# DEVA: Sensing Emotions in the Valence Arousal Space in Software Engineering Text

Md Rakibul Islam
University of New Orleans, USA
mislam3@uno.edu

Minhaz F. Zibran
University of New Orleans, USA
zibran@cs.uno.edu

## ABSTRACT

Existing tools for automated sentiment analysis in software engineering text suffer from either or both of two limitations. First, they are developed for non-technical domain and perform poorly when operated on software engineering text. Second, those tools attempt to detect valence only, and cannot capture arousal or individual emotional states such as *excitement, stress, depression,* and *relaxation.*

In this paper, we present the first sentiment analysis tool, DEVA, which is especially designed for software engineering text and also capable of capturing the aforementioned emotional states through the detection of *both* arousal and valence. We also create a ground-truth dataset containing 1,795 JIRA issue comments. From a quantitative evaluation using this dataset, DEVA is found to have more than 82% precision and more than 78% recall.

## CCS CONCEPTS

• **Software and its engineering → Programming teams**;

## KEYWORDS

Emotion; Sentiment; Valence; Arousal; Dictionary; Tool

**ACM Reference format:**
Md Rakibul Islam and Minhaz F. Zibran. 2018. DEVA: Sensing Emotions in the Valence Arousal Space in Software Engineering Text. In *Proceedings of ACM SAC Conference, Pau,France, April 9-13, 2018 (SAC'18),* 8 pages.
https://doi.org/10.1145/3167132.3167296

## 1 INTRODUCTION

Emotions are an inseparable part of human nature, which influence people's activities and interactions, and thus emotions affect task quality, productivity, creativity, group rapport and job satisfaction [14]. Software development being highly dependent on human efforts and interactions, is more susceptible to emotions of the individuals. Hence, a good understanding of the developers' emotions and their influencing factors can be exploited for effective collaborations, task assignments [17], and in devising measures to boost up job satisfaction, which, in turn, can result in increased productivity and projects' success.

Traditional approaches such as, interviews, surveys [55], and biometric measurements [33] for capturing developers' emotions are challenged for software projects involving distributed team settings. Moreover, those approaches in the workplace often make the developers suppress their natural emotional expressions and hinder their normal workflow [27].

Recently, attempts are made to sense emotions of authors from text. In software engineering, attempts are made to detect emotions from textual artifacts including issue comments [11, 15, 20, 24, 25, 30, 42, 45], email contents [19, 51], and forum posts [21, 40].

The techniques for automatic sentiment analysis in text appear to be highly sensitive to domain terms. Thus, the sentiment analysis tools (e.g., `SentiStrength` [49], `NLTK` [4], and `Stanford NLP` [47]), which are designed for general text do not perform well when applied to software engineering text [11, 15, 24, 28, 41, 45, 50, 51] largely due to the variations in meanings of domain-specific technical terms [27]. Hence, recent attempts [6, 10, 12, 27] devise automatic sentiment analysis techniques particularly meant for software engineering text.

All the existing tools are limited in capturing emotions at the necessary depth [41]. Existing approaches are able to detect *valence* (i.e., positivity and negativity of emotional polarities) only and fail to capture *arousal* or specific emotional states such as *excitement, stress, depression,* and *relaxation.* At work, software developers frequently experience these emotions [55], which can be attributed to their work progress. For example, a developer typically feels *relaxed*, if he makes enough progress in his assigned jobs. Otherwise, the developer feels *stressed.* Thus, these emotions need to be identified [36] where the existing approaches fall short [41].

Along this direction, this paper makes the following two contributions:

- We propose techniques realized in a prototype tool for detecting *excitement, stress, depression,* and *relaxation* expressed in software engineering text and thus able to compute both *valence* and *arousal.* Ours is the first tool, particularly crafted for software engineering text, and capable of automatic detection of emotions in *both* valence and arousal space.
- We produce a benchmark dataset consisting of 1,795 JIRA issue comments manually annotated with the four emotional states identified in those comments. This is also the first dataset of its kind.

We name our tool DEVA (**D**etecting **E**motions in **V**alence **A**rousal Space in Software Engineering Text), which includes a lexical approach with a number of heuristics. In empirical evaluations using the aforementioned dataset, DEVA demonstrates 82.19% precision, 78.70% recall, and 80.11% F-score. Both the DEVA tool and the dataset are made freely available online [2].

**Outline:** The rest of the paper is organized as follows. In Section 2, we briefly introduce the model of emotions used in this work.


*SAC'18, April 9-13, 2018, Pau,France*

ACM ISBN 978-1-4503-5191-1/18/04…$15.00
https://doi.org/10.1145/3167132.3167296

In Section 3, we introduce our tool, DEVA. Our approach for capturing arousal is discussed in Section 3.1. The techniques for capturing valence is presented in Section 3.2. A set of heuristics included in DEVA is described in Section 3.4. In Section 4, we describe how we empirically evaluate our tool. In Section 5, we discuss the limitations of this work. Related work is discussed in Section 6. Finally, Section 7 concludes the paper with future research directions.

## 2 EMOTIONAL MODEL

In this work, we use a simple bi-dimensional model [22, 29] of emotions, which is a variant of the dimensional framework, commonly known as VAD (aka PAD) model [46]. In the bi-dimensional model, as shown in Figure 1, the horizontal dimension presents the emotional *polarities* (i.e., positivity, negativity, and neutrality) known as *valence* and the vertical dimension indicates the levels of *reactiveness*, i.e., high and low *arousal*.

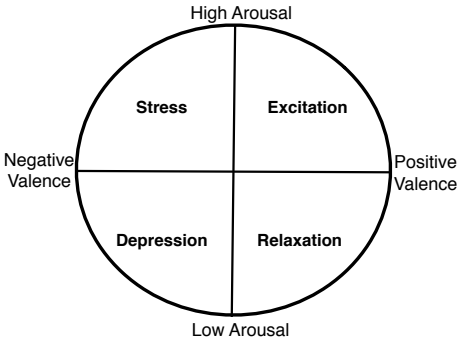


**Figure 1: Simple bi-dimensional model of emotions**

The dimensions are bipolar where the valence dimension ranges from negative to positive and the arousal dimension ranges from low to high. While many emotional states of a person can be determined by combining valence and arousal, we use a set of four major classes of emotional states that include *excitement, stress, depression,* and *relaxation*. For example, positive valence and high arousal, in combination, indicate the emotional state *excitement*. The four emotional states are very distinct, as each state constitutes emotions, which are quite different compared to the emotions of other states [22]. Thus, the model is unequivocal to recognize emotions, simple and easy to understand. This particular emotional model is also used in earlier work [29, 44].

## 3 DEVA

DEVA applies a dictionary-based lexical approach particularly designed for operation on software engineering text. For the capturing *both* arousal and valence, the tool uses two separate dictionaries (an arousal dictionary and a valence dictionary) that we develop by exploiting a general-purpose dictionary and two domain dictionaries especially crafted for software engineering text. DEVA also includes a preprocessing phase and several heuristics. At the preprocessing phase, DEVA identifies and discards source code contents from a given text input using regular expressions similar to what proposed by Bettenburg et al. [9]. The code elements are discarded because they typically are copy-pasted content that do not really carry the writer's emotions [27].

In the following sections, we first describe DEVA's dictionary-based approaches for capturing arousal and valence, and how they are combined to identify different emotional states. Then, we describe the heuristics, which guide the computation of DEVA towards high accuracy.

### 3.1 Capturing Arousal

For capturing arousal, we construct a *new* arousal dictionary for DEVA by combining the SEA (Software Engineering Arousal) [31] dictionary with the ANEW (Affective Norms for English Words) [54] dictionaries.

The SEA [31] dictionary is specifically developed to detect arousal in text in the software developer ecosystems. The dictionary contains 428 words. Each of the 428 words are assigned an arousal *score* $s_a^\omega$, which is a real number between +1 and +9. In this SEA dictionary, the arousal *level* of a word is interpreted as neutral, if $s_a^\omega = +5$. The arousal level of that word is considered high, if $s_a^\omega > +5$. Otherwise, that word is considered to have low arousal.

The ANEW [54] dictionary is a generic dictionary (i.e., not designed especially for any particular domain), which contains 13,915 words where each word is annotated with arousal, valence, and dominance scores, each also ranging between +1 and +9.

*3.1.1 Combining the SEA and ANEW dictionaries.* At first, all the words (along with their arousal scores) in the ANEW dictionary are included in the new arousal dictionary of DEVA. Then, we add any word to the new dictionary if that word is found in the SEA dictionary but not found in the ANEW dictionary. For example, the word 'ASAP' exists in the SEA but not in the ANEW, thus this word along with its arousal score is added to our new arousal dictionary. If a word is found in the both SEA and ANEW dictionaries, then for that word, the arousal score in the SEA dictionary is assigned to the arousal score in our new arousal dictionary. For example, the word 'Anytime' exists in the both dictionaries having the arousal scores 6.5 and 4.6 respectively in the SEA and ANEW dictionaries. Hence, in our new arousal dictionary, the word is assigned an arousal score 6.5. Thus, our newly constructed arousal dictionary includes 14, 084 emotional words.

*3.1.2 Adjusting the ranges of arousal scores.* To obtain an arousal scale consistent with valence scale (described later), first, the fractional value of $s_a^\omega$ is rounded to its nearest integer $\hat{s}_a^\omega$. Then, using the conversion scale in Table 1, we convert each integer arousal score $\hat{s}_a^\omega$ in the range [+1, +9] to $\mathcal{S}_a^\omega$ in the integer range [-5, +5]. For example, if the original arousal score of a word rounded to the closest integer is +2, it is converted to -4, according to the mappings shown in Table 1. For an arousal score $\mathcal{S}_a^\omega$ within the new range of [-5, +5], the arousal *level* $\mathcal{A}_\omega$ of a word $\omega$ is interpreted using Equation 1.

$$\mathcal{A}_\omega = \begin{cases} High, & \text{if } \mathcal{S}_a^\omega > +1 \\ Low, & \text{if } \mathcal{S}_a^\omega < -1 \\ Neutral, & \text{otherwise.} \end{cases} \quad (1)$$

This conversion between ranges does not alter the original arousal *levels* of the words.

**Table 1: Conversion of arousal scores from [+1,+9] to [-5,+5]**

| **Score in [+1,+9]** | +1 | +2 | +3 | +4 | +5 | +6 | +7 | +8 | +9 |
|---|---|---|---|---|---|---|---|---|---|
| **Score in [-5, +5]** | -5 | -4 | -3 | -2 | +/- 1 | +2 | +3 | +4 | +5 |

*3.1.3 Computing arousal score for* text. DEVA views an input text $t$ as a set of words such as $t = \{\omega_1, \omega_2, \omega_3, ..., \omega_n\}$ where $\omega_1, \omega_2, \omega_3, ..., \omega_n$ are distinct words in $t$. In computation of the arousal score for the entire text $t$, DEVA retrieves the arousal scores $\mathcal{S}_a^{\omega_1}, \mathcal{S}_a^{\omega_2}, \mathcal{S}_a^{\omega_3}, ..., \mathcal{S}_a^{\omega_n}$ of all the words in $t$ from the arousal dictionary we have constructed. At this particular stage of computation, a word in $t$ is disregarded if it is not found in the arousal dictionary. Then, for $t$, DEVA computes a pair $\langle h_t, \ell_t \rangle$ where,

$$h_t = \max\{\mathcal{S}_a^{\omega_1}, \mathcal{S}_a^{\omega_2}, \mathcal{S}_a^{\omega_3}, ..., \mathcal{S}_a^{\omega_n}\},$$
$$\ell_t = \min\{\mathcal{S}_a^{\omega_1}, \mathcal{S}_a^{\omega_2}, \mathcal{S}_a^{\omega_3}, ..., \mathcal{S}_a^{\omega_n}\}.$$

Finally, DEVA determines the overall arousal score $\mathcal{A}_t$ for the *entire* text $t$ using Equation 2.

$$\mathcal{A}_t = \begin{cases} h_t, & \text{if } |h_t| \geq |\ell_t| \\ \ell_t, & \text{otherwise.} \end{cases} \tag{2}$$

## 3.2 Capturing Valence

To capture valence in text, DEVA exploits the only available domain-specific valence dictionary named SentiStrength-SE[27], which is especially crafted for software engineering text. This dictionary contains 167 positively and 293 negatively polarized words. Each word $\omega$ is assigned a valence score $\mathcal{S}_v^{\omega}$ where $-5 \leq \mathcal{S}_v^{\omega} \leq +5$. Based on the score $\mathcal{S}_v^{\omega}$, the *polarity* (i.e., positivity, negativity, and neutrality) of valence $\mathcal{V}_{\omega}$ of a word is interpreted using Equation 3.

$$\mathcal{V}_{\omega} = \begin{cases} Positive, & \text{if } \mathcal{S}_v^{\omega} > +1 \\ Negative, & \text{if } \mathcal{S}_v^{\omega} < -1 \\ Neutral, & \text{otherwise.} \end{cases} \tag{3}$$

*3.2.1 Computing valence score for* text. The computation of valence scores for a text is similar to the computation of arousal score, except that the valence dictionary is used in place of the arousal dictionary. Thus, for a given text $t$, DEVA computes a pair $\langle \rho_t, \eta_t \rangle$ of integers, where

$$\rho_t = \max\{\mathcal{S}_v^{\omega_1}, \mathcal{S}_v^{\omega_2}, \mathcal{S}_v^{\omega_3}, ..., \mathcal{S}_v^{\omega_n}\},$$
$$\eta_t = \min\{\mathcal{S}_v^{\omega_1}, \mathcal{S}_v^{\omega_2}, \mathcal{S}_v^{\omega_3}, ..., \mathcal{S}_v^{\omega_n}\}.$$

Here, $\rho_t$ and $\eta_t$ respectively represent the positive and negative valence scores for the text $t$. Finally, the *overall* valence score $\mathcal{V}_t$ for the text $t$ is computed using Equation 4.

$$\mathcal{V}_t = \begin{cases} \rho_t, & \text{if } |\rho_t| \geq |\eta_t| \\ \eta_t, & \text{otherwise.} \end{cases} \tag{4}$$

## 3.3 Emotional States from Valence and Arousal

Upon computing the arousal score $\mathcal{A}_t$ and valence score $\mathcal{V}_t$ for a given text $t$, DEVA then maps the emotional scores to individual emotional states based on the bi-dimensional emotional model described in Section 2. In particular, the emotional state $\mathcal{E}_t$ (of the author) expressed in the text $t$ is determined using the mapping specified in Equation 5.

$$\mathcal{E}_t = \begin{cases} Excitement, & \text{if } \mathcal{V}_t \geq +2 \text{ and } \mathcal{A}_t \geq +2 \\ Stress, & \text{if } \mathcal{V}_t \leq -2 \text{ and } \mathcal{A}_t \geq +2 \\ Depression, & \text{if } \mathcal{V}_t \leq -2 \text{ and } \mathcal{A}_t \leq -2 \\ Relaxation, & \text{if } \mathcal{V}_t \geq +2 \text{ and } \mathcal{A}_t \leq -2 \\ Neutral, & \text{if } \mathcal{V}_t = \pm 1. \end{cases} \tag{5}$$

In addition to the above mentioned emotional states, a text may express only valence and no arousal, or vice versa. DEVA is also able to detect those scenarios in the text.

## 3.4 Heuristics in DEVA

While the underlying dictionaries play the major role in the lexical approach of DEVA, the tool also includes a number of heuristics to increase accuracy, as such was also hinted in earlier work [23, 27]. DEVA includes all the heuristics implemented in SentiStrength-SE [27], which is a recently released tool for the detection of *only valence* in software engineering text. For capturing *arousal* with high accuracy, DEVA also includes seven heuristics, which we have devised based on existing studies in psychology and software engineering [30, 57, 58] as well as our experience in the field. These seven heuristics for sensing *arousal* are discussed below with relevant examples excerpted from a dataset [43] composed of JIRA issue comments (described later in Section 4.1).

**$H_1$ : The exclamation mark (!) in a text implies high arousal.** The exclamation mark (!) in a text is commonly used to indicate high *arousal* of the text writer [7]. For example, in the following comment the commenter expresses *excitement* as the comment contains the word 'happy', which indicates positive *valence*. The three exclamation signs at the end express high *arousal*.

> "Very happy to see it is useful and used !!!."
> (Comment ID: 1927)

Thus, by combining positive *valence* (detected using the valence dictionary) and high arousal (detected using this heuristic $H_1$), DEVA correctly identifies (using Equation 5) the emotional state *excitement* expressed in the above comment. Without the heuristic $H_1$ the text would be incorrectly identified to have positive *valence* only.

**$H_2$ : Words with all capital letters indicate high arousal.** Words written with all capital letters often indicate high arousal state of the writer [7]. In the following comment, the commenter starts the comment with the word 'sorry' expressing negative valence. All capital letters in the word indicates high arousal state of the commenter.

> "SORRY Oliver, this is really my fault ... something like this way will not happen anymore."
> (Comment ID: 1802095)

Hence, DEVA detects negative valence and high arousal in the comment and identifies that the commenter is under *stress*.

However, in cases, such that API names and code elements are written in all capital letters but they do not express any emotional state of the writer. To distinguish such a scenario, DEVA checks the spellings of those words written in all capital letters against an English dictionary using the Jazzy [3] tool. A word written in all capital letters, is considered a name of an API or code element, if the word is misspelled. Thus, in the above comment, the word

**Table 2: Emoticons expressing different emotions**

| Emotions | Code of Emoticons |
|---|---|
| *Excitement* | :*, :?>, :x, :D, :)), 0:), @};- , =P~, :), :"> |
| *Stress* | :((, X-(, :O, =;, :-/ , (:|, >:P |
| *Depression* | :(, :-$, :-&, 8-), :-<, (:|, :-S, I-), :\| |
| *Relaxation* | B-), :>, :P, ;;), ;), =D>, ;)), :-?, [-o<, /:) |

'SOLR' will be identified as a name and DEVA will not interpret it to have expressed any arousal level.

**$H_3$ : Emoticons express emotional states.** Emotional icons, aka, emoticons are often used to express different emotions in informal text including software engineering text [22, 58]. For example, in the following comment the writer uses the emoticon ':(' to express *depression*.

"Oops, I did not run the run-install :("
(Comment ID: 521081)

DEVA is capable of identifying and interpreting emotional states expressed in the emoticons used in text. In interpreting the emoticons, DEVA exploits a list of emoticons mapped to the four categories of emotional states (i.e., *excitement, stress, depression/sadness*, and *relaxation*). The mapping, as presented in Table 2, was originally proposed by Yang et al. [57].

**$H_4$ : Interjections can indicate emotional states**. The interjections are special parts of speech (POS), which are meant for expressing emotional states [16]. For example, even if the above comment (ID: 521081) did not include the emoticon ':(', it would still express *depression* through the interjection 'oops', which DEVA would capture correctly using a list of interjections mapped to their meanings [1] as presented in Table 3.

**Table 3: Interjections expressing different emotions**

| Emotions | Interjections |
|---|---|
| *Excitement* | 'Gee', 'Hurray', 'Ooh', 'Oooh', 'Wee', 'Wow', 'Yah', 'Yeah', 'Yeeeeaah', 'Yeehaw' |
| *Stress* | 'Aah!', 'Aaah', 'Aaaahh', 'Argh', 'Augh', 'Bah', 'Boo', 'Boo!', 'Booh', 'Eek', 'Eep', 'Grr', 'Yikes' |
| *Depression* | 'Duh', 'Doh', 'Eww', 'Gah', 'Humph', 'Harumph', 'Oops', 'Oww', 'Ouch', 'Sheesh', 'Jeez', 'Yick' |
| *Relaxation* | 'Ahh', 'Phew' |

The list of interjections in Table 3, includes only those interjections, whose meanings are unambiguous and relevant to the emotional sates considered in this work. For example, the interjection 'Yahoo' is excluded since it sometimes expresses the name of a company.

**$H_5$ : Temporal terms indicate high arousal.** In psychology and management, it is generally accepted that because of time pressure a worker shows increased alertness or readiness i.e., high arousal [30]. This also applies to the software engineering field [31, 32, 39]. Thus, high arousal is expressed through temporal terms (e.g., asap, soon) in text. For example, in the following comment the commenter expresses high arousal by using the temporal term 'asap'.

"Sorry, I will fix these error **asap**."
(Comment ID: 1600615)

To capture such arousal states expressed through temporal terms, DEVA maintains a list of 12 temporal terms (Table 4) commonly used for referring to timelines, deadlines, or such.

**Table 4: Temporal terms included in the DEVA dictionary**

| |
|---|
| 'Soon', 'Sooner', 'ASAP', 'EOD', 'EOB', 'Today', 'Tomorrow', 'Tonight', "No later", "At earliest", "Tight schedule" |

**$H_6$ : Task completion leads to low arousal.** Typically, completion of a task makes one relaxed and at the state of low *arousal*. For example, the following comment indicating completion of a task also expresses the relaxation of the commenter.

"The two approaches seem complimentary to me. I'm happy to see this committed. Does anyone object?"
(Comment ID: 1667040)

The word 'happy' in the above comment indicate positive valence. But, the arousal state could be missed out if the word 'committed' is not considered to have expressed low arousal. DEVA takes into account both the words 'happy' and 'committed' and corrected identifies both positive valence and low arousal jointly mapped to *relaxation*. For capturing the task completion scenarios in software engineering, DEVA uses a collection of domain-specific words and phrases listed in Table 5.

**Table 5: Task completion indication terms in DEVA**

| |
|---|
| 'Fixed', 'Resolved', 'Solved', 'Done', "Patch looks good", "Working fine", "Working good", "Working properly", "Pushed in branch", "Pushed in trunk", 'Committed'. |

**$H_7$ : Negations reverse arousal state.** Generally, a negation (e.g., no, not ) is meant for reversing or weakening the meaning of the word it qualifies.

DEVA weakens the arousal level associated with a word when the word is found negated in text. Thus, *high* arousal level associated with a negated word is weakened to *low* arousal and the *low* arousal of a word is *neutralized*. For example, in the comment below the high arousal word 'worry' (it is also a negative *valence* word) is negated by 'not' and indicates low arousal.

"Lets not worry about this now" (Comment ID: 53698)

Again, in the following comment, the word 'good' is associated with a low arousal level in DEVA's arousal dictionary. Identifying the negation of the word with 'not', DEVA neutralizes the low arousal.

"Agreed, its not good. Improved in 1.1."
(Comment ID: 2263164)

## 4 EVALUATION

The accuracy of emotion detection of DEVA is measured in terms of *precision* ($\wp$), *recall* ($\Re$), and *F-score* ($\exists$) separately computed for each of the target emotional states as described in Section 2 and

formalized in Equation 5. Given a set $\mathcal{I}$ of texts, *precision* $\wp$, *recall* $\Re$, and *F-score* ($\exists$) for a particular emotional state $e$ is calculated as follows:

$$\wp = \frac{|\ \mathcal{I}_e \cap \mathcal{I}'_e\ |}{|\ \mathcal{I}'_e\ |}, \quad \Re = \frac{|\ \mathcal{I}_e \cap \mathcal{I}'_e\ |}{|\ \mathcal{I}_e\ |}, \quad \exists = \frac{2 \times \wp \times \Re}{\wp + \Re},$$

where $e \in \{excitement, stress, depression, relaxation, neutral\}$, $\mathcal{I}_e$ represents the set of texts expressing the emotional state $e$, and $\mathcal{I}'_e$ denotes the set of texts for which DEVA correctly captures the emotional state $e$.

Recall that DEVA is the first tool capable of automatic detection of the aforementioned emotional states in software engineering text, and no dataset is available for empirical evaluation of our tool. Hence, we first create a ground-truth dataset and compute the aforementioned metrics against that. Then we compare DEVA with a baseline approach we also implement. Finally, we compare our tool with a similar (but not identical) tool, TensiStrength [48].

## 4.1 Creation of Ground-Truth Dataset

The considered dataset [43] consists of two million JIRA issue comments over more than 1,000 projects. JIRA[1] is a commercial tool widely used by software developers for describing, tracking, and managing user-stories, bug-reports, feature requests, and other development issues. This dataset has also been used in many studies [27, 30, 37, 38, 42] on the social and emotional aspects of software engineering.

*4.1.1 Construction of a manageable subset.* We want to create a dataset by manually annotating the issue comments with their expressed emotional states. Manual annotation of two million issue comments could be a mammoth task. Hence, to minimize efforts, we create a subset of 2,000 issue comments for manual annotation using some criteria as described below.

The majority of issue comments in the above mentioned dataset are emotionally neutral [43]. Thus, a random selection is likely to include more neutral comments than those with other emotions. To avoid such a possibility, we first use a keyword-based searching method to collect from the original dataset a subset $\mathcal{G}_k$ of 50 thousand comments which are likely to contain *valence* and *arousal*. We use 68 unigram keywords (listed elsewhere [22]) and their 136 synonyms detected using WordNet [35]. The synonyms of a keyword include every synonym of all variations of the keyword with respect to POS. Such a keyword-based searching method is also used in another study [22] for a similar purpose.

Again, from the original dataset, we randomly select another subset $\mathcal{G}_r$ of 100 thousand issue comments. Then we create a set $\mathcal{G}_u$ such that $\mathcal{G}_u = \mathcal{G}_k \cup \mathcal{G}_r$. Then from $\mathcal{G}_u$, we filter out those comments, which have more than 100 letters resulting in another set $\mathcal{G}_{\hat{u}}$ consisting of 110 thousand comments. From the set $\mathcal{G}_{\hat{u}}$, we randomly select 2,000 comments for manual annotation by human raters.

*4.1.2 Manual annotation by human raters.* We employ three human raters (enumerated as A, B, C) for manually annotating the 2,000 issue comments with the emotions (i.e., *excitement, stress, depression*, *relaxation*, or *neutral*) they perceive in them. Each of these three human raters are graduate students in computer science having one to five years experience in software development in collaborative environments. Each of the human raters separately annotate each of the 2,000 issue comments.

[1]https://www.atlassian.com/software/jira

**Table 6: Inter-rater disagreements in categories of emotions**

| Emotions | Inter-rater Disagreements A, B | B, C | C, A | # of Issue Comments |
|---|---|---|---|---|
| *Excitement* | 06.57% | 07.79% | 07.54% | 411 |
| *Stress* | 06.75% | 17.46% | 14.68% | 252 |
| *Depression* | 06.92% | 11.07% | 07.61% | 289 |
| *Relaxation* | 05.72% | 09.69% | 05.72% | 227 |
| *Neutral* | 07.30% | 05.19% | 05.68% | 616 |
| Total number of issue comments: | | | | 1,795 |

We consider a comment conveying the emotional state $e$, if two of the three raters identify the same emotion in it. Total 205 issue comments are discarded since the human raters do not agree on the emotions they perceive in those comments. Thus, our ground-truth dataset ends up containing 1,795 issue comments. The number of issue comments expressing each of the emotional states are presented in the rightmost column of Table 6. This table also presents the emotion-wise percentage of cases where raters disagree. We also measure the degree of inter-raters agreement in terms of *Fleiss-*$\kappa$ [18] value. The obtained *Fleiss-*$\kappa$ value 0.728 signifies substantial agreement among the independent raters.

## 4.2 Measurement of Accuracy

We invoke DEVA to detect the emotional states in each of the 1,795 issue comments in our human-annotated ground-truth dataset. Then, for each of the issue comments, we compare DEVA's detected emotion with the human annotated emotion (i.e., ground-truth). We separately measure precision ($\wp$), recall ($\Re$), and F-score ($\exists$) for DEVA's detection of each of the emotional states, which are presented in the third column (from the left) of Table 7. As presented at the bottom three rows in the same column of the table, across all the emotional states, on average, DEVA achieves 82.19% precision, 78.70% recall, and 80.11% F-score.

## 4.3 Comparison with a Baseline

DEVA is the first tool especially designed for *software engineering text* to detect the emotional states in the bi-directional emotion model encompassing *both* valence and arousal. There exists no such other tool for direct comparison with DEVA. Hence, we implement a baseline approach based on the work of Mäntylä et al. [30] who used the ANEW dictionary to only *study* valence and arousal in software engineering text.

The baseline tool that we implement also exploits the ANEW dictionary. Thus, the baseline tool differs from DEVA in two ways. First, the baseline tool uses the regular ANEW dictionary while DEVA exploits a valence dictionary and an arousal dictionary especially designed for software engineering text. Second, DEVA applies a number of heuristics which are not included in the baseline tool. We want to verify if the crafted dictionaries and heuristics actually contribute to higher accuracy in the detection of emotional states.

**Table 7: Comparison beween DEVA and Baseline**

| Emotions | Metrics | DEVA | Baseline |
|---|---|---|---|
| ***Excitement*** | ℘ | 87.58 | 77.16 |
| | ℜ | 88.86 | 23.72 |
| | ∃ | 88.22 | 36.29 |
| ***Stress*** | ℘ | 72.29 | 48.48 |
| | ℜ | 66.53 | 12.74 |
| | ∃ | 69.29 | 20.18 |
| ***Depression*** | ℘ | 78.01 | 33.77 |
| | ℜ | 76.12 | 61.59 |
| | ∃ | 77.05 | 43.62 |
| ***Relaxation*** | ℘ | 85.63 | 19.63 |
| | ℜ | 65.63 | 66.76 |
| | ∃ | 74.31 | 30.33 |
| ***Neutral*** | ℘ | 87.44 | 72.45 |
| | ℜ | 96.37 | 31.63 |
| | ∃ | 91.69 | 44.03 |
| **Average** | ℘ | 82.19 | 50.30 |
| | ℜ | 78.70 | 39.27 |
| | ∃ | 80.11 | 34.87 |

**Hypothesis:** Upon operating DEVA and the baseline tool on the same software engineering dataset, DEVA must outperform the baseline, if the domain-specific dictionaries and heuristics included in it actually contribute to higher accuracies in the detection of emotional states in software engineering text.

We invoke the baseline tool to detect the emotional states in each of the issue comments in our ground-truth dataset. Then, we compute the precision (℘), recall (ℜ), and F-score (∃) for its detection of each emotional states (i.e., excitement, stress, depression, relaxation, and neutral) as shown in the rightmost column of Table 7. The overall average precision, recall, and F-score across all the emotional states are presented in the bottom three rows of the same column.

As we compare the accuracies of DEVA and the baseline approach in Table 7, our DEVA is found to have outperformed the baseline in all cases by a large margin except for the recall of relaxation where DEVA falls short by only 01.13%. In all cases, DEVA maintains a substantially higher F-score compared to the baseline. In other words, DEVA maintains a balance between precision and recall for each emotional state resulting in higher F-score for all cases. Overall, on average, across all the emotions, DEVA clearly outperforms the baseline.

Thus, the results of comparison imply that our hypothesis holds true, which means the domain-specific dictionaries and heuristics included in DEVA actually contribute to its superior performance.

## 4.4 Comparison with TensiStrength

Recently, TensiStrength [48] is released, which we find somewhat similar to our DEVA because both the tools are capable of detecting *stress* and *relaxation* in text. However, DEVA and TensiStrength are more different than they are similar. First, unlike DEVA, the TensiStrength tool is not especially designed for software engineering text. Second, TensiStrength cannot detect *excitement* and *depression*, which DEVA detects. Nevertheless, we compare TensiStrength's accuracies against those of DEVA in the detection of *stress* and *relaxation* only since these emotional states form a subset of the emotional states DEVA detects.

For a given text $t$, TensiStrength computes a pair $\langle \pi_t, \varsigma_t \rangle$ of integers, where $+1 \leq \pi_t \leq +5$ and $-5 \leq \varsigma_t \leq -1$. Here, $\pi_t$ and $\varsigma_t$ respectively represent the *relaxation* and *stress* scores for the given text $t$. A given text $t$ is considered expressing *relaxation* if $\pi_t > +1$. Similarly, a text is held conveying *stress* when $\varsigma_t < -1$. Besides, a text is considered neutral when the scores for the text appear to be $\langle 1, -1 \rangle$.

**Table 8: Comparison between DEVA and TensiStrength**

| Emotions | Metrics | DEVA | TensiStrength |
|---|---|---|---|
| ***Stress*** | ℘ | **72.29** | 35.70 |
| | ℜ | 66.53 | **92.03** |
| | ∃ | **69.29** | 51.44 |
| ***Relaxation*** | ℘ | **85.63** | 20.58 |
| | ℜ | **65.63** | 62.11 |
| | ∃ | **74.31** | 30.92 |
| ***Neutral*** | ℘ | **87.44** | 82.31 |
| | ℜ | **96.37** | 79.73 |
| | ∃ | **91.69** | 81.00 |
| **Average** | ℘ | **81.79** | 46.20 |
| | ℜ | 76.18 | **77.96** |
| | ∃ | **78.43** | 54.45 |

We execute TensiStrength on the ground-truth dataset. Then, we separately measure the precision (℘), recall (ℜ), and *F-score* (∃) for TensiStrength's detection of each of the three target emotional states (i.e., relaxation, stress, and neutral). Table 8 shows the precision, recall, and F-score of both the tools DEVA and TensiStrength in the detection of *stress*, *relaxation* and *neutral* comments. The overall average precision, recall, and F-score across the target emotional states are presented in the bottom three rows of the table.

As seen in Table 8, DEVA consistently achieves higher precision and F-score in the detection of all the emotional states. The recall of DEVA is also higher in all cases except for recall of *stress*, which affects the comparative overall recall of the tools. Still, DEVA maintains higher overall average F-score.

TensiStrength cannot differentiate between *depression* and *stress*. It cannot distinguish between *excitement* and *relaxation* either. These shortcomings are among the reasons for the tool's lower precision in the detection of *stress* and *relaxation*. For example, in the following comment, the commenter conveys *excitement*, but due to presence of the positive emotional word 'good', TensiStrength incorrectly determines the comment to have expressed *relaxation*.

```
"Good catch ! Will fix it asap."
```
(Comment ID: 1348887)

# 5 THREATS AND LIMITATIONS

From the empirical evaluations, DEVA is found superior to both the baseline approach and TensiStrength. Still, its accuracy is not 100% due to its shortcomings. Although DEVA captures negations very well, it still falls short in handling *complex structures* of negations. In the detection of subtle expressions of emotions in text,

even the human raters are often in disagreements, and DEVA also falls short in capturing them. The tool cannot distinguish irony and sarcasm in text, and fails to correctly identify emotions in such text. Capturing subtle emotional expressions, irony, and sarcasm in text is already recognized as a challenging problem in the area of Natural Language Processing (NLP).

The heuristics and domain-specific dictionaries included in DEVA contribute in correct identification of emotional states as verified in Section 4.3. However, in some cases, the heuristics may mislead the tool, although such cases are relatively rare compared to the common situations. The lists of task completion terms, temporal terms, interjections, and emoticons, included in DEVA, might not be complete to cover all possible scenarios. Similarly, the valence and arousal dictionaries in DEVA might also miss relevant emotional terms. One might question, instead of using the lexical approach for building DEVA's domain-specific dictionaries, if we could adopt any better approach, which could possibly minimize these limitations. However, a recent study [26] reports that, "lexicon-based approaches for dictionary creation work better for sentiment analysis in software engineering text."

One might argue that in construction of DEVA's arousal dictionary, the range conversion of arousal scores from [+1, +9] to [-5, +5] might have altered the original arousal levels of some words. We have considered this possibility and carefully designed the conversion scheme to minimize such possibilities. A random sanity check after the range conversion indicates absence of any such occurrence. The regular expressions used in the preprocessing phase of DEVA for filtering out source code elements in text might not be able to discard all code elements. However, studies show that light-weight regular expressions perform better than other heavyweight approaches (e.g., machine learning, island grammar) for this purpose [8].

Our ground-truth dataset manually annotated by three human raters are subject to human bias, experience, and understanding of the field. However, the human raters being computer science graduate students and having software development experience in collaborative environments limit this threat.

## 6 RELATED WORK

A comprehensive list of the tools and techniques developed and used to detect emotions can be found elsewhere [34, 36, 56]. To maintain relevance, we limit our discussion to only those tools and techniques that are attempted for *software engineering text*.

Earlier research involving sentiment analysis in software engineering text used three tools/toolkits, SentiStrength [49], Stanford NLP [5], and NLTK [4], while SentiStrength is used the most frequently [27]. All of the aforementioned three tools are developed and trained to operate on non-technical text and they do not perform well enough when operated in a technical domain such as software engineering. Domain-specific (e.g., software engineering) technical uses of inherently emotional words seriously mislead the sentiment analyses of those tools [28, 41, 45, 51] and limit their applicability in software engineering area.

Blaz and Becker [10] proposed three almost equally performing lexical methods, a Dictionary Method (DM), a Template Method (TM), and a Hybrid Method (HM) for sentiment analysis in "Brazilian Portuguese" text in IT (Information Technology) job submission tickets. Although their techniques might be suitable for *formally structured* text, those may not perform well in dealing with *informal* text frequently used in software engineering artifacts such as commit comments [27]. SentiStrength-SE [27], Senti4SD [12] and SentiCR [6] are three recent tools especially designed to deal with software engineering text. However, all the aforementioned tools and techniques are meant for detecting *valence only* and cannot capture arousal or other emotional states at a deeper level.

To detect emotions in more fine-grained levels, Murgia et al. [37] constructed a machine learning classifier specifically trained to identify six emotions *joy, love, surprise, anger, sad,* and *fear* in issue comments. Similar to their approach, Calefato et al. [13] also developed a toolkit to detect those six emotions. However, neither of these techniques are capable of detecting the emotional states *excitement, stress, depression,* and *relaxation* as captured in the well-established bi-directional emotional model encompassing both *valence* and *arousal* dimensions.

TensiStrength [48] is a recently released tool, which we have compared with our DEVA. As mentioned before, TensiStrength can detect *stress* and *relaxation* from text, but cannot capture *excitement* or *depression*, while DEVA is capable of detecting all of them. Unlike our DEVA, TensiStrength is not especially designed for any particular domain, and thus performs poorly for software engineering text as such is also found in our comparison with DEVA. Mäntylä et al. [30] studied both valence and arousal in software engineering text. For detection valence and arousal they also used a lexical approach, which is not especially designed for software engineering text. Their approach relies on the ANEW (Affective Norms for English Words) dictionary only, whereas DEVA uses two separate valence and arousal dictionaries especially crafted for software engineering text. Although their approach was never realized in a reusable tool, it inspired us in the implementation of the baseline tool that we have compared with DEVA.

## 7 CONCLUSION

In this paper, we have presented DEVA, a tool for automated sentiment analysis in text. DEVA is unique from existing tools in two aspects. First, DEVA is especially crafted for software engineering text. Second, DEVA is capable of detecting both valence and arousal in text and mapping them for capturing individual emotional states (e.g., *excitement, stress, depression, relaxation* and *neutrality*) conforming to a well-established bi-directional emotional model. *None* of the existing sentiment analysis tools have *both* the aforementioned capabilities/properties. DEVA applies a lexical approach with an arousal dictionary and a valence dictionary, both crafted for software engineering text. In addition, DEVA includes a set of heuristics, which help the tool to maintain high accuracy.

For empirical evaluation of DEVA, we have constructed a ground-truth dataset consisting of 1,795 JIRA issue comments, each of which are manually annotated by three human raters. This dataset is also a significant contribution to the community. From a quantitative evaluation using this dataset, DEVA is found to have achieved 82.19% precision and 78.70% recall. We have also implemented a baseline approach and compared against DEVA. A recently released similar

(but not identical) tool TensiStrength is also compared with our DEVA. From the comparisons, DEVA is found substantially superior to both the baseline and TensiStrength.

The current release of DEVA and our ground-truth dataset are freely available [2] for public use. We are aware of the existing limitations of our tool, which we have also discussed in this paper. Addressing all these limitations is within our future plan. In the future releases of DEVA, we will keep enriching the underlying dictionaries and enhancing the heuristics for further improving the tool's accuracy. Using DEVA and its future releases, we will conduct large scale studies of emotional variations and their impacts in software engineering. Moreover, we have plan to extend DEVA for *aspect-oriented* [52, 53] emotion analysis in software engineering text.

## REFERENCES


[1] verified: Aug 2017. *List of interjections.* https://www.vidarholen.net/contents/interjections/.
[2] verified: Dec 2017. *URL for downloading DEVA and Benchmark Dataset.* https://figshare.com/s/277026f0686f7685b79e.
[3] verified: Jan 2017. *Jazzy- The Java Open Source Spell Checker.* http://jazzy.sourceforge.net.
[4] verified: Sept 2017. *Natural Language Toolkit for Sentiment Analysis.* http://www.nltk.org/api/nltk.sentiment.html.
[5] verified: Sept 2017. *Stanford Core NLP Sentiment Annotator.* http://stanfordnlp.github.io/CoreNLP/sentiment.html.
[6] T. Ahmed, A. Bosu, A. Iqbal, and S. Rahimi. 2017. SentiCR: a customized sentiment analysis tool for code review interactions. In *ASE*. 106–111.
[7] T. Allen. 1988. Bulletin Boards of the 21st Century Are Coming of Age. *Smithsonian* 19, 6 (1988), 83–93.
[8] A. Bacchelli, M. Lanza, and R. Robbes. 2010. Linking e-mails and source code artifacts. In *ICSE*. 375–384.
[9] N. Bettenburg, B. Adams, and A. Hassan. 2011. A Lightweight Approach to Uncover Technical Information in Unstructured Data. In *ICPC*. 185–188.
[10] C. Blaz and K. Becker. 2016. Sentiment Analysis in Tickets for IT Support. In *MSR*. 235–246.
[11] F. Calefato and F. Lanubile. 2016. Affective Trust as a Predictor of Successful Collaboration in Distributed Software Projects. In *SEmotion*. 3–5.
[12] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli. 2017. Sentiment Polarity Detection for Software Development. *Empirical Software Engineering* (2017), 1—31.
[13] F. Calefato, F. Lanubile, and N. Novielli. 2017. EmoTxt: A Toolkit for Emotion Recognition from Text. In *ACII*.
[14] M. Choudhury and S. Counts. 2013. Understanding Affect in the Workplace via Social Media. In *CSCW*. 303–316.
[15] S. Chowdhury and A. Hindle. 2016. Characterizing Energy-Aware Software Projects: Are They Different?. In *MSR*. 508–511.
[16] Z. Chuang and Ch. Wu. 2002. Emotion Recognition from Textual Input Using an Emotional Semantic Network. In *ICSLP*.
[17] P. Dewan. 2015. Towards Emotion-Based Collaborative Software Engineering. In *CHASE*. 109–112.
[18] J. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76, 5 (1971), 378.
[19] D. Garcia, M. Zanetti, and F. Schweitzer. 2013. The Role of Emotions in Contributors Activity: A Case Study on the GENTOO Community. In *CCGC*. 410–417.
[20] E. Guzman, D. Azócar, and Y. Li. 2014. Sentiment Analysis of Commit Comments in GitHub: An Empirical Study. In *MSR*. 352–355.
[21] E. Guzman and B. Bruegge. 2013. Towards Emotional Awareness in Software Development Teams. In *ESEC/FSE*. 671–674.
[22] M. Hasan, E. Rundensteiner, and E. Agu. 2014. EMOTEX: Detecting Emotions in Twitter Messages. In *ASE*. 27–31.
[23] C. Hutto and E. Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *WSM*. 216–225.
[24] M. Islam and M. Zibran. 2016. Exploration and Exploitation of Developers' Sentimental Variations in Software Engineering. *Internation Journal of Software Innovation* 4, 4 (2016), 35–55.
[25] M. Islam and M. Zibran. 2016. Towards Understanding and Exploiting Developers' Emotional Variations in Software Engineering. In *SERA*. 185–192.
[26] M. Islam and M. Zibran. 2017. A Comparison of Dictionary Building Methods for Sentiment Analysis in Software Engineering Text. In *ESEM*. 478–479.
[27] M. Islam and M. Zibran. 2017. Leveraging Automated Sentiment Analysis in Software Engineering. In *MSR*. 203–214.
[28] R. Jongeling, S. Datta, and A. Serebrenik. 2015. Choosing Your Weapons: On Sentiment Analysis Tools for Software Engineering Research. In *ICSME*. 531–535.
[29] I. Khan, W. Brinkman, and R. Hierons. 2010. Do moods affect programmers' debug performance? *Cogn. Technol. Work* 13, 4 (2010), 245–258.
[30] M. Mäntylä, B. Adams, G. Destefanis, D. Graziotin, and M. Ortu. 2016. Mining Valence, Arousal, and Dominance – Possibilities for Detecting Burnout and Productivity. In *MSR*. 247–258.
[31] M. Mäntylä, N. Novielli, F. Lanubile, M. Claes, and M. Kuutila. 2017. Bootstrapping a Lexicon for Emotional Arousal in Software Engineering. In *MSR*. 1–5.
[32] M. Mäntylä, K. Petersen, T. Lehtinen, and C. Lassenius. 2014. Time pressure: A controlled experiment of test case development and requirements review. In *ICSE*. 83–94.
[33] D. McDuff, A. Karlson, A. Kapoor, A. Roseway, and M. Czerwinski. 2012. AffectAura: an intelligent system for emotional memory. In *CHI*. 849–858.
[34] W. Medhat, A. Hassan, and H. Korashy. 2014. Sentiment Analysis Algorithms and Applications: A survey. *Ain Shams Eng* 5, 4 (2014), 1093–1113.
[35] G. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (1995), 39–41.
[36] S. Muller and T. Fritz. 2015. Stuck and Frustrated or In Flow and Happy: Sensing Developers' Emotions and Progress. In *ICSE*. 688–699.
[37] A. Murgia, M. Ortu, P. Tourani, and B. Adams. 2017. An exploratory qualitative and quantitative analysis of emotions in issue report comments of open source systems. *Empirical Software Engineering* (2017), 1—44.
[38] A. Murgia, P. Tourani, B. Adams, and M. Ortu. 2014. Do Developers Feel Emotions? An Exploratory Analysis of Emotions in Software Artifacts. In *MSR*. 261–271.
[39] N. Nan and D. Harter. 2009. Impact of budget and schedule pressure on software development cycle time and effort. *IEEE Transactions on Software Engineering* 35, 5 (2009), 624–637.
[40] N. Novielli, F. Calefato, and F. Lanubile. 2014. Towards Discovering the Role of Emotions in Stack Overflow. In *SSE*. 33–40.
[41] N. Novielli, F. Calefato, and F. Lanubile. 2015. The Challenges of Sentiment Detection in the Social Programmer Ecosystem. In *SSE*. 33–40.
[42] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli. 2015. Are bullies more productive? Empirical study of affectiveness vs. issue fixing time. In *MSR*. 303–313.
[43] M. Ortu, A. Murgia, G. Destefanis, P. Tourani, R. Tonelli, M. Marchesi, and B. Adams. 2016. The Emotional Side of Software Developers in JIRA. In *MSR*. 480–483.
[44] R. Palacios, C. Lumbreras, P. Acosta, and A. Acosta. 2011. Using the Affect Grid to Measure Emotions in Software Requirements Engineering. *Journal of Universal Computer Science* 17, 9 (2011), 1281–1298.
[45] D. Pletea, B. Vasilescu, and A. Serebrenik. 2014. Security and Emotion: Sentiment Analysis of Security Discussions on GitHub. In *MSR*. 348–351.
[46] J. Russell and A. Mehrabian. 1977. Evidence for a Three-factor Theory of Emotions. *Journal of Research in Personality* 11, 3 (1977), 273–294.
[47] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts. 2013. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank. In *EMNLP*. 1631–1641.
[48] M. Thelwall. 2017. TensiStrength: Stress and relaxation magnitude detection for social media texts. *Information Processing and Management* 53, 1 (2017), 106–121.
[49] M. Thelwall, K. Buckley, and G. Paltoglou. 2012. Sentiment strength detection for the social web. *Journal of the American Society for Info. Science and Tech.* 63(1) (2012), 163–173.
[50] P. Tourani and B. Adams. 2016. The impact of human discussions on just-in-time quality assurance. In *SANER*. 189–200.
[51] P. Tourani, Y. Jiang, and B. Adams. 2014. Monitoring Sentiment in Open Source Mailing Lists – Exploratory Study on the Apache Ecosystem. In *CASCON*. 34–44.
[52] G. Uddin and F. Khomh. 2017. Automatic summarization of API reviews. In *ASE*. 159–170.
[53] G. Uddin and F. Khomh. 2017. Opiner: An opinion search and summarization engine for APIs. In *ASE*. 978–983.
[54] A. Warriner, V. Kuperman, and M. Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 English lemmas. *Behavior research methods* 45, 4 (2013), 1191–1207.
[55] M. Wrobel. 2013. Emotions in the software development process. In *HSI*. 518–523.
[56] A. Yadollahi, A. Shahraki, and O. Zaiane. 2017. Current State of Text Sentiment Analysis from Opinion to Emotion Mining. *Comput. Surveys* 50, 2 (2017), 1–33.
[57] C. Yang, K. Lin, and H. Chen. 2007. Building Emotion Lexicon from Weblog Corpora. In *ACL*. 133–136.
[58] C. Yang, K. Lin, and H. Chen. 2009. Writer Meets Reader: Emotion Analysis of Social Media from both the Writer's and Reader's Perspectives. In *WIIAT*. 287–290.