KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express xxx (xxxx) xxx

www.elsevier.com/locate/icte

Emotion-based analysis of programming languages on Stack Overflow

Stefano Cagnoni, Lorenzo Cozzini, Gianfranco Lombardo, Monica Mordonini, Agostino Poggi, Michele Tomaiuolo*

Department of Engineering and Architecture, University of Parma, Parma, Italy

Received 29 February 2020; received in revised form 29 May 2020; accepted 2 July 2020
Available online xxxx

Abstract

When developing a software engineering project, selecting the most appropriate programming language is a crucial step. Most often, feeling at ease with the possible options becomes almost as relevant as the technical features of the language. Therefore, it appears to be worth analyzing the role that the emotional component plays in this process.

In this article, we analyze the trend of the emotions expressed by developers in 2018 on the Stack Overflow platform in posts concerning 26 programming languages. To do so, we propose a learning model trained by distant supervision and the comparison of two different classifier architectures.

Keywords: Programming languages; Emotion detection; Machine learning

1. Introduction

Selecting the appropriate programming language is often a crucial step in the software development pipeline. This choice is naturally affected by technical considerations about the strengths and weaknesses of the programming language in addressing the problem of interest. The recent advent of social networks dealing with technical topics has involved developers in discussions about programming languages in which, often, strictly technical issues are on par with more emotionally-expressed personal views. Stack Overflow is a platform devoted entirely to developers and one of the largest sites posting discussions as well as questions and answers about software engineering. The importance of emotions and opinions in software engineering becomes even more evident if one considers their tight relationship with the quality of collective work and individual productivity [1,2].

In this work we analyze StackOverflow.com posts about programming languages from a sentiment-analysis viewpoint, to identify which emotions they most commonly express and to highlight trends in developers' opinions. To do so, we select the Stack Overflow posts about the 26 most popular programming, scripting, and markup languages in 2018, according to the social network rankings, to dynamically analyze the users' sentiment about them. Our analysis is based on a specific dataset we collected and labeled using a completely automatic process based on distant supervision. To go beyond considering only the polarity of a post, which prevents one from distinguishing the nuances of the emotions expressed therein, we aim at detecting the seven basic emotions of Parrott's model [3]. We compare a three-level hierarchical classifier consisting of four specialized classifiers to a flat model consisting of a seven-output classifier. The results of our analysis identify the languages for which the posts were most frequently associated with positive feelings, providing information that is interestingly complementary to the Developer Survey that Stack Overflow publishes yearly. The two main contributions of this article are: (*i*) a model trained by distant supervision for which we compare two possible classifier architectures; (*ii*) dynamic analysis of the opinions and emotions expressed by developers.

2. Related work

Sentiment analysis and opinion mining analyze people's opinions, sentiments, evaluations, attitudes, and emotions from

* Corresponding author.
E-mail addresses: stefano.cagnoni@unipr.it (S. Cagnoni), lorenzo.cozzini@studenti.unipr.it (L. Cozzini), gianfranco.lombardo@unipr.it (G. Lombardo), monica.mordonini@unipr.it (M. Mordonini), agostino.poggi@unipr.it (A. Poggi), michele.tomaiuolo@unipr.it (M. Tomaiuolo).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2020.07.002

written texts. Progress in Natural Language Processing (NLP) is leading to an increasing interest in this topic, with several transversal applications involving both academia and industry [4]. Various approaches have been proposed in the scientific literature, depending on the nature of the task. For example, while sentiment analysis is essentially a binary classification task, emotion detection is usually a multi-class or multi-label one. Recently, the widespread use of social media platforms allowed these tasks to play a key role also in knowledge discovery and to measure users' satisfaction. For example, in [5] the authors perform sentiment analysis on Twitter to get insights about a pop music event, using a hierarchical classifier to distinguish neutral, positive, and negative feelings expressed by users. In [6], a community of patients on Facebook is studied by detecting emotions of seven classes, to retrieve information about the emotional disclosure related with a rare disease condition.

In [7], the study highlights: (*i*) the relevance of performing sentiment analysis and opinion mining on Stack Overflow discussions, and (*ii*) the issues related to the specific language features of technical posts. In fact, different writing styles and the use of different dictionaries make it difficult for a classifier trained on general-interest social networks to perform well on data from networks dealing with technical topics [8]. Recently, Cabrera et al. [9] compare two different ensemble learning architectures based on decision trees to perform emotion analysis on Stack Overflow as a multi-label classification task, overcoming the limitations imposed by a small training set. In light of this, in this work we compare a three-level hierarchical classifier to a flat one in a multi-class context as in [10], using distant supervision [11] to label the training set. Specifically, we base classification on the six primary emotions described at the first level of the tree-structured Parrott's model [3], adding a class for the "objective" posts.

3. Methodology

Public datasets used for sentiment analysis most often include content published on popular social networks. However, models trained on those general data exhibit poor performances when applied to specialized topics [12], especially when these are highly technical as in social networks like Stack Overflow. Thus, analyzing such a content requires ad hoc models built from data collected from the very same platform.

Building a large dataset may be a daunting task, requiring a panel of experts to label many hundreds instances to avoid biases and subjective decisions. However, a distant supervision approach can help creating and polishing large datasets, in a fully automatic way and with good results [13]. In building a dataset for this work, we have assigned to each post one of the following seven labels: Love, Joy, Surprise, Sadness, Fear, Anger, or Objective. Fig. 1 illustrates the main steps of the data analysis workflow used in this project, as a custom application of the Knowledge Discovery in Databases process [14]. The next subsections give details about each step: Data Selection, Preprocessing and Transformation, Distant Supervision, and Mining. Results are discussed in a specific section. The software developed for this analysis has been written in Python using, in particular, the *nltk* and *sklearn* modules. All the code is publicly available,[1] along with an annotated dataset, consisting of 6000 objective instances and 1000 instances for each basic emotion.

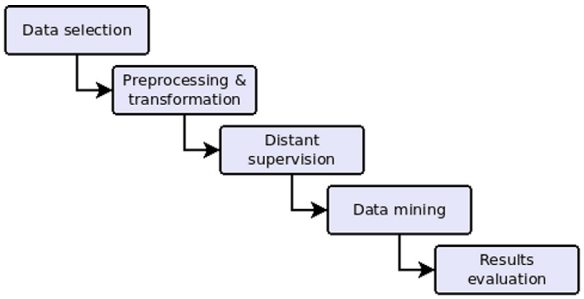

Fig. 1. Data analysis workflow.

3.1. Data selection

A dump of the whole Stack Overflow content is published regularly.[2] Dumps are available as compressed files and include questions, answers and comments. In this research, the analysis has regarded the file "stackoverflow.com-Posts.7z". At the time of download, the file size was about 70 GB (uncompressed) and contained more than 65 million posts from January 2008 to February 2019. The content is structured as an XML document, listing posts and all their relevant fields, including: Id, PostTypeId, AcceptedAnswerId, ParentId, CreationDate, Body, Title, Tags, AnswerCount, CommentCount.

In this research, we analyzed 2 415 694 posts published in 2018 and distributed almost uniformly over all months. These instances were filtered, retaining only those tagged as related with the most popular languages. The languages we chose are the most popular, according to the Developer Survey 2018, conducted by Stack Overflow with the participation of more than 100 000 developers[3]: Javascript, HTML, CSS, SQL, Java, Bash, Shell, Python, C#, PHP, C++, C, Typescript, Ruby, Swift, Assembly, Go, Objective-C, Vb.net, R, Matlab, Vba, Kotlin, Scala, Groovy, Perl.

3.2. Data preprocessing and transformation

Preprocessing consisted in removing HTML tags, codes example and punctuation from the texts, along with English stopwords, to limit the dictionary to common English words. Finally, we used the Snowball algorithm to reduce words to their stem form. We used the Word2Vec and Tf–Idf algorithms for vectorization and the Information Gain criterion for feature selection.

[1] http://sowide.unipr.it/datasets.
[2] https://archive.org/details/stackexchange.
[3] https://insights.stackoverflow.com/survey/2018/.

Please cite this article as: S. Cagnoni, L. Cozzini, G. Lombardo et al., Emotion-based analysis of programming languages on Stack Overflow, ICT Express (2020), https://doi.org/10.1016/j.icte.2020.07.002.

3.3. Distant supervision

Algorithm 1 Pseudo-code for adding a post to the training datasets $\mathcal{D}$.

```
function AddToDatasets(post)
    W ← StemAndTokenize(post)
    for all word ∈ W do
        emot ← ParrottStemmedDict(word)
        if emot then
            instance ← post − word
            D_subj ← D_subj ∪ {instance}
            D_[emot] ← D_[emot] ∪ {instance}
            if emot ∈ {love, joy, surprise} then
                D_pos ← D_pos ∪ {instance}
            else                ▷ emot ∈ {sadness, fear, anger}
                D_neg ← D_neg ∪ {instance}
            end if
            return
        end if
    end for
    D_obj ← D_obj ∪ {post}
end function
```

A first raw dataset was created automatically from the pre-processed data, as shown in Algorithm 1. Each post was annotated on the basis of the presence of specific keywords from Parrott's ontology of emotions. Each of the six basic emotions was associated to a list of terms, corresponding to its descendant sub-emotions in Parrott's ontology. These terms were stemmed, as done with the dataset words.

In fact, a taxonomy of emotions and keywords was built starting from the primary emotions of Parrott's model, creating six groups of keywords corresponding to them. Each group contained as terms a primary emotion and its related secondary end tertiary emotions. Stemming was used to match the keywords of this taxonomy with the words in the analyzed posts. We removed from the taxonomy some terms like "liking", "longing", and "contentment" to avoid confusing their stemmed version with other terms which do not have any emotional content. For example, "to like" may be confused with the preposition "like".

We matched each word in a post with those associated with emotions. According to such matches, we annotated posts as objective or subjective (first level), positive or negative (second level), and, finally, as conveying a precise emotion (third level). The decisive matches were also removed from the instance, before adding the latter to the dataset, to avoid introducing a methodological bias. As an additional refinement, instances containing keywords with contrasting polarity were annotated on a majority basis. At the end of this phase, we had collected, for each class: Joy, 12 059 posts; Love, 9101; Surprise, 1232; Fear, 2139; Anger, 1873; Sadness, 2738; Objective, 307 988. In total, 337 130 posts were annotated as conveying emotions (or being objective) through this process.

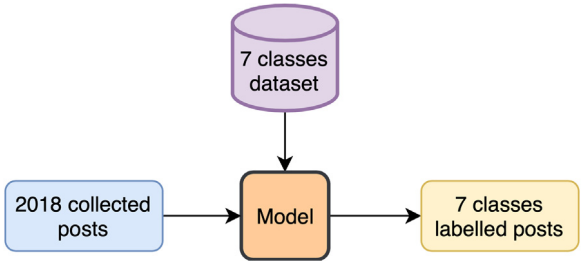

Fig. 2. Flat classifier.

Table 1
Comparison of classifiers' F-measure.

Classifier	F-measure
Flat, 7 classes	0.23
Hierarchical, 7 classes	0.41
Hierarchical, 3 classes	0.79

3.4. Data mining

We preliminarly considered the following algorithms for classification: Linear Regression, Logistic Regression, Random Forest, Naive Bayes, Gradient Boosting Trees, Support Vector Machines with Sequential Minimum Optimization (SMO). Finally we chose SMO for its consistently good accuracy and efficiency. We optimized the feature set for each classifier, after sorting it by the Information Gain criterion. The raw dataset was used to train a classifier and create an initial model that was applied to each instance in the raw dataset: we removed misclassified instances or those instances having the lowest confidence rate (i.e, outliers), producing a better performing clean dataset [13].

We compared two classifier architectures. The first one is based on a single "flat" model, distinguishing 7 possible classes (one of the six primary emotions, or "objective"). This classification model was trained over a balanced dataset, with 1000 instances for each class (see Fig. 2).

The other is a three-layered "hierarchical" modular classifier, consisting of 4 basic models (see Fig. 3). "Model 1", at the highest level, was trained over a balanced dataset of 12 000 instances to distinguish between subjective and objective posts. At the intermediate level, the polarity (positive or negative) of posts classified as subjective is determined by "Model 2", that was trained using a balanced set of 6000 subjective posts (3000 positive and 3000 negative). At the lowest level, "Model 3" and "Model 4" classify the three positive emotions, on one branch, and the three negative ones, on the other, respectively. Each model was trained using a specific balanced dataset of 3000 instances: 1000 posts expressing each of the positive emotions (Joy, Love. Surprise) for "Model 3" and 1000 posts expressing each of the negative emotions (Fear, Anger, Sadness) for "Model 4".

As shown in Table 1, the two architectures have very different classification quality. We compared them using the F-measure, defined as the harmonic mean of precision and recall;

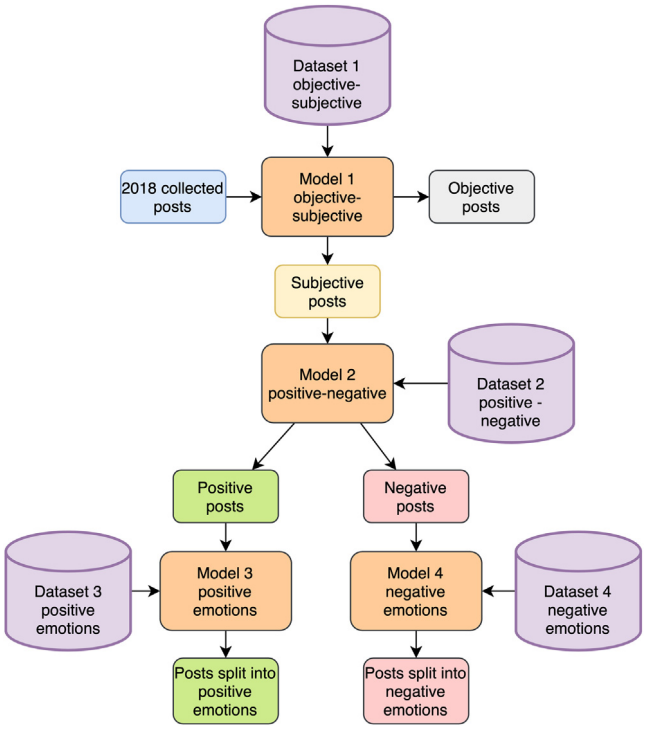

Fig. 3. Hierarchical classifier.

Table 2
Languages with most positive emotions in our analysis (left); most loved languages in the Stack Overflow 2018 survey (right).

Chart 1 (our analysis)	Positive emotions	**Chart 2** (dev. survey)	Positive answers
1. Matlab	74.57%	1. Rust	78.9%
2. R	74.53%	2. Kotlin	75.1%
3. Python	72.22%	3. Python	68.0%
4. Scala	71.74%	4. TypeScript	67.0%
5. SQL	71.73%	5. Go	65.6%
6. C	71.65%	6. Swift	65.1%
7. Assembly	71.36%	7. JavaScript	61.9%
8. Bash	70.08%	8. C#	60.4%
9. Perl	69.62%	9. F#	59.6%
10. Shell	69.17%	10. Clojure	59.6%

it reaches its best value at 1 (ideal case) with perfect precision and recall.

$$Fmeasure = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (1)$$

The hierarchical classifier is twice as accurate as the flat classifier. The intermediate classifier reaches an F-measure of 0.79 on the three classes (objective, positive, negative). The evaluations have been performed on a 30/70 split of the dataset. Classification at the lowest level is less accurate because of the number of classes and the subjectivity of the task: often, even humans do not agree about the most appropriate classification; moreover, a message may convey multiple emotions.

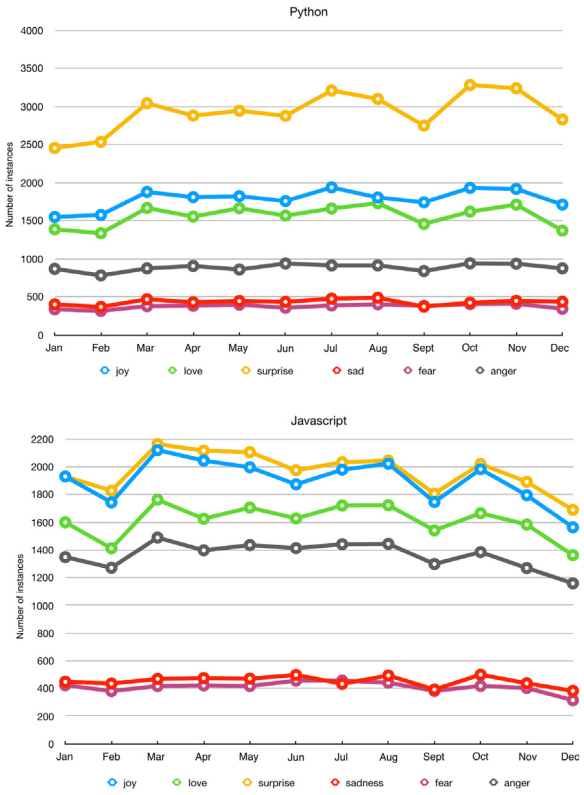

Fig. 4. Emotions of questions related to Python and Javascript.

4. Results

We used the hierarchical classifier to recognize emotions for all posts published in 2018 on Stack Overflow about the most popular languages, as described in Section 3.1. All 26 languages have been analyzed separately, often showing remarkably different profiles. As an example, Fig. 4 shows the emotions associated with questions related to Python and Javascript, as a comparison between two popular dynamic languages. As can be observed, Python-related posts tend to express anger less frequently and surprise more often than Javascript-related ones.

Table 2 shows the aggregated results of this analysis. Chart 1 (on the left) represents a list of the "happiest" languages, i.e., those for which most questions are associated with positive emotions, according to our analysis. As a comparison, Chart 2 (on the right) represents the most loved languages, according to Developer Survey 2018 (see Section 3.1). These two "charts" represent different data, both interesting for developers. The emotion analysis shows which languages are associated with positive posts by all kinds of platform users: professional developers, students, anyone curious about the language. Instead, the Stack Overflow survey included answers from "developers who are developing with that language or technology and have expressed interest in continuing to develop with it". As such, they show the degree of enthusiasm or commitment in a particular community (e.g., Rust developers), more than the general feeling about a language.

Please cite this article as: S. Cagnoni, L. Cozzini, G. Lombardo et al., Emotion-based analysis of programming languages on Stack Overflow, ICT Express (2020), https://doi.org/10.1016/j.icte.2020.07.002.

5. Conclusion

This work shows that it is possible to use an automatically labeled training set, obtained using a distant supervision approach, to train a model capable of identifying with good accuracy the emotions expressed by developers in the Stack Overflow posts about programming languages. Furthermore, it shows that the use of a hierarchical model consisting of multiple classifiers arranged on different levels gives better results than using a single flat classifier when performing emotion detection.

As a future extension, this work could also analyze replies and comments in addition to the original posts.

This approach to sentiment analysis seems to be general enough to be applied to other specific domains, different from the one addressed in this research.

CRediT authorship contribution statement

Stefano Cagnoni: Investigation, Validation, Writing - review & editing. **Lorenzo Cozzini:** Investigation, Software, Data curation, Visualization. **Gianfranco Lombardo:** Investigation, Methodology, Writing - review & editing. **Monica Mordonini:** Investigation, Methodology, Writing - review & editing. **Agostino Poggi:** Methodology, Resources, Writing - review & editing. **Michele Tomaiuolo:** Conceptualization, Methodology, Supervision, Writing - original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] E. Guzman, B. Bruegge, Towards emotional awareness in software development teams, in: Proceedings of ESEC/FSE 2013, 2013, pp. 671–674.

[2] M.R. Islam, M.F. Zibran, Leveraging automated sentiment analysis in software engineering, in: 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), IEEE, 2017, pp. 203–214.

[3] W.G. Parrott, Emotions in Social Psychology: Essential Readings, Psychology Press, 2001.

[4] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Hum. Lang. Technol. 5 (1) (2012) 1–167.

[5] P. Fornacciari, M. Mordonini, M. Tomaiuolo, A case-study for sentiment analysis on Twitter, in: WOA, 2015, pp. 53–58.

[6] G. Lombardo, P. Fornacciari, M. Mordonini, L. Sani, M. Tomaiuolo, A combined approach for the analysis of support groups on Facebook - The case of patients of Hidradenitis Suppurativa, Multimedia Tools Appl. 78 (3) (2019) 3321–3339.

[7] N. Novielli, F. Calefato, F. Lanubile, The challenges of sentiment detection in the social programmer ecosystem, in: Proceedings of the 7th International Workshop on Social Software Engineering, 2015, pp. 33–40.

[8] F. Calefato, F. Lanubile, F. Maiorano, N. Novielli, Sentiment polarity detection for software development, Empir. Softw. Eng. 23 (3) (2018) 1352–1382.

[9] L.A. Cabrera-Diego, N. Bessis, I. Korkontzelos, Classifying emotions in stack overflow and jira using a multi-label approach, Knowl.-Based Syst. 195.

[10] G. Angiani, S. Cagnoni, N. Chuzhikova, P. Fornacciari, M. Mordonini, M. Tomaiuolo, Flat and hierarchical classifiers for detecting emotion in tweets, in: Conference of the Italian Association for Artificial Intelligence, Springer, 2016, pp. 51–64.

[11] A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision, CS224N project report, Stanford 1 (12) (2009) 2009.

[12] P. Fornacciari, M. Mordonini, M. Tomaiuolo, Social network and sentiment analysis on Twitter: Towards a combined approach, in: KDWeb, 2015, pp. 53–64.

[13] S. Cagnoni, P. Fornacciari, J. Kavaja, M. Mordonini, A. Poggi, A. Solimeo, M. Tomaiuolo, Automatic creation of a large and polished training set for sentiment analysis on Twitter, in: International Workshop on Machine Learning, Optimization, and Big Data, Springer, 2017, pp. 146–157.

[14] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, From data mining to knowledge discovery in databases, AI Mag. 17 (3) (1996) 37.

Please cite this article as: S. Cagnoni, L. Cozzini, G. Lombardo et al., Emotion-based analysis of programming languages on Stack Overflow, ICT Express (2020), https://doi.org/10.1016/j.icte.2020.07.002.