NELEC at SemEval-2019 Task 3: Think Twice Before Going Deep

Parag Agrawal*
Microsoft
paragag@microsoft.com

Anshuman Suri*
Microsoft
ansuri@microsoft.com

Abstract

Existing Machine Learning techniques yield close to human performance on text-based classification tasks. However, the presence of multi-modal noise in chat data such as emoticons, slang, spelling mistakes, code-mixed data, etc. makes existing deep-learning solutions perform poorly. The inability of deep-learning systems to robustly capture these covariates puts a cap on their performance. We propose **NELEC** : **Ne**ural and **Le**xical **C**ombiner, a system which elegantly combines textual and deep-learning based methods for sentiment classification. We evaluate our system as part of the third task of 'Contextual Emotion Detection in Text' as part of SemEval-2019 (Chatterjee et al., 2019b). Our system performs significantly better than the baseline, as well as our deep-learning model benchmarks. It achieved a micro-averaged F_1 score of 0.7765, ranking 3^{rd} on the test-set leader-board. Our code is available at https://github.com/iamgroot42/nelec

1 Introduction

Sentiment analysis of textual data: Twitter data (Kouloumpis et al., 2011; Pak and Paroubek, 2010), movie reviews (Thet et al., 2010), and product reviews (Pang et al., 2008), is perhaps the most extensively explored problem, with a plethora of research to tackle it. Novel systems utilise deep learning architectures to achieve near-human performance on clean, well-formatted data. However, sentiment classification of chat data is significantly challenging. The presence of spelling errors, slang, emoticons, code-mixing, style of writing and abbreviations makes it significantly harder for existing deep-learning models to work on such data.

*Equal contribution, order determined by coin toss

Literature dealing with this problem comprises a wide range of approaches: from hand-crafted features to end-to-end deep-learning methods. Some rule-learning based methods use keyword-based analysis (Ko and Seo, 2000) and part-of-speech tagging (Agarwal et al., 2011). These procedures require extensive human-involvement for identifying keywords and designing rules and are thus not scalable.

Non-neural machine-learning methods utilize feature extraction algorithms like *n*-grams and Tf-Idf vectors, coupled with classification algorithms like Naive Bayes (Pang et al., 2002), Decision Trees (Bilal et al., 2016), SVM (Moraes et al., 2013). These approaches perform significantly better than rule-based approaches but fail to capture context well, since they ignore the order of words in text sequences.

Statistic	Train	Dev	Test
Emojis (%)	17.6	11.1	12.5
OOV (%)	3.7	4.9	4.9
OOV(processed) (%)	2.1	1.5	1.8
Avg.Length	13.6	12.7	12.7
Avg.Length(processed)	15.7	15.3	15.2
Happy emotion (%)	14.1	5.2	5.2
Sad emotion (%)	18.1	4.5	4.5
Angry emotion (%)	18.3	5.4	5.4

Table 1: Some statistics for the given training, development and test sets.

Neural, deep-learning based approaches use architectures such as variations of recurrent models: GRU (Chung et al., 2014), LSTM (Hochreiter and Schmidhuber, 1997), BiLSTM (Schuster and Paliwal, 1997) and Convolutional models (Mundra et al., 2017), performing significantly better than other machine-learning techniques. Their ability to generalise and capture context over long se-

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 266–271
Minneapolis, Minnesota, USA, June 6–7, 2019.

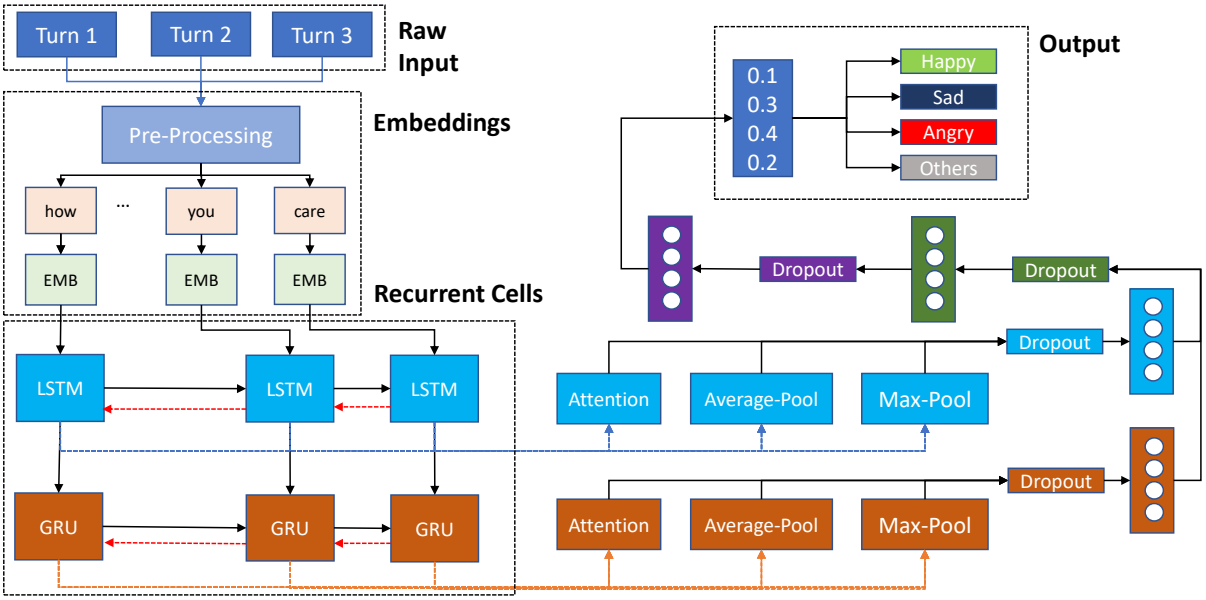

Figure 1: System diagram of the Deep-Learning model described in Section 2.1.

quences makes them a popular choice for text classification tasks.

We propose **NELEC**, a novel system specifically designed for sentiment classification. We combine lexical and neural features for sentiment classification, followed by class-specific thresholds for better labelling. Our system yields an F_1 score of 0.7765 on the test-set of Task 3 of SemEval 2019.

2 System Description

2.1 Deep Learning Model

We experiment with a two-layer, recurrent, deep-learning model with skip connections, bidirectional cells and attention (Figure 1). We trained our model for 100 epochs with Cyclic Learning Rate (Smith, 2017) scheduling. This model outperforms the baseline by a significant margin. An in-depth analysis of the cases where it fails reveals its shortcomings (along with that of a deep-learning model in general): it is not robust to misspellings and cannot capture the meaning of out-of-vocabulary words robustly. Even though pre-trained embeddings are available for most words, the context with which they are used in chat may vary from the corpora they were trained on, thus lowering their usability.

2.2 NELEC : Neural and Lexical Combiner

Since neural features have a lot of shortcomings, we shift our focus to lexical features. Using a combination of both lexical (*n*-gram features, etc.) and neural features (scores from neural classifiers), we trained a standard Light-GBM (Ke et al., 2017) Model for 100 iterations, with feature sub-sampling of 0.7 and data sub-sampling of 0.7 using bagging with a frequency of 1.0. We use $10^{-2} * \|weights\|_2$ as regularization. We also experimented with a logistic regression model, but it had a significant drop in performance for the 'happy' and 'angry' classes (Table 2). The total number of features used is 9270, out of which 9189 are sparse. The features we use in our model are described in the sections below:

2.2.1 Turn Wise Word *n*-Grams

Word level bi-grams and tri-grams (skip 1). These help capture patterns like "am happy" and automatically handles unseen data such as "am very happy" or "am so happy" because of the skip word. We take the term frequencies of these *n*-Grams as features. Word Grams not|good, hate, no|one had the highest feature gains.

2.2.2 Turn Wise Char *n*-Grams

Character level bi-grams and tri-grams. This feature helps capture character-level trends such as "haha" (and its variants), as well as emoticons. It helps with misspellings and makes the system robust to variants of several words like "haha". h|a|h, w|o|w had one of the highest feature gains.

2.2.3 Valence Arousal Dominance

We used Valence-Arousal-Dominance data (Mohammad, 2018) in the following manner:

1. Mean of Valence and Arousal values, along with turn-wise Maximum Dominance value for all words. Turn 3 Arousal for maximum dominant word had the highest feature gain.

2. Turn-wise mean of Valence, Arousal and Dominance values.

2.2.4 Emotion Intensity

We use EmoLex (Mohammad and Turney, 2010), which associates words to eight emotions and two sentiments. For each turn, we obtain the number of words having specific emotions and sentiment and use it as a feature.

Model	F_1			
	happy	sad	angry	μ_{avg}
Without Data Pre-Processing				
Deep	.5863	.5977	.6485	.6123
NELEC	.7382	.8047	.7873	**.7765**
Logistic	.6712	.7642	.7151	.7154
Baseline	.5461	.6149	.5945	.5861
With Data Pre-Processing				
Deep	.5710	.6630	.7350	.6651
NELEC	.7324	.8015	.7878	**.7736**
Logistic	.6782	.7680	.7120	.7177
Baseline	.5797	.5973	.6241	.6024

Table 2: Class-wise and micro-averaged F_1 scores for NELEC, our deep-learning model and existing baseline.

2.2.5 Neural Features

We used scores obtained by utilizing available pre-trained classifiers features:

1. Scores obtained by running conversations through a Sentiment Classifier trained on Twitter Data using SSWE embeddings (Tang et al., 2014).

2. Signals from Adult and Offensive Classifiers (Yenala et al., 2017), obtained via the Text Moderation API by Microsoft Cognitive Services. As observed in Table 2, this helps in 'Anger' detection. [1]

[1]https://docs.microsoft.com/en-in/azure/cognitive-services/content-moderator/text-moderation-api

2.3 Lexical Count Features

Lastly, we used certain count features such as the number of interrogation marks, exclamation marks, uppercase letters, the total number of words and letters for each turn. These features were observed to be very helpful while detecting anger and happiness.

3 Data Preparation

The training, development and test sets consist of 30160, 2755 and 5509 examples respectively. The final model is trained on the combined training and development set. For each instance, one of four class labels: {happy, angry, sad, other}, is provided. Table 1 provides some statistics for the given dataset.

We concatenate all three turns per conversation. For the Deep-Learning approach, a special $\langle eos \rangle$ token is inserted in between these turn-conversations.

3.1 Pre-processing for NELEC

1. **Lemmatization**: Contrary to intuition, using lemmatization decreased the final performance of our model. Further analysis suggests that emotion is highly sensitive to exact words: information captured by the word "hate" and "hated" are very different, even though a lemmatization system would reduce them to the same word, and similarly for "happy" versus "happiest". Using lemmatization drops the system's F_1 score by 0.0092.

2. **WordNet for Synonyms**: We also tried using synonyms for nouns using the Wordnet Graph (Miller, 1998). However, a similar issue plagues this approach. For instance "dog", "doggie" and "puppy" are all synonyms, but they do not express the same kind of emotion: words like "puppy" convey much more positive emotion. Using Wordnet drops the system's F_1 score by 0.0023.

3. **Normalization**: We try word tokenization and normalization by removing diacritics, numbers, stop-words, question marks etc. However, this also drops the F_1 score by 0.0046.

Character n-gram features can handle lemmatization as well as misspellings for most of the cases without discarding any additional information. Finally, we only lower-cased the sentences.

Feature Dropped	Features (#)	$F_{1_{\mu_{avg}}}$	Angry F_1	Sad F_1	Happy F_1	$F_{1_{\mu_{avg}}}$ gain
Word *n*-grams	4565	.7355	.7373	.7723	.6995	.0410
Character *n*-grams	4624	.6067	.6271	.6168	.5749	**.1698**
Valence-Arousal	15	.7444	.7125	.7426	.7160	.0321
Word-emotion Classifier	30	.7537	.7584	.7739	.7301	.0228
Pre-Built Classifier	9	.7524	.7373	.7756	.7481	.0241
Lexical Count Features	27	.7654	.7751	.8015	.7217	.0111
Turn 1 (All Features)	2578	.7417	.7173	.7716	.7106	.0348
Turn 2 (All Features)	3873	.7642	.7719	.8015	.7217	.0123
Turn 1 & 2 (All Features)	6451	.7191	.7304	.7539	.6750	.0574

Table 3: Micro-averaged F_1 scores when all features apart from these (per row) are dropped. F_1 gain here refers to the gain when using the feature mentioned, as opposed to dropping it.

3.2 Pre-processing for Deep-Learning based Approach

We use pre-trained GloVe (Pennington et al., 2014) embeddings. Some observations are:

- **Emoticons**: Around 15% of all conversations includes at least one emoticon. We use embeddings from a pre-trained emoji2vec (Eisner et al., 2016) model to handle emoticons.
- **Words with repeated characters**: This trend is common for chat-data. For example, "heelloo", "ookayy". We design specific regular expressions to handle such variations.
- **Abbreviations and slang**: tokens such as "idk", "irl" are converted to their full forms.

4 Experiments

To ascertain the novelty of our system, we report both class-wise and micro-averaged F_1 scores on the test set. We also compare our performance with the benchmarks provided by the contest organizers (Chatterjee et al., 2019a).

As mentioned in Section 3.2, data pre-processing on deep-learning models leads to significant performance gains, while leading to a drop in performance when using **NELEC**. **NELEC** outperforms both the baseline and our deep model by a considerable margin (Table 2).

4.1 Ablation Study

To analyze the usefulness of all features used by **NELEC**, we perform hold-one-out experiments on its features (Section 2.2). Results are reported in Table 3. There is a noticeable gain for most of the features, with character *n*-grams observing the maximum gain among them all.

One of the most intriguing patterns observed is the ease with which they detect sad emotion and an equal difficulty in detecting happiness.

- Words like "haha" and "okay" have several forms which all convey different magnitudes of emotion. While lemmatising such words, there is a significant loss of information.
- Most of the conversations labelled sad have easy-to-recognize signals such as negative emoticons, keywords like "lonely", which make detection easy. On the other hand, differentiating *happy* and *others* is non-trivial.
- Not using the second turn, along with its associated features, leads to a negligible drop in F_1 performance. This observation highlights the importance of the first user (in data) in analyzing sentiment. Moreover, we can utilize this information to make the feature set even smaller, making the model smaller and faster.

5 Conclusion

We propose a deep neural architecture to solve the problem of emotion detection in conversations from chat data. Although it outperforms the existing baseline, its performance is not satisfactory. To better capture lexical features and make the model robust to misspellings, abbreviations, emoticons, etc., we propose **NELEC**, a **Ne**ural and **Le**xical **C**ombiner. Our model utilises lexical features, along with signals from pre-trained neural models for sentiment and adult-offensive classification to boost performance. Our system performs at par with the existing state of the art, yielding a micro-averaged F_1 score of 0.7765 on the test set, ranking 3^{rd} on the test-set leader-board.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*, pages 30–38. Association for Computational Linguistics.

Muhammad Bilal, Huma Israr, Muhammad Shahid, and Amin Khan. 2016. Sentiment classification of roman-urdu opinions using naïve bayesian, decision tree and knn classification techniques. *Journal of King Saud University-Computer and Information Sciences*, 28(3):330–344.

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, USA.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154.

Youngjoong Ko and Jungyun Seo. 2000. Automatic text categorization by unsupervised learning. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 453–459. Association for Computational Linguistics.

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

George Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 174–184.

Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *HLT-NAACL 2010*.

Rodrigo Moraes, JoãO Francisco Valiati, and Wilson P GaviãO Neto. 2013. Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Systems with Applications*, 40(2):621–633.

Shreshtha Mundra, Anirban Sen, Manjira Sinha, Sandya Mannarswamy, Sandipan Dandapat, and Shourya Roy. 2017. Fine-grained emotion detection in contact center chat utterances. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 337–349. Springer.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.

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.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Leslie N Smith. 2017. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565.

Tun Thura Thet, Jin-Cheon Na, and Christopher SG Khoo. 2010. Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of information science*, 36(6):823–848.

Harish Yenala, Ashish Jhanwar, Manoj K Chinnakotla, and Jay Goyal. 2017. Deep learning for detecting inappropriate content in text. *International Journal of Data Science and Analytics*, pages 1–14.