An Empirical Analysis of the Role of Amplifiers, Downtoners, and Negations in Emotion Classification in Microblogs

Florian Strohm and Roman Klinger
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
70569 Stuttgart, Germany
Email: {roman.klinger,florian.strohm}@ims.uni-stuttgart.de

Abstract—The effect of amplifiers, downtoners, and negations has been studied in general and particularly in the context of sentiment analysis. However, there is only limited work which aims at transferring the results and methods to discrete classes of emotions, e.g., joy, anger, fear, sadness, surprise, and disgust. For instance, it is not straight-forward to interpret which emotion the phrase “not happy” expresses. With this paper, we aim at obtaining a better understanding of such modifiers in the context of emotion-bearing words and their impact on document-level emotion classification, namely, microposts on Twitter. We select an appropriate scope detection method for modifiers of emotion words, incorporate it in a document-level emotion classification model as additional bag of words and show that this approach improves the performance of emotion classification. In addition, we build a term weighting approach based on the different modifiers into a lexical model for the analysis of the semantics of modifiers and their impact on emotion meaning. We show that amplifiers separate emotions expressed with an emotion-bearing word more clearly from other secondary connotations. Downtoners have the opposite effect. In addition, we discuss the meaning of negations of emotion-bearing words. For instance we show empirically that “not happy” is closer to sadness than to anger and that fear-expressing words in the scope of downtoners often express surprise.

Keywords—emotion analysis; modifier detection; downtoner; amplifier; intensifier; negation; social media mining; sentiment analysis; twitter

I. INTRODUCTION

Emotion recognition in text is the task of associating words, phrases or documents with predefined emotions drawn from psychological models [1], [2]. In this paper, we phrase it as single label classification of joy, anger, fear, sadness, surprise, and disgust. It has been applied to, e.g., tales [3], blogs [4], and as a very popular domain, microblogs on Twitter [5]. The latter in particular provides a large source of data in the form of user messages [6], often used with self-assigned classes by the authors, as this can lead to a huge albeit noisy data set [7]. This procedure is often referred to as self-labeling, or, in general, as distant labeling.

Nowadays, state-of-the-art classification models for emotion prediction typically take into account sequential information, for instance with recurrent neural networks or convolutional neural networks [8], [9]. Clearly, these models are able to capture information expressed in phrases, for instance modifications of an emotion phrase, like in “I am slightly unhappy.” However, such models do not allow for obtaining a better semantic and linguistic understanding of the meaning of modifications of emotion expressions per se.

We aim in this paper at getting a better understanding of the impact and use of modifications of emotion words in Twitter. We perform modifier cue detection and subsequently identify their scope. Modifiers are commonly divided into intensifiers (which assign an intensity to a word) and negators (e.g., not), amongst other classes. Intensifiers are further separated into amplifiers (very, entirely, we do not distinguish maximizers and boosters) and downtoners (quite, slightly) [10]. We focus on these three modifiers: negations, amplifiers, and downtoners. From these, negations are most studied and most challenging in interpretation. For instance, “not sad” might express joy, fear, or anger, or none of the above. We will argue later that it is closer to expressing joy than to anger or fear.

Similarly, downtoners might change the prior emotion (i.e., the emotion of a word or phrase without considering context) of an expression. However, we will see that for instance “slightly sad” most likely still expresses the prior emotion sadness but also changes the other emotions which can be expressed by the same sentence at the same time. Intensifications (e.g., “very sad”) seem to be straight-forward in interpretation. We will argue that such formulations separate the prior emotion (sadness) of the word more clearly from a secondary emotion to be predicted (e.g., fear).

This research is similar to analyses of the meaning of negations in the context of sentiment [11]–[13]. However, the degree of freedom for interpretation is increased due to the greater set of classes (emotion categories vs. polarity). The only work in the context of emotions with modifiers we are aware of is by Carillo et al. [14]. They focus on the classification task of sentiment but treat modifiers emotion-specific. In contrast, we aim at classifying emotions particularly to analyze the role of modifiers. More specifically, we (1), select and evaluate an appropriate modifier scope detection method in the context of emotion words on a manually annotated
corpus which we make publicly available\(^1\), and (2), evaluate the impact of the best performing approach in a bag-of-words model showing its value for emotion classification. Finally, (3), as the main contribution, we develop a simple lexical model in which emotion words are weighted differently based on their modifier scope and prior emotion for the purpose of model introspection: The weights serve as a tool to study the meaning of modified emotion words.

II. BACKGROUND AND RELATED WORK

A. Emotion Analysis

Ekman defines joy, anger, fear, sadness, surprise and disgust as the minimal set of six basic emotions that can be differentiated by facial expressions, the set we use in this paper [15]. Plutchik adds anticipation and trust and the concepts of intensity, emotion mixtures and opposing classes to the model, which we analyze empirically here [2].

The first text collection which is nowadays used for emotion classification is the ISEAR corpus of descriptions of emotional events [16]. Alm et al. were the first discussing issues of annotation and prediction of emotions in tales [3]. Aman et al. built classifiers on top of blog posts [4]. Headlines were the subject of analysis in the SemEval competition on affect recognition [17].

Next to these manually built corpora, Wang et al. generated a training corpus by using the so-called self-labeling information provided by authors of tweets with their hashtags [7]. Their results show that the performance of an emotion classification system can be significantly improved with a large amount of data. Similarly, [18] use self-labeling with emoticons and hashtags. The first manually-annotated corpus of tweets for emotion analysis made publicly available was provided by [19], followed by a larger set with a focus on emotion intensity prediction [20]. The corpus by [21] provides multiple annotations of each instance and analyzes interactions between classes. It is a re-annotation of a SemEval corpus for stance detection [22].

B. Modifier Detection for Sentiment and Opinion Analysis

Negations have been extensively studied in different contexts. Chapman et al. use a list of negation cue phrases and assume the scope to include all tokens up to the next punctuation mark or to the next adversative conjunction [23]. Pang et al. include negation detection in a sentiment document classification system [24].

On a more fine-grained level, Councill et al. use a lexicon for negation cue detection and a linear-chain conditional random field for scope recognition, based on part of speech tags and dependency relations [25]. Reitan et al. use a similar approach on a tweet corpus [26]. Jia et al. use rules based on typed dependencies to determine the scope of a negation cue [27].

A straight-forward approach to modify features in a machine learning-based text classifier with negation information is to prepend modified entries in the bag of words (i. e., create an additional bag of modified words in addition to non-modified words, e. g., [24]). For a word-list-based classifier, Polanyi et al. propose to classify a document as positive or negative based on the sum of weights of positive and negative words [28]. Positive words have a weighting of +2, while negative words have a weighting of −2. If a word is negated, its weight is multiplied with −1. If a word is amplified, its weighting is modified additively (to +3 or −3, respectively) and, if it is modified by a downtoner accordingly (to +1 or −1, respectively). Kennedy et al. showed an improvement with this approach on movie review classification [29]. Follow-up work investigated the use of negations and modality in a linguistic experiment and also model multiple negations in the same expression [30]. Taboada et al. discuss lexicon-based methods for sentiment analysis in a broader context [31]. More recent work developed machine-learning-based classifiers to detect speculation and negations particularly for sentiment analysis [32].

We are not aware of any previous work on modifier detection for emotion expressions with the goal of emotion prediction. However, Carillo et al. build a model for sentiment classification in which they learn weights of modifications for an improved polarity prediction [14].

For a more comprehensive overview of previous work in negation and modifier detection in sentiment analysis, we refer to surveys and reviews previously published [11], [33], [34].

III. METHODS

We first aim at showing empirically that handling emotion words specifically with negations, amplifications, and downtoners improves the classification in contrast to a purely word-based model. We describe our modifier cue detection methods (Section III-A), explain the modifier scope detection (Section III-B) and present a simple bag-of-words based method to evaluate the impact of modifier detection (Section III-C).

A. Modifier Cue Detection

We limit ourselves to modifications of emotions, in which the modifier cue \(t\) is explicitly mentioned and build on top of existing modifier lists of negations (e. g., cannot, never, not), amplifiers (e. g., extremely, very, lot), and downtoners (e. g., few, less, rarely, some) and merge them [25], [35]–[39]. For a discussion of implicit emotion detection, we refer the reader to our recent work on the implicit emotion shared task [40]. We do not differentiate maximizers and boosters [41]. To focus our study to those terms which are predominantly used as modifier instead of other meanings, we calculate

\[
r_{mod}^t = \frac{\# \text{used as modifier}}{\# \text{used}}
\]

with mod \(\in\{\text{downtoner}, \text{amplifier}, \text{negation}\}\) and \# denoting the count. We estimate this value on a corpus subsample of 100 tweets for each \(t\). We accept \(t\) as modifier \(\iff r_{mod}^t > 0.5\) to ensure the main role of a term to be a modifier. For instance, we dismissed the amplifier too, as it is used more often in

\(^1\)The data used in this study is available at http://www.ims.uni-stuttgart.de/data/modifieremotion.
As we are specifically interested in the importance and meaning of modifiers on emotion terms (and not on other words), we only take them into account in the predictive models where appropriate. We therefore compare three approaches for modifier scope detection and select the best performing one.

1) Next-n Heuristic: As a combination of previous work for modifier handling, we define maximally \( n \) tokens as the scope which follow the cue up to the next punctuation mark or adversative conjunction [23], [24], [42]. For example, in the tweet “Happiness is not a goal; it is a by-product.” the words “a” and “goal” would be in the negation scope (with any \( n \geq 2 \), but not the words following the semicolon.

2) Dependency Tree Heuristic: We extend the approach by [27] to our set of modifiers and specifically emotion words using the REST and Streaming APIs provided by Twitter. The hashtags are #glad, #happiness, #happy, #joy, #lucky, #luck, and #pleasure for joy, #anger, #hate, #hatred, and #rage for anger, #afraid, #angst, #fear, #panic, #scare, and propagation.

3) Binary SVM: Similarly to a set of submissions to the shared task on negation scope detection [34], our third approach is a classification of tokens with linear support vector machines (SVM). For each modifier, we train one separate model to predict for a candidate token if it is modified or not. We assume that a token cannot be modified twice. The priority of our classifiers is negation detection, then amplifier detection, followed by downtoner detection.

We use features previously proposed [25] (cf. Table I). POS and dependencies are recognized with the Stanford CoreNLP tools. As an example, the features for the word “hate” in Figure 1 are: \textit{Word} = hate, \textit{POS} = VB, \textit{Right Dist.} = 0 (no modifier cue to the right), \textit{Left Dist.} = 3, \textit{Dep Dist.} = 0 (is leaf node), \textit{Dep1 POS} = VB, \textit{Dep1 Dist.} = 1, \textit{Dep2 POS} = null (first order parent is root node), \textit{Dep2 Dist.} = 0.

C. Emotion Classification

The classification task is to assign a tweet to one of the emotions from \textit{joy}, \textit{anger}, \textit{fear}, \textit{sadness}, \textit{surprise}, and \textit{disgust}. Note that we opt for not using a model which can take into account sequential information (e.g., a long short-term memory, a convolutional neural net, an \( n \)-gram model, or nonlinear kernels), because the impact of the modifier detection would be “hidden” in the handling of sequences in general. In contrast, we use a linear support vector machine with only unigram features such that the SVM is not able to capture modification effects itself. With this approach we might not reach highest performance but obtain a model suitable to study modification effects.

IV. EMOTION CLASSIFICATION EXPERIMENTS UNDER CONSIDERATION OF MODIFIERS

In the following Section IV-A, we discuss the corpora used for our evaluation shown in Section IV-B, which shows and discusses the results of our experiments.

A. Corpora

1) Self-Labeling for Emotion Classification: To generate corpora of substantial size, we use a self-labeling approach: we retrieve tweets with specific hashtags for each emotion using the REST and Streaming APIs provided by Twitter. The hashtags are #glad, #happiness, #happy, #joy, #lucky, #luck, and #pleasure for joy, #anger, #hate, #hatred, and #rage for anger, #afraid, #angst, #fear, #panic, #scare, and

<table>
<thead>
<tr>
<th>emotion</th>
<th>TRAIN</th>
<th>TEST</th>
<th>TRAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>joy</td>
<td>597.992</td>
<td>299.028</td>
<td>1.000</td>
</tr>
<tr>
<td>anger</td>
<td>59.591</td>
<td>29.501</td>
<td>1.000</td>
</tr>
<tr>
<td>fear</td>
<td>68.886</td>
<td>34.504</td>
<td>1.000</td>
</tr>
<tr>
<td>sadness</td>
<td>207.026</td>
<td>103.607</td>
<td>1.000</td>
</tr>
<tr>
<td>surprise</td>
<td>24.582</td>
<td>12.483</td>
<td>1.000</td>
</tr>
<tr>
<td>disgust</td>
<td>1.923</td>
<td>877</td>
<td>1.000</td>
</tr>
<tr>
<td>total</td>
<td>960.000</td>
<td>480.000</td>
<td>6.000</td>
</tr>
</tbody>
</table>

TABLE II

<table>
<thead>
<tr>
<th>emotion</th>
<th>TRAIN</th>
<th>TEST</th>
<th>TRAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>joy</td>
<td>597.992</td>
<td>299.028</td>
<td>1.000</td>
</tr>
<tr>
<td>anger</td>
<td>59.591</td>
<td>29.501</td>
<td>1.000</td>
</tr>
<tr>
<td>fear</td>
<td>68.886</td>
<td>34.504</td>
<td>1.000</td>
</tr>
<tr>
<td>sadness</td>
<td>207.026</td>
<td>103.607</td>
<td>1.000</td>
</tr>
<tr>
<td>surprise</td>
<td>24.582</td>
<td>12.483</td>
<td>1.000</td>
</tr>
<tr>
<td>disgust</td>
<td>1.923</td>
<td>877</td>
<td>1.000</td>
</tr>
<tr>
<td>total</td>
<td>960.000</td>
<td>480.000</td>
<td>6.000</td>
</tr>
</tbody>
</table>
I do **not love** but **hate** you very much.

#worry for fear, #bitter, #grief, #misery, #sad, #sadness, and #sorrow for sadness, #surprise and #surprised for surprise, and #disgust for disgust. We assume this hashtags to denote the label of the respective tweets to create a large dataset. We replace hashtags, URLs, and usernames by the same strings, respectively. Table II shows our separation of the crawled data into train and test sets for emotion classification. The corpora TRAINREPR and TESTREPR are uniformly sampled randomly. We use these two corpora to train our emotion classifier and to evaluate the real world performance and modifier impact. Additionally, we create the corpus TRAINdowntown which will be discussed in Section V-A.

2) **Manually-annotated Corpora:** To select the best performing modifier scope detection method and to estimate their performance, we manually annotate a corpus which is also used for the SVM scope detection model training. The annotation is performed by one author of the paper. The task is to categorize pairs of an emotion-bearing word $z_e$ with a modifier word $z_{mod}$ into “$z_{mod}$ modifies $z_e$” or “not”. For instance, Figure 2 visualizes that **not** modifies love and very modifies hate. However, not does not modify hate and very does not modify love. We therefore have four instances with two positive and two negative annotations for two different modifiers and two emotion words.

The resources we create should be valuable also outside of our specific parameter setting. For instance, our selection of dictionary entries cannot be complete. Therefore, in the annotation process, the annotators do not see automatically detected modifiers or automatically recognized emotion terms but need to mark them themselves such that the corpus quality is not decreased by error propagation from preprocessing steps.

Therefore, more specifically, we use three different sampling methods to obtain a corpus densely populated with relevant instances, but not limited to those detected with our resources: Equally-sized subsets are sampled based on the occurrence of (1) both modifier cue and emotion word, (2) only modifier cue, (3) only emotion word. Using different sampling methods enables us to expand our emotion and modifier lexicons with emotion-bearing words and modifier cues found during annotation. We annotate 1,000 tweets resulting in 1,913 modifier-emotion word pairs. Table III summarizes the annotation, split into subcorpora for training the modifier detectors and an evaluation set: The corpus MODEVAL contains one-third of the annotations from each modifier type. We use this corpus to evaluate the performance of the different modifier scope detection approaches. Furthermore, we create three corpora TRAINNEG, TRAINAMP and TRAINDOWN containing the remaining two thirds of annotations for scope detection model training. The table also shows that of all detected modifiers, downtoners are the least common ones.

### B. Results

#### 1) **Modifier Scope Detection:** The results of the selection of parameter $n$ in the **next-$n$ method** (Section III-B1) on the training corpora are shown in Figure 3. The best result is obtained for $n = 2$. This value goes against our expectations,
as [26] detected an average of \( n = 3.8 \) to work best for negations on Twitter. One reason might be that we do not consider the full scope of a modifier but limit our analysis to emotion words only.

Table IV shows the results on modEval for emotion scope detection. The simplest method, the next-2 heuristic, shows best results throughout all modifier types. The performance for downtoners is substantially lower than for negations and amplifiers. The SVM method (83.7 % F\(_1\) for negations) achieves comparable results to the approach by [25] (80 % F\(_1\) on product reviews). The main source of errors for the DepTree approach are errors in the dependency trees because of missing punctuation. The source of errors for the comparably low performance on downtoner scope detection depends on the method. For the next-2 approach, a challenge is that downtoner cues appear more often after the scope. For the DepTree approach, we observe that downtoner cues are more often not a direct child of associated emotion words. Regarding the SVM approach, we presume two main reasons for the limited performance: Firstly, we prioritize negations and amplifiers and secondly, because we have a limited training set for downtoner.

2) Emotion Classification under Consideration of Modifiers: In this paper, we aim at analyzing the impact of negations, amplifiers, downtoners and to understand their contribution to emotion analysis, mainly as a justification to further inspect their role on emotion-bearing words. Therefore, in this result section, we show that our hypothesis that they affect the interpretation of emotion words actually holds.

Table V shows the results for SVM classification on four different subsets of data, namely the full data set for training (TRAINREPR) and testing (TESTREPR) (called “all data” in the table), the subset of data which contains at least one negator, one amplifier, or one downtoner, respectively. For these subsets, only the respective modifier detection is applied.

Altogether, the classifier is best performing on joy, followed by sadness and fear. The modifier detection contributes consistently, though partially only to a limited degree, to all class predictions (on all data for joy with +-1.4, fear with +-1.7, sadness with +2). Most of the improvement originates from an increase in recall when training and testing on all data. When limiting the experiment to different modifiers, we see that this is likely a result of the negation detection, while amplifiers and downtoners contribute partially to precision and partially to recall, depending on the respective emotion.

Inspecting the contribution by modification, we observe the strongest contribution over the model without handling modifications for downtoners, with an improvement of +5.5 percentage points (pp). Here, 14 pp improvement originate from the emotion anger and 18 from disgust.

Across all modifiers, most important is the special handling of fear, with 3.5 pp in negations and 3 pp in amplifiers.

V. Analysis of the Impact of Modifiers in the Context of Emotion Words

We showed in the previous section that modifier detection improves classification in a bag-of-words model. Now we come to the main contribution of this paper, a deeper analysis of the meaning of negators, amplifiers, and downtoners on emotion words.

A. Experimental Setting: Weighted Emotion Lexicon

For this analysis, we extend the work by Polanyi et al. from shifting values in one dimension of polarity according to different modifiers to multiple dimensions, i.e., six fundamental emotions [28]. In addition, instead of proposing a fixed set of weights, we estimate these from data. We use the NRC lexicon for emotion word recognition, similar to lists of positive/negative words [28], [44].

The parameters of the model are represented in four \( 6 \times 6 \) matrices \( W_{\text{no-mod}}, W_{\text{amp}}, W_{\text{down}}, W_{\text{neg}} \). In each matrix, one cell \( w_{ij} \) corresponds to the weight which a word of emotion \( e_j \) in the respective modification scope contributes to the emotion \( e_i \). This data structure is visualized in Figure 4. Input text is represented as four count vectors of length 6 (\( x_{\text{no-mod}}, x_{\text{amp}}, x_{\text{down}}, x_{\text{neg}} \)) of words whose scope contains emotion words of the respective emotion. For instance, \( x_{\text{down},i} \) is the count of downtonned words which belong to \( e_i \). The posterior emotion score vectors resulting from words of specific modification scopes for an input text \( x \) are then

\[
\vec{c} = W^T \times \vec{x}
\]
TABLE V
RESULTS ON TESTREPR CORPUS AND SUBCORPORA LIMITED TO INSTANCES WITH RESPECTIVE MODIFIERS, WITH AND WITHOUT MODIFIER DETECTION. THE SVM BAG-OF-WORDS MODEL IS TRAINED WITH UNIGRAM FEATURES ON TRAINREPR. THE LEXICON WEIGHT MATRICES ARE TRAINED ON TRAIN±.

<table>
<thead>
<tr>
<th>emotion</th>
<th>size</th>
<th>w/o mod. det.</th>
<th>w/ next2 heure.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>all data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>joy</td>
<td>299,028</td>
<td>82.0</td>
<td>94.6</td>
</tr>
<tr>
<td>anger</td>
<td>29,501</td>
<td>68.3</td>
<td>32.2</td>
</tr>
<tr>
<td>fear</td>
<td>34,504</td>
<td>77.4</td>
<td>50.7</td>
</tr>
<tr>
<td>sadness</td>
<td>103,607</td>
<td>74.1</td>
<td>66.6</td>
</tr>
<tr>
<td>surprise</td>
<td>12,483</td>
<td>75.3</td>
<td>32.3</td>
</tr>
<tr>
<td>disgust</td>
<td>877</td>
<td>18.8</td>
<td>3.5</td>
</tr>
<tr>
<td>Macro</td>
<td>480,000</td>
<td>66.0</td>
<td>46.6</td>
</tr>
<tr>
<td>negative</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>joy</td>
<td>22,459</td>
<td>70.5</td>
<td>83.9</td>
</tr>
<tr>
<td>anger</td>
<td>5,866</td>
<td>61.9</td>
<td>35.1</td>
</tr>
<tr>
<td>fear</td>
<td>6,685</td>
<td>75.1</td>
<td>50.0</td>
</tr>
<tr>
<td>sadness</td>
<td>24,299</td>
<td>75.0</td>
<td>79.0</td>
</tr>
<tr>
<td>surprise</td>
<td>1,122</td>
<td>39.8</td>
<td>12.3</td>
</tr>
<tr>
<td>disgust</td>
<td>165</td>
<td>31.3</td>
<td>3.1</td>
</tr>
<tr>
<td>Macro</td>
<td>60,416</td>
<td>58.9</td>
<td>43.9</td>
</tr>
<tr>
<td>amplifier</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>joy</td>
<td>23,622</td>
<td>79.6</td>
<td>90.8</td>
</tr>
<tr>
<td>anger</td>
<td>3,300</td>
<td>61.1</td>
<td>29.9</td>
</tr>
<tr>
<td>fear</td>
<td>3,017</td>
<td>72.6</td>
<td>48.7</td>
</tr>
<tr>
<td>sadness</td>
<td>15,773</td>
<td>76.9</td>
<td>77.0</td>
</tr>
<tr>
<td>surprise</td>
<td>872</td>
<td>50.4</td>
<td>17.5</td>
</tr>
<tr>
<td>disgust</td>
<td>109</td>
<td>28.6</td>
<td>3.7</td>
</tr>
<tr>
<td>Macro</td>
<td>46,693</td>
<td>61.5</td>
<td>44.6</td>
</tr>
<tr>
<td>downtoner</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>joy</td>
<td>7,900</td>
<td>78.2</td>
<td>91.1</td>
</tr>
<tr>
<td>anger</td>
<td>979</td>
<td>51.9</td>
<td>22.0</td>
</tr>
<tr>
<td>fear</td>
<td>980</td>
<td>71.6</td>
<td>43.3</td>
</tr>
<tr>
<td>sadness</td>
<td>4,232</td>
<td>73.5</td>
<td>72.5</td>
</tr>
<tr>
<td>surprise</td>
<td>370</td>
<td>54.3</td>
<td>15.5</td>
</tr>
<tr>
<td>disgust</td>
<td>25</td>
<td>30.0</td>
<td>4.0</td>
</tr>
<tr>
<td>Macro</td>
<td>14,486</td>
<td>63.2</td>
<td>41.4</td>
</tr>
</tbody>
</table>

with mod ∈ {no-mod, amp, down, neg}. The overall emotion score is then the element-wise sum across rows of these vectors. Finally, the decision for an input text is

\[ e(x) = \arg \max_i (e_i(x)) \]

where \( i \) corresponds to one of the basic emotions.

Based on this setting, we optimize the weights on a balanced corpus TRAIN± to further develop an understanding of the meaning of modifiers by model inspection. The weights are not influenced by different training set sizes which would make interpretation difficult. It only includes tweets containing at least one emotion and a modifier word. As optimization paradigm, we use hill climbing and F\(_1\) as the objective function. We do random restarts with initialization of \( w \sim \mathcal{N}(0, 1) \) and take the best matrix from the set of optimization results. The slice \( W_{\text{mod}} \) for each modifier is optimized for \( \approx 120 \) hours, resulting in 28 optimization runs with 2720 epochs on average for the neutral matrix, 49 optimization runs with 1391 epochs for the negative matrix, 53 optimization runs with 1248 epochs for the amplifier matrix and 64 optimization runs with 990 epochs for the downtoner matrix. Weight updates are performed as \( w' = w + r \sim \mathcal{N}(0, 1) \). We stop optimization if no improvement is observed in 500 epochs.

B. Analysis of Weighting Matrices in the Lexical Model

The results of this optimization procedure are shown in Figure 5. We discuss the results based on the following hypotheses: Words outside of modifier scope mainly contribute positively to the emotion classification corresponding to their prior emotion and negatively to emotions of opposing polarity. Words in negation scope contribute to emotions of their opposing polarity or express no emotion. Words in amplifier scope contribute more to emotions of their prior emotion than words outside of modifier scope. Words in downtoner scope contribute less to emotions of their prior emotion than words outside of modifier scope.

a) Emotion words outside of modification scope: The hypothesis is supported by the matrix; i.e., each word of each emotion mainly contributes to their prior emotion with the highest weight for surprise, followed by fear, anger/disgust, and joy (i.e., in a text with joy and fear words, both outside of modifier context, the classification output would be fear). We

\(^2\)We do not report the results of the prediction of this model on independent data as it is outperformed by the SVM classification. Instead, we focus on the analysis of the model parameters in the following.
observe a positive contribution of emotion words to other than their prior emotion for those of same polarity, namely anger to fear (0.8), fear to sadness (0.4), and disgust to sadness\(^3\) (0.2). Contrasting our expectation, surprise contributes to sadness (0.8), disgust (0.6), and joy\(^4\) (0.5), showing that surprise can be divided into positive and negative realizations. The negative contribution of anger words is striking for joy (−3.1) and surprise (−3.0), supporting the second of the hypotheses.

b) Emotion words inside a negation scope: The hypothesis that negated words mainly contribute to emotions of opposite polarity holds for joy to sadness\(^5\) (1.2) and disgust (0.7), and sadness to joy\(^6\) (1.0). However, some emotions do not show this flip in polarity, for instance in fear and surprise. For the class surprise, a reason is that tweets often use comparisons like the phrase “... no party like ...”, with “no party” indicating negated surprise\(^7\). Examples for fear appearing in negated context include those whose authors encourage people not to have fear but still use fear-related hashtags\(^8\). Altogether, the weights are lower than in the matrix for emotion words outside of a modifier scope, backing our hypothesis that partially no emotion is expressed with a negated emotion word.

c) Emotion words in the scope of amplifiers: Most diagonal weights of the amplifier matrix show an increased value in comparison to the matrix for emotion words outside of scope, as hypothesized (for joy by factor 2\(^9\), fear by factor 1.9, sadness by 1.6). For some emotions, in addition to the hypothesis, the amplifier clearly strengthens the non-
occurrence of another emotion: an amplified joy word is a clear signal for non-occurrence of anger (−3.8), while it has nearly no contribution without modification (−0.1). This pattern can also be observed for joy and fear (−1.6 instead of 0.2 without modification) but only to a lesser degree for other emotions. For anger words, the contribution to sadness flips from a negative to a positive contribution. Interestingly, amplified words of fear contribute positively to all emotions.

d) Emotion words in the scope of downtoners: The weights on the diagonal for emotion words in the scope of downtoners is lower than for words out of scope of a modifier, however, higher than for negations. Therefore, downtoners can partially be interpreted as “light version” of negations.\(^10\) However, as expected, they do not flip the polarity. Counter examples are downtoned words associated with fear and their impact on surprise. Most of such tweets contain a phrase similar to “little surprise”, which has a meaning similar to negation. While on average the weights are lower than for other modifications and no modifications, striking is the highest weight in all matrices for sadness contributing negatively to anger (−4.3). A reason could be that practically no tweet occurs in the corpus that contains a downtoned word for sadness and is labeled as anger.

### VI. Summary, Conclusion and Future Work

In this paper, we showed that modifier detection and handling has an impact on the prediction of emotions. This impact differs by emotion and by modifier: the prediction of disgust and anger are most affected by downtoners, while joy and anger are most affected by negations. Amplifications are most relevant to fear. Across all emotions the prediction of surprise and sadness are not that strongly affected.

A deeper look on the impact of negations, amplifiers, and downtoners on separate emotions discloses results which are mostly in line with the models by Plutchik and Russell [45].
Interesting results include that modifiers influence different pairs of emotions to different degrees: highest weights (−3.7) can be observed for disgust–surprise (observation–prediction) without modifiers. Amplifying words denoting surprise, however, does not increase such weights but decreases them – amplifiers separate (some) emotions stronger from all than their prior emotions. This is particularly the case for fear, where the weight increases from 1.9 to 3.6 (without modifier to amplifier). For negations, which are probably the most challenging modifiers to understand emotions, we see the highest (negative) weights for disgust and fear, surprise and anger–“not surprised” definitely does not mean anger, and “not disgusted” definitely does not mean fear. More intuitively are positive weights which are, again, in line with psychological models.

Future work includes more detailed parameter tuning in our models. We made the assumption that a maximal $F_1$ of scope detection is optimal for classification and therefore set $n=2$. However, a different ratio of precision and recall might be beneficial. Therefore, jointly optimizing parameters of emotion scope detection in the downstream task might uncover a different parameter setting.

One source of error in the scope detection are mistakes in the parse tree generation. An evaluation of different parsers and optimizing them for the task at hand might lead to improved performance.

The weight matrices in our lexical model were optimized separately for each modifier. However, we represent them as a 3D tensor already. Therefore, a next step will be a joint optimization of all parameters. We assume that interactions between them might lead to improved results.

Our study is built on top of document-level classification. We propose follow-up studies to investigate the word level and subword level with the use of distributional semantics. In addition, we did not take into account implicit modifications and modifying inflections and derivations. This strain of work will connect our results in this paper to the initiatives of predicting sentiment, emotion and sarcasm, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.


ACKNOWLEDGEMENTS

This research has been partially funded by the German Research Council (DFG), project SEAT (Structured Multi-Domain Emotion Analysis from Text, KL 2869/1-1). We thank Evgeny Kim and Laura Bostan for proof-reading and fruitful discussions.

REFERENCES


