ConSSED at SemEval-2019 Task 3: Configurable Semantic and Sentiment Emotion Detector

Rafał Poświata
National Information Processing Institute
al. Niepodległości 188b, 00-608 Warsaw, Poland
rafal.poswiata@opi.org.pl

Abstract

This paper describes our system participating in the SemEval-2019 Task 3: EmoContext: Contextual Emotion Detection in Text. The goal was to for a given textual dialogue, i.e. a user utterance along with two turns of context, identify the emotion of user utterance as one of the emotion classes: Happy, Sad, Angry or Others. Our system: ConSSED is a configurable combination of semantic and sentiment neural models. The official task submission achieved a micro-average F1 score of 75.31 which placed us 16th out of 165 participating systems.

1 Introduction

Emotion detection is crucial in developing a "smart" social (chit-chat) dialogue system (Chen et al., 2018). Like many sentence classification tasks, classifying emotions requires not only understanding of single sentence, but also capturing contextual information from entire conversations. For the competition we were invited to create a system for emotion detection of user utterance from short textual dialogue i.e. a user utterance along with two turns of context (Chatterjee et al., 2019b). The number of emotion classes has been limited to four (Happy, Sad, Angry and Others).

The rest of the paper is organized as follows. Section 2 briefly shows the related work. Section 3 elaborates on our approach. It shows preprocessing step and architecture of our system. Section 4 describes the data set, used word embeddings and hyper-parameters, adopted research methodology and experiments with results. Finally, Section 5 concludes our work.

2 Related Work

Detection of emotions in dialogues can be divided into two types: based only on the text of the dialogue (Chen et al., 2018) and based on many channels (video, speech, motion capture of a face, text transcriptions) (Busso et al., 2008). Regardless of the type, the most common solution is the use of neural networks, in particular variations of Recurrent Neural Networks, such as LSTMs (Hochreiter and Schmidhuber, 1997), BiLSTMs (Schuster and Paliwal, 1997) and GRUs (Cho et al., 2014) or Convolutional Neural Networks (Krizhevsky et al., 2012). Our solution uses LSTMs and BiLSTMs and is based on the ideas from SS-BED system (Chatterjee et al., 2019a).

3 Our Approach

Figure 1 provides an overview of our approach. We wanted to create a system that would benefit from the advantages of semantic and sentiment embeddings (like SS-BED). At the same time, it would be easily configurable both in terms of the selection of parameters/network architecture as well as the change of applied embeddings, both static and dynamic. In the next subsections, we describe in details our approach.

3.1 Preprocessing

For the preprocessing, we adjusted the ekphrasis tool (Baziotis et al., 2017). We use this tool for tokenization and to do the following:

- Normalize URLs, emails, percent/money/time/date expressions and phone numbers.
- Annotate emphasis and censored words and phrases with all capitalized letters.
- Annotate and reduce elongated (e.g. Whaaaat becomes <elongated> What) and repeated words (e.g. !!!!!!!!! becomes <repeated> !).

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 175–179
Minneapolis, Minnesota, USA, June 6–7, 2019.

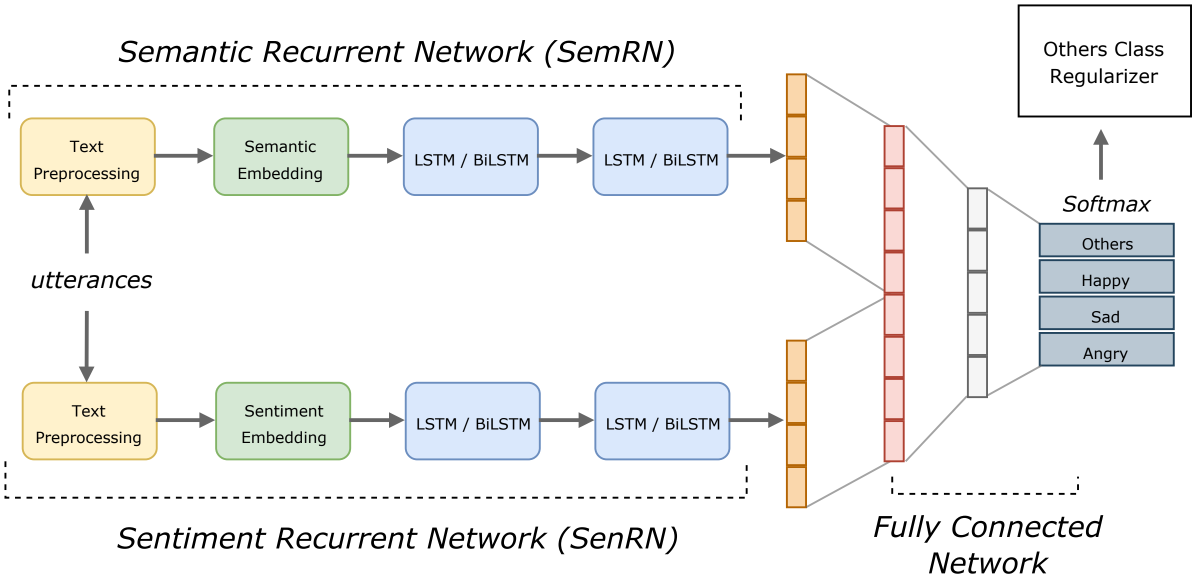

Figure 1: High level architecture of Configurable Semantic and Sentiment Emotion Detector (ConSSED).

- Unpack hashtags (e.g. #GameTime becomes <hashtag> game time </hashtag>) and contractions (e.g. "didn't" becomes "did not").
- Simplify emoticons e.g. :-] is changed to :).

We also prepare and apply dictionaries with common abbreviations and mistakes to reduce vocabulary size and deal with Out of Vocabulary (OOV) issue.

3.2 Model

Our model contains four parts: Semantic Recurrent Network (SemRN), Sentiment Recurrent Network (SenRN), Fully Connected Network and Others Class Regularizer. SemRN and SenRN are independent of each other and have similar architecture: Text Preprocessing, suitable Word Embedding and 2-layer LSTM or bidirectional LSTM (BiLSTM) - which is configurable. Outputs of those two modules are concatenated and become input for Fully Connected Network. This network has one hidden layer and Softmax layer which represents probabilities of classes. The last element of our model is Others Class Regularizer (used only during the prediction on validation/test set).

3.3 Others Class Regularizer

This component was created due to the fact that a real-life distribution is about 4% for each of Happy, Sad and Angry class and the rest is Others class. This component works by grouping records into three sets, depending on whether they are predicted as Happy, Sad or Angry. Next, for all of these sets, it checks if there are more representatives than the assumed percentage of all records. If yes, it increases the probability for Others class by 0.01 (independently in each set) until it reaches the number of representatives lower than the assumed percentage. The assumed percentage value was defined as 5.5% taking into account the validation set.

4 Experiments and Results

4.1 Data

In our work on the system, we used only official data sets made available by the organizers. However, we noticed that there are cases when conversations occur twice, but with different labels. We have removed these records and received sets which are shown in Table 1.

	Number of records
train	29977
validation	2755
test	5509

Table 1: Data sets statistics.

4.2 Word Embeddings

For our experiments, we chose five word embeddings: three semantic and two sentiment. Semantic embeddings are GloVe (Pennington et al., 2014) trained on Twitter data[1], Word2Vec (Mi-

[1] https://nlp.stanford.edu/projects/glove/

Hyper-parameter name	Possible values
SEM_LSTM_DIM	[200, 230, 256, 280, 300, 320]
SEM_FIRST_BIDIRECTIONAL	[False, True]
SEM_SECOND_BIDIRECTIONAL	[False, True]
SEN_LSTM_DIM	[200, 230, 256, 280, 300, 320]
SEN_FIRST_BIDIRECTIONAL	[False, True]
SEN_SECOND_BIDIRECTIONAL	[False, True]
HIDDEN_DIM	[100, 128, 150]
LSTM_DIM	[200, 230, 256, 280, 300, 320]
BATCH_SIZE	[32, 64, 80, 100, 128]
DROPOUT	(0.1, 0.5)
RECURRENT_DROPOUT	(0.1, 0.5)
LEARNING_RATE	(0.001, 0.004)
OTHERS_CLASS_WEIGHT	(1.0, 3.0)

Table 2: The names of hyper-parameters with possible values.

kolov et al., 2013) with ten affective dimensions trained by NTUA-SLP team as part of their solution for SemEval2018 (Baziotis et al., 2018)[2] (we call it NTUA_310) and ELMo (Peters et al., 2018) trained on 1 Billion Word Benchmark[3]. As sentiment embeddings we chose Sentiment-Specific Word Embedding (SSWE) (Tang et al., 2014)[4] and Emo2Vec (Xu et al., 2018)[5].

4.3 Hyper-parameters Search

In order to tune the hyper-parameters of our model, we adopt a Bayesian optimization by using Hyperopt library[6]. The names of hyper-parameters with possible values (list or range) are shown in Table 2. Parameters with SEM prefix apply to the Semantic Recurrent Network, and with SEN prefix to the Sentiment Recurrent Network. LSTM_DIM parameter is for BiLSTM baseline systems. In order to cope with the differences in the distribution of classes in the training set and the validation and test sets, as well as the previously mentioned actual distribution of emotion classes in relation to the Others class, apart from the use of Others Class Regularizer we also used class weight for Others class (OTHERS_CLASS_WEIGHT parameter).

4.4 Methodology

We train all models using the training set and tune the hyper-parameters using the validation set. Due to the time frame of the competition, we limited the search of hyper-parameters to 10 iterations for each model. Then, for the best parameters (found in a limited number of iterations), we once again learned this model with a training and validation set. The final model validation took place on the test set. During all experiments, we used the preprocessing described in section 3.1.

4.5 Experiments

The results of our experiments are shown in Table 3. We have divided them into two stages: validation of the baseline systems and our solution.

For the first stage, we used the 2-layer bidirectional LSTM model (BiLSTM) with all the word embedding presented in section 4.2 and compared this approach to the baseline model prepared by the organizers (Baseline). The model using NTUA_310 embedding (73.34) performed best, compared to the Baseline, we have an improvement of about fifteen percent. The second best model was a solution using ELMo embedding (72.42). From sentiment embeddings the best was Emo2Vec (71.18).

The second stage was focused on the validation of the ConSSED model. In this experiment, we trained six models to verify all possible pairs of semantic embedding-sentiment embedding. The results show that the use of the ConSSED model allows better results than corresponding baseline systems. As we could have guessed from the first stage, the best was a combination of NTUA_310 and Emo2Vec (75.31), which was our official solution during the competition. In parentheses, we presented the results without the use of Others Class Regularizer. As we can see, the use of this component improves the results but only slightly. In addition, after the competition, we have rerun the search for hyper-parameters (this time increasing the number of iterations) for the ConSSED-

[2] https://github.com/cbaziotis/ntua-slp-semeval2018
[3] https://tfhub.dev/google/elmo/2
[4] http://ir.hit.edu.cn/~dytang/
[5] https://github.com/pxuab/emo2vec_wassa_paper
[6] https://hyperopt.github.io/hyperopt/

	Happy F1	Sad F1	Angry F1	Avg. F1
Baseline	54.61	61.49	59.45	58.61
BiLSTM-GloVe	59.62	67.16	73.64	67.39
BiLSTM-ELMo	67.99	74.69	74.35	72.42
BiLSTM-NTUA_310	70.29	77.21	73.07	73.34
BiLSTM-SSWE	66.34	71.54	69.07	68.86
BiLSTM-Emo2Vec	69.48	73.27	70.93	71.18
ConSSED-GloVe-SSWE	68.48 (67.86)	74.91 (69.69)	76.54 (74.00)	73.30 (70.62)
ConSSED-GloVe-Emo2Vec	68.46 (68.46)	77.51 (77.51)	73.21 (71.39)	72.90 (72.18)
ConSSED-ELMo-SSWE	69.27 (69.16)	79.30 (79.30)	74.88 (73.32)	74.27 (73.60)
ConSSED-ELMo-Emo2Vec	71.30 (71.30)	76.05 (76.05)	76.67 (76.50)	74.69 (74.68)
ConSSED-NTUA_310-SSWE	70.69 (70.69)	78.13 (78.13)	75.54 (74.92)	74.66 (74.45)
ConSSED-NTUA_310-Emo2Vec	**69.69 (69.69)**	**78.39 (78.39)**	**77.67 (76.95)**	**75.31 (75.10)**
*ConSSED-NTUA_310-Emo2Vec	72.66 (72.66)	79.60 (79.60)	77.80 (76.83)	76.64 (76.31)

Table 3: Results of our experiments on the test set. The values without the use of Others Class Regularizer are shown in parentheses. Bolded model indicate our official solution in the competition. Experiment with an asterisk was carried out after the end of the competition.

	Competition Model	Best Model
Avg. F1	75.31	76.64
SEM_LSTM_DIM	320	320
SEM_FIRST_BIDIRECTIONAL	True	True
SEM_SECOND_BIDIRECTIONAL	False	False
SEN_LSTM_DIM	256	280
SEN_FIRST_BIDIRECTIONAL	True	True
SEN_SECOND_BIDIRECTIONAL	True	True
HIDDEN_DIM	150	150
BATCH_SIZE	100	100
DROPOUT	0.30328	0.34468
RECURRENT_DROPOUT	0.31007	0.29362
LEARNING_RATE	0.00338	0.00333
OTHERS_CLASS_WEIGHT	2.41235	2.63698

Table 4: Comparison between two ConSSED-NTUA_310-Emo2Vec models: official **Competition Model** and **Best Model** trained after the end of the competition.

NTUA_310-Emo2Vec model, which give us a better result than our official competition result (76.64). Hyper-parameters found for ConSSED-NTUA_310-Emo2Vec models and differences between them are shown in Table 4.

4.6 Competition Results

The best result we have obtained on official leaderboard is equal to 75.31 according to micro-averaged F1 score. Our solution is ranked 16th out of 165 participating systems.

5 Conclusion

In this paper, we present Configurable Semantic and Sentiment Emotion Detector (ConSSED) - our system participating in the SemEval-2019 Task 3. ConSSED has achieved good results, and subsequent studies show that it can achieve even better which results from a further search for hyper-parameters. We think that the use of fine-tuned ELMo model (e.g. by Twitter data) would improve the result even more. In addition, we would like to integrate our system with the BERT embedding (Devlin et al., 2018).

For developing our system we used Keras[7] with TensorFlow[8] as backend. We make our source code available at https://github.com/rafalposwiata/conssed.

[7] https://keras.io/
[8] https://www.tensorflow.org/

References

Christos Baziotis, Athanasiou Nikolaos, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. 2018. NTUA-SLP at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, New Orleans, Louisiana, June 5-6, 2018*, pages 245–255.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task

4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeannette Chang, Sungbok Lee, and Shrikanth Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42:335–359.

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.

Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Ting-Hao K. Huang, and Lun-Wei Ku. 2018. Emotionlines: An emotion corpus of multi-party conversations. *CoRR*, abs/1802.08379.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45:2673–2681.

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)*, pages 1555–1565, Baltimore, Maryland. Association for Computational Linguistics.

Peng Xu, Andrea Madotto, Chien-Sheng Wu, Ji Ho Park, and Pascale Fung. 2018. Emo2vec: Learning generalized emotion representation by multi-task training. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 292–298. Association for Computational Linguistics.