

An Ensemble Based Method for Predicting Emotion Intensity of Tweets

Sreekanth Madisetty^(✉) and Maunendra Sankar Desarkar

Department of Computer Science and Engineering,
Indian Institute of Technology Hyderabad, Hyderabad, India
{cs15resch11006,maunendra}@iith.ac.in

Abstract. Recently, user generated contents have increased tremendously in social media. Twitter is a popular micro-blogging platform in which users share their feelings, opinions, feedback, etc. It has been observed that microblogs are often associated with emotions. Several studies have focused on assigning a given tweet to one of the available emotion categories (e.g., *anger*, *fear*, *joy*, *sadness*). It is often useful in applications to find the intensity of emotion in the tweets. The focus on identifying emotion intensity is less in the literature. In this paper, we focus on determining the level of emotion intensity in the tweets. We use an ensemble of three methods: Convolution Neural Networks (CNN) with word embedding features, XGBoost with word n-gram and char n-gram features, and Support Vector Regression (SVR) with lexicon and word embedding features. The final prediction of the given tweet is obtained by the average of predictions of individual methods in the ensemble. The performance of ensemble is better than the methods in the ensemble due to diverse features. Our experimental results outperform baseline methods.

Keywords: Social media · Sentiment analysis · Emotion detection

1 Introduction

Nowadays social media plays a very important role among Internet users. Twitter is a social micro-blogging platform where people express their opinions, feelings, arguments about different topics across the world. Tweets often contain sentiments and emotions expressed by the users. Several lines of research that focus on tweets try to understand emotion or sentiment attached to it. Sentiment analysis describes whether the tweet is positive, negative or neutral. Emotion detection assigns the tweet to one of the given emotion categories (*anger*, *fear*, *joy*, and *sadness*). Existing research in this context have mainly focused on either sentiment analysis or emotion detection in Twitter [1, 27]. The focus on emotion intensity prediction is limited in the literature. It is often useful to find the intensity of emotion in text in various applications, e.g., crisis management, product quality, event reporting, etc.

In this paper, we focus on the following problem: Given a tweet and emotion category, predict the intensity of that emotion in the tweet. We use three different families of machine learning algorithms, Convolution Neural Networks (CNN), XGBoost, and Support Vector Regression (SVR) to find the emotion intensity in the tweets. Each algorithm is very popular in handling various machine learning tasks. The predictions of each algorithm are averaged to get the final prediction.

Recently, a dataset is published in WASSA-2017 shared task in emotion intensity [17] where the tweets are labeled with four emotion categories, *anger*, *fear*, *joy*, and *sadness*. For each tweet, the intensity of that emotion is also provided. Few example instances from that dataset are presented in Table 1. We use this dataset to evaluate our proposed method.

Table 1. Example tweets showing emotion intensity

Id	Text	Category	Intensity
10000	I asked for my parcel to be delivered to a pick up store not my address #fuming #poorcustomerservice	anger	0.896
20000	Job interview in the afternoon #nervous #ek	fear	0.917
30000	Today I reached 1000 subscribers on YT!, #goodday, #thankful	joy	0.926
40000	My #Fibromyalgia has been really bad lately which is not good for my mental state. I feel very overwhelmed #anxiety #bipolar #depression	sadness	0.946

Rest of the paper is organized as follows. Related literature for current work is described in Sect. 2. Next in Sect. 3, problem statement of our work is defined. Details of the proposed method are presented in Sect. 4. Experimental evaluation of the method is described in Sect. 5. We conclude the work by providing directions for future research in Sect. 6.

2 Related Work

A large amount of work has been done to detect sentiments from twitter data. Although, sentiment analysis is different from emotion intensity prediction, features which are used in sentiment analysis can also be used in emotion intensity prediction. Hence, in this section, we present related work from literature for both sentiment analysis and emotion intensity prediction tasks.

Sentiment Analysis: Part-of-speech tag, lexicons, bag-of-words, emoticons, linguistic features, semantic features, etc. are some of the common features used in sentiment analysis. A hybrid approach which uses both corpus-based and dictionary-based methods to find the semantic orientation of the opinion words

in tweets is described in [13]. Agarwal et al. [1] used POS-specific prior polarity features and tree kernel for sentiment analysis. Bag-of-words features along with Sentiwordnet, lexicons, emoticons, etc. are used in [25]. Semantic feature is added along with traditional features for sentiment analysis in [28]. Kouloumpis et al. [12] used linguistic features and lexical resources. However, in all the above methods emotion category is not considered.

Emotion Detection: A method with distant supervision for emotion classification is described in [26]. The public mood is modeled using Twitter messages in [4]. A dataset for emotion detection in Twitter is developed in [27]. The authors have considered seven emotion categories, namely, *anger*, *disgust*, *fear*, *joy*, *love*, *sadness*, and *surprise*. Another large dataset containing instances of $\langle \textit{tweet}, \textit{emotion category} \rangle$ annotation is created in [31]. The authors have used emotion-related hashtags which are present in the tweets for the creation of dataset. They have used unigrams, bigrams, sentiment words, and part-of-speech features for emotion detection. They have also considered seven emotion categories similar to Roberts et al. [27] but used *thankfulness* category instead of *anger*. However, in all the above methods intensity of the emotion is not considered. Word-emotion association lexicon is built using crowdsourcing in [20]. An annotation scheme is introduced for finding the emotion intensities in the blog posts in [2]. A supervised framework is developed for identifying the emotional expressions and intensities in [7]. However, the emotion intensities are categorical (high, medium, and low). An ensemble method for predicting emotion intensities is described in [14]. The authors have used two SVR methods with different features and a neural network method in the ensemble. However, word embedding features are not used.

3 Problem Definition

We now briefly define the problem addressed in this paper: *Given a tweet T and an emotion E , determine the intensity $Y_{T,E}$ of emotion E felt by the author of the tweet T . $Y_{T,E}$ is a real-valued score between 0 and 1.* Here 1 is the maximum possible score, and it means the maximum amount of emotion E felt by the speaker of the tweet T . Similarly, 0 is the minimum possible score, and it means the least amount of emotion E .

4 Methodology

We model the problem of predicting emotion intensity as a regression problem. We identified three methods, namely, Convolution Neural Networks (CNN), XGBoost, and Support Vector Regression (SVR) from three different family of algorithms for this prediction. These methods are selected due to their wide acceptability in the machine learning literature for performing various predictive analytics. These three methods are combined in an ensemble to retain the predictive power of the individual algorithms as well as to exploit the synergy between them.

Tweets often contain noise in the form of slang words, elongated words, spelling mistakes, abbreviations, @ mentions, etc. The maximum length of tweet is 140 characters long. We apply the following text preprocessing steps to get better performance of the model. URLs are removed, all words are converted to lower case, @ mentions and numbers are also removed as part of the preprocessing step. These preprocessed tweets are given to each of the individual methods in the ensemble for training and testing. We now describe these methods in detail.

4.1 Convolution Neural Networks (CNN)

Convolution Neural Networks (CNN) are popular in computer vision for various tasks, e.g., face recognition, image classification, action recognition, human pose estimation, scene labeling, etc. CNNs are also used in many Natural Language Processing (NLP) tasks, named entity recognition, part-of-speech tagging, chunking, etc. We used CNN for our problem on the similar lines of approach given in [10]. CNN architecture has five layers, namely, input layer, convolution layer, pooling layer, hidden layer, and output layer.

The input to the model is tweets. Let each tweet be comprised of sequence of words: $\{term_1, term_2, term_3, \dots, term_n\}$. Then tweet vector is represented as

$$T_v = w_1 \circ w_2 \circ w_3 \circ \dots \circ w_n \quad (1)$$

Where w_i is the word embedding vector of $term_i$, and \circ is the concatenation operator. Each $w_i \in \mathbb{R}^{1 \times d}$ is associated with their corresponding pre-trained word vectors. These word embeddings can be looked up in a vocabulary of the embedding matrix $W \in \mathbb{R}^{V \times d}$, where V is the number of words in the vocabulary. Words are mapped to indices from 1 to V , and the embedding matrix is created in such a way that at index i , the word embedding corresponding to the word associated with index i is present. Tweet matrix $T_m \in \mathbb{R}^{n \times d}$ is given as input to the model where each word is represented by word embedding $w_i \in \mathbb{R}^{1 \times d}$. Glove Twitter word embeddings are used in our method. These word embeddings are publicly available¹ [23]. Tweet lengths may vary, so necessary padding is applied to have equal lengths for all the tweet vectors. Next layer is convolution layer. Convolution feature maps are created to extract emotion features. Convolution feature is calculated as follows.

$$o_i = g(\alpha \cdot w_{i:i+h-1} + \beta) \quad (2)$$

where α is a convolution filter, $\beta \in \mathbb{R}$ is bias term, h is window size, $w_{i:i+h-1}$ is the concatenation of embeddings for the terms occurring in a window of length h , from positions i to $i+h-1$, and g is a non-linear function such as the hyperbolic tangent. This convolution filter is applied to each possible window of words in the tweet to produce a convolution feature map $c \in \mathbb{R}^{n-h+1}$. Next layer is max pooling layer. The main idea in this layer is to capture most important

¹ <https://nlp.stanford.edu/projects/glove/>.

activation. Let $o_1, o_2, o_3, \dots \in \mathbb{R}$ denote the output values for our filter. Max-over-time pooling is computed as follows.

$$c = \max_i(o_i) \in \mathbb{R} \tag{3}$$

The output of max-pooling layer is given as input to the dense hidden layer. The output of hidden layer is passed through the final output layer using sigmoid as the activation function. Values output by this sigmoid activation function is emitted as the prediction of the emotion intensity for the input tweet. To avoid overfitting, dropout parameter is used.

The dataset used in our experiments contains four emotion categories. Four CNNs are used for these four emotion categories. Each CNN is trained separately for each emotion category, and emotion intensities for that category are predicted. Same configuration (filter length, number of filters, word embedding dimension size, dropout rate, number of neurons in hidden layer, number of layers, etc.) is used for all the categories to train the model. This CNN model is static where the word embeddings are not changed throughout the model.

4.2 Extreme Gradient Boosting (XGBoost)

This is the second method in the ensemble. XGBoost is based on original Gradient Boosting Machine (GBM) framework [6]. It is a supervised learning algorithm. It is a tree ensemble model and is a set of Classification and Regression Trees (CARTs). Normally, a single tree is not strong enough for classification in practice. In tree ensemble, predictions of multiple trees are added to get the final prediction. Mathematically, model is written as

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in F \tag{4}$$

where K is the number of trees, f is a function in the functional-space F , F is the set of all possible CARTs, x_i is training data, and \hat{y}_i is the prediction. If y_i is target variable then the objective function can be written as

$$Obj = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \tag{5}$$

The first part in the above equation is training loss and second part is regularization. Additive training is used for training the model. XGBoost is often used in many of the data science competitions. It does computations parallelly and is very fast. Word n-gram and character n-gram features are used in this model.

4.3 Support Vector Regression (SVR)

This is the third method used in our ensemble which is taken from [16]. Features used are word n-grams, char n-grams, word embeddings, and lexicons. Word

embeddings are trained from Edinburgh Twitter corpus [24] using *Word2Vec* with 400 dimensions. Lexicons used in this method are AFINN [22], BingLiu [9], MPQA [32], NRC Affect Intensity Lexicon [15], NRCWord-Emotion Association Lexicon [20], NRC10 Expanded [5], NRC Hashtag Emotion Association Lexicon [18], NRC Hashtag Sentiment Lexicon [19], Sentiment140 [19], SentiWordNet [3], and SentiStrength [29]. If the lexicon consists of categorical labels for the words then number of words matching each category in the tweet are counted. If the lexicon provides scores for the words then the scores of each word in the tweet are added. SVM Regression model is trained by using these features for each category separately and emotion intensities are predicted.

4.4 Ensemble

Ensemble methods have been proved to be very successful for classification problems. A system named Webis has achieved the best performance in SemEval-2015 subtask B, “Sentiment Analysis in Twitter” [8]. In the Netflix competition [30] and KDD Cup 2009 [21], the winners have used ensemble-based methods. There are several ways to combine the classifiers, e.g., bagging, boosting, simple averaging, majority voting, stacking, etc. We tested our methods with some of them, and simple averaging performed better than the other ensemble methods. Our ensemble method works as follows. CNN with word embedding features is trained on each category separately in the training data, and it is applied to the testing data and predictions are noted. Similarly, XGBoost with word n-gram and char n-gram features is trained, and predictions of testing data are saved. In a similar fashion, SVR with lexicon and word embedding features is trained and is applied on testing data and predictions are noted. Finally, for each tweet, the average of prediction values of individual methods is considered as final prediction.

5 Experiments

5.1 Data

The dataset used in our experiments is obtained from [16]. Statistics of the data is described in Table 2. Each row of the dataset contains id, text, emotion category, and emotion intensity as described in Table 1. The emotion intensity is a real value between 0 and 1. There are four categories of emotions, namely, *anger*, *fear*, *joy*, and *sadness*. The dataset is created by using a technique called best-worst-scaling (BWS) which improves the annotation consistency and reliable emotion intensity values.

5.2 Evaluation Metrics

In this section we describe the evaluation metrics used in our approach.

Table 2. Number of tweets in each category

Emotion	Training	Validation	Testing	All
anger	857	84	760	1701
fear	1147	110	995	2252
joy	823	74	714	1611
sadness	786	74	673	1533
All	3613	342	3142	7097

– **Pearson correlation (PC):**

It measures the correlation between two variables. Pearson correlation is calculated between predicted values and gold values. Pearson correlation coefficient is calculated as

$$PC = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \tag{6}$$

In our problem, n is the number of test tweets, x_i is predicted emotion intensity value for i^{th} test tweet, y_i is ground truth value, \bar{x} is mean of x , and \bar{y} is mean of y .

– **Spearman rank correlation (SC):**

It measures the relationship between two rankings. Let X denote the set of actual intensity values and Y denote the set of predicted intensity values. Let X and Y are converted to ranks rgX and rgY respectively. Spearman rank correlation coefficient is calculated as

$$SC = \frac{cov(rgX, rgY)}{\sigma_{rgX} \sigma_{rgY}} \tag{7}$$

where $cov(rgX, rgY)$ is the co-variance of rank variables, σ_{rgX} , σ_{rgY} are the standard deviations of the rank variables.

Sometimes, the tweets which are having high emotion content are relevant. So, it is useful to identify the high emotion related content. To test this kind of tweets, we use two additional metrics, Pearson 0.5 to 1.0 (*PCH*) and Spearman 0.5 to 1.0 (*SCH*). Pearson 0.5 to 1.0 is calculated by considering the instances only with ground truth emotion intensities which are greater than or equal to 0.5, and the rest are ignored. Similarly, Spearman 0.5 to 1.0 is calculated.

5.3 Results and Discussions

The first method used in the ensemble is CNN. Glove Twitter word embeddings are used with dimensions, 25, 50, 100, and 200 [23]. We have used 100 as maximum sentence length, window size 3, 250 filters, hidden layer with 200 neurons, dropout 0.2 as regularization parameter in our setting. The results of CNN with 25D, 50D, 100D, and 200D word embeddings are reported in Tables 3, 4, 5, and 6 respectively. We observe that the increase in dimensions results in increase in

Table 3. CNN with Glove 25D.

Emotion	PC	SC	PCH	SCH
anger	0.540	0.511	0.410	0.386
fear	0.615	0.593	0.476	0.451
joy	0.525	0.520	0.377	0.387
sadness	0.600	0.586	0.465	0.450
Average	0.570	0.552	0.432	0.419

Table 4. CNN with Glove 50D.

Emotion	PC	SC	PCH	SCH
anger	0.616	0.581	0.493	0.477
fear	0.664	0.642	0.512	0.480
joy	0.591	0.590	0.421	0.447
sadness	0.689	0.682	0.523	0.503
Average	0.640	0.624	0.488	0.477

Table 5. CNN with Glove 100D.

Emotion	PC	SC	PCH	SCH
anger	0.672	0.644	0.537	0.518
fear	0.684	0.657	0.561	0.518
joy	0.604	0.600	0.395	0.401
sadness	0.707	0.703	0.524	0.524
Average	0.667	0.651	0.504	0.491

Table 6. CNN with Glove 200D.

Emotion	PC	SC	PCH	SCH
anger	0.670	0.639	0.548	0.540
fear	0.691	0.664	0.579	0.525
joy	0.643	0.635	0.434	0.423
sadness	0.727	0.728	0.545	0.544
Average	0.683	0.667	0.526	0.508

Table 7. XGBoost.

Emotion	PC	SC	PCH	SCH
anger	0.571	0.521	0.486	0.446
fear	0.599	0.546	0.517	0.448
joy	0.572	0.567	0.394	0.379
sadness	0.666	0.662	0.471	0.449
Average	0.602	0.574	0.467	0.431

Table 8. SVR.

Emotion	PC	SC	PCH	SCH
anger	0.636	0.627	0.502	0.472
fear	0.633	0.621	0.484	0.441
joy	0.650	0.654	0.379	0.365
sadness	0.713	0.714	0.555	0.534
Average	0.658	0.654	0.480	0.453

performance. For example, CNN with 50D performance is better than CNN with 25D. Similarly, CNN with 100D is performing better than CNN with 50D, and the performance of CNN with 200D is greater than CNN with 100D. Therefore, CNN with 200D is used in our method.

The second method used in the ensemble is XGBoost. The parameters in this method are learning rate = 0.1, number of estimators = 100, booster is gradient boosting tree, and maximum depth is 3. The results of four emotion categories are reported in Table 7. The Pearson coefficient and Spearman coefficient values are higher than CNN with 25D but lesser than the CNN with other dimensional word embeddings (50D, 100D, 200D). The final method used in the ensemble is SVR (Table 8). The parameters used in this model are linear kernel, $C = 0.001$

Table 9. Ensemble (proposed method).

Emotion	PC	SC	PCH	SCH
anger	0.718	0.696	0.609	0.584
fear	0.729	0.709	0.606	0.546
joy	0.717	0.721	0.480	0.470
sadness	0.771	0.772	0.600	0.585
Average	0.734	0.725	0.574	0.546

Table 10. Comparison of our proposed method with other approaches.

Method	PC	SC	PCH	SCH
SVR (word n-grams)	0.501	0.492	0.390	0.382
SVR (Saif et al. [16])	0.658	0.654	0.480	0.453
IMS [11]	0.722	0.712	0.514	0.503
XGBoost+CNN	0.703	0.691	0.551	0.527
XGBoost+SVR	0.704	0.697	0.540	0.513
CNN+SVR	0.724	0.713	0.558	0.535
Ensemble	0.734	0.725	0.574	0.546

(penalty term). Radial Basis Function (RBF) and polynomial kernels are also tested. However, linear kernel is performing better. The evaluation metric values are better than XGBoost and CNN with 25D and 50D. An ensemble is created by averaging the predictions of three methods described in Sect. 4 and results are reported in Table 9. It can be observed that the Pearson coefficient for both 0 to 1 and 0.5 to 1.0, the ensemble values are higher than all other methods. Similarly, the Spearman coefficient for both 0 to 1 and 0.5 to 1.0, the ensemble method values are higher.

Comparison with SVR [16], IMS [11] and combination of methods in the ensemble is presented in Table 10. We observe that our ensemble method significantly outperforms the baselines and other combinations. This shows efficacy our proposed method. Category-wise comparison of our approach with other methods for four emotion categories, *anger*, *fear*, *joy*, and *sadness* is presented in Fig. 1a, b, c, and d respectively. For *anger*, *fear*, and *joy* categories, our method performs better than other methods. This is due to the presence of diverse features in individual methods of the ensemble. For *sadness* category, our proposed method values are higher for *PC* and *SC* whereas CNN+SVR combination method values are slightly higher for *PCH* and *SCH*.

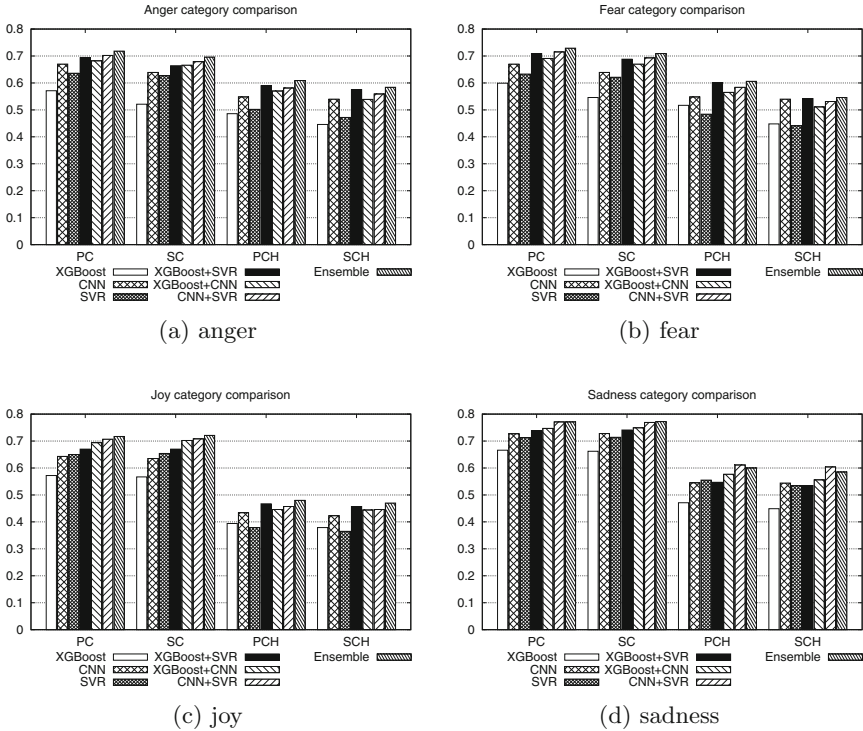


Fig. 1. Category-wise comparison of four emotion categories

6 Conclusion

In this paper, we presented an ensemble approach to predict the emotion intensity in tweets. The three methods are Convolution Neural Networks (CNN), XGBoost, and Support Vector Regression (SVR). Glove Twitter word embeddings are used with different dimensions for training the CNN model. The presence of diverse features in each of these three methods make the ensemble more stronger in predicting the better emotion intensities. Experimental results show that our method significantly outperforms other methods. For future work, we would like to identify new features and new methods to include in the ensemble.

References

1. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of twitter data. In: Proceedings of the Workshop on Languages in Social Media, pp. 30–38. Association for Computational Linguistics (2011)
2. Aman, S., Szipakowicz, S.: Identifying expressions of emotion in text. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 196–205. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74628-7_27

3. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: LREC, vol. 10, pp. 2200–2204 (2010)
4. Bollen, J., Mao, H., Pepe, A.: Modeling public mood and emotion: twitter sentiment and socio-economic phenomena. In: ICWSM 2011, pp. 450–453 (2011)
5. Bravo-Marquez, F., Frank, E., Mohammad, S.M., Pfahringer, B.: Determining word-emotion associations from tweets by multi-label classification. In: WI 2016, pp. 536–539. IEEE Computer Society (2016)
6. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
7. Das, D., Bandyopadhyay, S.: Identifying emotional expressions, intensities and sentence level emotion tags using a supervised framework. In: PACLIC, vol. 24, pp. 95–104 (2010)
8. Hagen, M., Potthast, M., Büchner, M., Stein, B.: Webis: an ensemble for twitter sentiment detection. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 582–589 (2015)
9. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004)
10. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint [arXiv:1408.5882](https://arxiv.org/abs/1408.5882) (2014)
11. Köper, M., Kim, E., Klinger, R.: 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, pp. 50–57 (2017)
12. Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: the good the bad and the omg! In: ICWSM 2011, vol. 164, pp. 538–541 (2011)
13. Kumar, A., Sebastian, T.M.: Sentiment analysis on twitter. *Int. J. Comput. Sci. Issues (IJCSI)* **9**(4), 372 (2012)
14. Madisetty, S., Desarkar, M.S.: NSEmo at EmoInt-2017: an ensemble to predict emotion intensity in tweets. In: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 219–224 (2017)
15. Mohammad, S.M.: Word affect intensities. arXiv preprint [arXiv:1704.08798](https://arxiv.org/abs/1704.08798) (2017)
16. Mohammad, S.M., Bravo-Marquez, F.: Emotion intensities in tweets. In: Proceedings of the Sixth Joint Conference on Lexical and Computational Semantics (*Sem), Vancouver, Canada (2017)
17. Mohammad, S.M., Bravo-Marquez, F.: WASSA-2017 shared task on emotion intensity. In: Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA), Copenhagen, Denmark (2017)
18. Mohammad, S.M., Kiritchenko, S.: Using hashtags to capture fine emotion categories from tweets. *Comput. Intell.* **31**(2), 301–326 (2015)
19. Mohammad, S.M., Kiritchenko, S., Zhu, X.: NRC-Canada: building the state-of-the-art in sentiment analysis of tweets. arXiv preprint [arXiv:1308.6242](https://arxiv.org/abs/1308.6242) (2013)
20. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word-emotion association lexicon. *Comput. Intell.* **29**(3), 436–465 (2013)
21. Niculescu-Mizil, A., Perlich, C., Swirszcz, G., Sindhvani, V., Liu, Y., Melville, P., Wang, D., Xiao, J., Hu, J., Singh, M., et al.: Winning the KDD cup orange challenge with ensemble selection. In: Proceedings of the 2009 International Conference on KDD-Cup 2009, vol. 7, pp. 23–34. JMLR.org (2009)

22. Nielsen, F.Å.: A new ANEW: evaluation of a word list for sentiment analysis in microblogs. arXiv preprint [arXiv:1103.2903](https://arxiv.org/abs/1103.2903) (2011)
23. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014). <http://www.aclweb.org/anthology/D14-1162>
24. Petrovic, S., Osborne, M., Lavrenko, V.: The Edinburgh twitter corpus. In: Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, pp. 25–26 (2010)
25. Poursepanj, H., Weissbock, J., Inkpen, D.: uOttawa: system description for SemEval 2013 task 2 sentiment analysis in twitter. In: SemEval@NAACL-HLT, pp. 380–383 (2013)
26. Purver, M., Battersby, S.: Experimenting with distant supervision for emotion classification. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 482–491. Association for Computational Linguistics (2012)
27. Roberts, K., Roach, M.A., Johnson, J., Guthrie, J., Harabagiu, S.M.: EmpaTweet: annotating and detecting emotions on twitter. In: LREC 2012, pp. 3806–3813 (2012)
28. Saif, H., He, Y., Alani, H.: Semantic sentiment analysis of twitter. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012. LNCS, vol. 7649, pp. 508–524. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35176-1_32
29. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. *J. Am. Soc. Inform. Sci. Technol.* **63**(1), 163–173 (2012)
30. Töscher, A., Jahrer, M., Bell, R.M.: The BigChaos solution to the Netflix grand prize. Netflix prize documentation, pp. 1–52 (2009)
31. Wang, W., Chen, L., Thirunarayan, K., Sheth, A.P.: Harnessing twitter “big data” for automatic emotion identification. In: 2012 International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 International Conference on Social Computing (SocialCom), pp. 587–592. IEEE (2012)
32. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 347–354. Association for Computational Linguistics (2005)