E-LSTM at SemEval-2019 Task 3: Semantic and Sentimental Features Retention for Emotion Detection in Text

Harsh Patel
Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar
201701021@daiict.ac.in

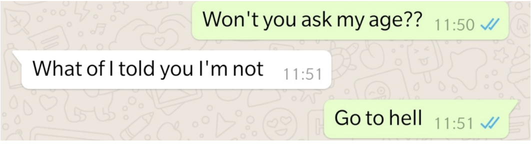
Abstract

This paper discusses the solution to the problem statement of the SemEval19: EmoContext competition(Chatterjee et al., 2019b) which is ”Contextual Emotion Detection in Texts”. The paper includes the explanation of an architecture that I created by exploiting the embedding layers of Word2Vec and GloVe using LSTMs as memory unit cells which detects approximate emotion of chats between two people in the English language provided in the textual form. The set of emotions on which the model was trained was Happy, Sad, Angry and Others. The paper also includes an analysis of different conventional machine learning algorithms in comparison to E-LSTM.

1 Introduction

Emotions are the basic human quality that almost every human possesses. According, to a recent study by Glasgow University[1], human emotions can be divided into six basic classes which are happiness, sadness, anger, fear, surprise and disgust, surprise being the most difficult one as both positive and negative statements can lead to a sense of surprise. For example, the statement *Your application for CSE branch in Stanford University is accepted* is positive and it leads to surprise whereas the statement *Your brother met with an accident* is a negative statement which all leads to a surprise.

Problem Statement: *Given a text for three turn conversation, classify the emotion of the text in the following four categories - Happy, Sad, Angry, Others.*


Won't you ask my age?? 11:50

What of I told you I'm not 11:51

Go to hell 11:51

Figure 1. Example of three turn conversation[2]

Detecting human emotions only from the text is very difficult as the emotions are a combination of the situation and the facial expressions of a person(Cowie et al., 2001). So, merely classifying it from the conversation is not a very accurate way.

In this paper, I have proposed an extended approach to the original model(Chatterjee et al., 2019a) which combines deep learning along with some techniques used in Natural Language Processing(NLP) using semantic and embedding approach (Franco-Salvador et al., 2018; Shivhare and Khethawat, 2012) called as ”Emotion LSTM” or E-LSTM to detect emotions in the provided training set. The E-LSTM is a combination of both count-based and predictive techniques which are widely used in Natural Language Processing.

2 Approach

My approach in solving the given problem statement was to maintain the semantic and sentimental relationship among the words(Gupta et al., 2017). So, as shown in Figure 2, I modeled the architecture such that the lower part contains the embeddings for sentiment analysis whereas the upper part contains the embeddings for maintaining a semantic relationship. The embeddings are then passed onto a network of LSTM layers which memorize the relationship among the words. The output of the final LSTM cell is then flattened and is combined with the output of the LSTM cell in the other half. The combined matrix is then passed as an input to a dense network with two sub-levels

[1] https://www.gla.ac.uk/news/archiveofnews/2014/february/headline_306019_en.html

[2] WhatsApp is used as a messaging platform to illustrate the three turn conversation approach

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 190–194
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

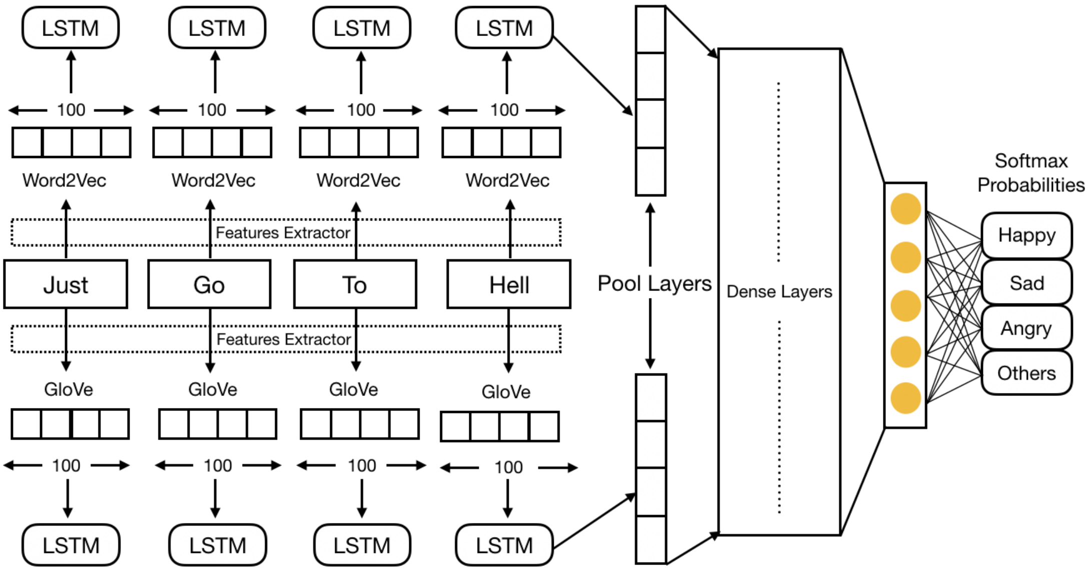


Figure 2. Architecture of E-LSTM model

Data	Labels	Happy	Sad	Angry	Others	Total
First Phase	#	4243	5463	5506	14948	30160
	%	14.07	18.11	18.26	49.56	100
Final Phase	#	142	125	150	2338	2755
	%	5.15	4.53	5.44	84.86	100

Table 1. Statistics of Training Dataset

whose output is then treated as a probability for the given four possible emotions using Softmax function.

2.1 Training Dataset

For the EmoContext SemEval-2019 Task 3, I was provided initially with a training dataset of about 30,000 entries containing 3 turn conversation and labels corresponding to each conversation. After successfully completing the first round, I was then provided with a final training dataset of about 2,700 entries. Statistics of both the datasets are shown in Table 1.

For the first phase, I proceeded with the provided dataset as a whole fro training whereas, in the second phase, I merged the provided new dataset with the dataset of Phase I and then used it for the model training.

2.2 Handling Repetition and Emoticons

After thoroughly analyzing the provided dataset, it was observed that emoticons were frequently used in the statements to describe the feeling or to end the statement. Similarly, special characters like . and * were also frequently used along with repetition. For example, You've got me blushing...☺ and Go to hell☹ statements. So, the first step of my data preprocessing was to remove the multiple instances of special characters and emoticons. So, the statement You've got me blushing...☺ after preprocessing became You've got me blushing. Other than normal preprocessing, the emoticons were also stored according to the sentence index in a dictionary and were used at the last step to verify if the predicted emotion matches partly or fully with the emotion depicted by used emojis using a weighted approach.

Word1,Word2	Word2Vec	GloVe
sad,:(	0.25	0.78
better,great	0.81	0.19

Table 2. Comparison of Word2Vec and GloVe embeddings in classifying relation among two words

2.3 Embedding Layers

The main challenge in the architecture of the model was to identify a proper embedding layer

#	Turn 1	Turn 2	Turn 3	True Label	Comments
1	You broke my heart	It was never mine to break !	See you are arrogant	sad	LSTM-Word2Vec failed because of word ”arrogant”
2	I like to cry	why are you crying	It was a joke	happy	Almost all model failed except E-LSTM model
3	You're not giving me coupon nor photo	your phone is on mute hahahha	☹☹☹☹	sad	All the models failed but the last emoji comparing technique passed for the E-LSTM model
4	its only being childish	Your username is sad. '-' hug =/	how?	others	Counting based models failed becasue of negative words

Table 3. Qualitative Analysis of baseline models along with proposed E-LSTM model

to increase the model accuracy. The initial evaluations were passed only by using the baseline structure of GloVe embedding along with LSTM layers which proved to be costly as the micro F1 score that I got was comparatively less (about 0.57 for phase I and 0.61 for phase II) whereas the training time for significantly high. So, the accuracy of the model was improved through maintaining the semantic and syntactic features of statements intact by using the two novel types of research in the Natural Language Processing field which are Word2Vec(Word to Vector)(Mikolov et al., 2013; Rahmawati and Khodra, 2016) and GloVe(Global Vectors)(Baroni et al., 2014; Pennington et al., 2014) embedding layers. The Word2Vec embedding layers maintained the sentiments of the provided text whereas the GloVe embedding maintained the semantic feature of the text. As shown in Table 2, Word2Vec was better in classifying a relationship between sad and :(as it is a predictive model and was thus trained accordingly, whereas GloVe embedding was better in classifying relation between words better and great as the approach is completely based on counting i.e. counting involved in matrices operation.

2.4 Model Training

For training my E-LSTM model, I have used Keras library. As the data was limited, I have used the K-fold cross-validation method to train the model better. For training, I used K=5 i.e. 5 fold cross-validation. This number was chosen specifically after training on the data multiple times and comparing the obtained accuracy with the training time. The most optimal hyperparameters for my model were using CrossEntropy with Softmax as my loss function along with SGD(Stochastic Gradient Descent) as an optimizer with a learning rate of 0.003. For fully connected dense layers, I used a dropout of 0.3 to prevent over-fitting of the model. The batch size that I used while training the model was 800. Apart from hyperparameters, the main thing to note while concatenation of results obtained from the LSTM layers is the Leaky-ReLU layer that I have used. Reason being some negative input values which were completely discarded by normal ReLU layer.

3 Experimental Setup

In this section, I have described the statistics of my testing data along with a comparison of the obtained results with other models. I have also discussed some of the glitches that are evident in my model in the latter half.

3.1 Test Dataset

Similar to training dataset, test dataset was also provided in both the phases i.e. initial phase and final phase. But before the System-Design submission, one gold test dataset was also provided to test the model if it's changed before paper submission. All the three test dataset files contained an index number and three turn conversation as their entry.

Model	**Happy**			**Sad**			**Angry**		
	Precision	Micro F1	F1	Precision	Micro F1	F1	Precision	Micro F1	F1
NB	45.4	56.32	50.27	74.22	70.1	72.10	43.21	38.21	40.57
SVM	75.21	32.1	45	94.45	66.66	78.16	92.11	62.21	74.26
CNN	64.3	49.32	55.82	76.21	70.12	73.04	74.12	49.44	59.32
CNN-GloVe	57.9	58.43	58.16	92.11	77.43	84.13	73.11	74.47	73.78
GloVe-LSTM	69.31	49.87	58	82.6	87.42	84.94	79.12	64.21	70.89
W2V-LSTM	75.42	45.55	56.8	84.32	78.12	81.1	80.2	64.34	71.4
E-LSTM	76.68	61.3	**64.47**	92.11	82.12	**86.83**	94.32	69.89	**80.29**

Table 4. Comparison of accuracy of different models ran on validation dataset of Task-2

3.2 Baseline Approaches

For comparison and proving my model better, I compared it with two different categories - 1. Machine Learning based and 2. Deep Learning based

For Machine Learning based baseline models, I have used Naive Bayes(NB) and Support Vector Machine(SVM). As the used models are inefficient with large datasets, so I used a subset of provided dataset to train them.

For Deep Learning based baseline models, I have used normal Convolutional Neural Networks(CNNs), CNN combined with Long Short Term Memory(LSTM) memory unit cells for data remembering, CNN combined with embedding layer of Global Vectors(GloVE) which maintains the sentiments in the text, LSTMs combined with GloVe embedding layer to find the sentiments among words and Word2Vect embedding layer combined with LSTMs which is used to maintain the semantic features in a statement. For all the deep learning baseline architectures, text in batches was given as input.

4 Results

As seen in the table Table 4, E-LSTM model outperformed all other models in both F1 score and average F1 score for all classes of emotion. Hence, it can be concluded that combining semantic and sentiment features of a statement can lead to better accuracy of emotion detection. It is also evident that Deep Learning models like CNNs, LSTMs, and RNNs are better than normal Machine Learning models like SVMs.

4.1 Qualitative Analysis

It is evident from Table 3 that E-LSTM model performed best as it tackled all the cases where counting based models when actual emotion is different from the words used in the conversation. Sentimental features also provided wrong results sometimes due to the predicted and true emotions being very close. The third entry in the table involves conversation which is highly contradicting from the true emotion. Thus, almost all the models failed in this type of case. But the verification of predicted emotion with the emoticons as described earlier saved the E-LSTM model from failing. Thus, the handcrafted features at the end of the model are very useful in this type of scenarios.

5 Conclusion

Evaluation of the given test data set shows that my model outperforms classical machine learning algorithms and also simple CNN and LSTM layers based models. Thus, it can be concluded that maintaining the semantic and syntactic relationship among words can be useful to identify emotions from texts accurately.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 238–247.

Ankush Chatterjee, Umang Gupta, Manoj Kumar Chinnakotla, Radhakrishnan Srikanth, Michel Galley, and Puneet Agrawal. 2019a. Understanding emotions in text using deep learning and big data. *Computers in Human Behavior*, 93:309–317.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019b. Semeval-2019 task 3: Emocontext: Contextual emotion detection in text. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval-2019)*, Minneapolis, Minnesota.

Roddy Cowie, Ellen Douglas-Cowie, Nicolas Tsapatsoulis, George Votsis, Stefanos Kollias, Winfried Fellenz, and John G Taylor. 2001. Emotion recognition in human-computer interaction. *IEEE Signal processing magazine*, 18(1):32–80.

Marc Franco-Salvador, Sudipta Kar, Thamar Solorio, and Paolo Rosso. 2018. Uh-prhlt at semeval-2016 task 3: Combining lexical and semantic-based features for community question answering. *arXiv preprint arXiv:1807.11584*.

Umang Gupta, Ankush Chatterjee, Radhakrishnan Srikanth, and Puneet Agrawal. 2017. A sentiment-and-semantics-based approach for emotion detection in textual conversations. *arXiv preprint arXiv:1707.06996*.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Dyah Rahmawati and Masayu Leylia Khodra. 2016. Word2vec semantic representation in multilabel classification for indonesian news article. In *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, pages 1–6. IEEE.

Shiv Naresh Shivhare and Saritha Khethawat. 2012. Emotion detection from text. *CoRR*, abs/1205.4944.