Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Classifying emotions in Stack Overflow and JIRA using a multi-label approach[☆]

Luis Adrián Cabrera-Diego [*], Nik Bessis, Ioannis Korkontzelos

Edge Hill University, St Helens Rd, Ormskirk L39 4QP, United Kingdom

ARTICLE INFO

Article history:
Received 14 October 2019
Received in revised form 4 February 2020
Accepted 7 February 2020
Available online xxxx

Keywords:
Multi-label classification
Emotion classification
Stack Overflow
Jira Issue Tracker

ABSTRACT

A forum or social media post can express multiple emotions, such as love, joy or anger. Emotion classification has been proven useful for measuring aspects such as user satisfaction. Despite its usefulness, research in emotion classification is limited, because the task is multi-label and publicly available data sets and lexica are very limited. A number of emotion classifiers for general-domain text have been proposed recently, but only a few for text in the domain of Open Source Software (OSS), such as EmoTxt. In this paper, we explore different lexica and two multi-label algorithms for classifying emotions in text related to OSS. We trained various multi-label classifiers using *HOMER* and *RAkEL* on a data set of *Stack Overflow* posts and a data set of *JIRA Issue Tracker* comments. The classifiers have been enriched with features derived from different state-of-the-art lexica. We achieved multi-label *Micro F-scores* up to 0.811 and Subset 0/1 Loss of 0.290. These results represent a statistically significant improvement over the state-of-the-art.

1. Introduction

Emotions, such as *love*, *joy* and *anger*, are complex states of mind caused by internal or external events [1]. For many years, they have attracted research interest in psychology [2]. Researchers in Natural Language Processing (NLP) have shown interest in their applications. One of them, emotion classification, is explored in this paper for the domain of Open Source Software (OSS).

In psychology, multiple theories have been proposed to understand emotions. During 1972, the first formal theory proposed six basic universal emotions: *anger*, *disgust*, *fear*, *joy*, *sadness* and *surprise* [3]. In 1980, the *Wheel of Emotions* model arranged the eight primary bipolar emotions in four axes: *joy vs. sadness*, *anger vs. fear*, *trust vs. disgust* and *surprise vs. anticipation* [4]. Secondary emotions are identified as intensity variants or combinations of the primary ones. In 1987, emotions were represented in a tree with six main branches: *anger*, *fear*, *joy*, *love*, *sadness* and *surprise* [5]. The branches could be bifurcated into secondary and tertiary branches to model sub-emotions. In 2012, the *Hourglass of Emotions (HOE)* [1], a theory based on the Wheel of Emotions was proposed. Emotions were modelled in four dimensions: *pleasantness*, *attention*, *sensitivity* and *aptitude*. Each dimension can have positive or negative polarity and a different activation level. Depending on how active it is, different emotions are represented.

In Computer Science and NLP, automatic emotion mining has attracted research attention. In particular, the following tasks have been popular [2]:
A. Emotion detection: The task of determining whether a text conveys any emotion(s), without specifying which one(s).
B. Emotion classification: the identification of particular emotions, such as love or sadness, triggered in a non-neutral text.
C. Emotion cause detection: The task of determining the causes that stimulated the emotions expressed in a text.

Emotion classification, the most popular among these tasks [2], has been applied to quantify customer satisfaction of products or services [6], to prevent suicide [7] and to analyse newspapers articles [8] or tweets [9]. The classification of emotions in text related to OSS is less explored, as in [10,11].

In general, emotion classification is less popular than *sentiment analysis*, i.e. classification of text as positive, negative and neutral. Annotated data sets are few, small and rare due to their cost, subjectivity and exposure to disagreements [12]. In addition, there are multiple theories about emotions, as discussed previously. Resources for emotion classification, such as lexica, are scarce, probably because annotation is hard and expensive, and also due

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105633.

[*] Correspondence to: Laboratoire L3i, Faculté des Sciences et Technologies, La Rochelle Université, Avenue Michel Crépeau, La Rochelle, 17042, France.
E-mail addresses: luis.cabrera_diego@univ-lr.fr (L.A. Cabrera-Diego), nik.bessis@edgehill.ac.uk (N. Bessis), yannis.korkontzelos@edgehill.ac.uk (I. Korkontzelos).

https://doi.org/10.1016/j.knosys.2020.105633

to the multitude of emotion theories. Finally, emotion classification is a multi-label problem, i.e. more than one emotion can be expressed in a piece of text, and it is more challenging than single-label ones.

This paper presents our experiments towards building an emotion classifier for OSS-related text.[1] We used two multi-label classification algorithms and various emotion lexica. Methods were evaluated on two different data sets, one consisting of JIRA Issue Tracker comments and one of Stack Overflow posts. Moreover, the performance of classifiers was compared against a random baseline, the most frequent label baseline and EmoTxt [11], a state-of-the-art tool. Experimental results show that the classification methods explored outperform the baselines. They also perform statistically better than EmoTxt, when trained and tested on JIRA comments. We did not observe a statistical difference between EmoTxt and classifiers trained and tested on Stack Overflow posts.

This paper is structured as follows: Sections 2 and 3 discuss motivation for this work and multi-label classifiers, respectively. Section 4 presents work related to automatic emotion classification. Our methodology is explained in Section 5, whereas data is discussed in Section 6. In Section 7, we discuss the experimental and evaluative settings. Results are presented and discussed in Sections 8 and 9, respectively. Some threats to the validity of our conclusions are analysed in Section 10. Section 11 concludes and proposes some future work.

2. Motivation

The motivation for this work is to create an emotion classifier for text related to OSS projects, i.e. forums posts, issue tracker messages and mailing lists, which in combination with other tools can help developers to analyse OSS projects in terms of elements, such as user experience or project management quality. As it will be shown in Section 4, most of the existing tools for emotion classification have been created for processing general domain texts. However, the words in text related to OSS projects can have different connotations or senses. For example, some general domain classifiers indicate that the phrase "Every time I call this method, Java eats my RAM" expresses emotions different from anger or sadness.[2] Therefore, it is essential to create an emotion classifier that knows how specific words are used in the OSS domain.

Specifically, this research is part of the CROSSMINER project [13], which targets to help OSS developers in creating complex software systems by enabling monitoring, analysing and assisting them to select components, such as libraries, while facilitating knowledge extraction from their repositories.

3. Multi-label classifiers

Multi-label classification is the process of classifying data instances, each of which may belong to one or more classes [14]. It contrasts single-label classification, in which an instance can only be categorised into one class. Due to the varying number of labels that have to be predicted for an instance, multi-label classification problems are considered harder than single-label ones [15]. Apart from emotion classification, other multi-label classification tasks concern genres of films or books, elements in images or music styles in songs. Multi-label classifiers can be divided into three groups, following the approach they use [8]:

A. Problem transformation Methods transform a multi-label task into multiple single label ones. There are two types of transformation: *Binary Relevance* and *Label Powerset*. In the former, each instance annotated with several labels is copied multiple times, each of which is annotated with one label. *Label Powerset* assigns a unique label to each multi-label combination. Usually, only combinations in the training data are considered.

B. Ensemble Algorithms improve upon problem transformation methods by using multiple classifiers, trained on subsets of the original training data. Examples of these methods are *HOMER* [16, 17], *RAkEL* [18] and *ECC* [19].

C. Algorithm adaptation This group consists of methods created or adapted to perform inherently multi-label classification tasks. Some examples are *Backpropagation for Multi-label Learning (BP-MLL)* [20] (neural network), *Clus* [21] (decision trees) and *Multi-Label K-Nearest Neighbour (ML-kNN)* method [22] (k-nearest neighbours). In some cases, adapted algorithms, such as *AdaBoost.MH* [23], use, at their core, transformations to solve the problem [24]. Moreover, in recent years the number of multi-label algorithms based on neural networks has increased. Examples in the domain of image processing are Wei et al. [25] and Wang et al. [26], and in text classification we can name FastText[3] [27] and Nam et al. [28].

Furthermore, recently there is research interest on what is known as Extreme Multi-label Classification, where thousands, even millions, of possible labels have to be processed [29–32].

In this work, we experimented with two methods: *HOMER* and *RAkEL*. Specifically, we used the implementations in Mulan [33], a multi-label extension of the machine learning library Weka [34].

The *Hierarchy Of Multilabel classifiERs (HOMER)* [16] is a machine learning algorithm that "addresses a multi-label task by breaking down the entire label set recursively into several disjoint smaller sets that contain similar labels" [17]. These smaller sets of labels are then used to train multiple multi-label classifiers, arranged hierarchically, and which only focus on smaller sub-classification tasks. In its default instantiation, HOMER transforms the labels using Binary Relevance and uses internally *C4.5 Decision Trees* as the main classification algorithm.

RAndom k-labELsets (RAkEL) [18] is another machine learning algorithm that creates multiples multi-label classifiers. However, unlike HOMER, RAkEL splits the set of labels on disjoint subsets that are selected randomly and non-recursively. By default, RAkEL uses a Label Powerset transformation in an attempt to improve predictions by finding correlations between labels. Moreover, the default internal classification algorithm is a C4.5 Decision Tree.

Both HOMER and RAkEL can support any multi-label classification algorithm, such as BP-MLL or CLUS, or single-label classifiers with transformed labels, either with Binary Relevance or Label Powerset.

4. Related work

Recently, researchers have shown interest in automatic classification of emotions in text. In this section, we review the state-of-the-art.

The *Affect Analysis Model* [35] is an unsupervised emotion classification method. It relies on rules and a manually annotated database. Among others, it contains affective strength for emoticons, affect words, common abbreviations and acronyms. *Feeler* [36] is also unsupervised and based on the cosine similarity of high-dimensional vectors. Its features encode TF-IDF-weighted

[1] The source code and data sets are publicly available and can be found in: github.com/creat89/EmotionsJiraStackoverflow

[2] We tested the on-line demos: paralleldots.com/emotion-analysis, tone-analyzer-demo.ng.bluemix.net, depechemood.eu/DepecheMood.html. Only the former identified that the phrase expresses *anger*; the others identified *inspiration*, *amusement* or *confidence*.

[3] Originally this classifier only supported single-label classification problems, but since April 2019 it supports multi-label ones too.

unigrams and are enriched using lexica, such as the *WordNet Affect Lexicon* [37]. The unsupervised method presented in [38] uses this last lexicon too and reduction tools, such as *Latent Semantic Analysis* and *Non-negative Matrix Factorisation*.

In contrast to these methods, supervised learning has been combined with a psychological approach in [39]. In particular, a *Hidden Markov Model (HMM)* was used to simulate how mental state sequences affect or cause emotions.

A multi-label classifier was employed to detect emotions in suicide notes [40]. It used Label Powerset and a one-vs.-all radial basis Support Vector Machine (SVM) that represented text using unigrams. The classifier detected 15 emotions, e.g. *hopelessness* and *guilt*, and also the lack of emotion. Many multi-label classifiers, e.g. BP-MLL, RAkEL and HOMER, have been evaluated for emotion identification in short Brazilian Portuguese texts [8]. Words that did not occur in a stoplist were weighted by TF-IDF and the *SenticNet* lexicon [41] was used.

Several state-of-the-art systems were proposed to address tasks in the *Semantic Evaluation series (SemEval)*. Task 4 in *SemEval 2007* was about classifying emotions and polarity of news headlines [42]. Out of three participants, the best-performing system in the emotion classification sub-task was *UPAR7* [43], a rule-based system that uses dependency graphs enriched with information from the WordNet Affect Lexicon and *SentiWordNet* [44]. Task 1 in *SemEval 2018* [9] also focussed on emotion classification among others. Most participants used *Convolution Neural Network (CNN)*, *Recurrent Neural Network (RNN)* or *Long-Short Term Memory Network (LSTM)* architectures, along with external resources such as lexica, word embeddings or word *n*-grams. The best classifier, NTUA-SLP [45], consists of a *Bidirectional LSTM (BiLSTM)* that uses attention and embeddings, trained on a large corpus of unlabelled tweets. As the task only provided a small training set, transfer learning was implemented by pre-training on the sentiment analysis corpus in *Task 4* of *SemEval 2017* [46].

Recently, there is research interest in the classification of emotions in texts related to software engineering and development. For instance, in Murgia et al. [47], the authors performed a qualitative and quantitative analysis regarding the feasibility of applying automatic classification techniques, such as Naïve-Bayes, SVM or k-Nearest Neighbours, for classifying emotions in issue comments.

In [10], emotions, such as *anger*, *love*, *sadness* and *joy*, conveyed in posts from the JIRA Issue Tracker[4] were detected by multiple Linear SVMs. It is not indicated if the multi-label task was addressed and how. Apart from text, features encode information from the *WordNet Affect Lexicon*, *SentiStrength* [49] and a politeness detection tool [50].

EmoTxt[5] [11] is an emotion classifier based on the principles in [10], separately trained on two corpora: JIRA Issue Tracker posts [48] and Stack Overflow comments [51]. EmoTxt uses six binary SVMs for classifying *joy*, *love*, *sadness*, *anger*, *surprise*, and *fear* following a one-vs.-all approach with Binary Relevance.[6] Each SVM can assign a specific emotion, only. Apart from the features in [10], the authors added TF-IDF. Although EmoTxt can be seen as multi-label if the output of all classifiers is merged, it was evaluated using uniquely single-label metrics, i.e. precision, recall and F-score. EmoTxt has been implemented in *EMTk* [52] and in *EmoD* [53], two toolkits that analyse emotions and sentiment in software engineering documents and data sources related to repositories.

DEVA [54] is based on a bi-dimensional theory of emotions, in which *excitement*, *stress*, *depression* and *relaxation* are determined in accordance to arousal and valence values. To determine them, it uses dictionaries along with heuristics, such as the detection of exclamation marks, capital letters or interjections. DEVA was evaluated on a manually annotated corpus of ∼1800 JIRA comments. An improved version of *DEVA* uses machine learning [55], e.g. *Adaptive Boosting* and *Gradient Boosting Tree*, instead of lexicons and heuristics, only.

As discussed, most state-of-the-art emotion classifiers are unsupervised. Some explore supervised machine learning and mainly rely on single-label classifiers.

5. Methodology

We apply two multi-label classifiers, HOMER and RAkEL, and evaluate them comparatively on OSS-related text. We used the *NLP4J*[7] lemmatiser to explore if lemmatisation affects the outcome. We also investigate the use of lexical resources to enrich classification vectors. The vectors are composed mainly of word *n*-grams, skip-bigrams,[8] and extra text-based and lexicon-based heuristic features. The extra text-based features concern the number of:

- Positive emoticons
- Negative emoticons
- Consecutive positive emoticons
- Consecutive negative emoticons
- Question words
- Negation words
- Ellipsis
- Words uniquely in capital letters
- Elongated words
- Question marks
- Exclamation marks
- Consecutive exclamation marks
- Consecutive question marks
- Alternated question and exclamation marks

We also included binary features that represent the presence of meaningful symbols, next to the first and last token of a text:

- Question mark
- Exclamation mark
- Negative emoticons
- Positive emoticons
- Ellipsis
- Full stop (last sequence only)

Features were extracted from three lexica: *SenticNet 5* [41], the *NRC Word-Emotion Association Lexicon* [56] and the *NRC Affect Intensity Lexicon* [57]. The lexica were organised in two groups, and each contributed different features:

A. NRC Emotion: This group consists of the NRC Word-Emotion Association Lexicon and the NRC Affect Intensity Lexicon. The former has been annotated by crowd-sourcing for emotions (anger, fear, anticipation, trust, surprise, sadness, joy and disgust) and polarity (positive or negative) associated to a list of words that come from other lexica. From this lexicon we obtain six features: the number of words related to four specific emotions (anger, fear, surprise, sadness and joy) and the number of neutral words, i.e words that appeared in the lexicon but were not linked to any polarity or emotion. The latter lexicon is a manually annotated collection of ∼6k words linked to their intensities about each emotion (anger, joy, sadness, fear). For each emotion in this lexicon, we calculate the number of words related to it in an instance, the average and maximum emotional strength and strength of the last emotional word. This group contributed a total of 22 features.

[4] The data set contains 4000 entries and is not described in detail. Most probably, same data set presented in [48].

[5] EmoTxt is freely available: github.com/collab-uniba/Emotion_and_Polarity_SO.

[6] Models for *surprise* and *fear* were not generated on JIRA posts.

[7] emorynlp.github.io/nlp4j

[8] The numbers of *n*-grams and skip-bigrams to be considered were determined during the optimisation process, described in Section 7.

Please cite this article as: L.A. Cabrera-Diego, N. Bessis and I. Korkontzelos, Classifying emotions in Stack Overflow and JIRA using a multi-label approach, Knowledge-Based Systems (2020) 105633, https://doi.org/10.1016/j.knosys.2020.105633.

B. SenticNet 5 is a collection of 100,000 entries, that range from unigrams to pentagrams, annotated according to the axes of the Hourglass of Emotions. SenticNet 5 also contains polarity annotations (positive or negative), polarity strength, moods, i.e. *surprise*, *interest*, *disgust*, and related concepts. It was generated using a LSTM neural network that extended previous SenticNet versions by discovering *conceptual primitives*, i.e. ensembles of verb-noun pairs. We calculate 27 features by matching word n-grams between text and the lexicon.[9] Eight concern polarity: the number of n-grams with polarity in a text, the average and maximum polarity strength, the strength of the last word with polarity. The remaining 19 features concern the axes of the Hourglass of Emotions: the number of n-grams in an instance related to the axe, their average, maximum and minimum strength and the strength of the last word related to the axe.

Term search and matching was done on lemmatised texts, regardless if the classifiers used lemmatisation to extract features. This was done to maximise the number of matching words and generate numerical features accurately.

6. Data sets

To the best of our knowledge, there are two data sets for emotion detection in the domain of OSS. The first is a collection of ∼5.8k JIRA Issue Tracker comments divided in three groups that correspond to varying levels of granularity, i.e. sentences vs. full comments [48]. The annotated emotions are *joy*, *love*, *anger*, *sadness*, *surprise* and *fear*; *neutral* instances are also included. This data was used in EmoTxt [11] and for a sentiment analysis tool [58]. The second data set contains 4.8k Stack Overflow posts [51], manually annotated with the same emotions. As all instances are assigned two labels or fewer, Table 1 shows the number of instances annotated with each combination, for each corpus.

Following EmoTxt [11], we aim at determining which emotions are present in text, and we did not consider the neutral instances. The non-neutral instances of both corpora were split using stratified sampling into training and test parts using an 80%–20% proportion. In the split process, we prioritised multi-label instances towards the training part. For instance, in JIRA, there is only one instance labelled with both *anger* and *surprise*, therefore, it was assigned to the training part.

7. Settings

We conducted 16 experiments, in which we use different multi-label classifiers, lexica for vector enrichment, lemmatisation settings and data sets for training and testing. Furthermore, we optimised parameters using *Bayesian Optimisation* [59], a lazy learning method that models the hyper-surface generated by an objective function and a parameter set. We optimised the following parameters:

A. Word n-grams Represent contiguous character sequences linked by white-space, e.g. "*I am happy*". We considered n-grams of length one, two and three.

B. Skip-bigrams A variation of n-grams, in which a gap of predefined size is skipped to generate bigrams. For example, in the phrase "*... a frequent coding issue*", "*frequent issue*" is a skip-bigram with one token gap. We considered skip-bigrams with one or two tokens gap and also no skip-bigrams at all.

C. Minimum frequency of occurrence We explored a threshold in the range [1, 50]. N-grams or skip-bigrams that exceed it are considered as features.

D. Subsets As discussed in Section 3, HOMER and RAkEL subdivide the training data set and create multiple single-label classifiers that deal with smaller label sets. We optimised the number of subdivisions used by each method. These ranged between two and five, i.e. the number of labels minus one.

As Bayesian Optimisation objective function, we used the minimum between the median and the average of the multi-label Macro F-score (see Eq. (3)) calculated in a 10-fold cross validation setting. We have chosen the Macro F-score as it considers the proportion of labels in the data set.[10] Table 2 shows the parameters obtained by Bayesian Optimisation for all experiments.

As most multi-label classifiers, HOMER and RAkEL are probabilistic. Thus, to consider an emotion label as triggered, its probability needs to surpass a threshold. We used the default threshold set in HOMER and RAkEL, 0.5.

To extend evaluation, we compared HOMER and RAkEL against EmoTxt, which we trained on the JIRA and Stack Overflow data sets for *love*, *joy*, *sadness*, *anger*, *surprise* and *fear*. Its parameters were tuned using the integrated optimisation facility. For each instance, the predictions of all EmoTxt models were merged into a single vector, compatible with multi-label evaluation metrics.

We considered two baselines: assigning random labels and assigning the most frequent label, i.e. *love*, to all test instances. The random baseline consists of six boolean aleatory generators, that randomly determine which labels are activated in a prediction vector.[11] The scores are averaged over ten executions.

Multi-label classification is challenging, not only in developing methods but also evaluating results because "*it is difficult to tell which of the following mistakes is more serious: one instance with three incorrect labels vs. three instances each with one incorrect label*" [60].

Unlike single-label metrics, multi-label ones use vector sets instead of confusion matrices to represent all instances. Let us consider a corpus of n manually annotated instances. Let the vector sets $T = \{t_1, .., t_n\}$ and $P = \{p_1, .., p_n\}$ represent the ground truth and predictions, respectively. Each t_i or p_i, $i \in [1..n]$, is a binary vector of length l equal to the number of possible labels; 1 denotes triggered labels and 0 inactive ones. Below, we discuss the metrics we employ.

Hamming Loss calculates how different the prediction is from the expected outcome. For each incorrect label prediction, 1 is added to the loss function:

$$\text{Hamming Loss}(T, P) = \frac{1}{nl} \sum_{i=1}^{n} \sum_{j=1}^{l} t_{ij} \oplus p_{ij} \tag{1}$$

For this metric, all errors are equally important. Predicting *anger* instead of *love* or *joy* instead of *love* are equally wrong. Hamming Loss is affected by corpus imbalance, i.e. wrong prediction of infrequent labels can be under-estimated.

Subset 0/1 Loss counts predictions with at least one incorrect label as wrong:

$$\text{Subset 0/1 Loss}(T, P) = \frac{1}{n} \sum_{i=1}^{n} t_i \oplus p_i \tag{2}$$

[9] The number of n-grams, one to five, used in this matching is independent of the number of n-grams used by the machine learning algorithms.

[10] Using a metric that does not consider the proportion of labels can be misleading. For example, a classifier that assigns the most frequent label to all instances can generate low values of Hamming Loss, because it hides classification errors for infrequent class instances.

[11] The random baseline can generate null vectors, in which none of the emotions is triggered. Theoretically, a multi-label classifier cannot generate null vectors. In practice null vectors are possible, because probabilities are generated for each label independently. Unlike single-label classifiers, normalisation functions, such as softmax, are not applicable.

Table 1
Corpora instances annotated with zero (Neutral), one (diagonal) or two emotions.

Corpus	Neutral	Emotions	Joy	Love	Anger	Surprise	Sadness	Fear
JIRA	3885	Joy	280	80			2	
		Love		744			10	
		Anger			340	1	5	
		Surprise				30	1	
		Sadness					439	
		Fear						13
Stack Overflow	1959	Joy	404	71	4	6	4	
		Love		1,130	13	4	2	
		Anger			851	2	11	1
		Surprise				30	1	2
		Sadness					202	10
		Fear						91

Table 2
Optimised parameters for experimentation.

		Corpus	Lemma	n-grams	Skip-bigrams	Min.	Subsets
HOMER	NRC	JIRA	NO	2	1	10	4
			YES	2	2	15	5
		Stack Overflow	NO	1	0	4	4
			YES	1	1	25	4
	SenticNet	JIRA	NO	3	0	1	2
			YES	3	0	32	2
		Stack Overflow	NO	1	0	5	5
			YES	1	1	15	3
RAkEL	NRC	JIRA	NO	3	0	1	5
			YES	1	2	15	5
		Stack Overflow	NO	3	1	40	4
			YES	1	1	35	5
	SenticNet	JIRA	NO	1	2	1	5
			YES	1	2	1	5
		Stack Overflow	NO	1	1	20	5
			YES	1	0	25	5

Macro F-score[12] evaluates label prediction accuracy. It takes into account the proportion of each label class in the data set:

$$\text{Macro F-Score}(T, P) = \frac{2}{l}\sum_{i=1}^{l}\frac{\sum_{j=1}^{n} t_{ij}\cdot p_{ij}}{\sum_{j=1}^{n} t_{ij}+p_{ij}} \tag{3}$$

Micro F-score[12]: evaluates how well on average labels and instances have been predicted:

$$\text{Micro F-Score}(T, P) = 2\,\frac{\sum_{i=1}^{l}\sum_{j=1}^{n} t_{ij}\cdot p_{ij}}{\sum_{i=1}^{l}\sum_{j=1}^{n} t_{ij}+p_{ij}} \tag{4}$$

Instance F-score: assesses how well on average instances have been predicted:

$$\text{Instance F-Score}(T, P) = \frac{2}{n}\sum_{j=1}^{n}\frac{\sum_{i=1}^{l} t_{ij}\cdot p_{ij}}{\sum_{i=1}^{l} t_{ij}+p_{ij}} \tag{5}$$

The range of Hamming and Subset 0/1 Loss values is [0, 1]. As they are loss functions, zero means that all predictions were correct. Macro, Micro and Instance F-score also range in [0, 1], however, one indicates perfect performance.

Although multi-label metrics are suitable for this task, we use standard single-label metrics to evaluate the global performance of each classifier per emotion. Let $I_c(E)$ be the number of instances where E was predicted correctly, $I_p(E)$ the number of instances predicted with E and $I_a(E)$ the actual number of E instances. Precision and recall are defined as:

$$\text{Precision}(E) = \frac{I_c(E)}{I_p(E)} \qquad \text{Recall}(E) = \frac{I_c(E)}{I_a(E)} \tag{6}$$

F-score is the harmonic mean of precision and recall. The average F-score of all emotions is equal to the value of the Macro F-score, defined in Eq. (3).

In addition, we assess the statistical significance of performance differences, in terms of Subset 0/1 Loss, among EmoTxt, HOMER and RAkEL, using *Cochran's Q Test* with $\alpha = 0.05$. If p value refutes the null hypothesis, i.e. the results are statistically different, we apply as *post hoc* a pairwise *McNemar Test* with $\alpha = 0.05$ and *False Discovery Rate* correction. We calculate the effect size for method pairs that show a statistical significant difference using *Cramér's V*. It is considered as small if $V = 0.1$, medium if $V = 0.3$ and large if $V = 0.5$ [61].

8. Results

Table 3 shows the evaluation results for each classifier and the number of "*null*" vectors predicted. In null vectors, none of the emotions was predicted with a probability higher than the 0.5 threshold. Results for EmoTxt and the two baselines, discussed in Section 7, are also shown.

We observe that models trained and tested on JIRA perform better than models trained and tested on Stack Overflow. The performance is lower for models trained and tested on different data set. The number of null vectors fluctuates remarkably, but most are predicted by models trained on Stack Overflow. SenticNet 5 produces the fewest null vectors when combined with HOMER.

Subset 0/1 Loss expresses the percentage of wrongly predicted instances. We can observe that the model with ID = 13 predicted at least one emotion wrongly in 29% of the JIRA test instances, whereas the model with ID = 15 predicted wrongly 44.2% of the Stack Overflow test instances. The random baseline predicted wrongly 98% of the instances in both test data sets. The Most

[12] This metric is different than the eponymous metric for single-label classification.

Please cite this article as: L.A. Cabrera-Diego, N. Bessis and I. Korkontzelos, Classifying emotions in Stack Overflow and JIRA using a multi-label approach, Knowledge-Based Systems (2020) 105633, https://doi.org/10.1016/j.knosys.2020.105633.

Table 3
Evaluation results of the 16 emotions classifiers that were considered. The scores presented for the Random label baseline are the average of ten executions.

		Training set	Lemma	ID	F-score			Loss		Null	
					Macro	Micro	Instance	Subset 0/1	Hamming		
RAkEL	NRC	JIRA	NO	1	0.589	**0.811**	0.757	0.292	**0.063**	57	JIRA Test Set
			YES	2	0.551	0.789	0.743	0.319	0.070	51	
		Stack Overflow	NO	3	0.366	0.568	0.503	0.556	0.136	95	
			YES	4	0.280	0.556	0.483	0.592	0.141	107	
	SenticNet	JIRA	NO	5	0.527	0.767	0.723	0.334	0.079	51	
			YES	6	0.538	0.772	0.723	0.341	0.077	58	
		Stack Overflow	NO	7	0.349	0.588	0.514	0.546	0.128	99	
			YES	8	0.304	0.528	0.473	0.592	0.153	85	
HOMER	NRC	JIRA	NO	9	0.585	0.783	0.753	0.317	0.073	42	
			YES	10	0.571	0.804	0.763	0.295	0.065	49	
		Stack Overflow	NO	11	0.308	0.583	0.536	0.543	0.140	64	
			YES	12	0.435	0.575	0.501	0.565	0.134	104	
	SenticNet	JIRA	NO	13	0.578	0.784	**0.779**	**0.290**	0.076	**12**	
			YES	14	**0.590**	0.777	0.772	0.300	0.078	15	
		Stack Overflow	NO	15	0.271	0.519	0.440	0.629	0.147	122	
			YES	16	0.347	0.561	0.520	0.534	0.143	67	
EmoTxt		JIRA		17	0.488	0.755	0.673	0.373	0.077	93	
		Stack Overflow		18	0.340	0.644	0.578	0.465	0.110	91	
Random label				–	0.254	0.224	0.240	0.983	0.503	5.3	
Most frequent label				–	0.417	0.100	0.411	0.624	0.200	0	
RAkEL	NRC	JIRA	NO	1	0.311	0.436	0.354	0.758	0.179	188	Stack Overflow Test Set
			YES	2	0.259	0.437	0.365	0.755	0.185	163	
		Stack Overflow	NO	3	0.497	**0.683**	0.627	0.452	0.104	107	
			YES	4	0.465	0.671	0.625	0.454	0.109	96	
	SenticNet	JIRA	NO	5	0.228	0.378	0.307	0.762	0.189	200	
			YES	6	0.214	0.347	0.268	0.798	0.191	234	
		Stack Overflow	NO	7	0.418	0.664	0.608	0.479	0.110	114	
			YES	8	0.438	0.676	0.623	0.449	0.106	111	
HOMER	NRC	JIRA	NO	9	0.273	0.423	0.343	0.753	0.180	196	
			YES	10	0.268	0.451	0.362	0.723	0.165	213	
		Stack Overflow	NO	11	0.496	0.681	0.629	0.451	0.106	99	
			YES	12	**0.509**	0.675	0.640	0.454	0.111	83	
	SenticNet	JIRA	NO	13	0.306	0.406	0.379	0.716	0.210	**44**	
			YES	14	0.282	0.403	0.379	0.687	0.205	57	
		Stack Overflow	NO	15	0.467	**0.683**	0.632	**0.442**	0.103	104	
			YES	16	0.484	0.682	**0.642**	0.451	0.108	86	
EmoTxt		JIRA		17	0.225	0.333	0.238	0.816	0.178	302	
		Stack Overflow		18	0.431	0.680	0.596	0.445	**0.097**	152	
Random label				–	0.260	0.225	0.250	0.980	0.501	9.1	
Most frequent label				–	0.419	0.100	0.418	0.604	0.198	0	

Frequent Label baseline predicted wrongly at least 60% of the instances.

In general, EmoTxt performs worse than HOMER and RAkEL methods when trained and tested on the same source. Otherwise, performance differences are not constant and no particular pattern can be observed.

All models outperform the random baseline, with respect to all metrics. Models trained and tested on the same source perform twice as well as the baseline in terms of Macro F-score, thrice for Micro F-score and Subset 0/1 Loss, and five times for Hamming loss. Similarly, these models perform better than the most frequent label baseline. The baselines generate less null vectors. However, this does not mean that the vectors they predict are always correct.

With respect to the statistical analysis of Subset 0/1 results for methods tested on JIRA, Cochran's Q Test showed that at least one method pair exhibits a statistically significant performance difference ($p_{\text{value}} = 6.49 \times 10^{-166}$). The *post hoc* test results are shown in the bottom-left part of Table 4. Methods trained on JIRA perform statistically different than models trained on Stack Overflow. Models trained on JIRA perform better than those trained on Stack Overflow, when tested on JIRA. EmoTxt trained on JIRA, ID = 17, performs statistically different than all methods except for those trained on JIRA using SenticNet 5 and RAkEL, i.e. ID = 5 ($p_{\text{value}} = 0.175$) and ID = 6 ($p_{\text{value}} = 0.100$). This means that models with ID = 1, 2, 9, 10, 13, 14 are better than 17. Moreover, the effect size is between 0.12 and 0.19, i.e. small or small-medium. For instance, the difference between the model trained on JIRA using SenticNet 5, HOMER and no lemmatisation, i.e. ID = 13, and EmoTxt trained on JIRA, i.e. ID = 17, has a $p_{\text{value}} = 2.01 \times 10^{-4}$ and small-medium effect size ($V = 0.19$).

Concerning the results of Cochran's Q Test applied to the models tested on Stack Overflow, we obtained a $p_{\text{value}} = 3.22 \times 10^{-271}$, which means that at least one pair of methods is statistically different in terms of Subset 0/1 Loss. The outcomes of the *post hoc* test are shown in the top-right part of Table 4. EmoTxt trained on Stack Overflow, ID = 18, is statistically different from models trained on JIRA (ID = 1, 2, 5, 6, 9, 10, 13, 14, 17). However, no statistical difference was found between any flavour of EmoTxt trained on Stack Overflow, i.e. ID = 18, and any other method trained on Stack Overflow, i.e. ID = 3, 4, 7, 8, 11, 12, 15, 16. In

Table 4
Results of applying a *post hoc* test over the outcomes obtained for methods tested on the Stack Overflow data set (top-right part) and on the JIRA data set (bottom-left part). A dash (–) designates no statistical difference found, * designates that $p_{\text{value}} < 0.05$, ○ that $p_{\text{value}} < 0.01$, and ● that $p_{\text{value}} < 0.001$. Cramér's V effect size is shown next to the p_{value}.

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1		-	● .43	● .43	-	-	● .41	● .44	-	-	● .45	● .43	- .07	○ .12	● .45	● .44	* .11	● .44
2	-		● .43	● .43	-	* .09	● .41	● .45	-	-	● .45	● .44	-	○ .12	● .45	● .45	○ .11	● .44
3	● .44	● .39		-	● .45	● .49	-	-	● .46	● .41	-	-	● .36	● .33	-	-	● .51	-
4	● .47	● .42	* .11		● .45	● .49	-	-	● .46	● .40	-	-	● .36	● .33	-	-	● .52	-
5	* .10	-	● .35	● .39		* .09	● .42	● .46	-	-	● .46	● .45	-	● .14	● .47	● .46	* .10	● .47
6	* .12	-	● .34	● .39	-		● .46	● .50	* .09	● .16	● .50	● .49	● .16	● .22	● .51	● .51	-	● .50
7	● .42	● .37	-	* .11	● .33	● .33		-	● .43	● .37	-	-	● .33	● .30	-	-	● .48	-
8	● .46	● .44	-	-	● .42	● .41	* .11		● .46	● .42	-	-	● .37	● .34	-	-	● .53	-
9	-	-	● .40	● .44	-	-	● .37	● .44		-	● .47	● .46	-	○ .12	● .47	● .47	○ .12	● .46
10	-	-	● .42	● .45	-	* 0.12	● .39	● .46	-		● .41	● .40	-	-	● .42	● .42	● .17	● .41
11	● .41	● .38	-	* .12	● .33	● .33	-	* .12	● .37	● .40		-	● .37	● .33	-	-	● .53	-
12	● .44	● .40	-	-	● .36	● .35	-	-	● .40	● .43	-		● .36	● .33	-	-	● .52	-
13	-	-	● .41	● .46	* .10	* .12	● .40	● .45	-	-	● .39	● .42		-	● .38	● .37	● .17	● .37
14	-	-	● .40	● .44	-	-	● .40	● .44	-	-	● .39	● .41	-		● .34	● .34	● .22	● .34
15	● .52	● .48	● .18	-	● .46	● .45	● 0.21	-	● .50	● .51	● .20	○ .14	● .51	● .51		-	● .52	-
16	● .38	● .34	- -	○ .14	● .32	● .31	-	○ .14	● .34	● .38	-	-	● .39	● .38	● .22		● .52	-
17	● .19	* .12	● .30	● ● .34	-	-	● .28	● .34	* * .12	● .18	● .27	● .30	● .19	○ .15	● .39	● .25		● .53
18	● .34	● .29	● .18	● .24	● .26	● .25	○ .15	● .24	● .29	● .33	○ .14	● .20	● .36	● .33	● .33	○ .13	● .19	

Table 5
Precision (P), Recall (R) and F-score (F_1), for every emotion, obtained by each classifier trained and tested on one specific corpus, either the JIRA data set or the Stack Overflow data set. SN stands for SentiNet.

			Lemma	ID	Joy	Love	Sadness	Anger	Surprise	Fear
					(P R F_1)	(P R F_1)	(P R F_1)	(P R F_1)	(P R F_1)	(P R F_1)
JIRA	RAkEL	NRC	NO	1	.**82** .62 .**70**	.91 .94 .92	.89 .70 .78	.76 .70 .73	.**60** .30 .40	0 0 0
			YES	2	.74 .**67** .70	.89 .**94** .92	.83 .68 .75	.70 .66 .68	.40 .20 .27	0 0 0
		SN	NO	5	.72 .60 .66	.92 .92 .92	.80 .71 .75	.61 .59 .60	.29 .20 .24	0 0 0
			YES	6	.69 .59 .64	.88 .90 .89	.88 .70 .78	.70 .68 .685	.29 .20 .24	0 0 0
	HOMER	NRC	NO	9	.79 .57 .66	.**92** .94 .**93**	.75 .**80** .77	.64 .62 .63	.56 .**50** .**53**	0 0 0
			YES	10	.75 .59 .66	.91 .91 .91	.88 .7 .**80**	.76 .77 .**77**	.50 .20 .29	0 0 0
		SN	NO	13	.75 .60 .67	.88 .96 .92	.74 .78 .76	.60 .69 .64	.57 .40 .47	0 0 0
			YES	14	.75 .56 .64	.90 .94 .92	.78 .78 .78	.56 .**76** .65	.33 .10 .15	.**50** .**33** .**40**
	EmoTxt			17	.71 .48 .57	.88 .83 .85	.**93** .69 .79	.**85** .62 .72	0 0 0	0 0 0
Stack Overflow	RAkEL	NRC	NO	3	.62 .33 .43	.79 .78 .78	.67 .43 .52	.74 .73 .74	0 0 0	.59 .45 .51
			YES	4	.50 .35 .41	.78 .81 .79	.72 .38 .50	.71 .71 .71	0 0 0	.60 .27 .37
		SN	NO	7	.53 .37 .44	.81 .79 .80	.58 .45 .51	.70 .69 .70	0 0 0	.17 .045 .07
			YES	8	.54 .37 .**44**	.78 .78 .78	.67 .43 .52	.76 .74 .75	0 0 0	.33 .09 .10
	HOMER	NRC	NO	11	.48 .39 .43	.81 .78 .797	.**76** .47 .**58**	.73 .73 .73	0 0 0	.57 .36 .44
			YES	12	.43 .36 .39	.77 .**84** .**80**	.58 .38 .46	.72 .72 .72	.**33** .**10** .**15**	.55 .**50** .**52**
		SN	NO	15	.54 .31 .39	.77 .80 .79	.68 .40 .51	.77 .**76** .**76**	.20 .**10** .13	.60 .14 .22
			YES	16	.45 .**40** .43	.79 .81 .80	.68 .**49** .57	.75 .74 .75	.17 .**10** .13	.36 .18 .24
	EmoTxt			18	.**76** .19 .30	.**82** .79 .**80**	.68 .40 .51	.**81** .67 .73	0 0 0	**1**.**0** .14 .24

other words, models trained on JIRA and tested on Stack Overflow perform worse than EmoTxt trained and tested on Stack Overflow. With respect to models trained and tested on Stack Overflow, we cannot determine statistically which classifier is better or worse, as the statistical test did not indicate a difference. All models trained on JIRA are statistically different to models trained on Stack Overflow. This means that models trained on JIRA perform worse, in terms of Subset 0/1 Loss, than models trained on Stack Overflow when tested on the latter.

Post hoc tests did not find statistical differences between using lemmatisation and not, at least when models are tested on the same data set.[13] In those cases where there is statistical difference, the effect sizes were small. Effect sizes lesser than 0.10 mean that differences are trivial in practice or difficult to notice without further analysis. For instance, methods ID = 6 and ID = 5, tested on Stack Overflow, are statistically different ($p_{\text{value}} = 4.87 \times 10^{-2}$) but the effect size is only 0.09.

Table 5 presents evaluation results, in terms of precision, recall and F-score, of models trained and tested on one specific corpus, either JIRA or Stack Overflow. In the JIRA part of Table 5, we observe that all models perform better than EmoTxt, especially in terms of F-Score. Most methods, including EmoTxt, have issues with predicting *fear*, except for HOMER using no lemmatisation and vectors enriched with SenticNet 5. EmoTxt has issues in predicting *love* and *surprise*, whereas for *sadness* and *anger*, it is the most precise. Furthermore, based on the number of null vectors presented in Table 3, EmoTxt shows that it is a conservative tool in general, which overall affects recall. In the Stack Overflow part of Table 5, EmoTxt is the most precise method for almost all emotions. However, EmoTxt achieves low recall, especially for *joy* or *fear*. *Surprise* is hard to predict for all methods, except for HOMER, which can predict some instances. The F-scores can explain why the Macro F-score values in Table 3 are low. Most methods fail to predict correctly at least one emotion, affecting Macro F-scores severely.

9. Discussion

A poor vocabulary intersection between JIRA and Stack Overflow data may be a reason why models trained on JIRA did not perform well when tested on Stack Overflow (see Table 3). We see this reason as not very probable, because the two data sets are from the same domain, software engineering. Another possibility is that people express themselves differently on the two means, although they belong to the same domain. For example, on Stack Overflow, people may be more straightforward and less emotionally expressive, than on JIRA, where discussions can easily get longer.

A further reason could be that annotators may perceive emotions differently. For the Stack Overflow data set, inter-annotator agreement is moderate, as the *Fleiss' Kappa* score ranges between 0.30 and 0.66 for different emotions, with an average of 0.47 [62], which means that the annotation was not easy and so is emotion classification.

Comparing the Instance F-scores and Subset 0/1 Loss values in Table 3, we can determine how precisely classifiers dealt with the emotions, and their multi-label aspect. For example, in Stack Overflow, model ID = 16 predicts emotions more accurately than model ID = 15. However, the latter predicts more instances correctly based on Subset 0/1 Loss values. This indicates that ID = 16 manages the multi-label aspect better, but is less precise than ID = 15 in detecting emotions. Model ID = 13 is best for predicting emotions in JIRA, because it achieved the highest Instance F-score and the lowest Subset 0/1 Loss.

All models fail to predict *fear* and *surprise* correctly because they are the least frequent. The Stack Overflow data set contains more of these instances than JIRA, however, evidently not enough. For solving this issue, we could use a classification algorithm that has been designed for dealing with class imbalance rather than HOMER's and RAkEL's default algorithm, a C4.5 Decision Tree. For instance, *DECOC* (*Diversified Error Correction Output Codes*) [63] is an algorithm that follows the ideas of *Error-Correcting Output Codes* (*ECOC*) [64], i.e. to use multiple combinations of binary classifiers that are merged before the final output, but that uses different weights in order to prioritise minority classes. This weighted approach has shown to perform better than other similar imbalance classification algorithms [63].

Another possible solution for the latter problem, is to make use of an oversampling algorithm, such as *SMOTE* (*Synthetic Minority Over-sampling Technique*) [65]. However, rather than applying it to the whole training data set before passing it to the classifiers, we could embed it into HOMER and RAkEL. Specifically, HOMER and RAkEL generate internally subsets of labels, thus, we could apply SMOTE to oversample the less frequent labels within these label subsets and, in consequence, improve the performance of the classifiers. This approach would be similar to the one proposed in [66], where they embed SMOTE into an *AdaBoost SVM* algorithm to improve the classification of imbalance classes.

To interpret the results of the proposed models and EmoTxt, in Table 6 we manually analysed some incorrectly predicted instances. In example *A*, some classifiers predicted all emotions correctly, while others only predicted some or none. In example *B*, most classifiers predicted *love* instead of *joy*, probably because *love* was assigned to most JIRA instances that contain the word "thanks", e.g. "Wow, fast. Thanks!" or "Thanks, Ashish!". In example *C*, only models trained on Stack Overflow ($ID = 18, 11, 12$) assigned the correct emotion. We believe that this text may be sarcastic, expressing *sadness* or *anger*. In examples *D* and *E* opposite emotions were predicted, e.g. *love* vs. *anger*. Due to the short length of these two examples, it is hard to determine which words or elements activated the emotions. In *F*, some classifiers assigned more than one emotion, sometimes other than the actual annotated emotion. Depending on the message context, wrongly predicted emotions may seem relevant. If the context is criticism on Windows and the praise of Unix, *F* may express sadness or anger, apart from surprise. If *F* is related to a severe Linux bug, it can be sarcastic and, in consequence, may just express surprise.

In example *G*, most classifiers correctly predicted *anger*, but none detected *sadness*. In example *H*, only few classifiers, ($ID = 3, 11, 14$), predicted *anger* correctly. Most classifiers failed because the most representative negative word occurs with an elongated suffix. Out-of-dictionary words affect feature calculations. To address this, elongated words need to be normalised to their original form. Discovering how classifier $ID = 14$ correctly predicted *anger* requires further analysis. However, by observing examples *D*, *H* and *I*, we suspect a bias to the emotion *anger*. Examples *I* and *J* show that classifiers cannot distinguish *love* and *joy* well. Example *K* is incomplete, thus a part of the context is lost. We suspect that the missing text was in code tags, which are frequent in Stack Overflow, and it was removed. Text in code tags should not represent natural language. Finally, example *L* shows how positive words, e.g. *hope*, can mislead emotion prediction.

Related to the last point, it would make sense to integrate a negation management tool, similar to those used in sentiment analysers. It would prevent from predicting opposite emotions, e.g. *fear* instead of *joy*, as in example *L* in Table 6. The integration may be complex as it is hard to compute which emotion are negated. For instance, the phrase "*I'm not afraid of what will happen to GitHub after being purchased by Microsoft.*" is hard to annotate for emotions even for humans. For lexicon-based classifiers, "*afraid*" may trigger *fear*. However, due to the negation it should be handled differently.

Reverting elongated words into their original form can be complicated. It implies detecting which words are truly elongated, e.g. "looove" vs. "issue". Thus, rules about words that contain repeated letters, should be applied. Then, the rules need to be validated against a dictionary. However, elongated words can represent out-of-dictionary words, e.g. superlatives, software names and acronyms, making validation harder. Moreover, disambiguation is needed to determine which word matches the

[13] For observing this pattern in Table 4 use the following coordinates (x, y). For the top-right part: $x = \{x \in \mathbb{N} | x > 1, x \equiv 0 \pmod 2\}$, $y = x + 1$. For the bottom-right part: $x = y + 1$, $y = \{y \in \mathbb{N} | y > 0, y \equiv 1 \pmod 2\}$.

Table 6
Examples of instances, from both testing data sets, on which the classifiers had problem to correctly predict the emotions.

	Example		Actual emotions	Prediction		
				Emotions (ID)	Emotions (ID)	Emotions (ID)
JIRA data set	A	Im stuck with IE6 unfortunately.	Anger, sadness	- (10)	Anger, sadness (13)	Anger (17)
	B	Works for me. Thank you Paulex.	Joy	Love, anger (4)	Love (13)	Love (17)
	C	It will be great if you could confirm it either way first.	Love	Joy (17)	Anger (13)	Love (12)
	D	Looks fine, thank you, George.	Love, joy	Love (13)	Love, anger (14)	Love, joy (17)
	E	My patch wouldn't compile.	Sadness	Love, anger (3)	Anger (13)	- (17)
	F	I wonder why not so many people use Linux?	Surprise	Anger, sadness (5)	Surprise, anger (13)	- (17)
Stack Overflow data set	G	Fails horribly for, e.g., domain = google.co.uk	Anger, sadness	Anger, surprise (13)	Anger (15)	Anger (18)
	H	HOW CAN I BE SUCH A BIG IDIOTT!! but thankx anyways...:D	Love, anger	Anger (14)	- (15)	- (18)
	I	I've been very happy with bulk-loader. We've integrated it with great success.	Joy	Love, sadness, anger (14)	Joy (17)	Love (18)
	J	Whoops, neglected to look at the dates. Oh well, I still think it's a valid answer.	Joy, surprise	Love, anger (11)	Sadness (13)	- (18)
	K	Have a look at . (Except for it being the foundation of their incredibly great IncrediBuild product, I haven't used it, though.)	Love	Sadness (13)	Love (15)	- (18)
	L	OMG I hope it isn't all in one file!	Surprise, fear	Joy, anger (8)	Surprise (15)	Joy (18)

context, in cases such as "os" (operating system) and "oss" (open source software).

EmoTxt performed significantly different than in [11] for some emotions. For models trained and tested on JIRA, the maximum F-score difference is for *joy*, 57.3% vs. 86% in [11]. The reasons remain unclear. The corpus in [11] is smaller, but contains the a similar number of neutral instances. We encountered 362 instances of *joy*, 834 of *love* and 457 of *sadness*, whereas 124, 166 and 302 are respectively declared in [11]. It is also mentioned that the JIRA data set does not contain *fear* or *surprise* instances, contradicting the description of JIRA [48] and our findings. The reported data size in [48] including neutral instances is 5992, whereas we counted 5830. In [11], 4916 instances are declared, of which 4000 are neutral. It is not clear why the data set was truncated and how it was split for training and testing EmoTxt. This may have affected the results.

We also observed F-score differences for all emotions against EmoTxt trained and tested on Stack Overflow in [11]. For *joy*, we obtained 30.4% instead of 77%, 80.4% vs. 92% for *love*, 50.6% vs. 79% for *sadness*, 73.4% vs. 86% for *anger*, 0% vs. 58% for *surprise* and 24% vs. 86% for *fear*. We noticed small differences in the number of instances labelled with each emotion in our corpora and the one used in [11], e.g. 1200 vs. 1220 for *love*, or 106 vs. 104 for *fear*.

Another possible reason for the disagreement regarding EmoTxt performance in this work and in [11] can reside on how the data was split for training and testing. As we focused on the multi-label aspect, we split both data sets considering all labels assigned to an instance, i.e. using stratified sampling. For example, we see the 71 Stack Overflow instances, labelled with *joy-love* (Table 1), as different from instances labelled with *joy* or *love*, only. Thus, our training data contains 56 *joy-love* instances, 904 *love*-only instances and 323 *joy*-only ones. This consideration, along with the moderate inter-annotator agreement, may explain the F-score differences.

The lack of statistical difference between models that used or excluded lemmatisation may suggest that minor feature variations do not affect emotion classification. This may also hold for the number of n-grams and skip-bigrams. Similarly, the lack of statistical difference between using SenticNet 5 or NRC lexica may indicate that, despite the variation of theories or annotations, emotions are represented equally. Moreover, model parameters are tuned using Bayesian Optimisation, maximising performance, despite using varying features.

Two statistically indifferent models may not perform exactly the same. Maybe the test data was not large enough to reveal a difference. However, testing on larger data does not guarantee that the difference will be observable. The effect size may be very small, meaning that in practice, despite a statistical difference, the performance will be similar.

We expected that SenticNet 5 would boost performance, due to its large size and its OSS-related terms, e.g. "memory leak" and "open source". It seems that the number of lexicon entries is less important than their quality and annotation calibre. We may need to represent lexicon information into more complex features.

Investigating the correlation of labels, e.g. *love* and *joy*, can improve multi-label classification performance. This task is not straight-forward for machine learning algorithms. In methods based on Binary Relevance, such as HOMER, label dependency is lost, since each label is considered separately. Methods based on Label Powerset, such as RAkEL, preserve the correlation between labels, as multi-label instances contribute new, composite labels. However, this action can reduce the label density in training data sets, i.e the number of instances decreases with respect to the number of total possible labels [24]. Furthermore, Label Powerset sees the correlation of labels as a conditional dependency, which may not be always true [14].

10. Threats to validity

Trained models may not perform consistently on other data sets. Performance may be affected by domain variations, annotators and diverse annotation guidelines. This has been observed in Section 8, where models trained on a data set do not perform well when tested on another, even in the same domain.

Parameter settings different from the optimised ones may lead to better performance. Bayesian Optimisation was chosen for its excellent balance between speed and quality. Finer models could have been used, at the cost of longer optimisation time. We enriched classification features with good indications of emotions in text, in our view. Different extra features may improve performance.

11. Conclusions and future work

Emotion classification is the task of determining which emotions are expressed in non-neutral text. The task is complex because text can express multiple emotions. The state-of-the-art lacks freely available annotated data and external resources and only offers few classification tools, especially in the domain of Open Source Software (OSS). We explored two multi-label classifiers, i.e. HOMER and RAkEL, and lexica, i.e. SenticNet 5, the NRC Word-Emotion Association Lexicon and the NRC Affect Intensity Lexicon, to develop an emotion classifier for text related to OSS. The classifiers were evaluated on collections of JIRA Issue Tracker comments and Stack Overflow posts. We evaluated against EmoTxt, a state-of-the-art emotion classifier for OSS-related text, a random baseline and the most frequent class one. We used multi-label and single-label metrics, as well as statistical significance testing.

HOMER and RAkEL models outperformed both baselines. They perform statistically better than EmoTxt, when trained and tested on the JIRA data set. The effect size of the performance difference between EmoTxt and the proposed methods is small or small-medium. In general, our models achieved multi-label Micro F-scores, multi-label Macro F-scores and Subset 0/1 Loss up to 81.1%, 59% and 29%, respectively. When trained and tested on Stack Overflow, our models performed similarly to EmoTxt and no statistical difference was found. This means that it is necessary to perform further comparisons to determine a statistical difference, although they may show that the effect size of the difference is imperceptible or trivial. We conclude that using either HOMER or RAkEL does not affect the results significantly. Similarly, the size of lexica did not affect performance either. Thus, maybe the quality of lexica annotations rather than size is key in improving performance.

In the future, we plan to explore other classification methods. For example, we would like to determine how an algorithm such as *Stochastic Gradient Boosting Trees*, which has been observed to perform consistently good on different data sets [67], could behave in multi-label tasks when used alone (with transformed labels) or as the main classifier of RAkEL and HOMER. We contemplate to investigate algorithms such as *Diversified Error Correcting Output Codes* [63] to deal with the imbalance of specific emotions. In addition, we would like to explore how an embedded version of *SMOTE*, such as in [66], could improve the performance of HOMER and RAkEL. We would also experiment with algorithms that have been specifically designed for multi-label tasks, such as FastText [27].

The inclusion of other external resources based on word embedding or other lexica, could help in improving performance. Related to this, the exploration of new extra features for vector enrichment, as well as the inclusion of methods for detecting and managing negations, may improve performance to a great extent. Finally, the use of neural networks along with transfer learning, from another domain, may also contribute to improve the performance of an emotion classifier for texts related to Open Source Software.

CRediT authorship contribution statement

Luis Adrián Cabrera-Diego: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Nik Bessis:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Ioannis Korkontzelos:** Conceptualization, Methodology, Validation, Investigation, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgments

This research has been carried out as part of the *CROSSMINER Project*, which has received funding from the *European Union's Horizon 2020 Research and Innovation Programme* under *Grant Agreement No.* 732223.

References

[1] E. Cambria, A. Livingstone, A. Hussain, The hourglass of emotions, in: A. Esposito, A.M. Esposito, A. Vinciarelli, R. Hoffmann, V.C. Müller (Eds.), Cognitive Behavioural Systems, Springer Berlin Heidelberg, Dresden, Germany, 2012, pp. 144–157, http://dx.doi.org/10.1007/978-3-642-34584-5_11.

[2] A. Yadollahi, A.G. Shahraki, O.R. Zaiane, Current state of text sentiment analysis from opinion to emotion mining, ACM Comput. Surv. 50 (2) (2017) 25:1–25:33, http://dx.doi.org/10.1145/3057270.

[3] P. Ekman, W.V. Friesen, P. Elssworth, Emotion in the Human Face: Guidelines for Research and an Integration of Findings, in: Pergamon General Psychology Series, vol. 11, Pergamon, 1972, http://dx.doi.org/10.1016/C2013-0-02458-9.

[4] R. Plutchik, H. Kellerman, Emotion: Theory, Research, and Experience, in: Theories of Emotion, vol. 1, Academic Press, 1980.

[5] P. Shaver, J. Schwartz, D. Kirson, C. O'Connor, Emotion knowledge: Further exploration of a prototype approach, J. Personal. Soc. Psychol. 52 (6) (1987) 1061–1086, http://dx.doi.org/10.1037/0022-3514.52.6.1061.

[6] N. Gupta, M. Gilbert, G.D. Fabbrizio, Emotion detection in email customer care, Comput. Intell. 29 (3) (2013) 489–505, http://dx.doi.org/10.1111/j.1467-8640.2012.00454.x.

[7] B. Desmet, V. Hoste, Emotion detection in suicide notes, Expert Syst. Appl. 40 (16) (2013) 6351–6358, http://dx.doi.org/10.1016/j.eswa.2013.05.050.

[8] A.M. de Almeida, R. Cerri, E. Cabrera Paraiso, R. Gomes Mantovani, S. Barbon Junior, Applying multi-label techniques in emotion identification of short texts, Neurocomputing 320 (2018) 35–46, http://dx.doi.org/10.1016/j.neucom.2018.08.053.

[9] S. Mohammad, F. Bravo-Marquez, M. Salameh, S. Kiritchenko, SemEval-2018 task 1: Affect in tweets, in: M. Apidianaki, S.M. Mohammad, J. May, E. Shutova, S. Bethard, M. Carpuat (Eds.), Proceedings of the 12th International Workshop on Semantic Evaluation, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 1–17, http://dx.doi.org/10.18653/v1/S18-1001.

[10] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, R. Tonelli, Are bullies more productive?: Empirical study of affectiveness vs. issue fixing time, in: M. Di Penta, M. Pinzger, R. Robbes (Eds.), Proceedings of the 12th Working Conference on Mining Software Repositories, MSR'15, IEEE Press, Florence, Italy, 2015, pp. 303–313, http://dx.doi.org/10.1109/MSR.2015.35.

[11] F. Calefato, F. Lanubile, N. Novielli, EmoTxt: A toolkit for emotion recognition from text, in: 2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos, ACIIW, San Antonio, TX, USA, 2017, pp. 79–80, http://dx.doi.org/10.1109/ACIIW.2017.8272591.

[12] C.O. Alm, D. Roth, R. Sproat, Emotions from text: Machine learning for text-based emotion prediction, in: R.J. Mooney (Ed.), Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT'05, Association for Computational Linguistics, Vancouver, Canada, 2005, pp. 579–586, http://dx.doi.org/10.3115/1220575.1220648.

[13] A. Bagnato, K. Barmpis, N. Bessis, L.A. Cabrera-Diego, J. Di Rocco, D. Di Ruscio, T. Gergely, S. Hansen, D. Kolovos, P. Krief, I. Korkontzelos, S. Laurière, J.M. López de la Fuente, P. Maló, R.F. Paige, D. Spinellis, C. Thomas, J. Vinju, Developer-centric knowledge mining from large open-source software repositories (CROSSMINER), in: M. Seidl, S. Zschaler (Eds.), Software Technologies: Applications and Foundations, STAFF 2017, Springer International Publishing, Marburg, Germany, 2018, pp. 375–384, http://dx.doi.org/10.1007/978-3-319-74730-9_33.

[14] K. Dembczyński, W. Waegeman, W. Cheng, E. Hüllermeier, On label dependence and loss minimization in multi-label classification, Mach. Learn. 88 (1) (2012) 5–45, http://dx.doi.org/10.1007/s10994-012-5285-8.

[15] F. Herrera, F. Charte, A.J. Rivera, M.J. del Jesus, Multilabel Classification: Problem Analysis, Metrics and Techniques, Springer International Publishing, Switzerland, 2016, http://dx.doi.org/10.1007/978-3-319-41111-8.

Please cite this article as: L.A. Cabrera-Diego, N. Bessis and I. Korkontzelos, Classifying emotions in Stack Overflow and JIRA using a multi-label approach, Knowledge-Based Systems (2020) 105633, https://doi.org/10.1016/j.knosys.2020.105633.

[16] G. Tsoumakas, I. Katakis, I. Vlahavas, Effective and efficient multilabel classification in domains with large number of labels, in: Proceedings of ECML/PKDD 2008 Workshop on Mining Multidimensional Data, MMD'08, Antwerp, Belgium, 2008, pp. 53–59.

[17] Y. Papanikolaou, G. Tsoumakas, I. Katakis, Hierarchical partitioning of the output space in multi-label data, Data Knowl. Eng. 116 (2018) 42–60, http://dx.doi.org/10.1016/j.datak.2018.05.003.

[18] G. Tsoumakas, I. Katakis, I. Vlahavas, Random k-labelsets for multilabel classification, IEEE Trans. Knowl. Data Eng. 23 (7) (2011) 1079–1089, http://dx.doi.org/10.1109/TKDE.2010.164.

[19] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (3) (2011) 333, http://dx.doi.org/10.1007/s10994-011-5256-5.

[20] M.-L. Zhang, Z.-H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351, http://dx.doi.org/10.1109/TKDE.2006.162.

[21] H. Blockeel, L.D. Raedt, J. Ramon, Top-down induction of clustering trees, in: J.W. Shavlik (Ed.), Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, Morgan Kaufmann Publishers Inc., Madison, WI, USA, 1998, pp. 55–63.

[22] M. Zhang, A k-nearest neighbor based multi-instance multi-label learning algorithm, in: É. Grégoire (Ed.), Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence, Vol. 2, ICTAI 2010, Arras, France, 2010, pp. 207–212, http://dx.doi.org/10.1109/ICTAI.2010.102.

[23] R.E. Schapire, Y. Singer, BoosTexter: A boosting-based system for text categorization, Mach. Learn. 39 (2) (2000) 135–168, http://dx.doi.org/10.1023/A:1007649029923.

[24] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, Int. J. Data Warehous. Min. 3 (3) (2007) 1–13, http://dx.doi.org/10.4018/jdwm.2007070101.

[25] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, S. Yan, HCP: A flexible CNN framework for multi-label image classification, IEEE Trans. Pattern Anal. Mach. Intell. 38 (9) (2016) 1901–1907, http://dx.doi.org/10.1109/tpami.2015.2491929.

[26] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, W. Xu, CNN-RNN: A unified framework for multi-label image classification, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Las Vegas, Nevada, USA, 2016, pp. 2285–2294, http://dx.doi.org/10.1109/CVPR.2016.251.

[27] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Vol. 2, Valencia, Spain, 2017, pp. 427–431.

[28] J. Nam, J. Kim, E. Loza Mencía, I. Gurevych, J. Fürnkranz, Large-scale multi-label text classification — Revisiting neural networks, in: T. Calders, F. Esposito, E. Hüllermeier, R. Meo (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer Berlin Heidelberg, Nancy, France, 2014, pp. 437–452, http://dx.doi.org/10.1007/978-3-662-44851-9_28.

[29] R. Babbar, B. Schölkopf, Data scarcity, robustness and extreme multi-label classification, Mach. Learn. 108 (8) (2019) 1329–1351, http://dx.doi.org/10.1007/s10994-019-05791-5.

[30] R. You, Z. Zhang, Z. Wang, S. Dai, H. Mamitsuka, S. Zhu, AttentionXML: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification, in: H. Wallach, H. Larochelle, A. Beygelzimer, F.d. Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019, pp. 5812–5822.

[31] F. Gargiulo, S. Silvestri, M. Ciampi, G.D. Pietro, Deep neural network for hierarchical extreme multi-label text classification, Appl. Soft Comput. 79 (2019) 125–138, http://dx.doi.org/10.1016/j.asoc.2019.03.041.

[32] J. Liu, W.-C. Chang, Y. Wu, Y. Yang, Deep learning for extreme multi-label text classification, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Association for Computing Machinery, Shinjuku, Tokyo, Japan, 2017, pp. 115–124, http://dx.doi.org/10.1145/3077136.3080834.

[33] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, I. Vlahavas, Mulan: A Java library for multi-label learning, J. Mach. Learn. Res. 12 (2011) 2411–2414.

[34] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: An update, SIGKDD Explor. Newsl. 11 (1) (2009) 10–18, http://dx.doi.org/10.1145/1656274.1656278.

[35] A. Neviarouskaya, H. Prendinger, M. Ishizuka, Textual affect sensing for sociable and expressive online communication, in: A.C. Paiva, R. Prada, R.W. Picard (Eds.), Affective Computing and Intelligent Interaction, Springer Berlin Heidelberg, Lisbon, Portugal, 2007, pp. 218–229, http://dx.doi.org/10.1007/978-3-540-74889-2_20.

[36] T. Danisman, A. Alpkocak, Feeler: Emotion classification of text using vector space model, in: C. Mellish (Ed.), Proceedings of the AISB 2008 Symposium on Affective Language in Human and Machine, Vol. 2, Aberdeen, Scotland, UK, 2008, pp. 53–59.

[37] C. Strapparava, A. Valitutti, WordNet-affect: an affective extension of wordNet, in: M.T. Lino, M.F. Xavier, F. Ferreira, R. Costa, R. Silva (Eds.), Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC'04, European Language Resources Association (ELRA), Lisbon, Portugal, 2004, pp. 1083–1086.

[38] S.M. Kim, A. Valitutti, R.A. Calvo, Evaluation of unsupervised emotion models to textual affect recognition, in: D. Inkpen, C. Strapparava (Eds.), Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, CAAGET '10, Association for Computational Linguistics, Los Angeles, California, 2010, pp. 62–70.

[39] D.T. Ho, T.H. Cao, A high-order hidden Markov model for emotion detection from textual data, in: D. Richards, B.H. Kang (Eds.), Knowledge Management and Acquisition for Intelligent Systems, Springer Berlin Heidelberg, Kuching, Malaysia, 2012, pp. 94–105, http://dx.doi.org/10.1007/978-3-642-32541-0_8.

[40] K. Luyckx, F. Vaassen, C. Peersman, W. Daelemans, Fine-grained emotion detection in suicide notes: a thresholding approach to multi-label classification, Biomed. Inform. Insights 5 (Suppl. 1) (2012) 61–69, http://dx.doi.org/10.4137/BII.S8966.

[41] E. Cambria, S. Poria, D. Hazarika, K. Kwok, SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18, Association for the Advancement of Artificial Intelligence, New Orleans, LA, USA, 2018, pp. 1795–1802.

[42] C. Strapparava, R. Mihalcea, SemEval-2007 task 14: Affective text, in: E. Agirre, L. Marquez, R. Wicentowski (Eds.), Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval '07, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 70–74.

[43] F.-R. Chaumartin, UPAR7: A knowledge-based system for headline sentiment tagging, in: E. Agirre, L. Marquez, R. Wicentowski (Eds.), Proceedings of the 4th International Workshop on Semantic Evaluations, SemEval '07, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 422–425.

[44] S. Baccianella, A. Esuli, F. Sebastiani, SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in: N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, D. Tapias (Eds.), Proceedings of 7th Language Resources and Evaluation Conference, LREC'10, La Valleta, Malta, 2010, pp. 2200–2204.

[45] C. Baziotis, A. Nikolaos, A. Chronopoulou, A. Kolovou, G. Paraskevopoulos, N. Ellinas, S. Narayanan, A. Potamianos, NTUA-SLP at semEval-2018 task 1: Predicting affective content in tweets with deep attentive RNNs and transfer learning, in: M. Apidianaki, S.M. Mohammad, J. May, E. Shutova, S. Bethard, M. Carpuat (Eds.), Proceedings of the 12th International Workshop on Semantic Evaluation, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 245–255, http://dx.doi.org/10.18653/v1/S18-1037.

[46] S. Rosenthal, N. Farra, P. Nakov, SemEval-2017 task 4: Sentiment analysis in twitter, in: Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval-2017, Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 502–518, http://dx.doi.org/10.18653/v1/S17-2088.

[47] A. Murgia, M. Ortu, P. Tourani, B. Adams, S. Demeyer, An exploratory qualitative and quantitative analysis of emotions in issue report comments of open source systems, Empir. Softw. Eng. 23 (1) (2018) 521–564, http://dx.doi.org/10.1007/s10664-017-9526-0.

[48] M. Ortu, A. Murgia, G. Destefanis, P. Tourani, R. Tonelli, M. Marchesi, B. Adams, The emotional side of software developers in JIRA, in: 2016 IEEE/ACM 13th Working Conference on Mining Software Repositories, MSR, Austin, Texas, USA, 2016, pp. 480–483, http://dx.doi.org/10.1109/MSR.2016.059.

[49] M. Thelwall, K. Buckley, G. Paltoglou, Sentiment strength detection for the social web, J. Am. Soc. Inf. Sci. Technol. 63 (1) (2012) 163–173, http://dx.doi.org/10.1002/asi.21662.

[50] C. Danescu-Niculescu-Mizil, M. Sudhof, D. Jurafsky, J. Leskovec, C. Potts, A computational approach to politeness with application to social factors, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Vol. 1: Long Papers, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 250–259.

[51] N. Novielli, F. Calefato, F. Lanubile, A gold standard for emotion annotation in stack overflow, in: A. Zaidman, Y. Kamei, E. Hill (Eds.), Proceedings of the 15th International Conference on Mining Software Repositories, MSR '18, ACM, Gothenburg, Sweden, 2018, pp. 14–17, http://dx.doi.org/10.1145/3196398.3196453.

[52] F. Calefato, F. Lanubile, N. Novielli, L. Quaranta, EMTk - the emotion mining toolkit, in: 2019 IEEE/ACM 4th International Workshop on Emotion Awareness in Software Engineering, SEmotion, Montréal, Québec, Canada, 2019, pp. 34–37, http://dx.doi.org/10.1109/SEmotion.2019.00014.

[53] K.P. Neupane, K. Cheung, Y. Wang, EmoD: An end-to-end approach for investigating emotion dynamics in software development, in: 2019 IEEE International Conference on Software Maintenance and Evolution, ICSME, Cleveland, Ohio, USA, 2019, pp. 252–256, http://dx.doi.org/10.1109/ICSME.2019.00038.

[54] M.R. Islam, M.F. Zibran, DEVA: Sensing emotions in the valence arousal space in software engineering text, in: Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC'18, ACM, Pau, France, 2018, pp. 1536–1543, http://dx.doi.org/10.1145/3167132.3167296.

[55] M.R. Islam, M.K. Ahmmed, M.F. Zibran, MarValous: Machine learning based detection of emotions in the valence-arousal space in software engineering text, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC '19, ACM, Limassol, Cyprus, 2019, pp. 1786–1793, http://dx.doi.org/10.1145/3297280.3297455.

[56] S.M. Mohammad, P.D. Turney, Crowdsourcing a word-emotion association Lexicon, Comput. Intell. 29 (3) (2012) 436–465, http://dx.doi.org/10.1111/j.1467-8640.2012.00460.x.

[57] S. Mohammad, Word affect intensities, in: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, European Language Resources Association (ELRA), Miyazaki, Japan, 2018, pp. 174–183.

[58] 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, Buenos Aires, Argentina, 2017, pp. 203–214, http://dx.doi.org/10.1109/MSR.2017.9.

[59] J. Močkus, V. Tiešis, A. Žilinskas, The application of Bayesian methods for seeking the extremum, in: G.P. Szegö, L.C.W. Dixon (Eds.), Towards Global Optimisation, Vol. 2, North-Holland, 1978, pp. 117–128.

[60] X.-Z. Wu, Z.-H. Zhou, A unified view of multi-label performance measures, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 70, PMLR, Sydney, Australia, 2017, pp. 3780–3788.

[61] J. Cohen, Statistical Power Analysis for the Behavioral Sciences, second ed., Lawrence Earlbaum Associates, Hillsdale, USA, 1988.

[62] J.R. Landis, G.G. Koch, The measurement of observer agreement for categorical data, Biometrics 33 (1) (1977) 159–174, http://dx.doi.org/10.2307/2529310.

[63] J. Bi, C. Zhang, An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme, Knowl.-Based Syst. 158 (2018) 81–93, http://dx.doi.org/10.1016/j.knosys.2018.05.037.

[64] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artificial Intelligence Res. 2 (1) (1995) 263–286, http://dx.doi.org/10.1613/jair.105.

[65] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (1) (2002) 321–357, http://dx.doi.org/10.1613/jair.953.

[66] J. Sun, H. Li, H. Fujita, B. Fu, W. Ai, Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting, Inf. Fusion 54 (2020) 128–144, http://dx.doi.org/10.1016/j.inffus.2019.07.006.

[67] C. Zhang, C. Liu, X. Zhang, G. Almpanidis, An up-to-date comparison of state-of-the-art classification algorithms, Expert Syst. Appl. 82 (2017) 128–150, http://dx.doi.org/10.1016/j.eswa.2017.04.003.