An Empirical Investigation of Sentiment Analysis of the Bug Tracking Process in Libre Office Open Source Software

Apostolos Kritikos(✉), Theodoros Venetis, and Ioannis Stamelos

School of Informatics, Aristotle University of Thessaloniki, University Campus, 54124 Thessaloniki, Greece
{akritiko,venetheo,stamelos}@csd.auth.gr
http://www.csd.auth.gr/en/

Abstract. In this work we are studying the sentiment in Open Source Software projects and more specifically in the process of bug reporting, to investigate the human factor, namely, the feedback from the community (end-users, developers, testers, etc.). One of the characteristics for which Open Source Software has gained attention, over the years, is the fact that it is continuously being tested and maintained by its community of volunteers. Sentiment analysis, a rapidly growing field, can enrich software evaluation with a social aspect. Results suggest that FLOSS projects' bug reports can potentially constitute a rich emotionally - imbued information source.

1 Introduction

Since the beginning of Open Source Software, the community around a FLOSS project is one of the greatest strengths of open source software development over closed source development. Eric S. Raymond [1] highlights this strength; *"Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone"*. In the community of an Open Source Software project we find developers who usually contribute both in the development and test process. In addition, community members can also be people who do not possess development skills but use the project as end users and contribute to testing. Cerone et al. [2] stress the importance of defining qualitative metrics for indicators of collaboration effectiveness within Open Source Software communities.

In this paper we will be analyzing the sentiment from bug reports posted to Bugzilla. The proposed methodology is being applied to various major versions of the Open Source Software project Libre Office.

The rest of the paper is organized as follows: Sect. 2 provides background information for the basic terms discussed in this work. Section 3 presents scientific work done by other researchers relevant with the aim of this paper. Section 4 analyzes our case study plan. Section 5 provides the statistical analysis conducted to the data we collected. In Sect. 6 we discuss the findings on our statistical analysis, while Sect. 7 speculates on threats to

© IFIP International Federation for Information Processing 2020
Published by Springer Nature Switzerland AG 2020
V. Ivanov et al. (Eds.): OSS 2020, IFIP AICT 582, pp. 36–46, 2020.
https://doi.org/10.1007/978-3-030-47240-5_4

validity. Finally, in Sect. 8 we conclude our work by summarizing our findings and we refer to possible future work. Please note that all the original data for this scientific work are available at http://users.auth.gr/akritiko/datasets/kritikosOSS2020_dataset.zip.

2 Background

This section presents an overview of the research state of the art on sentiment analysis (also known as opinion mining).

Sentiment analysis (often referred to as *opinion mining*) is the process of determining the polarity of an input text. It can be performed mainly at three different levels of granularity [3]: *document-level analysis*, classifying the sentiment expressed in a whole document as positive, negative or neutral; *sentence-level analysis*, which determines the sentiment per sentence in a document; *aspect-level analysis*, which delivers more fine-grained results and is based on the idea that an opinion consists of a sentiment and a respective target, which typically refers to an entity or a specific aspect of it.

The approaches mainly used for deducing the sentiment expressed in a document is the lexicon-based [4] and the machine learning-based [5] approach. Besides these two, there is a third one that involves the deployment of ontology-based methodologies for performing aspect-level sentiment analysis [6].

3 Related Work

In the software engineering domain, there have been the following relevant approaches for studying emotions and related factors in Open Source Software projects. Guzman et al. [7] study emotions expressed in commit comments of several open source projects and analyze their relationship with different factors. Rousinopoulos et al. [8] attempt to analyse the evolution of the sentiment of developers of a Open Source Software project (openSUSE Factory) by applying sentiment analysis techniques on the e-mails sent to the mailing list of the project. Similarly, Garcia et al. [9] perform sentiment analysis on the e-mail archives of another FLOSS project (Gentoo) and attempt to investigate the relation between the emotions in the mailing list and the respective activity of contributors in the project. Finally, the authors of [10] and [11] analyse comments in an issue tracking system, in order to investigate whether a relationship exists between human affectiveness (emotion, sentiment and politeness) and developer productivity (i.e. time needed for fixing an issue).

4 Case Study Plan

We have chosen Libre Office as a candidate to conduct our analysis for a variety of reasons: (1) It is an Open Source Software project with a rich code base (until the time of writing it has had 486,472 commits made by 1,880 contributors representing 9,517,407 lines of code [12]), (2) it has an active community that shows no sign of long-term decline and has attracted the long-term and most active committers in OpenOffice.org (from which Libre Office was originated as a fork).

In this work we will be studying sixteen (16) major versions of the Libre Office productivity suite, namely: 3.3, 3.4, 3.5, 3.6, 4.0, 4.1, 4.2, 4.3, 4.4, 5.0, 5.1, 5.2, 5.3, 5.4, 6.0, 6.1.

4.1 Data Collection – Bugs

We collected our data from the official Libre Office bug tracking tool which is implemented with Bugzilla [13]. We gathered a total of 740 bugs that are related to the versions. Each bug is a threaded discussion with comments related to the bug reported. These 740 bugs are consisted of 6960 comments in total. Apart from the text of the comments to which we have applied the sentiment analysis process we also gather the following information for each bug [14]:

- ASSIGNED_TO: The person to which the bug was assigned to be fixed.
- CREATED: The date of the creation of the bug.
- CREATOR: The creator of the bug.
- LAST_CHANGED: When was the last time that this bug was updated.
- OPEN: If the bug is open or closed.
- COMPONENT_ID: Which component does the bug refer to (i.e. Writer).
- VERSION_ID: Which version does the bug refer to (i.e. 3.3).

As far as the comments of a bug are concerned (threaded discussion) we have gathered the following information:

- CREATED: Date and time that the comment was created.
- CREATOR: The creator of the comment (can be different than the creator of the bug)
- COMMENT_TEXT: The text of the comment.
- COUNT_ORDER: The order of the comment in the discussion (needed to be able to follow the flow of the threaded discussion).
- BUG_CREATOR: Used to identify when a comment is an answer to the thread by the creator of the bug.
- BY_SYSTEM: Used to identify system automated responses (which show zero sentiment by default).

4.2 Data Collection – Sentiment Analysis

As far as the sentiment analysis part is concerned, we used two (2) different tools, namely VADER [15] and MeaningCloud [16].

We consider both tools valuable for our investigation. VADER (Valence Aware Dictionary and sEntiment Reasoner) is an academic product [17] that serves "as a lexicon and rule-based sentiment analysis tool". MeaningCloud, on the other hand is a commercial solution for sentiment analysis that provides a free plan for 20.000 requests per month. Since we are interested in investigating if and how sentiment affects the bug tracking process in open source software, by using both tools we, without loss of generality, get safer results concerning the existence of sentiment in the bug tracking process of open source software whereas, at the same time, we are utilizing the different kinds of information that each tool provides with the aim of succeeding a wider scope in our investigation.

VADER sentiment analyzer provided us with the following data per comment:

- COMPOUND_SCORE: As the authors of the tool describe it, a "normalized, weighted composite score. This score is the sum of the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between −1 (most extreme negative) and +1 (most extreme positive).".
- NEGATIVE, NEUTRAL, POSITIVE: These scores are ratios that show the percentage of the comment that is positive, negative and/or neutral.

MeaningCloud analyzer provided us with the following data per comment:

- SCORE: The sentiment score that can be one of the following values: none (no sentiment), N+ (very negative), N (negative), neutral, P (positive), P+ (very positive).
- CONFIDENCE: Percentage to which the analysis is accurate [0–100].
- IRONY: If the comment is ironic (IRONIC, NONIRONIC).
- SUBJECTIVITY: If the opinion on the comment is considered subjective or objective (SUBJECTIVE, OBJECTIVE).
- AGREEMENT: If the sentences of the comment are in agreement with each other or there are ambiguous meanings (AGREEMENT, DISAGREEMENT).

5 Statistical Analysis

This section presents the statistical analysis of the data collected. As a first step we tried to identify possible outliers that might be misleading for our results to consider possible data reduction [18]. In the following image you can see a visualization of the number of comments distribution per version of Libre Office and the box plots regarding our variables (Fig. 1).

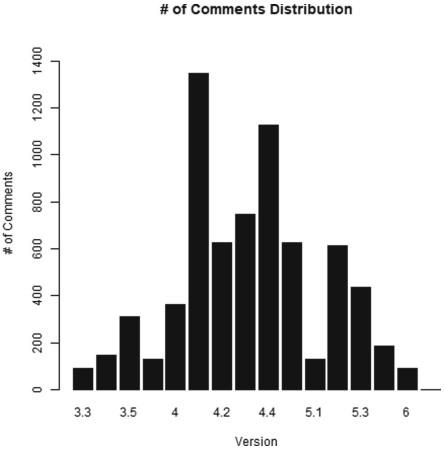

Fig. 1. NOF comments distribution

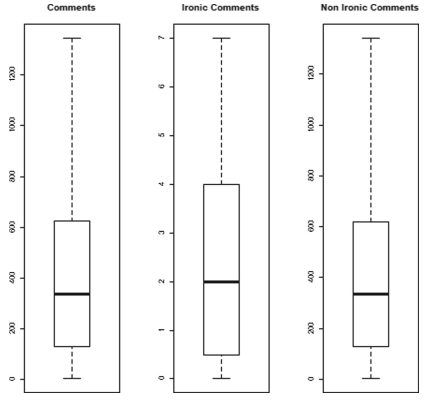

Fig. 2. Box plots for comments, ironic, non ironic

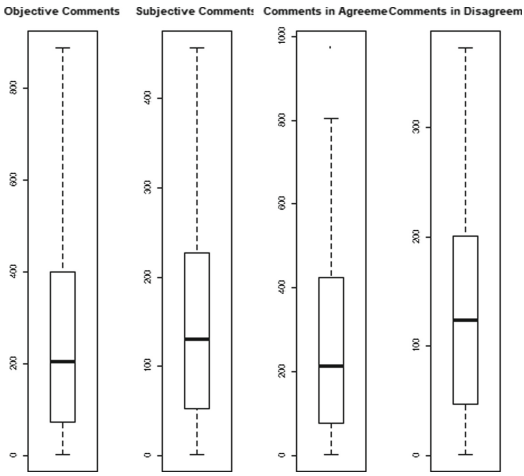

Fig. 3. Box plots for objective, subjective, agreement and disagreement comments

From Fig. 2 and Fig. 3 we can see that our means are being affected by low values of our population. This also becomes obvious by the descriptive statistics provided to Table 1 following.

Table 1. Discriptive analysis for variables.

Variable	n	mean	sd	median	min	max	skew	kyrtosis
Comments	16	435.00	390.32	336.50	2.0	1345.0	0.91	−0.26
Ironic	16	2.44	2.06	2.00	0.0	7.0	0.47	−0.78
NonIronic	16	432.56	389.11	334.50	2.0	1341.0	0.92	−0.24
Subjective	16	279.56	259.58	205.50	1.0	889.0	0.95	−0.20
Objective	16	155.44	131.34	131.00	1.0	456.0	0.83	−0.40
Agreement	16	299.94	282.60	212.50	1.0	973.0	0.99	−0.10
Disagreement	16	135.06	108.60	124.00	1.0	372.0	0.69	−0.67
Vader (Positive)	16	269.88	248.98	200.00	2.0	869.0	1.03	0.00
Vader (Neutral)	16	101.75	92.38	81.00	0.0	311.0	0.84	−0.45
Vader (Negative)	16	63.38	50.76	55.50	0.0	165.0	0.46	−1.13
MeaningCloud (Positive)	16	241.38	229.88	177.50	1.0	808.0	1.10	0.20
MeaningCloud (Neutral)	16	40.94	31.86	35.50	0.0	107.0	0.55	−0.87
MeaningCloud (Negative)	16	60.75	47.12	54.50	0.0	156.0	0.56	−0.93
MeaningCloud (No Sent.)	16	91.94	84.51	67.00	1.0	274.0	0.72	−0.80

Since this work studies the bugs reported to Libre Office per version, we decided not to proceed to data reduction since the versions that seem to act like outliers are provide valuable information. For example, the increased activity of comments in some of the 4.x versions is since 3.6 version was the version that followed the fork from Open Office after which redesign and refactor processes took place. We also chose to let versions with low concentration of comments, i.e. 6.1, to further investigate if this behavior is due to successful bug fixing or it was just a version with low bug tracking activity. In Table 1 you can see the descriptive analysis of the variables we gathered for the sentiment analysis using the tools described in Sect. 4 (Figs. 4 and 5).

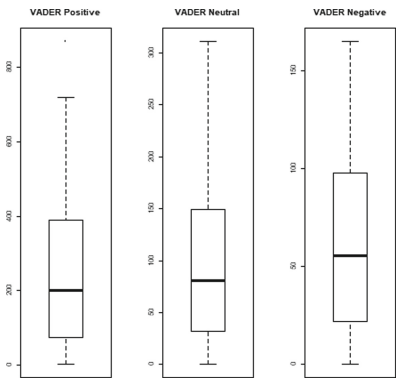

Fig. 4. Box plots for VADER sentiment analyzer

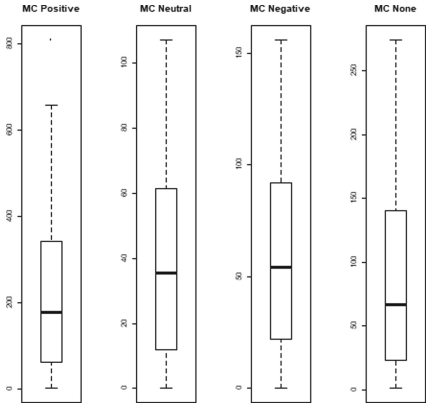

Fig. 5. Box plots for MeaningCloud sentiment analyzer

Finally, we conducted a descriptive analysis separately for the confidence variable per version of the Libre Office project.

Having concluded with the descriptive analysis of the chosen variables we then proceeded in studying our selected variables regarding their normality. Since the descriptive analysis is showing (Table 1) skewness with values around or bigger than 1 we have the indication that some of our variables might not follow the normal distribution. Therefore, we proceeded in applying the Shapiro-Wilk normality test in order to identify which variables follow the normal distribution and which do not from the results we are seeing that there are variables with p-value > 0.05 therefore parametric statistical analysis cannot be carried for them in this study since their data might not follow the normal distribution. Therefore, we will be conducting our statistical tests with the use of nonparametric tests and more specifically using the Spearman's Rank Correlation Coefficient (Table 3).

In Table 4 we have applied the Spearman's Rank Correlation to investigate possible statistical similarities in the distributions of VADER (Positive, Neutral, Negative) and Meaning Cloud (Positive, Neutral, Negative). Our results indicate that all three correlations are statistically significant (p-value < 0.05) and very strong correlated ($\rho > 0{,}9$).

Table 2. Descriptive analysis for confidence.

Version	n	mean	sd	median	min	max	skew	kyrtosis
v3.3	89	94.83	6.32	100	76	100	−0.75	−0.83
v3.4	145	95.03	6.26	100	76	100	−0.85	−0.52
v3.5	309	95.83	5.24	100	84	100	−0.87	−0.64
v3.6	129	96.05	5.26	100	86	100	−0.94	−0.62
v4.0	364	95.83	5.71	100	74	100	−1.06	−0.06
v4.1	1345	96.98	5.03	100	76	100	−1.41	0.53
v4.2	624	96.78	4.99	100	74	100	−1.48	1.47
v4.3	746	96.79	5.11	100	76	100	−1.42	0.96
v4.4	1126	97.05	4.83	100	83	100	−1.41	0.61
v5.0	624	96.73	4.84	100	73	100	−1.40	1.38
v5.1	128	96.02	4.94	100	76	100	−1.16	1.12
v5.2	615	96.66	5.05	100	73	100	−1.43	1.37
v5.3	435	96.65	5.23	100	84	100	−1.23	−0.06
v5.4	186	97.72	3.86	100	86	100	−1.57	1.63
v6.0	93	96.82	5.09	100	76	100	−1.67	2.52
v6.1	2	93.00	9.90	93	86	100	0.00	−2.75

Table 3. Shapiro-Wilk normality tests.

Variable	W	p-value
Comments	0.87727	0.0352
Ironic	0.90422	0.09393
NonIronic	0.87637	0.03408
Subjective	0.89079	0.05734
Obective	0.87399	0.03131
Agreement	0.86491	0.02275
Disagreement	0.90727	0.1051
VADER (Positive)	0.85854	0.01824
VADER (Neutral)	0.88603	0.04822
VADER (Negative)	0.92277	0.1869
MeaningCloud (Positive)	0.85216	0.01466
MeaningCloud (Neutral)	0.93041	0.2238
MeaningCloud (Negative)	0.92766	0.2238
MeaningCloud (No Sent.)	0.89165	0.05915

Table 4. Spearman's rank correlation for VADER versus MeaningCloud

	Vader POS	Vader NEU	Vader NEG
MC POS	p-value = 2.2e−16 $\rho = 0.9617647$		
MC NEU		p-value = 2.2e−16 $\rho = 0.9588235$	
MC NEG			p-value = 1.252e−10 $\rho = 0.9757177$

6 Discussion

In this work we are studying the sentiment in Open Source Software projects and more specifically in the process of bug reporting in the Libre Office project via its bug tracking platform, Bugzilla. We conducted our experiment in 16 major versions of the software and gathered the respective data (as analyzed in Sect. 4) for 740 bugs consisted of threaded discussions with 6960 comments in total. We proceeded in sentiment analysis of these comments using two different tools (Vader & Meaning Cloud) and statistically analyzed the results.

Our first goal in this investigation was to get a result on whether the sentiment analysis of comments in bug report is meaningful. Given that bug reporting is a technical process, usually made by technical oriented people (developers, and so forth) one might argue that the text of the reports could be standardized with technical language and therefore, possible lack of sentiment. In Table 2, showing the descriptive analysis of the confidence metric of the sentiment analysis across the versions of Libre Office, we can see that the mean is $> 90\%$ for all the versions. This gives us a first indicator that the confidence of the sentiment analysis (which is done with the MeaningCloud tool) is strong across all our data. In Table 4 we performed a nonparametric test to get results regarding the similarities for the positive, neutral and negative groups of comments between the two tools we have used. We see that the tests show significant similarities in the distributions of Positive, Neutral and Negative groups of comments for both Vader and Meaning Cloud.

From Table 1 we see that the number of ironic comments (variable Irony) are significantly fewer (min: 0, max: 7, mean: 2.44) than the non-ironic comments (min: 2, max: 1341, mean: 432,56). This seems to be intuitively logical if we consider that Bugzilla is a tool to which a community of Libre Office users (many of them voluntarily) report bugs. Having a closer look to random comments tagged as ironic from MeaningCloud we observed that such comments usually contain capitalized words and although the message does not seem to be ironic, however the tool scores them as such (i.e. *[…] however, it's happened on EVERY release since it started happening 2 years ago (I always move to the n+ 1.0.0 release when it comes out)*).

Looking at the subjectivity related variables we are observing similarly interesting comments as with the irony variable. Starting a bug report with a reference to ourselves is common in bug reporting (i.e. *I have experienced this think while I was doing that*).

However, in some cases if we strip this first-person reference in the beginning of the sentence, we are left with the bug itself (i.e. in the previous example: this is happening when someone does that). In the comments collected there are similar "misunderstandings" by the MeaningCloud which could be considered objective comments otherwise (i.e. *In order to limit the confusion between ProposedEasyHack and EasyHack and to make queries much easier* ***we are changing*** *ProposedEasyHack to NeedsDevEval. Thank you and apologies for the noise*).

7 Threats to Validity

We are conducting our experiment to Libre Office. This may be a threat to validity since including all of the components of the project would results in a much bigger dataset with a bigger variety of functionality and thus, variety in bugs.

A significant difficulty we came across during our work involves the domain-specific vocabulary used in the input text (i.e. bug reports) we analysed. This often may result to inaccurate ratings by the sentiment analyser, since most popular sentiment analysis tools have been trained on more generic texts (i.e. Social Media). However, this problem is not new; it constitutes a commonly recognized threat to the validity of sentiment analysis results, not only in the domain we are working on, but in any other specialized field.

Another difficulty lies in the fact that, besides specialized terminology, content like bug reports may often contain extensive use of "every-day" (i.e. Jargon) expressions, abbreviations and even emoticons (sequences of symbols representing an emotion). These expressions are typically very dynamic, changing constantly and are being frequently replaced by other expressions, following each time the popular trends. Thus, even training the sentiment analyser accordingly would unfortunately generate dubious results.

Moreover, attempting to perform sentiment analysis in informal text generated by numerous human users may often involve subjectivity and sarcasm, both of which lead to inaccurate sentiment measurements. A solution to the former problem may involve integrating a subjectivity classifier [19, 20]. The latter problem, on the other hand, is even more challenging and typically appears in online discussions (like e.g. bug reports), although it is not that frequent in reviews of products and services. There have been some initial investigations (e.g. [21] and [22]), although the topic has not been extensively studied yet.

Finally due to space limitation we decided to make this first attempt to run our experiment per version of the Libre Office. This means that our original dataset was "compressed" in groups and, inevitably some of the fields that would otherwise be part of the experiment were omitted. It is however our intention to extend the version in this direction as well, as mentioned in the future work section.

8 Conclusion and Future Work

In this work we investigated the semantic analysis of bugs reported to the Libre Office Open Source Software, a tool written in C++ programming language.

As future work, we plan to replicate our case study on projects written in various programming languages for both desktop and web development (i.e. java, php, and so forth). We also intent to include metrics related with both the structural properties of the source code of the Open Source project alongside with its quality and resilience characteristics.

Additionally, we would like to attempt to enrich the human based dataset (for the sentiment analysis part) by using, additionally to the text originated from the bug reports (which mainly tend to be technical reports), the user ratings (both score and the text of the review) that various FLOSS projects provide. This way we believe we will be capturing the sentiment of both the developers and the end users of the software.

Finally, it is our intention to study other sentiment analyzers found in literature and possibly compare and contrast their application to our dataset.

References

1. Raymond, E.S.: The Cathedral and the Bazaar, 1st edn. Tim O'Reilly (Ed.). O'Reilly & Associates, Inc., Sebastopol (1999)
2. Cerone, A., Fong, S., Shaikh, S.A.: Analysis of collaboration effectiveness and individuals' contribution in FLOSS communities (2012)
3. Liu, B.: Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. **5**(1), 1–167 (2012)
4. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Comput. Linguist. **37**(2), 267–307 (2011)
5. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retriev. **2**(1–2), 1–135 (2008)
6. Kontopoulos, E., Berberidis, C., Dergiades, T., Bassiliades, N.: Ontology-based sentiment analysis of Twitter posts. Expert Syst. Appl. **40**(10), 4065–4074 (2013)
7. Guzman, E., Azócar, D., Li, Y.: Sentiment analysis of commit comments in GitHub: an empirical study. In: Proceedings of the 11th Working Conference on Mining Software Repositories, pp. 352–355. ACM, May 2014
8. Rousinopoulos, A.I., Robles, G., González-Barahona, J.M.: Sentiment analysis of free/open source developers: preliminary findings from a case study. Rev. Eletrôn. Sist. Inf. **13**(2) (2014). https://doi.org/10.5329/resi. ISSN 1677-3071
9. Garcia, D., Zanetti, M.S., Schweitzer, F.: The role of emotions in contributors activity: a case study on the GENTOO community. In: 2013 Third International Conference on Cloud and Green Computing (CGC), pp. 410–417. IEEE, September 2013
10. Ortu, M., Adams, B., Destefanis, G., Tourani, P., Marchesi, M., Tonelli, R.: Are bullies more productive? Empirical study of affectiveness vs. issue fixing time (2015)
11. Murgia, A., Tourani, P., Adams, B., Ortu, M.: 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, May 2014
12. Libre Office Project on Open Hub. https://www.openhub.net/p/libreoffice
13. Libre Office Bugzilla. https://bugs.documentfoundation.org/
14. Libre Office Bugzilla Fields Documentation. https://wiki.documentfoundation.org/QA/Bugzilla/Fields
15. VADER, Github Repository. https://github.com/cjhutto/vaderSentiment
16. MeaningCloud Tool. https://www.meaningcloud.com/

17. Hutto, C.J., Gilbert, E.E.: VADER: a parsimonious rule-based model for sentiment analysis of social media text. In: Eighth International Conference on Weblogs and Social Media (ICWSM 2014), Ann Arbor, MI, June 2014
18. Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslen, A.: Experimentation in Software Engineering, 1st edn. Kluwer Academic Publishers, Boston/Dordrecht/London (2000)
19. Wiebe, J., Riloff, E.: Creating subjective and objective sentence classifiers from unannotated texts. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 486–497. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30586-6_53
20. Barbosa, L., Feng, J.: Robust sentiment detection on Twitter from biased and noisy data. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters (COLING 2010) (2010)
21. González-Ibánez, R., Muresan, S., Wacholder, N.: Identifying sarcasm in Twitter: a closer look. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-vol. 2, pp. 581–586. Association for Computational Linguistics, June 2011
22. Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., Huang, R.: Sarcasm as contrast between a positive sentiment and negative situation. In: EMNLP, pp. 704–714 (2013)