**MINOR PROJECT PROGRESS REPORT**

**University School of Information and Communication Technology**

**GGS Indraprastha University, Delhi-78**

Submitted in partial fulfilment of the requirements for the award

of the degree of Bachelor of Technology in

**Information Technology**

**Under the guidance of**

Prof. Arvinder Kaur

&

Mr. Amrit Pal Singh

**Submitted by**

Divesh Bisht

(04616401514)

7th Semester

**1. ABSTRACT**

Sentiment analysis refers to the use of [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), [text analysis](https://en.wikipedia.org/wiki/Text_analytics), [computational linguistics](https://en.wikipedia.org/wiki/Computational_linguistics), and [biometrics](https://en.wikipedia.org/wiki/Biometrics) to systematically identify, extract, quantify, and study affective states and subjective information. It is one of the fastest growing research areas in computer science, making it challenging to keep track of all the activities in the area. Recent years have witnessed an increasing attention to social aspects of software engineering. This includes studies of emotions and sentiments which are experienced and expressed by the software developers.

We present a computer-assisted literature review where we utilize both text mining and qualitative coding to answer whether developers feel emotions. The study is carried out on dataset which consists of comments posted by software developers on a software version maintaince website like Apache Jira.

We study whether the sentiment analysis tools agree with the sentiment recognized by human evaluators. We also study to what extent the results obtained from different sentiment analysis tools agree with each other. Furthermore, we evaluate the impact of the choice of a sentiment analysis on software engineering studies for positive, negative and neutral texts.

**2. INTRODUCTION**

The study consists of two prime processes that is Emotion Mining and Sentiment Analysis. Emotion mining tries to identify the presence of human emotions like joy, love, sadness, fear etc from text, voice and video artifacts produced by humans. As such, it is different from sentiment analysis, which instead evaluates a given emotion as being positive or negative. Absence of emotion is marked as neutral sentiment.

Recently, software development field has become increasingly social. Therefore, it becomes necessary to know that do software developers feel emotions i.e. whether emotions can be extracted from comments posted by developers on software development ecosystems.

Our study was divided into 5 stages which are described below :

* Finding and understanding previously published research papers that deal with the issue of emotion mining and sentiment analysis.
* Collecting data through python scripts and cleaning the data obtained so that it can be processed easily.
* Manually mining emotions from the collected data and grouping various emotions into positive, negative or neutral sentiments.
* Sentiment analysis of the dataset carried out using Natural Language Processing tools such as NLTK, Sentistrength and WatsonNLU( Natural Language Understanding ).
* Calculating various evaluation metrics such as weighted kappa and adjusted rand index for agreement between manual labelling and sentiment analysis tools.

**3. RELATED WORK**

Murgia et al. analysed the emotions present in software developer comments by labelling them manually through evaluators. There was no use of tools involved.

Jongeling et al. further performed sentiment analysis on the dataset made available by Murgia’s study and calculated various agreement statistics on the results obtained.

We try to address the following research questions through our study:

*RQ 1 : To what extent do different sentiment analysis tools agree with emotions of software developers?*

*RQ 2 : To what extent do results from different sentiment analysis tools agree with each other?*

*RQ 3 : Do different sentiment analysis tools lead to contradictory results in a software engineering study?*

**4. RESEARCH METHODOLOGY**

This section discusses the dataset used in our analysis, the general procedure used to rate extracted comments and description of various sentiment analysis tools used.

**4.1 DATASET**

Having hands on experience with Python, we automated the whole process. Python scripts were written using *BeautifulSoup* and *Urllib* libraries in order to extract the software developers’ comments along with username, time of comment and issue ID from **Apache Jira’s** open source projects. For the sake of simplicity, we took only three open source projects in consideration i.e. MTOMCAT, RAT and LUCENE. We extracted 1117 comments.

A total of 500 comments were included in the final dataset and all the code snippets in the comments were removed. These comments were taken in the groups of ten or more so as to maintain the context flow and help human raters familiarize themselves with the ongoing conversations so as to get an abstruce information about the emotions present.

**4.2 EMOTION MINING**

For the whole emotion mining process, the complete dataset of 500 comments was distributed among 4 raters. Each evaluator rated the particular issue comment as having a particular emotion or nil in case no emotion was found.

We used Parrott’s six primary emotions i.e. love, joy, surprise, anger, sadness and fear. The raters were given a guide containing 2-3 examples of each emotion to help them understand the labeling process.

Once all the labeling was done, final mapping values were obtained using the following guidelines:

* We consider the comment as positive if at least three evaluators have indicated a positive sentiment and no evaluator has indicated negative sentiments.
* We consider the comment as negative if at least three evaluatorshave indicated a negative sentiment and no evaluator has indicated positive sentiments.
* We consider the comment as neutral when three or more evaluators have neither indicated a positive sentiment nor a negative sentiment.
* Comment is considered contradictory for all remaining cases.

There were 2 comments for which 3 or more evaluators had given ‘surprise’ as emotion. These were removed from the dataset since surprise can be treated as both positive and negative.

We found that 112 comments have been labeled contradictory and they were also removed from the dataset leaving 386 comments in the dataset.

**4.3 SENTIMENT ANALYSIS TOOLS**

We exclude tools that require training before they can be applied. We chose the following three tools to perform sentiment polarity analysis:

* **NLTK** - It uses a simple bag of words model and returns for each text three probabilities: a probability of the text being negative, one of it being neutral and one of it being positive. If neutral is greater than 0.5 then the label will be neutral. Otherwise, the label will be positive or negative, whichever has the greater probability. We use the API provided at text-processing.com to use NLTK.
* **SENTISTRENGTH** – It assigns an integer value between 1 and 5 for the positivity of a text and a value between −1 and−5 for the negativity, n. A text is considered positive when (p+n)>0 , negative when (p+n)<0 and neutral if p+n=0 & p>=4.
* **WATSON NLU –** The API returns for a text fragment a score which is in the range [−1,1]. For negative scores, the type is negative, positive for positive scores and neutral for 0. The status reflects the analysis success and it is either “OK” or “ERROR”.

**4.4 PERFORMANCE MEASURES**

Out of 386 comments, 11 showed ERROR status with the WatsonNLU and so they were removed from the dataset. Confusion matrices were calculated on remaining 375 comments.

Since more than 58% comments were neutral, so traditional metrics such as accuracy might be misleading as the dataset is unbalanced. Thus, we go for weighted kappa and ARI.

**Weighted kappa** (κ) as recommended by Bakeman and Gottman is a measure of interrater agreement. The agreement due to chance is factored out using kappa.

Since the sentiments can be seen as ordered, from positive through neutral to negative, and disagreement between positive and negative is more severe than between positive and neutral or negative and neutral, so our

weighting scheme is as shown in Table.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Neutral | Positive | Negative |
| Neutral | 0 | 1 | 1 |
| Positive | 1 | 0 | 2 |
| Negative | 1 | 2 | 0 |

|  |  |
| --- | --- |
| **Kappa Value** | **Interpretation of Agreement between Raters** |
| <0 | Poor |
| 0 - 0.20 | Slight |
| 0.21 - 0.40 | Fair |
| 0.41 - 0.60 | Moderate |
| 0.61 - 0.80 | Substantial |
| 0.81 - 1.0 | Almost perfect |

**Adjusted Rand Index** (ARI) by Hubert and Arabie corrects for the possibility that pairs of observations have been put in the same category by chance. The expected value of ARI for independent partitions is 0 and 1 for identical partitions, the closer the value of ARI to 1 the better the correspondence between the partitions.

**5. RESULTS**

Agreement of sentiment analysis tools with manual labelling and with each other

|  |  |  |
| --- | --- | --- |
| **TOOLS** | **KAPPA** | **ARI** |
| NLTK vs Manual | 0.3 | 0.06 |
| SentiStrength vs Manual | 0.39 | 0.139 |
| WNLU vs Manual | 0.43 | 0.177 |
| NLTK vs SentiStrength | 0.3 | 0.069 |
| NLTK vs WNLU | 0.49 | 0.221 |
| SentiStrength vs WNLU | 0.49 | 0.202 |

Agreement of groups of sentiment analysis tools with manual labelling

|  |  |  |
| --- | --- | --- |
| **TOOLS** | **KAPPA** | **ARI** |
| NLTK, SentiStrength | 0.6 | 0.297 |
| NLTK, WNLU | 0.51 | 0.24 |
| SentiStrength,WNLU | 0.55 | 0.283 |
| NLTK, SentiStrength, WNLU | 0.66 | 0.405 |

**RQ1:** Only moderate and fair agreement can be obtained between tools and manual labelling. WatsonNLU scores best with kappa=0.43 and ARI=0.177

**RQ2:** Kappa values obtained between different tools is slightly better compared to manual labelling. Also, agreement in tools increases when two or more are combined for analysis.

**RQ3:** Different sentiment analysis tools do not lead to contradictory results. There agreement with manual labelling, although less, is quite close to each other.

**6. REFERENCES**

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.

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.

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

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.

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.

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

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

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