasilescu, & A. Serebrenik (2014, May). Security and emotion: sentiment analysis of security discussions on GitHub. In *Proceedings of the 11th working conference on mining software repositories* (pp. 348-351). ACM.

[10] J. Cohen (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, *70*(4), 213.

[11] W. G. Parrott (Ed.). (2001). *Emotions in social psychology: Essential readings*. Psychology Press.

[12] Porter, M. E., & V. E. Millar (1985). How information gives you competitive advantage.

[13] W. G. Parrott, (Ed.). (2001). *Emotions in social psychology: Essential readings*. Psychology Press.