Affective Representations for Sarcasm Detection

Ameeta Agrawal and Aijun An
York University, Toronto, Canada
{ameeta,aan}@cse.yorku.ca

ABSTRACT

Sarcasm detection from text has gained increasing attention. While one thread of research has emphasized the importance of affective content in sarcasm detection, another avenue of research has explored the effectiveness of word representations. In this paper, we introduce a novel model for automated sarcasm detection in text, called Affective Word Embeddings for Sarcasm (AWES), which incorporates affective information into word representations. Extensive evaluation on sarcasm detection on six datasets across three domains of text (tweets, reviews and forum posts) demonstrates the effectiveness of the proposed model. The experimental results indicate that while sentiment affective representations yield best results on datasets comprising of short length text such as tweets, richer representations derived from fine-grained emotions are more suitable for detecting sarcasm from longer length documents such as product reviews and discussion forum posts.

ACM Reference Format:
Ameeta Agrawal and Aijun An. 2018. Affective Representations for Sarcasm Detection. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3209978.3210148

1 INTRODUCTION

Sarcasm and irony [1] are a sophisticated form of symbolic or nonliteral language use where one says or writes the opposite of what they mean. Due to this intentional ambiguity, detecting sarcasm, especially in written communication where the usual cues such as the tone of voice or facial expression are unavailable, is a particularly challenging task. Consider a few examples of sarcastic text utterances presented in Table 1.

Extensive research in psychology points towards a strong correlation between affect and sarcasm [3, 8], and while some recent models of computational sarcasm detection [10, 12, 24, 26] incorporate affective features, the affective information is derived through extensive feature engineering and limited-sized affective resources.

(i) *A little nervous to start school, 5 classes in one day should be fun...*
(ii) *Guess what they said??? we dont replace phones with "physical damage"*
(iii) *Good luck getting this on once you've filled it, and good luck filling it*

Table 1: Examples of sarcastic text

Moreover, the distinct role of sentiment versus emotion spectrums of affect remains unexplored.

While word representations trained from larger corpora of text can overcome the issues of limited training data and manual feature engineering, and provide a wider vocabulary coverage [9, 13], most of these word embeddings have been obtained using only contextual information, without incorporating any affective information.

In this paper, we seek to benefit from bridging the two avenues of research (word representations and affective knowledge) for detecting sarcasm. We propose Affective Word Embeddings for Sarcasm (AWES), a framework for jointly modeling affective as well as contextual information, in order to obtain affectively richer word representations making them more suitable for detecting sarcasm in text. We investigate the use of information stemming from two different spectrums of affect: sentiment and emotion. The proposed model projects words with similar affective orientations into neighboring regions of the embedding space. In particular, to prepare for training affective word embeddings, we use distant supervision to automatically label two large corpora of product reviews with (noisy) sentiment or emotion labels. Then, a Bidirectional Long Short-Term Memory (BLSTM) neural network model is trained using one of the labeled corpora for incorporating affective and contextual information into word representations, where the affective knowledge is derived via the noisy affective labels, and the sequences of words capture the contextual information.

The main contributions of our work include: (i) a framework for learning two types of novel affective word representations[2] (sentiment-aware and emotion-aware) for sarcasm detection; (ii) an extensive evaluation on six benchmark sarcasm datasets across three domains (tweets, product reviews and forum posts); (iii) a novel finding that sentiment-aware representations are most effective for short text sarcasm detection and emotion-aware representations are most effective for detecting sarcasm in longer texts.

2 RELATED WORK

Early work in sarcasm detection from text mainly relied on lexical features such as *n*-grams [15, 16] and syntactic and pattern-based features [4, 28]. Another popular thread of research in sarcasm detection explores the role of affective knowledge, in the form of binary sentiment categories [2, 12, 24] to more fine-grained categories of emotions [5, 10, 14, 23, 26], where the affective information is derived from a handful of limited-sized lexicons which provide limited vocabulary coverage through manual feature engineering which requires considerable time and effort.

[1] In general, verbal irony is often called sarcasm, but in the absence of an agreement among researchers (linguists, psychologists, computer scientists) on the formal definition and structure of sarcasm or irony [7], in this work, we treat sarcasm and irony as similar concepts.

[2] https://www.dropbox.com/s/p46y3icr95b4rsh/awes_emo.txt?dl=0

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210148

Thus, an alternative approach of inducing relevant features involves exploiting text representations learned automatically via neural network models, which help avoid the feature sparsity problem of discrete models [9, 13]. However, most of the existing text representations are obtained using only contextual information, without considering any affective information.

While one recent work [6] obtained word representations from a corpus of noisy labels derived from emojis, our approach is significantly different in several regards. Unlike their approach which uses tweets data, we use multiple affect lexicons to automatically label the data, which allows us to use a corpus of product reviews, which do not usually have emojis but have richer content than tweets. Moreover, while their model is evaluated only on datasets of forum posts, we demonstrate the effectiveness of our affective word representations through a variety of datasets generated from three different domains including tweets, reviews as well as forum posts. Lastly, our text representations learned from a much smaller corpus of 200,000 reviews outperform their representations which were derived from a corpus of 1.5 billion tweets.

3 AFFECTIVE REPRESENTATIONS FOR SARCASM DETECTION

Figure 1 depicts an overview of the proposed framework: creating weakly labeled data through distant supervision (§3.1), affective word representation learning (§3.2) and sarcasm detection (§3.3).

3.1 Weakly Labeled Data

In order to learn affective word embeddings, we require a large corpus of training instances (e.g., sentences or documents such as reviews) along with their corresponding affective labels. Creating large scale manually annotated affective datasets involves many challenges [20]. Therefore, we leverage distant supervision, where data is labeled automatically based on heuristics or rules, to create large training datasets, albeit with noisy annotations.

3.1.1 Data. Our data is extracted from a corpus of Amazon product reviews [17]. All the reviews are tokenized using NLTK preserving punctuation as separate tokens, and reviews containing less than 5 tokens are filtered out.

3.1.2 Affect Labeling. Let $D = \{d_1, d_2, ..., d_{|D|}\}$ denote the set of unlabeled text documents. Given a document $d = \{w_1, w_2, ..., w_{|d|}\}$, $d_i \in D$, consisting of a sequence of words, the goal is to compute a corresponding affect label $l_i \in L$ for d_i, where L denotes a predefined finite set of discrete affect labels.

Primarily, there are two broad categories of affect: *sentiment*, consisting of binary labels such as positive and negative, and *emotion*, involving a more fine-grained spectrum of emotions such as happiness, sadness, anger, and so on. In this work, we seek to assess the effectiveness of both of these models of affect. Consequently, we create two sets of annotated datasets, D^{senti} and D^{emo}, following sentiment and emotion labeling, respectively. In the case of sentiment annotation, the set of labels $L^{sent} = \{positive, negative\}$, $|L| = 2$, whereas, typically, emotion annotation assumes $|L| > 2$.

First, for each word $w_i \in d$, an affect vector $\mathbf{a}(w) = \langle a_1, a_2, ..., a_{|L|} \rangle$ is computed, where a_j indicates the intensity of an affect. Three affect lexicons are leveraged for deriving affect knowledge and computing $\mathbf{a}(w)$. (i) **EmoLex** [20]: For a given word w, the lexicon contains its binary association scores with positive and negative

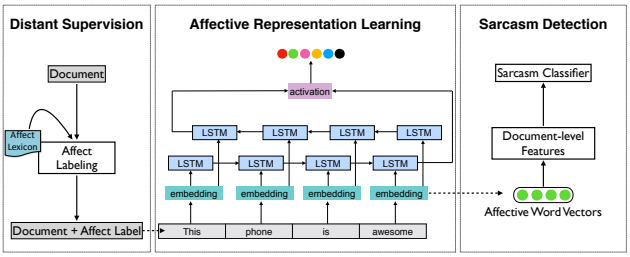

Figure 1: Overview of the proposed framework

sentiment for calculating sentiment label, and scores corresponding to the six categories of Ekman's model for computing emotion label. (ii) **SentiWordNet** (SWN) [1]: Unlike EmoLex, which only marks binary association, the strength of the association in SWN is between the range of 0 and 1. For a given word w, we extract its corresponding sentiment association. (iii) **WordNetAffect** (WNA) [25]: For each of Ekman's six emotions emotion, WNA specifies a list of words associated with that emotion. Here $a_j = 1$ if w is associated with emotion s_j, and $a_j = 0$ otherwise.

The affect vector of a document $\mathbf{a}(d)$ is then, the sum of the affect vectors of all its words, $\mathbf{a}(d) = \sum_i \mathbf{a}(w_i)$. If the document has an association with at least one affect, i.e., $\exists a_j \in \mathbf{a}(d) | a_j > 0$, then, $l = \text{argmax}\, \mathbf{a}(d)$, where $l \in L$. In other words, documents assigned zero affect score (i.e., neutral) are not considered. In case multiple emotion labels have the maximum value, we randomly select one emotion label from the set of labels with the maximum score.

For D^{senti}, the sentiment labels are computed using EmoLex and SWN, while, for D^{emo}, the emotion labels are derived from EmoLex and WNA. In order to mitigate some noise in noisy labels, we constrain the labeling criteria such that only those documents where the labels output by both the lexicons in each case agree, are considered. For each of the datasets, a balanced set of training instances is extracted, with an equal number of documents labeled with each affect. Finally, D^{senti} consists of 400k reviews with $V =$ 445k, whereas D^{emo} comprises of 216k reviews and $V = 183$k.

3.2 Learning Affective Word Representations

Recurrent neural networks such as LSTM can be effectively leveraged to obtain word representations, which are essentially the weights of the connections between the input and the hidden layers. Since both left and right contexts of surrounding words can contain useful contextual information, we consider a Bidirectional LSTM (BLSTM) model [11], which captures the *context* information (left-to-right as well as right-to-left sequence of words) by modeling the long-range dependencies between the words of a text document. The middle of Figure 1 shows an unfolded BLSTM for an input sentence of 4 words. The embeddings are trained together with LSTM for classifying a sentence into its sentiment/emotion category, thus integrating *affective* information.

During the affective representation learning phase, first, all the words of the input text document are converted to their vector representation using an embedding matrix, which is sequentially fed (left-to-right and right-to-left) to the Bidirectional LSTM model. The outputs of the BLSTM are then flattened and connected to the output layer which is used to predict the target label. The objective function minimizes the loss between the predicted and

true labels, and the training error derivatives are backpropagated to the embeddings in the first layer during the training process.

Given a document $d = \{w_1, w_2, ..., w_{|d|}\}$, where w_i denotes a word drawn from a vocabulary $\mathcal{V}$, the goal is to learn an affective word representation $e_i \in \mathbb{R}^n$ for $w_i \in \mathcal{V}$. The full embedding matrix is represented as $E \in \mathbb{R}^{n \times V}$, where V is the size of the vocabulary. We initialize E with pre-trained word vectors from GloVe [22]. We introduce two model architectures of AWES: one for capturing the sentiment information along binary dimensions such as positive and negative (AWES-senti), and the other for encoding a richer spectrum of emotions (AWES-emo).

3.2.1 AWES-senti. Essentially, AWES-senti follows the single-class setting, where an instance belongs to one of two classes (positive and negative). In other words, the target label y is binary represented, where $y = \{0, 1\}$. To predict the sentiment label of the input document, an output layer with a sigmoid activation function, which squashes the inputs into a probability range of $[0, 1]$ for every class, is added to the last layer.

The loss objective over a batch of m documents is calculated via binomial cross-entropy, minimized as follows:

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log\left(y_i'\right) + (1 - y_i)\log\left(1 - y_i'\right)\right] \quad (1)$$

where i denotes the ith training sample, y is the binary representation of true sentiment label, and y' is the predicted probability.

3.2.2 AWES-emo. Assuming an emotion model of k classes, AWES-emo considers the multi-class setting where an instance can belong to one of the k emotion classes. Given an annotated document with its associated emotion label, the target value y is represented as a one-hot vector, where the values of all the indices but one are 0, i.e., if a document d_i is labeled with emotion l_i, then:

$$y_j = \begin{cases} 1, & \text{if } y_j = l_i \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

To predict the emotion label of the input document, an output layer with a softmax activation function which gives a probability distribution over the k classes is added on top of the hidden layer for modeling multi-class probabilities. The softmax function converts the classification result into label probabilities, i.e. $y_t' \in [0, 1]^k$.

The final training objective is to minimize the multinomial cross-entropy loss of the predicted and the true label distributions in order to fit the multi-class emotion labels, where the error is:

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k} y_{ij}\log\left(y_{ij}^{'}\right) \quad (3)$$

where i denotes the ith training sample, j denotes the jth class, y is the true distribution, and y' is the predicted probability distribution, $y_{ij}' \in [0, 1]$ and $\sum_j y_{ij}' = 1$.

3.3 Document Representation for Sarcasm Detection

The purpose of our learning affective word embeddings is sarcasm detection, i.e., to classify a piece of text (e.g., sentence or a paragraph) as sarcasm or non-sarcasm. For such a purpose, we derive a fixed size representation of a document by computing the element-wise minimum, maximum and average along each dimension of all the affective word vectors of all the words in the document. In other words, the final size of document representation is $3 \times n$, where n is the size of word embedding. The intent is to capture the sentiment/emotion variations in a document. Given a set of labeled documents represented by such document vectors plus sarcasm labels, a supervised learning method can be used to learn a sarcasm detection model.

dataset	domain	sarcastic	non-sarcastic	total
SASI-TW	tweets	73	107	180
RILOFF	tweets	112	498	610
ELECT	tweets	938	938	1876
SASI-AM	reviews	67	113	180
FILATOVA	reviews	437	437	874
IAC-SARC	forum posts	1630	1630	3260

Table 2: Statistics of evaluation datasets.

4 EXPERIMENTS

4.1 Datasets

The following six sarcasm datasets (summarized in Table 2) are used for evaluating the proposed model. **(i) SASI-AM** [28] and **(ii) SASI-TW** [4]: The SASI-Amazon (SASI-AM) dataset comprises of 180 sentences from Amazon product reviews annotated by three annotators as sarcastic or non-sarcastic. Similar annotation scheme was followed for creating SASI-Twitter (SASI-TW), a dataset of 180 tweets. **(iii) RILOFF** [24]: This dataset contains sarcasm annotations for tweets. **(iv) ELECT** [19]: This dataset includes crowd-sourced annotations of tweets pertaining to the 2012 US presidential elections. We assume 938 tweets annotated as sarcasm to be sarcastic, and extract 938 simple statements for denoting non-sarcastic instances. **(v) FILATOVA** [7]: This consists of a corpus of reviews, marked as sarcastic or not, using crowdsourcing. **(vi) IAC-SARC** [21]: The sarcasm corpus v2 contains quote-response pairs from a dataset of discussion forum posts. We extract 1,630 responses per class (sarcastic and non-sarcastic).

4.2 Baselines

All the approaches are evaluated using 10-fold cross validation and the results are reported in terms of macro-averaged F-measure over the two classes, *sarcasm* and *non-sarcasm*. The $l2$-regularized logistic regression model implemented in the scikit library is used for classification. In particular, the following baselines are considered: **(i) *n*-grams**: n-grams are one of the most effective features leveraged in sarcasm detection [15, 16]. We implement models exploiting n-grams features including unigrams, bigrams and trigrams, indicating the presence or absence of each n-grams. **(ii) Riloff** [24]: We reimplement their rule-based algorithm where an instance is labeled as sarcastic if it contains both a positive and a negative sentiment term in any order. **(iii) Joshi** [13]: This baseline models sarcasm as a discordance between semantic similarity, obtained via word embeddings, i.e., unigrams, bigrams and trigrams features [15] augmented with similarity features computed from `word2vec` word vectors [18]. **(iv) Word vectors**: Another relevant baseline compares contextual word vectors including **GloVe** [22] trained on a corpus of 42B words from Common Crawl, and **`word2vec`** [18] CBOW model trained on 100B words of Google news data, as well as affective word vectors including **Sentiment Specific Word Embeddings (SSWE)** [27] unified model trained on a corpus of 10M tweets, and **DeepMoji** [6] vectors from 1.5B tweets.

method	short text			long text			average		
	SasiTW	Riloff	Elect	SasiAM	Filatova	IAC	short	long	all
n-grams	0.58	0.54	0.59	0.50	0.67	0.62	0.57	0.60	0.58
Riloff [24]	0.54	0.63	0.48	0.50	0.46	0.43	0.55	0.46	0.51
Joshi [13]	0.58	0.57	0.61	0.52	0.67	0.64	0.59	0.61	0.60
GloVe [22]	0.54	0.75	0.60	0.60	0.72	0.69	0.63	0.67	0.65
word2vec [18]	0.52	0.73	0.60	0.59	0.72	0.70	0.62	0.67	0.64
SSWE [27]	0.59	0.63	0.59	0.57	0.66	0.67	0.60	0.63	0.62
DeepMoji [6]	0.53	0.64	0.60	0.65	0.71	0.71	0.59	**0.69**	0.64
AWES-senti	0.57	0.76	0.62	0.61	0.74	0.70	**0.65**	0.68	**0.67**
AWES-emo	0.55	0.76	0.61	0.64	0.74	0.70	0.64	**0.69**	**0.67**

Table 3: Results (macro F-measure) of sarcasm detection across six datasets.

4.3 Results

The main experimental results summarized in Table 3 indicate a few general observations such as *n*-grams features and all the word vectors methods perform better on long text documents than on short texts, and, on average across all the six datasets, our proposed model (AWES) outperforms all the other baselines.

In order to assess the effectiveness of our proposed approach, we compare the results of AWES against those of four pre-trained word embeddings (GloVe, word2vec, SSWE and DeepMoji). On average, AWES-senti achieves the overall best result on short text documents, while AWES-emo (along with DeepMoji) obtains the best result on long text documents. However, we observe that, on average, the performance of DeepMoji word vectors is worse than all the other word embeddings on short text documents, and while SSWE obtains the highest score on SASI-TW, it falls short on all the remaining five datasets. On the other hand, AWES performs consistently well on *both* short and long text domains.

One of the most interesting observations of this study, however, is as follows: while both DeepMoji and SSWE were trained on short text data (tweets), their performance is strikingly different. DeepMoji, trained on a set of emojis performs better on long text documents, whereas, SSWE, trained on binary sentiment categories is better on short texts. This observation is in line with the performance of AWES, where AWES-senti obtains the best average score on short texts, while AWES-emo achieves the best average score on long texts. In summary, the results suggest that the choice of affect model used to train the word vectors (sentiment versus emotion) is a more distinctive factor than the domain of training data, and, allow us to conclude, albeit counterintuitively to a certain extent, that richer models of emotions do not always lead to additional gains, and that "simpler" models of affect along the axes of positive and negative sentiment are actually better for short text domains.

5 CONCLUSIONS

In this paper, we introduced a novel method for detecting sarcasm in text by leveraging affective word representations obtained from weakly labeled data. In particular, it was observed that sentiment affective word representations are more suitable for short text documents such as tweets, whereas emotion word representations benefit sarcasm detection in long documents such as product reviews and discussion posts.

REFERENCES

[1] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0. In *LREC'10* (19-21). ELRA, Valletta, Malta.

[2] Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling sarcasm in twitter, a novel approach. *ACL 2014* (2014), 50.

[3] John D Campbell and Albert N Katz. 2012. Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes* (2012).

[4] Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised Recognition of Sarcastic Sentences in Twitter and Amazon. In *CONLL '10*. ACL, 10.

[5] Delia Irazú Hernańdez Farías, Viviana Patti, and Paolo Rosso. 2016. Irony Detection in Twitter: The Role of Affective Content. *ACM Transactions on Internet Technology (TOIT)*, Article 19 (2016), 24 pages. https://doi.org/10.1145/2930663

[6] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *EMNLP*.

[7] Elena Filatova. 2012. Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing.. In *LREC*. 392–398.

[8] Ruth Filik, Christian Hunter, and Hartmut Leuthold. 2015. When language gets emotional. *Acta psychologica* (2015).

[9] Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. 2015. Sarcastic or Not: Word Embeddings to Predict the Literal or Sarcastic Meaning of Words. In *EMNLP*.

[10] Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying Sarcasm in Twitter: A Closer Look. In *Human Language Technologies: Short Papers*. ACL, 6. http://dl.acm.org/citation.cfm?id=2002736.2002850

[11] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

[12] Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing Context Incongruity for Sarcasm Detection. In *ACL and IJCNLP*. 757–762.

[13] Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark James Carman. 2016. Are Word Embedding-based Features Useful for Sarcasm Detection?. In *EMNLP*. http://aclweb.org/anthology/D/D16/D16-1104.pdf

[14] Maria Khokhlova, Viviana Patti, and Paolo Rosso. 2016. Distinguishing between irony and sarcasm in social media texts. In *ISMW FRUCT*.

[15] CC Liebrecht, FA Kunneman, and APJ van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets# not. (2013).

[16] Stephanie Lukin and Marilyn Walker. 2013. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. In *Proceedings of the Workshop on Language Analysis in Social Media*.

[17] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *SIGIR*. ACM.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR* abs/1301.3781 (2013).

[19] Saif M. Mohammad, Svetlana Kiritchenko, and Joel Martin. 2013. Identifying Purpose Behind Electoral Tweets. In *WISDOM*. Article 1, 9 pages.

[20] Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a Word-Emotion Association Lexicon. 29, 3 (2013), 436–465.

[21] Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. Creating and Characterizing a Diverse Corpus of Sarcasm in Dialogue. , 31–41 pages. https://doi.org/10.18653/v1/W16-3604

[22] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*.

[23] Antonio Reyes, Paolo Rosso, and Davide Buscaldi. 2012. From Humor Recognition to Irony Detection: The Figurative Language of Social Media. *DKE* (April 2012).

[24] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *EMNLP*.

[25] Carlo Strapparava and Alessandro Valitutti. 2004. WordNet-Affect: An affective extension of WordNet. In *LREC*. 1083–1086.

[26] Emilio Sulis, Delia Irazú Hernández Farías, Paolo Rosso, Viviana Patti, and Giancarlo Ruffo. 2016. Figurative Messages and Affect in Twitter. *Knowledge Based Systems* (2016), 132–143. https://doi.org/10.1016/j.knosys.2016.05.035

[27] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *ACL*. http://aclweb.org/anthology/P/P14/P14-1146.pdf

[28] Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. ICWSM-A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews. In *ICWSM*.