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

---

[1]The data used in this study is available at http://www.ims.uni-stuttgart.de/data/modifieremotion.

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 investaged 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

$$r^t_{\mathrm{mod}} = \frac{\#t \text{ used as modifier}}{\# \text{ used}}$$

with mod $\in \{$downtoner, amplifier, 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 $r^t_{\mathrm{mod}} > 0.5$ 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

TABLE I
FEATURES FOR MODIFIER SCOPE CLASSIFICATION
(PROPOSED BY [25] EXCEPT FOR *).

| Feature	Description
Word	Normalized string of a token.
POS	Part of speech of a token.
Right Dist.	Token distance to the nearest explicit modifier cue in the sentence to the right of a token.
Left Dist.	Token distance to the nearest explicit modifier cue in the sentence to the left of a token.
Dep Dist. *	Minimum number of edges that must be traversed in the dependency tree from a token to an explicit modifier cue.
Dep1 POS	Part of speech of the the first order parent of a token.
Dep1 Dist.	Minimum number of edges that must be traversed in the dependency tree from the first order parent of a token to an explicit modifier cue.
Dep2 POS	Part of speech of the second order parent of a token.
Dep2 Dist.	Minimum number of edges that must be traversed in the dependency tree from the second order parent of a token to an explicit modifier cue.
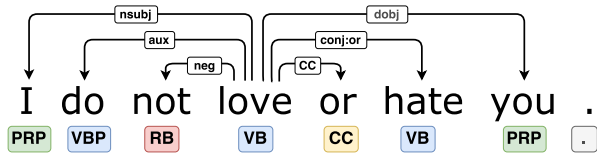
Fig. 1.   Dependency tree example.

a non modifying context. The resulting dictionaries have 39 negation terms, 69 amplifier terms and 36 downtoner terms.

B. Modifier Scope Detection

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 in a heuristic on dependency trees (generated with Stanford CoreNLP 3.7.0, [43]): We flag every parent as modified if its direct child corresponds to a modifier cue. For instance, in Figure 1, "love" is recognized as negated because "not" is in our negation lexicon. To recognize "hate" as being in scope as well, we propagate the modification information along conjunction edges. Adversative conjunctions block this

TABLE II
EMOTION CLASSIFICATION CORPORA.

emotion	TRAINREPR	TESTREPR	TRAIN⚖
joy	597.992	299.028	1.000
anger	59.591	29.501	1.000
fear	68.886	34.504	1.000
sadness	207.026	103.607	1.000
surprise	24.582	12.483	1.000
disgust	1.923	877	1.000
total	960.000	480.000	6.000

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: *Word = hate*, *POS = VB*, *Right Dist. = 0* (no modifier cue to the right), *Left Dist. = 3*, *Dep Dist. = 0* (is leaf node), *Dep1 POS = VB*, *Dep1 Dist. = 1*, *Dep2 POS = null* (first order parent is root node), *Dep2 Dist. = 0*.

C. Emotion Classification

The classification task is to assign a tweet to one of the emotions from *joy*, *anger*, *fear*, *sadness*, *surprise*, and *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 non-linear 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

Modifier	MODEVAL	TRAINNEG	TRAINAMP	TRAINDOWN
negation	315	630	0	0
amplifier	249	0	497	0
downtoner	74	0	0	148
total	638	630	497	148

I do ***not* love** but **hate** you ***very*** much.

Fig. 2.   Annotation example.

Fig. 3.   Different values of n for next-n modifier detection, evaluated on the TRAINNEG, TRAINAMP, TRAINDOWN corpora.

#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 TRAIN⚖ 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
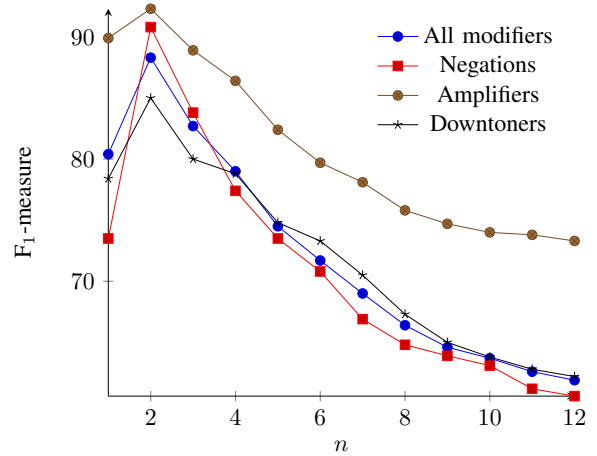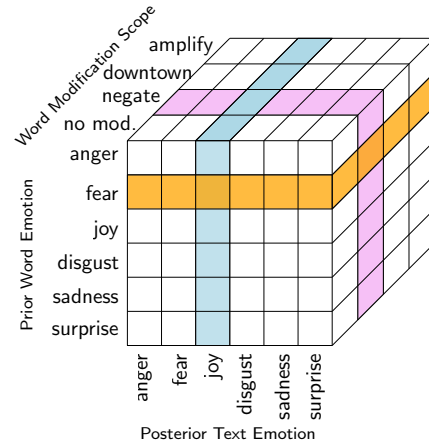

Fig. 4.   Visualization of our four weight matrices for dictionary-based emotion recognition with modifiers. The orange slice corresponds to occurrences of input words with prior emotion fear, the blue slice to output emotion joy and the red slice to input words in the scope of a negation. The cell in all three slices contains the weight such word contributes to the overall classification.

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,

TABLE IV

Modifier	Next-2			DepTree			SVM		
	P	R	F_1	P	R	F_1	P	R	F_1
Negator	**93.6**	87.9	**90.7**	93.0	80.4	86.2	78.7	**89.4**	83.7
Amplifier	**91.7**	**93.7**	**92.7**	90.7	83.0	86.7	91.4	89.4	90.4
Downtoner	72.8	**88.9**	**80.0**	**75.0**	50.0	60.0	66.7	55.6	60.7
Macro-avg.	86.0	**90.2**	**87.8**	**86.3**	71.1	77.7	78.9	78.2	78.3

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.

To achieve that, we test our systems on a uniform subsample from Twitter, namely TRAINREPR/TESTREPR, which has a real-world distribution of modifiers and non-modified emotions. For inclusion of modifier detection, the bag-of-word features of tokens in the scope are prefixed with respective abbreviations (amp, down, neg) and use the next-2-heuristic.

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 $+.3$, *anger*

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 6×6 matrices $W_{\text{no-mod}}$, W_{amp}, W_{down}, W_{neg}. In each matrix, one cell w_{ij} corresponds to the weight which a word of emotion e_i in the respective modification scope contributes to the emotion e_j. This data structure is visualized in Figure 4. Input text is represented as four count vectors of length 6 ($\vec{x}_{\text{no-mod}}$, $\vec{x}_{\text{amp}}$, $\vec{x}_{\text{down}}$ $\vec{x}_{\text{neg}}$) of words whose scope contains emotion words of the respective emotion. For instance, $x_{\text{down},i}$ is the count of downtoned 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

$$\vec{e}_{\text{mod}} = W_{\text{mod}}^T \times \vec{x}_{\text{mod}}$$

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⚖.

| | emotion | size | SVM bag of words | | | | | |
| | | | w/o mod. det. | | | w/ next2 heur. | | |
			P	R	F_1	P	R	F_1
all data	joy	299,028	82.0	94.6	87.9	**83.2**	93.8	**88.2**
	anger	29,501	68.3	32.2	43.7	65.3	**34.5**	**45.1**
	fear	34,504	77.4	50.7	61.3	76.6	**53.5**	**63.0**
	sadness	103,607	74.1	66.6	70.1	72.6	**68.2**	**70.3**
	surprise	12,483	75.3	32.3	45.2	72.3	**33.3**	**45.6**
	disgust	877	18.8	3.5	5.8	17.3	3.2	5.4
	Macro	480,000	66.0	46.6	52.3	64.6	**47.8**	**52.9**
negations	joy	22,459	70.5	83.9	76.7	**72.3**	**85.4**	**78.3**
	anger	5,686	61.9	35.1	44.8	**64.3**	**37.4**	**47.3**
	fear	6,685	75.1	50.0	60.0	70.5	**57.8**	**63.5**
	sadness	24,299	75.0	79.0	77.0	**77.6**	**79.1**	**78.3**
	surprise	1,122	39.8	12.3	18.8	**40.7**	**12.5**	**19.1**
	disgust	165	31.3	3.1	5.6	22.8	3.1	5.4
	Macro	60,416	58.9	43.9	47.2	58.1	**45.9**	**48.7**
amplifiers	joy	23,622	79.6	90.8	84.9	**80.6**	90.0	**85.0**
	anger	3,300	61.1	29.9	40.2	**64.0**	29.2	40.1
	fear	3,017	72.6	48.7	58.3	67.9	**55.8**	**61.3**
	sadness	15,773	76.9	77.0	77.0	76.1	**77.5**	76.8
	surprise	872	50.4	17.5	25.9	**51.4**	16.9	25.4
	disgust	109	28.6	3.7	6.6	23.9	**04.6**	**7.7**
	Macro	46,693	61.5	44.6	48.8	60.7	**45.7**	**49.4**
downtoners	joy	7,900	78.2	91.1	84.1	**79.8**	90.7	**84.9**
	anger	979	51.9	22.0	30.9	**62.6**	**35.0**	**44.9**
	fear	980	71.6	43.3	54.0	63.4	**48.6**	**55.0**
	sadness	4,232	73.5	72.5	73.0	**74.7**	72.1	**73.4**
	surprise	370	54.3	15.5	24.0	46.0	15.2	22.8
	disgust	25	50.0	4.0	7.5	**66.7**	**16.0**	**25.9**
	Macro	14,486	63.2	41.4	45.6	**65.5**	**46.3**	**51.1**

with mod $\in$ {no-mod, amp, down, neg}. The overall emotion score is then the element-wise sum across rows

$$\vec{e}(x) = \sum_{\text{mod}} \vec{e}_{\text{mod}}$$

of these vectors. Finally, the decision for an input text is

$$e(x) = \underset{i}{\operatorname{argmax}}(e_i(x)),$$

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_{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

$w' = w + r$ with $r \sim \mathcal{N}(0,1)$. We stop optimization if no improvement is observed in 500 epochs.[2]

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.

No Modifier

	joy	anger	fear	sadness	surprise	disgust
disgust	0.3	-0.8	-1.9	0.2	-3.7	1.7
surprise	0.5	-3.0	-2.0	0.8	2.0	0.6
sadness	-1.7	0.2	-0.6	0.7	-0.9	0.1
fear	-1.9	-0.1	1.9	0.4	0.4	-0.1
anger	-3.1	1.7	0.8	-0.8	-3.0	-0.8
joy	1.0	-0.1	0.2	-0.2	-0.2	-0.8

Negation

	joy	anger	fear	sadness	surprise	disgust
disgust	0.2	0.1	-0.8	-0.7	-0.0	0.1
surprise	0.3	0.9	-0.9	1.0	2.5	1.8
sadness	1.0	0.5	-0.0	-0.3	-0.7	-0.2
fear	-0.1	-0.0	0.6	-0.5	-1.4	-2.8
anger	-1.3	-0.0	-0.4	-1.0	-2.6	-0.4
joy	-0.3	-0.0	-2.4	1.2	0.9	0.7

Downtoner

	joy	anger	fear	sadness	surprise	disgust
disgust	0.6	-0.1	-0.4	-0.2	-3.4	1.0
surprise	0.3	-0.0	1.0	0.7	1.8	-0.5
sadness	-0.3	-4.3	0.3	0.7	-2.0	-2.4
fear	-2.2	1.8	1.0	0.4	2.3	1.5
anger	1.0	1.5	1.0	0.6	-0.1	0.9
joy	0.8	-3.0	-1.7	-1.5	1.0	-0.7

Amplifier

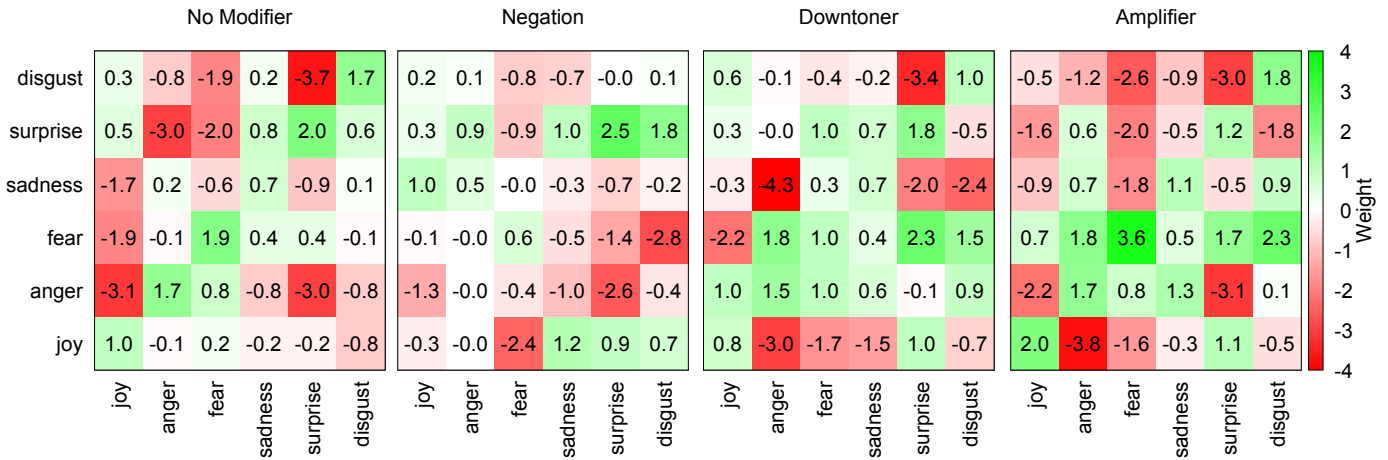|	joy	anger	fear	sadness	surprise	disgust
disgust	-0.5	-1.2	-2.6	-0.9	-3.0	1.8
surprise	-1.6	0.6	-2.0	-0.5	1.2	-1.8
sadness	-0.9	0.7	-1.8	1.1	-0.5	0.9
fear	0.7	1.8	3.6	0.5	1.7	2.3
anger	-2.2	1.7	0.8	1.3	-3.1	0.1
joy	2.0	-3.8	-1.6	-0.3	1.1	-0.5

Fig. 5. Weighting matrices for the lexical model. Columns correspond to the predicted emotion, rows correspond to the prior emotion of the observed words.

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].

[3]Example: "They are 'terrorists' not 'Islamists', you pathetic excuse for a journalist !!!! #hate..."

[4]Example: "Still can't believe my cute baby shower #afternoontea #surprise #ourgirl"

[5]Example: "Not sure how this happened but in two days I've somehow gained 5 lbs...so not happy about this. #ugly #fatty #depressed #sad"

[6]Example: "Yes! I'm about to eat this piece of cheesecake and I don't feel guilty about it. #indulgingalittle #cheesecake #happy"

[7]Example: "Ain't no party like a birthday party when @LJ_Rader shows up #surprise"

[8]Examples: "Don't worry, let God take control. #worry", ""No fear is stronger than you are." - Mark David Gerson #fear #quote #spirituality"

[9]Example: "Wishing you a very happy day! #happiness #positivity"

[10]Example for downtoned sadness with impact on joy: "pray more and worry less #pray #faith #love #peace #happiness...", and vice versa: "Just a bit happy to be back in Ibiza..."

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 the intensities of whole tweets, as shown by Mohammad *et al.* in previous work [20]. In addition, the analysis and comparison with sequence-based classifiers including attention mechanisms will allow for a deeper analysis of end-to-end systems. We assume that it is more challenging to obtain knowledge regarding modifiers from these methods, however, given the work in this paper, we will analyze if our hypotheses also manifest in these approaches.

ACKNOWLEDGEMENTS

REFERENCES

[1] P. Ekman, "Basic emotions," in *Handbook of Cognition and Emotion*, M. Dalgleish, T; Power, Ed. Sussex, UK: John Wiley & Sons, 1999.

[2] R. Plutchik, "The nature of emotions: Human emotions have deep evolutionary roots," *American Scientist*, vol. 89, pp. 344–350, 2001.

[3] C. O. Alm, D. Roth, and R. Sproat, "Emotions from text: Machine learning for text-based emotion prediction," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, October 2005, pp. 579–586. [Online]. Available: http://www.aclweb. org/anthology/H05/H05-1073

[4] S. Aman and S. Szpakowicz, "Identifying expressions of emotion in text," in *Text, Speech and Dialogue: 10th International Conference, TSD 2007, Pilsen, Czech Republic, September 3-7, 2007. Proceedings*, V. Matoušek and P. Mautner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 196–205.

[5] P. S. Dodds, K. D. Harris, I. M. Kloumann, C. A. Bliss, and C. M. Danforth, "Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter," *PLOS ONE*, vol. 6, no. 12, pp. 1–1, 12 2011. [Online]. Available: https://doi.org/10.1371/journal.pone.0026752

[6] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "Concept drift awareness in twitter streams," in *2014 13th International Conference on Machine Learning and Applications*, Dec 2014, pp. 294–299.

[7] W. Wang, L. Chen, K. Thirunarayan, and A. P. Sheth, "Harnessing Twitter Big Data' for Automatic Emotion Identificatio," in *2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust and 2012 ASE/IEEE International Conference on Social Computing, SocialCom/PASSAT 2012*, 2012, pp. 587–592.

[8] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 1615–1625. [Online]. Available: https://www.aclweb.org/anthology/D17-1169

[9] M. Köper, E. Kim, and R. Klinger, "Ims at emoint-2017: Emotion intensity prediction with affective norms, automatically extended resources and deep learning," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 50–57. [Online]. Available: http://www.aclweb.org/anthology/W17-5206

[10] R. Quirk, S. Greenbaum, G. Leech, , and J. Svartvik, *A Comprehensive Grammar of the English Language*. Longman Group Ltd., 1985.

[11] M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A. Montoyo, "A survey on the role of negation in sentiment analysis," in *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*. Uppsala, Sweden: University of Antwerp, July 2010, pp. 60–68. [Online]. Available: http://www.aclweb.org/anthology/W/W10/W10-3111

[12] S. Kiritchenko and S. Mohammad, "The effect of negators, modals, and degree adverbs on sentiment composition," in *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 43–52. [Online]. Available: http://www.aclweb.org/anthology/W16-0410

[13] J. Ruppenhofer, J. Brandes, P. Steiner, and M. Wiegand, "Ordering adverbs by their scaling effect on adjective intensity," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*. INCOMA Ltd. Shoumen, BULGARIA, 2015, pp. 545–554. [Online]. Available: http://www.aclweb.org/anthology/R15-1071

[14] J. CarrillodeAlbornoz and L. Plaza, "An emotion-based model of negation, intensifiers, and modality for polarity and intensity classification," *Journal of the American Society for Information Science and Technology*, vol. 64, no. 8, pp. 1618–1633, 2013.

[15] P. Ekman, "An argument for basic emotions," *Cognition & Emotion*, vol. 6, pp. 169–200, 1992.

[16] K. Scherer and H. Wallbott, "The ISEAR questionnaire and codebook," Geneva Emotion Research Group, 1997.

[17] C. Strapparava and R. Mihalcea, "Semeval-2007 task 14: Affective text," in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 70–74. [Online]. Available: http://www.aclweb.org/anthology/S/S07/S07-1013

[18] M. Purver and S. Battersby, "Experimenting with distant supervision for emotion classification," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France: Association for Computational Linguistics, April 2012, pp. 482–491. [Online]. Available: http://www.aclweb.org/anthology/E12-1049

[19] S. M. Mohammad, X. Zhu, S. Kiritchenko, and J. Martin, "Sentiment, emotion, purpose, and style in electoral tweets," *Information Processing & Management*, vol. 51, no. 4, pp. 480 – 499, 2015.

[20] S. Mohammad and F. Bravo-Marquez, "Wassa-2017 shared task on emotion intensity," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 34–49. [Online]. Available: http://www.aclweb.org/anthology/W17-5205

[21] H. Schuff, J. Barnes, J. Mohme, S. Padó, and R. Klinger, "Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 13–23. [Online]. Available: http://www.aclweb.org/anthology/W17-5203

[22] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 31–41. [Online]. Available: http://www.aclweb.org/anthology/S16-1003

[23] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan, "A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries," *Journal of Biomedical Informatics*, vol. 34, pp. 301–310, 2001.

[24] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, July 2002, pp. 79–86. [Online]. Available: http://www.aclweb.org/anthology/W02-1011

[25] I. Councill, R. McDonald, and L. Velikovich, "What's great and what's not: learning to classify the scope of negation for improved sentiment analysis," in *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*. Uppsala, Sweden: University of Antwerp, July 2010, pp. 51–59. [Online]. Available: http://www.aclweb.org/anthology/W/W10/W10-3110

[26] J. Reitan, J. Faret, B. Gambäck, and L. Bungum, "Negation scope detection for twitter sentiment analysis," in *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Lisboa, Portugal: Association for Computational Linguistics, September 2015, pp. 99–108. [Online]. Available: http://aclweb.org/anthology/W15-2914

[27] L. Jia, C. Yu, and W. Meng, "The effect of negation on sentiment analysis and retrieval effectiveness," *Proceeding of the 18th ACM conference on Information and knowledge management - CIKM '09*, pp. 1827–1830, 2009.

[28] L. Polanyi and A. Zaenen, "Contextual valence shifters," *Computing attitude and affect in text: Theory and Applications*, vol. 20, pp. 1–10, 2006.

[29] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters," *Computational intelligence*, vol. 22, no. 2, pp. 110–125, 2006.

[30] F. Benamara, B. Chardon, Y. Mathieu, V. Popescu, and N. Asher, "How do negation and modality impact on opinions?" in *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics*. Jeju, Republic of Korea: Association for Computational Linguistics, July 2012, pp. 10–18. [Online]. Available: http://www.aclweb.org/anthology/W12-3802

[31] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-Based Methods for Sentiment Analysis," *Computational Linguistics*, vol. 37, pp. 267–307, 2011.

[32] N. P. Cruz, M. Taboada, and R. Mitkov, "A machine-learning approach to negation and speculation detection for sentiment analysis," *Journal of the Association for Information Science and Technology*, vol. 67, no. 9, pp. 2118–2136, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23533

[33] X. Zhu, H. Guo, S. Mohammad, and S. Kiritchenko, "An empirical study on the effect of negation words on sentiment," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 304–313. [Online]. Available: http://www.aclweb.org/anthology/P14-1029

[34] R. Morante and E. Blanco, "*sem 2012 shared task: Resolving the scope and focus of negation," in **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*. Montréal, Canada: Association for Computational Linguistics, 7-8 June 2012, pp. 265–274. [Online]. Available: http://www.aclweb.org/anthology/S12-1035

[35] Wikipedia, "Intensifier – Wikipedia," 2017, last access: 2017-12-10. [Online]. Available: \url{https://en.wikipedia.org/w/index.php?title=Intensifier}

[36] English Club, "Adverbs of degree list," Online: `https:// www. englishclub. com/ vocabulary/adverbs-degree.htm`, 2017, last access: 2017-11-13.

[37] A. J. Thomson and A. V. Martinet, *A practical English grammar*. Oxford University Press, 1986.

[38] S. Romero, "This is so cool! – A Comparative Corpus Study on Intensifiers in British and American English," Pro Gradu Thesis, School of Language, Translation and Literary Studies, University of Tampere, November 2012. [Online]. Available: https://tampub.uta.fi/bitstream/handle/10024/84065/gradu06287.pdf

[39] F. Benamara, C. Cesarano, A. Picariello, D. Reforgiato, and V. S. Subrahmanian, "Sentiment analysis: Adjectives and adverbs are better than adjectives alone," in *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.

[40] R. Klinger, O. de Clercq, S. M. Mohammad, and A. Balahur, "Iest: Wassa-2018 implicit emotions shared task," in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Brussels, Belgium: Association for Computational Linguistics, November 2018.

[41] R. Ito and S. Tagliamonte, "Well weird, right dodgy, very strange, really cool: Layering and recycling in english intensifiers," *Language in Society*, vol. 32, no. 2, pp. 257–279, 2003.

[42] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[43] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010

[44] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word-emotion association lexicon," *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, 2013.

[45] J. Posner, J. A. Russell, and B. S. Peterson, "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology," *Development and psychopathology*, vol. 17, no. 3, pp. 715–734, 2005.