Textmining at EmoInt-2017: A Deep Learning Approach to Sentiment Intensity Scoring of English Tweets

Hardik Meisheri and **Rupsa Saha** and **Priyanka Sinha** and **Lipika Dey**
(hardik.meisheri,rupsa.s,priyanka27.s,lipika.dey)@tcs.com

Abstract

This paper describes our approach to the Emotion Intensity shared task. A parallel architecture of Convolutional Neural Network (CNN) and Long short term memory networks (LSTM) alongwith two sets of features are extracted which aid the network in judging emotion intensity. Experiments on different models and various features sets are described and analysis on results has also been presented.

1 Introduction

Sentiment analysis is an area of active research in the field of natural language processing. It aims to identify the sentiment expressed by the author of some form of textual data. Apart from the entities available in text, identification of opinion, sentiment, nuances, sarcasm etc., provide important contextual clues that help in natural language understanding and more complex information extraction tasks. The strength of the emotions expressed in text help quantify and compare subjective expressions and can be used downstream as well. Traditional fact-based approaches are rule based and prove insufficient for modern-day NLP requirements especially with large amounts of polarized short, noisy text from social media platforms such as Twitter. Twitter has become a rich source of user opinions and spread of information on this social site has far reaching consequences. Emotion Intensity task in WASSA-2016 aims to explore various approaches of determining the intensity of certain emotions expressed by a speaker via a tweet (Mohammad and Bravo-Marquez, 2017). Our approach is to explore the use of a Deep Learning framework for the same.

A significant amount of research in Natural Language Processing focuses on identifying the sentiment polarity of a given text, rather than the degree to which a given emotion is present in a text. A similar task was proposed in SemEval 2016 Task 7, and on a smaller scale in SemEval-2015 Task 10 'Sentiment Analysis in Twitter' Subtask E (Rosenthal et al., 2015).

The data for this task consists of tweets across various domains, classified into four emotions : joy, sadness, anger and fear. The training data additionally carries a real-valued score between 0 and 1 per tweet, indicating the degree of the emotion (that the tweet is classified as) the present in the tweet.

2 Related Work

In SemEval 2016 Task 7 the objective was to attribute an intensity score to English and Arabic phrases (Kiritchenko et al., 2016). Mostly supervised methods were used, with a variety of features, including different sentiment lexicons, word embeddings, point wise mutual information (PMI) scores between terms (single words and multi-word phrases), lists of words which express negation, modifiers etc. Team ECNU (Wang et al., 2016) approached it as a ranking task, using Random Forest algorithm. UWB, iLab-Edinburgh and NileTMRG all treated the task as a regression problem, and had supervised approaches. UWB used Gaussian Regression (Hercig et al., 2016), while iLab-Edinburgh went in for linear regression (Refaee and Rieser, 2016). Team LSIS (Htait et al., 2016) had a completely unsupervised approach, using sentiment lexicons and PMI scores.

Similar approaches, that is, usage of sentiment lexicons in a supervised setup, word embeddings, etc. were also seen in the proposed systems of SemEval 2015 Task 10 (Subtask E) (Rosenthal et al., 2015).

Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 193–199
Copenhagen, Denmark, September 7–11, 2017. ©2017 Association for Computational Linguistics

3 Methodology

3.1 Preprocessing

Text from tweets are inherently noisy. They contain twitter specific words along with hashtags and username mentions. Cleaning the text before further processing helps to generate better features and semantics. We employ the following preprocessing steps.

- **Hashtags** are important markers for determining sentiment or user intent. The "#" symbol is removed and the word itself is retained.
- **Username mentions**, i.e.words starting with "@", generally provide no information in terms of sentiment. Hence such terms are removed completely from the tweet. If however, the text contains multiple tweets as part of a single conversation, the user mentions would have been an important aspect.
- **Emoticons** (for example, ':(',':)', ':P' etc) are removed during embedding generation although they are retained while feature extraction.
- Extra spaces are removed.

3.2 Feature Generation

For extracting **Lexicon Features**, we follow the procedure as per the baseline system provided in the WASSA Emotion Intensity Task. The knowledge sources that have been used are: MPQA subjective lexicon (Wilson et al., 2005), Bing Liu lexicon (Ding et al., 2008), AFINN (Nielsen, 2011), Sentiment140 (Kiritchenko et al., 2014), NRC Hashtag Sentiment Lexicon (Mohammad and Kiritchenko, 2015), NRC Hashtag Emotion Association Lexicon (Mohammad et al., 2013), NRC Word-Emotion Association Lexicon (Mohammad and Turney, 2013), NRC-10 Expanded Lexicon (Bravo-Marquez et al., 2016) and the SentiWordNet (Esuli and Sebastiani, 2007). Two more features are calculated on the basis of emoticons (obtained from AFINN (Nielsen, 2011)) and negations present in the text.

Following the baseline system, we generate 45 features for each tweet, which we term as Feature Set A.

In addition to this, we use the **SentiNeuron** model proposed by (Radford et al., 2017) to generate another feature. It is an unsupervised method of generating sentiment signals. LSTM based network with 4096 units have been trained on a 82 million large Amazon reviews dataset to predict next word. Output of 2388th unit, which is sentiment signal is used as feature. This feature is then normalized between 0 to 1, and further referred to as Feature Set B.

Thus for each tweet, we arrive at 46 features generated as above. Parallel architecture of CNN and LSTM layers are used to extract important words as well as the temporal information contained in the sentence. Details of the parallel architecture are presented in subsection 3.6

3.3 Embeddings

The processed text is then converted to word embeddings. Converting text into word embeddings represents each word of the text into a d dimensional vector (Mikolov et al., 2013). We use available pre-trained embeddings which are trained on large data set. The following modules were used:

GloVe Word Embeddings - trained on 2 billion tweets from twitter (Pennington et al., 2014), vectors of 25, 50, 100 and 200 dimensions are provided as part of the pre-trained model. For this work, we use the 200 dimensional vectors. GloVe embeddings are used for the datasets corresponding to anger, fear and joy emotions.

Edinburgh Embeddings - trained on 10 million tweets for sentiment classification, they provide 400 dimensional vectors (Petrovic et al., 2010). We use them for sadness emotion.

Each tweet can further be divided in words, and we assume maximum number of words in any tweet be 35. This assumption is in line with the 140 characters limit on each tweet. Each tweet is thus represented as a $\langle 35 \times d \rangle$ matrix, where d is the output dimension of embeddings of a single word.

3.4 CNN Model

Convolution Neural Network based models have been used extensively in extracting textual features in NLP (Poria et al., 2015) (Kim, 2014). Three parallel CNN layers are employed to get bigrams, trigrams and 4-grams (Johnson and Zhang, 2014). With each of these layers two convolution filters are used to traverse through entire matrix. The width of each filter is fixed to d (the dimension of embeddings for each word), hence one dimensional convolution is used. To get a single value

from the outputs of the filters, we use Max Pooling. As mentioned earlier maximum number of words that tweet contains is assumed to be 35, max pooling values for bigrams, trigrams and 4-grams are 34, 33 and 32 respectively. Max pooling layer selects single value from each filter, therefore output of CNN architecture is 6 features for each tweet. Figure 1 shows the CNN architecture with an example sentence.

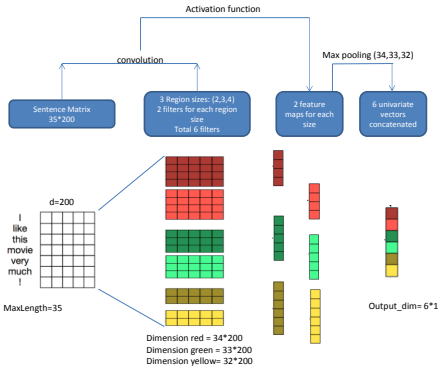

Figure 1: CNN Architecture

3.5 LSTM model

The inherent characteristics of sequence in text makes extraction of textual features a prime candidate for the use of Recurrent Neural Networks. RNNs are suited for capturing temporal relationships, which, in our case, are exhibited by words. Long short term memory networks (LSTMs) are a type of Recurrent Neural Networks which can easily capture long term dependences in a sequence, overcoming the common problem of vanishing gradient (Goldberg, 2016). Figure 2 shows the LSTM architecture with an example. Similar to CNN architecture, LSTM also receives a matrix for a tweet as input. At each step, embeddings of single word is provided. The number of LSTMs is a hyper parameter, fixed at 10 for this task. The model outputs a feature vector of dimension 10.

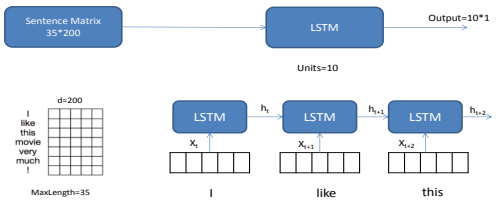

Figure 2: LSTM Architecture

3.6 Unified Model

Proposed system architecture is presented in Figure 3, which integrates convolutional neural network (CNN) and Long short term memory networks (LSTM). As shown, output of CNN and LSTM is merged, along with feature sets A and B. Before merging output of CNN layer is flatten to match dimension of other features. This is achieved through the Merge layer as shown. Output of merge layer is then propagated to fully connected neural network layer with 10 hidden units. Finally, output layer is defined with single hidden unit.

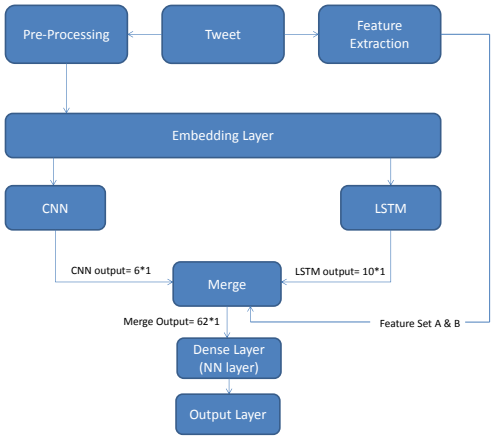

Figure 3: Merged Architecture

4 Results and Discussion

4.1 Results

Training, development and test sets each had individual files for each emotion namely, anger, fear, joy and sadness. We have trained the model separately for each emotion. Final submission for

the test set was done with unified model (CNN + LSTM + Features) with joy and anger trained with Mean Square Error as loss function and fear and sadness trained with the custom loss function. This model secured 8th rank in task.

Separate experiments were performed using CNN and LSTM layers, as well as a combination of each with features, followed by our proposed model. Pearson Correlation Coefficient and Spearman's Correlation Coefficient are used as metrics.

- LSTM layer followed by dense layer is trained with mean square error as loss function. RMSProp (Hinton et al., 2012) was used as optimizer as it is effective for Recurrent Neural Networks (RNNs). Two experiments done for this, one with features and one without.

- CNN layer followed by dense layer is trained with mean square error as loss function. Adam (Kingma and Ba, 2014) is used as the optimizer. Two experiments done for this, one with features and one without.

- The unified model, described previously, is also used in two experiments. In one, it is trained with mean square error as loss function, irrespective of emotion, and uses Adam as optimizer. The second experiment with the unified model is the proposed system, where Mean Square Error loss function is used for joy and anger and custom loss function is used for fear and sadness.

Results on the development dataset are shown in Table 1. Along with models defined above baseline results are also shown.

In order to demonstrate the difference brought about by the separate feature sets used, Table 3 shows Pearson Score on the development set with and without different sets.An identical set of experiments are conducted replacing the mean square error function with a custom loss function. Custom loss is defined as

$$loss = 1 - Pearson\ Correlation$$

Table 4 compares the results on the development set for each emotion based on the loss function used.

Table 3: Pearson Correlation results on Development Set

	$SetA\&B$	$SetB$	$SetA$	$None$
Anger	0.690	0.567	0.681	0.390
Fear	0.637	0.542	0.628	0.625
Joy	0.764	0.650	0.738	0.670
Sadness	0.556	0.527	0.573	0.372
Avg	0.661	0.571	0.655	0.514

All the above experiments are replicated on the test set. Figure 5 and Figure 4 shows experiments with different set of features with mean square error as loss function and custom loss function respectively. It is evident that trend which was evident in development set about fear and sadness emotion performing better does not hold true for test set.

Table 4: Results on Development Set

	Custom Loss		MSE	
	Pearson	*Spearman*	*Pearson*	*Spearman*
Anger	0.563	0.594	0.690	0.626
Fear	0.690	0.689	0.636	0.592
Joy	0.666	0.671	0.764	0.755
Sadness	0.649	0.658	0.556	0.573
Avg	0.642	0.653	0.661	0.636

Table 2 shows the results of different data on test set. It is observed that LSTM model outperform the unified model on test set. This points to the disparity in test and development data in terms of words. Although vocabulary was expanded to include words in test set, the sentiment relatedness is hard to capture using CNN.

4.2 Analysis

It can be seen that different feature sets play an important role in guiding the model. In Table 3 feature set A provided a significant improvement in the results whereas feature set B alone degraded the performance of the system, albeit when merged with feature set A, the results improve. Table 4 compares the results on the development set for each emotion based on the loss function used. It shows that the custom loss function performs better in fear and sadness emotions.

Table 1: Comparison of different approaches on development data

Model	Avg Pearson	Avg Spearman	Anger		Fear		Joy		Sadness	
			Per.	Spr.	Per.	Spr.	Per.	Spr.	Per.	Spr.
Baseline	0.611	0.601	0.605	0.562	0.574	0.558	0.703	0.689	0.562	0.597
CNN	0.285	0.286	-0.17	-0.08	0.278	0.231	0.636	0.628	0.395	0.361
LSTM	0.582	0.565	0.566	0.528	0.567	0.524	0.733	0.736	0.461	0.473
CNN + Features	0.650	0.641	0.674	0.668	0.539	0.508	0.753	0.728	0.630	0.658
LSTM + Features	0.671	0.653	0.668	0.612	0.638	0.596	**0.77**	**0.762**	0.609	0.642
CNN + LSTM +features	0.661	0.637	**0.690**	**0.626**	0.637	0.592	0.764	0.755	0.556	0.573
Submitted Model	**0.698**	**0.674**	**0.690**	**0.626**	**0.69**	**0.658**	0.764	0.755	**0.649**	**0.658**

Table 2: Comparison of different approaches on test data

Model	Average Pearson	Average Spearman	Anger		Fear		Joy		Sadness	
			Per.	Spr.	Per.	Spr.	Per.	Spr.	Per.	Spr.
CNN	0.384	0.382	0.237	0.255	0.364	0.361	0.391	0.396	0.544	0.516
LSTM	0.621	0.609	0.598	0.583	0.677	0.652	0.567	0.571	0.641	0.631
CNN + Features	0.645	0.630	0.597	0.586	0.651	0.629	0.648	0.639	0.682	0.667
LSTM + Features	**0.703**	**0.691**	**0.669**	**0.652**	**0.723**	**0.705**	**0.71**	**0.705**	**0.711**	**0.702**
CNN + LSTM + features	0.680	0.668	0.646	0.631	0.702	0.684	0.674	0.668	0.697	0.687
Submitted Model	0.649	0.638	0.604	0.593	0.663	0.645	0.66	0.658	0.668	0.657

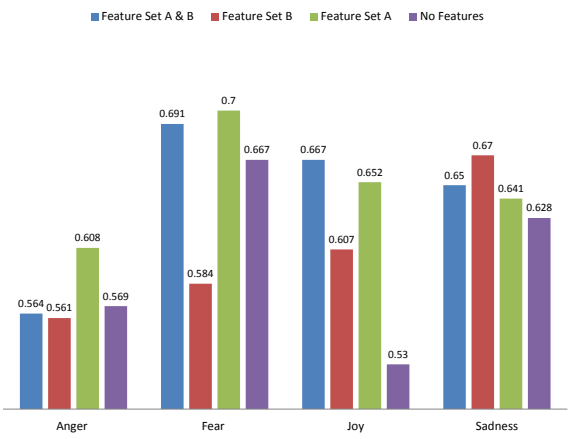

Figure 4: Results on test data using custom loss function

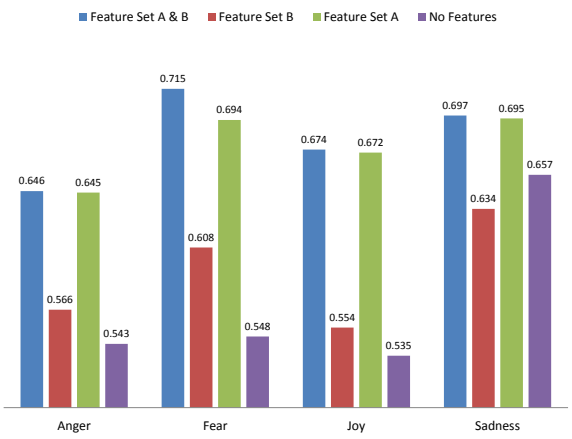

Figure 5: Results on test data using Mean Square Error function

5 Conclusion

We have applied a unified deep learning model to the emotion intensity task on twitter data. Two sets of features have been extracted using traditional NLP methods and recent deep learning based feature generation. LSTM and CNN based models have been implemented for regression task. A mixture of LSTM and CNN has been proposed. Experiments on combination of feature set on models are presented. Results shows that features help as indicated by the higher correlation. In addition to that mixture model performs better on development set while on test set LSTM model proves to be more accurate.

References

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.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*. ACM, pages 231–240.

Andrea Esuli and Fabrizio Sebastiani. 2007. Sentiwordnet: A high-coverage lexical resource for opinion mining. *Evaluation* pages 1–26.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research* 57:345–420.

T Hercig, T Brychcın, L Svoboda, and M Konkol. 2016. Uwb at semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics, San Diego, California*. pages 354–361.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent .

Amal Htait, Sebastien Fournier, and Patrice Bellot. 2016. Lsis at semeval-2016 task 7: Using web search engines for english and arabic unsupervised sentiment intensity prediction. *Proceedings of SemEval* pages 469–473.

Rie Johnson and Tong Zhang. 2014. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058* .

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Svetlana Kiritchenko, Saif M Mohammad, and Mohammad Salameh. 2016. Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases. *Proceedings of SemEval* pages 42–51.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research* 50:723–762.

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 and Felipe Bravo-Marquez. 2017. Wassa-2017 shared task on emotion intensity. In *Proceedings of the EMNLP 2017 Workshop on Computational Approaches to Subjectivity, Sentiment, and Social Media (WASSA), September 2017, 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.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543.

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.

Soujanya Poria, Erik Cambria, and Alexander F Gelbukh. 2015. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *EMNLP*. pages 2539–2544.

A. Radford, R. Jozefowicz, and I. Sutskever. 2017. Learning to Generate Reviews and Discovering Sentiment. *ArXiv e-prints* .

Eshrag Refaee and Verena Rieser. 2016. ilab-edinburgh at semeval-2016 task 7: A hybrid approach for determining sentiment intensity of arabic twitter phrases. *Proceedings of SemEval* pages 474–480.

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.

Feixiang Wang, Zhihua Zhang, and Man Lan. 2016. Ecnu at semeval-2016 task 7: An enhanced supervised learning method for lexicon sentiment intensity ranking. *Proceedings of SemEval* pages 491–496.

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.