NSEmo at EmoInt-2017: An Ensemble to Predict Emotion Intensity in Tweets

Sreekanth Madisetty and **Maunendra Sankar Desarkar**
Department of Computer Science and Engineering
IIT Hyderabad, Hyderabad, India
{cs15resch11006, maunendra}@iith.ac.in

Abstract

In this paper, we describe a method to predict emotion intensity in tweets. Our approach is an ensemble of three regression methods. The first method uses content-based features (hashtags, emoticons, elongated words, etc.). The second method considers word n-grams and character n-grams for training. The final method uses lexicons, word embeddings, word n-grams, character n-grams for training the model. An ensemble of these three methods gives better performance than individual methods. We applied our method on WASSA emotion dataset. Achieved results are as follows: average Pearson correlation is 0.706, average Spearman correlation is 0.696, average Pearson correlation for gold scores in range 0.5 to 1 is 0.539, and average Spearman correlation for gold scores in range 0.5 to 1 is 0.514.

1 Introduction

Twitter is a popular microblogging platforms in which users share their opinions, feelings on different topics which are happening across the world.

The aim of sentiment analysis is to detect the positive, negative, or neutral feelings from the text, whereas the aim of emotion analysis is to detect the types of feelings in the text, such as anger, fear, joy, sadness, disgust, and surprise. In this paper, we focus on emotion analysis in tweets. Sentiment analysis of Twitter data is very challenging. Users who are posting on Twitter often do not follow grammar rules. This results in noise in the Twitter data. This noisy nature of Twitter data is in the form of spelling mistakes, use of slang words, sentence mistakes, abbreviations, elongated words, etc. Moreover, the text limit is 140 characters long. In this paper, four emotions are considered. They are anger, fear, joy, and sadness. The task is to predict the emotion intensity of each test instance in a range between 0 and 1. The emotion intensity 1 indicates the maximum emotion whereas 0 indicates the least emotion felt by the author of the tweet.

We use an ensemble of three methods, namely, Support Vector Regression (SVR), Neural Networks, and Baseline to predict the emotion intensity in tweets. The performance of ensemble approach is better than that of the individual methods.

There is a growing interest in sentiment analysis of tweets across variety of domains such as health (Chew and Eysenbach, 2010), stock market (Bollen et al., 2011), disaster management (Mandel et al., 2012), and presedential elections (Wang et al., 2012).

The rest of the paper is organized as follows. Related literature for current work is presented in Section 2. Next in Section 3, problem statement and details of the methods used in this paper are defined. Experimental evaluation of the method is shown in Section 4. We conclude the work by providing directions for future research in Section 5.

2 Related Work

With the increase of user-generated contents in social media, blogs, discussion fora, etc. people are focusing on the problem of analyzing the sentiments expressed in these contents. Go et al. (2009) used emoticons as labels for training data and distance supervision to classify tweets into positive or negative class. Pak and Paroubek (2010) presented a method for automatic collection of a corpus that can be used to train a sentiment classi-

Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 219–224
Copenhagen, Denmark, September 7–11, 2017.

fier. The authors have classified the tweets into three classes, namely, positive, negative, and neutral using trained classifier. Kouloumpis et al. (2011) used linguistic and lexical features to detect the sentiments of Twitter messages. The authors showed that Part-Of-Speech (POS) features might not be useful for sentiment analysis in the Twitter domain.

Khan et al. (2015) proposed a method for combining lexicon-based and learning-based methods for Twitter sentiment analysis. There has been a lot of work done in the SemEval Twitter sentiment analysis tasks (Rosenthal et al., 2014, 2015; Nakov et al., 2016; Rosenthal et al., 2017).

Combining classifiers has been proved to be very successful for classification problems. A system named Webis achieved top-rank in SemEval-2015 subtask B, task 10 "Sentiment Analysis in Twitter" (Hagen et al., 2015). The authors reproduced four state-of-the-art Twitter sentiment classification methods with diverse feature sets. The predictions of four classifiers are combined by taking the average of classifiers' individual confidence scores for the three classes and predicts the label with the highest score. In the Netflix competition, the winner used an ensemble method to implement a collaborative filtering algorithm (Töscher et al., 2009). In KDD Cup 2009 also, the winner used an ensemble method (Niculescu-Mizil et al., 2009). Zhang et al. (2016) used a classifier fusion based method for polarity classification in Twitter. The authors have used four classifiers in the ensemble method.

3 System Description

In this section, we describe the methodology used for WASSA 2017 shared task on emotion intensity. The WASSA 2017 shared task (Mohammad and Bravo-Marquez, 2017b) problem definition is as follows: *Given a tweet and an emotion E, determine the intensity of the emotion E felt by the author of the tweet.* The intensity is a real-valued score between 0 and 1. The maximum possible emotion intensity 1 stands for feeling the maximum amount of emotion E and the minimum possible emotion intensity 0 stands for feeling the least amount of emotion E. There are four categories of emotion given in the task, namely, anger, fear, joy, and sadness. We combine the three methods (Support vector regression, Neural networks, and Baseline) for predicting the emotion intensity.

3.1 Data Preprocessing

For any machine learning algorithm preprocessing the data is a very important step. As discussed in Section 1 tweets often contain a lot of noise. Before applying the model to the data, preprocessing should be done. Removal of unnecessary tokens from the text will improve the performance of the model. All words are converted to lower case, URLs are removed, numbers, and @ mentions are also removed as these tokens do not contribute in predicting the sentiment of the tweet. Hashtags, emoticons, punctuation marks (?, !) are retained because they will help in predicting the sentiment.

3.2 Support Vector Regression

This is the first method used for predicting emotion intensity in tweets. First, we define the features used in this work.

3.2.1 Features

- No. of hashtags: The number of hashtags present in the tweet.
- Length: Length of the tweet
- Word n-grams: We used word n-grams with n ranging from 1 to 3 i.e., unigrams, bigrams, and trigrams. All these n-grams are word level n-grams.
- Char n-grams: We also used character n-grams. These n-grams include the existence of two, three, four, five, and six consecutive sequence of characters.
- Punctuation: Number of punctuation symbols (?, !) present in the tweet.
- Emoticons: Number of emoticons present in the tweet.
- Elongated words: The number of words with one character repeated more than twice, for example, 'haaapy'.
- Lexicon: NRC Affect Intensity Lexicon (Mohammad, 2017) is used.

All the above features are used for training the model.

3.3 Neural Networks

This is the second method used to determine the emotion intensity in tweets. A multi-layered neural network with two hidden layers is used. These hidden layers consist of 125 and 25 neurons respectively. We used Keras for developing this multi-layered neural network model. Keras is a useful Python library for developing deep learning models. TensorFlow is used as backend for Keras. Word n-grams and character n-grams are used in this model.

3.4 Baseline

This method was given in WASSA 2017 shared task as the baseline method (Mohammad and Bravo-Marquez, 2017a). The authors have created the datasets of tweets annotated for anger, fear, joy, and sadness emotion intensities. They have used the best-worst scaling technique to improve annotation consistency and obtained reliable scores. They created a regression system, *AffectiveTweetsPackage* for the Weka machine learning workbench, to automatically determine emotion intensity and related tasks. The following features are used in this baseline system.

- word n-grams: This feature will check whether the word n-grams are present in the tweet or not, with n values 1, 2, 3, and 4.
- char n-grams: It will check whether the char n-grams are present in the tweet or not, with n values 3, 4, and 5.
- Word Embeddings: *Word2Vec* (Mikolov et al., 2013) is used to create word embeddings with negative sampling skip-gram model. Vector for the tweet is created by averaging the individual word embeddings of the tweet. Word vectors are trained from the Edinburgh Twitter Corpus (Petrovic et al., 2010). Number of dimensions used is 400.
- Lexicons: Lexicons used in this system are AFINN (Nielsen, 2011), BingLiu (Hu and Liu, 2004) , MPQA (Wilson et al., 2005), NRC Affect Intensity Lexicon (Mohammad, 2017), NRC Word-Emotion Association Lexicon (Mohammad and Turney, 2013), NRC10 Expanded (Bravo-Marquez et al., 2016), NRC Hashtag Emotion Association Lexicon (Mohammad and Kiritchenko, 2015), NRC Hashtag Sentiment Lexicon (Mohammad et al., 2013), Sentiment140 (Mohammad et al., 2013), SentiWordNet (Baccianella et al., 2010), SentiStrength (Thelwall et al., 2012).

3.5 Ensemble Combination

Ensemble methods use several learning algorithms to obtain better predictive performance than any other individual method used in the ensemble combination. There are several ways to combine the learning models such as bagging, boosting, majority voting, simple averaging, stacking, etc. Bagging trains each model in the ensemble using a subset of the training data drawn randomly, whereas boosting builds an ensemble in such a way that new model performance will improve for instances that are misclassified by previous models.

In majority voting, each model makes a prediction for the test instance, and the final prediction of the model is the one which is predicted by more models. Simple averaging is also another method for combining predictions of learned models, in which the prediction of the model for each test instance is the average of the predictions of the individual models. Stacking is another approach where the models are combined using another machine learning algorithm. The predictions of the individual model are the input to another learning algorithm (meta-learning algorithm).

We tested different ways of combining the individual regressors to an ensemble method. We observed that each method tries to predict the emotion intensity closer to the actual predictions for some test instances that others fail for. This is because of having different feature sets for different methods which are used in an ensemble. When we combine the individual regression methods, the performance of an ensemble will increase because of individual strengths of the methods. Finally, we observed that simple averaging performs better than other methods.

Our ensemble works as follows: SVR is trained separately for each class, anger, fear, joy, and sadness by considering train and dev data. Testing is performed on test data, and predictions of each class are saved in separate files. These predictions are real-valued scores between 0 and 1. We used all features that are listed in Section 3.2.1 for this method. Next, a multi-layered neural network is trained on the same data as SVR. Two hidden layers are used with 125 and 25 neurons. Num-

Table 1: Number of tweets in each phase.

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

Table 2: Submitted results for the competition.

Result	Pearson 0to1	Spearman 0to1	Pearson .5to1	Spearman .5to1
Submitted Results	0.525	0.528	0.373	0.369

ber of features is the input to the input layer, and the output is a real value between 0 and 1. For this reason, sigmoid activation function is used in the output layer. Word n-grams and character n-grams are used as features for this model. Then, we directly used the baseline algorithm given in the shared task. It is trained on the same data as SVR and neural network models.

Word embeddings of Edinburgh corpus, lexicons, word n-grams, char n-grams are used as features. Word embeddings are available for 50 dimensions and 400 dimensions. However, we found 400 dimension word embedding to perform better in our experiments. Predictions for each class are obtained from each of the trained models. Finally, the average of individual methods prediction for each test instance is considered as final prediction. The final prediction value is also in between 0 and 1.

4 Experiments

4.1 Data

There are four emotion categories, namely, anger, fear, joy, and sadness in the dataset given in the shared task (Mohammad, 2017). Details of number of tweets in each category for training, validation, and testing are shown in Table 1.

4.2 Results

In this section, we describe the results obtained by our methods. For evaluating the proposed methods, two evaluation metrics Pearson correlation and Spearman correlation are used. Pearson correlation for two sets is equal to 1 if they have a high positive correlation, -1 if they have a high negative correlation, and 0 if there is no correlation.

Table 2 shows our submitted results to the competition before the deadline. Word unigrams, and some limited features (lexicon, hashtags, punctuation) related to the sentiment are used, and SVR is used for learning and predicting the emotion intensities. Later, we improved our method using extra features and using different approaches. Table 3 shows the SVR model using polynomial kernel function. Table 4 shows SVR model using RBF kernel, and Table 5 shows SVR model using linear kernel function. We observe that SVR using linear kernel function is performing better than SVR with RBF and SVR with polynomial kernel function. So, we used SVR with linear kernel in the ensemble. The parameters used in SVR are gamma = 0.1 (kernel coefficient for rbf, poly), and C = 0.001 (penalty term)

Table 6 describes the results using neural networks model with word n-grams and char n-grams as features. The parameters used in this experiment are as follows: loss function is entropy, optimization algorithm is stochastic gradient descent, rectifier activation function is used in the hidden layers whereas sigmoid activation function is used in the output layer. Table 7 presents the results of the baseline method using 50 dimensional word embeddings of Edinburgh corpus whereas baseline method with 400 dimensional word embeddings are presented in Table 8.

The results of ensemble combination of SVR using linear kernel, neural networks, baseline method with 400 dimensional word embeddings are presented in Table 9. We have achieved the following results in the ensemble: average Pearson correlation is 0.706, average Spearman correlation is 0.696, average Pearson correlation for gold scores in range 0.5 to 1 is 0.539, and Spearman correlation for gold scores in range 0.5 to 1 is 0.514. Comparison of proposed method with baseline methods is presented in Table 10. We observe that our proposed method correlation values are higher than two variations of baselines (50d, 400d). We also observe that ensemble method is performing better than any other individual method used in combination. This is due to different feature sets used in the methods mentioned in Section 3.

5 Conclusion

We created two methods Support Vector Regression and Neural Networks and used baseline method from the shared task to detect the emotion

Table 3: SVR with polynomial kernel.

Emotion	Pearson 0to1	Spearman 0to1	Pearson .5to1	Spearman .5to1
anger	0.405	0.455	0.278	0.276
fear	0.333	0.466	0.239	0.250
joy	0.416	0.487	0.283	0.354
sadness	0.482	0.552	0.438	0.465
Average	0.409	0.490	0.310	0.336

Table 4: SVR with rbf kernel.

Pearson 0to1	Spearman 0to1	Pearson .5to1	Spearman .5to1
0.591	0.583	0.431	0.422
0.606	0.571	0.491	0.428
0.572	0.580	0.374	0.396
0.656	0.656	0.543	0.533
0.606	0.597	0.460	0.445

Table 5: SVR with linear kernel.

Pearson 0to1	Spearman 0to1	Pearson .5to1	Spearman .5to1
0.601	0.590	0.426	0.416
0.617	0.589	0.491	0.425
0.603	0.621	0.377	0.399
0.665	0.679	0.535	0.531
0.622	0.620	0.457	0.443

Table 6: Neural Networks.

Emotion	Pearson 0to1	Spearman 0to1	Pearson .5to1	Spearman .5to1
anger	0.570	0.557	0.432	0.436
fear	0.601	0.567	0.492	0.451
joy	0.571	0.565	0.350	0.329
sadness	0.642	0.630	0.499	0.491
Average	0.596	0.580	0.443	0.427

Table 7: Baseline with 50d word embeddings.

Emotion	Pearson 0to1	Spearman 0to1	Pearson .5to1	Spearman .5to1
anger	0.631	0.620	0.502	0.469
fear	0.622	0.606	0.477	0.431
joy	0.635	0.641	0.368	0.354
sadness	0.710	0.713	0.537	0.521
Average	0.649	0.645	0.471	0.444

Table 8: Baseline with 400d word embeddings.

Emotion	Pearson 0to1	Spearman 0to1	Pearson .5to1	Spearman .5to1
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

Table 9: Ensemble model combining Support Vector Regression using linear kernel, Neural Networks, Baseline method.

Emotion	Pearson 0to1	Spearman 0to1	Pearson .5to1	Spearman .5to1
anger	0.687	0.672	0.548	0.523
fear	0.703	0.676	0.574	0.517
joy	0.693	0.696	0.435	0.429
sadness	0.739	0.741	0.601	0.587
Average	**0.706**	**0.696**	**0.539**	**0.514**

Table 10: Comparison of proposed method with baseline method. The number in the brackets is the percentage difference with closest competitor (Baseline2).

Method	Pearson 0to1	Spearman 0to1	Pearson .5to1	Spearman .5to1
Baseline1 (50d)	0.649	0.645	0.471	0.444
Baseline2 (400d)	0.658	0.654	0.480	0.453
Ensemble	0.706 (**7.29%**)	0.696 (**6.42%**)	0.539 (**12.29%**)	0.514 (**13.47%**)

intensity in tweets. The predictions of these three methods are averaged to get the final prediction of each test instance for each class. The results of ensemble method show that average Pearson correlation, average Spearman correlation values are higher than the baseline method, SVR, neural networks.

For future work, we would like to see other learning methods which can improve the performance of the ensemble, and also we want to identify additional features for predicting the emotion intensity. We would like to use different Twitter word embeddings other than Edinburgh corpus in future.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*. volume 10, pages 2200–2204.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science* 2(1):1–8.

Felipe Bravo-Marquez, Eibe Frank, Saif M Mohammad, and Bernhard Pfahringer. 2016. Determining word–emotion associations from tweets by multi-label classification. In *WI'16*. IEEE Computer Society, pages 536–539.

Cynthia Chew and Gunther Eysenbach. 2010. Pandemics in the age of twitter: content analysis of tweets during the 2009 h1n1 outbreak. *PloS one* 5(11):e14118.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1(12).

Matthias Hagen, Martin Potthast, Michel Büchner, and Benno Stein. 2015. Webis: An ensemble for twitter sentiment detection .

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowl-*

edge discovery and data mining. ACM, pages 168–177.

Aamera ZH Khan, Mohammad Atique, and VM Thakare. 2015. Combining lexicon-based and learning-based methods for twitter sentiment analysis. *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)* page 89.

Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *Icwsm* 11(538-541):164.

Benjamin Mandel, Aron Culotta, John Boulahanis, Danielle Stark, Bonnie Lewis, and Jeremy Rodrigue. 2012. A demographic analysis of online sentiment during hurricane irene. In *Proceedings of the Second Workshop on Language in Social Media*. Association for Computational Linguistics, pages 27–36.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Saif M Mohammad. 2017. Word affect intensities. *arXiv preprint arXiv:1704.08798* .

Saif M. Mohammad and Felipe Bravo-Marquez. 2017a. Emotion intensities in tweets. In *Proceedings of the sixth joint conference on lexical and computational semantics (*Sem)*. Vancouver, Canada.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017b. 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.

Saif M Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence* 31(2):301–326.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242* .

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence* 29(3):436–465.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. *Proceedings of SemEval* pages 1–18.

Alexandru Niculescu-Mizil, Claudia Perlich, Grzegorz Swirszcz, Vikas Sindhwani, Yan Liu, Prem Melville, Dong Wang, Jing Xiao, Jianying Hu, Moninder Singh, et al. 2009. Winning the kdd cup orange challenge with ensemble selection. In *Proceedings of the 2009 International Conference on KDD-Cup 2009-Volume 7*. JMLR. org, pages 23–34.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903* .

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*. volume 10.

Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2010. The edinburgh twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*. pages 25–26.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 493–509.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*. pages 451–463.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*. Dublin, Ireland, pages 73–80.

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology* 63(1):163–173.

Andreas Töscher, Michael Jahrer, and Robert M Bell. 2009. The bigchaos solution to the netflix grand prize. *Netflix prize documentation* pages 1–52.

Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. 2012. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, pages 115–120.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, pages 347–354.

Zhengchen Zhang, Chen Zhang, Fuxiang Wu, Dong-Yan Huang, Weisi Lin, and Minghui Dong. 2016. I2rntu at semeval-2016 task 4: Classifier fusion for polarity classification in twitter. *Proceedings of SemEval* pages 71–78.