CLaC Lab at SemEval-2019 Task 3: Contextual Emotion Detection Using a Combination of Neural Networks and SVM

Elham Mohammadi, Hessam Amini and Leila Kosseim
Computational Linguistics at Concordia (CLaC) Lab
Department of Computer Science and Software Engineering
Concordia University, Montréal, Québec, Canada
first.last@concordia.ca

Abstract

This paper describes the *CLaC Lab* system at SemEval 2019, Task 3 (*EmoContext*), which focused on the contextual detection of emotions in a dataset of 3-round dialogues. For our final system, we used a neural network with pretrained *ELMo* word embeddings and POS tags as input, GRUs as hidden units, an attention mechanism to capture representations of the dialogues, and an SVM classifier which used the learned network representations to perform the task of multi-class classification. This system yielded a micro-averaged F1 score of 0.7072 for the three emotion classes, improving the baseline by approximately 12%.

1 Introduction

Automatic emotion detection has been the focus of much research in a variety of fields, including emotion detection based on images (Rao et al., 2019), speech signals (Davletcharova et al., 2015), electroencephalography (EEG) signals (Ackermann et al., 2016), and texts (Tafreshi and Diab, 2018).

With the advent of social media, emotion detection from text has been used to track bloggers' mental health and has been explored using different techniques, such as lexicon-based approaches and machine learning (Canales and Martínez-Barco, 2014). Lexicon-based approaches include keyword-based and ontological approaches. In a keyword-based approach (e.g. Ma et al., 2005), a specific set of opinion terms and their WordNet synonyms and antonyms are used to determine the emotion class of the text. On the other hand, ontology-based approaches (e.g. Sykora et al., 2013) try to detect emotions by taking into account the knowledge of concepts, the interconnection between them, and their final emotional impact. To achieve better generalization, the resources used in lexicon-based approaches can be employed as input features to supervised or unsupervised machine learning models.

The drawback with supervised machine learning approaches lies in the need for a large corpus of labelled data. Abdul-Mageed and Ungar (2017) collected a labelled twitter dataset of 1/4 billion tweets, spanning over 8 primary emotions, using distant supervision, i.e. collecting tweets with emotion hashtags that can be used as labels. Then using a gated recurrent neural network for classification, they achieved an accuracy of 95.68%, superior to the best previously published results by Volkova and Bachrach (2016).

Although much research has focused on the detection of emotions in tweets and blog posts (e.g. Mohammad, 2012; Desmet and Hoste, 2013; Liew and Turtle, 2016), emotion detection in dialogues, as well as in single utterances has received very little attention. This topic can have significant impact for the development of social chatbots, with the aim of creating an emotional connection between a user and a chatbot (Banchs, 2017). Task 3 of SemEval 2019 (*EmoContext*) has focussed on contextual emotion detection over 4 classes: *happy*, *sad*, *angry*, and *others* (Chatterjee et al., 2019b). We participated in this shared task under the name *CLaC Lab* and used a combination of artificial neural networks (with recurrent units and attention mechanism) and Support Vector Machines (SVM) to address this multi-class classification task.

The rest of the paper is organized as follows: Section 2 describes the overall methodology. Section 3 presents a detailed explanation of the different components of the system. Section 4 presents the results of the system. Section 5 discusses some interesting findings from our participation to this task. Finally, Section 6 is dedicated to the conclusion of this work.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 153–158
Minneapolis, Minnesota, USA, June 6–7, 2019.

2 Methodology

This section presents the overall methodology used to perform the task of contextual emotion detection. An overview of the system architecture is presented; while more detailed explanation can be found in Section 3.

2.1 Neural Architecture

The core of our system is a neural network that is trained to learn the feature representations necessary to train our final classifier. Figure 1 shows the overall architecture of the neural network that we used.

The Input As shown in Figure 1, each input sample consists of three consecutive utterances of a dialogue between two interlocutors. We consider each utterance as a sequence of tokens (words). As a result, each utterance is represented as a vector, such as $[x_{i,1}, x_{i,2}, \ldots, x_{i,t}, \ldots, x_{i,n}]$, where $x_{i,t}$ is the vector representation of the t-th word in the i-th utterance, and n is the length of the i-th utternace.

The Recurrent Component The input layer is followed by a bidirectional hidden recurrent component. Each utterance is fed to a separate hidden component, who is responsible to process that specific utterance in a forward and backward pass.

For the forward pass, the content value of the hidden component at a specific time-step, h_t, relies on both the value of the current input, x_t, and the content value of the hidden component itself at the previous time-step, h_{t-1} (Equation 1). The content value produced at this stage is then passed through another mapping function, f_y, which generates the output value of the hidden component at the current time-step, y_t (Equation 2).

$$h_t = f_h(x_t, h_{t-1}) \tag{1}$$

$$y_t = f_y(h_t) \tag{2}$$

The backward pass differs from the forward pass, in that in Equation 1, h_{t-1} is replaced by h_{t+1}, meaning that the content value of the hidden component relies on the content value at its next time-step instead of the previous one. The output calculation is identical to the forward pass (see Equation 2).

Attention Mechanism Vaswani et al. (2017) describes an attention mechanism as the weighted sum of several values (i.e. vectors), where the weight assigned to each value can be computed using a compatibility function. Using this description, the overall function of the attention mechanism for our task can be defined using Equation 3, where $\omega(y_{t'})$ refers to the weight assigned to the output of the hidden layer at time-step t', and N is the number of time-steps (i.e. the length of the utterance).

$$attn = \sum_{t'=1}^{N} y_{t'} \omega(y_{t'}) \tag{3}$$

Although originally developed for the task of machine translation (Bahdanau et al., 2014), attention mechanisms have been shown to significantly improve text classification tasks (e.g. Yang et al., 2016; Zhou et al., 2016; Wang et al., 2016; Cianflone et al., 2018). Following these works, we used attention in our system (see Figure 1).

Classification Once the neural network has created a representation of the input, a final feed-forward classification network, which takes as input the concatenated vectors from the attention units of the three utterances, performs the classification task.

2.2 Support Vector Classifier

The neural network was not used to do the final classification, but was used only as a feature extractor. This is illustrated in Figure 1 by the dotted connections between the attention units and the classifier. The extracted features were fed to an SVM (Cortes and Vapnik, 1995), which acted as the classifier.

Our main drive for using an SVM was due to explicit handling of margin size versus misclassification rate (i.e. variance versus bias). This, alongside the deterministic nature of an SVM and its faster training process (in comparison to a neural network), enabled us to play with its several configurations in order to find the optimal one for our task.

3 System Overview

In this section, we provide detailed information on the final system's architecture.

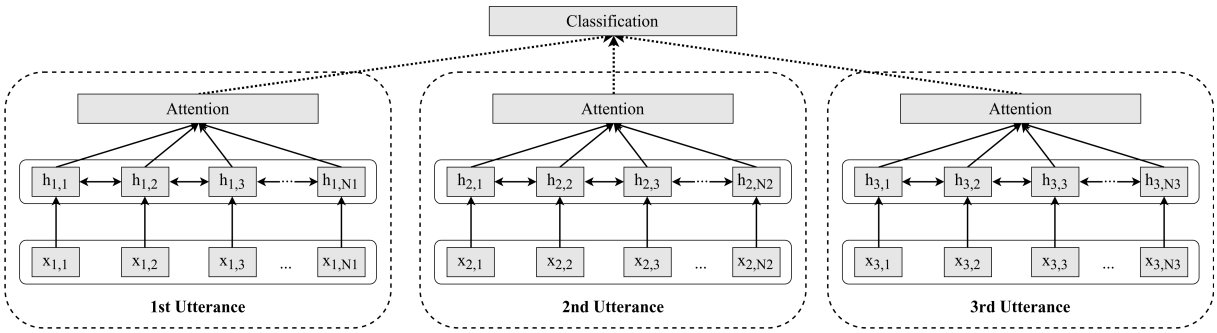

Figure 1: The overall framework of the neural network model.

3.1 The Neural Network

We developed the neural network using *PyTorch* (Paszke et al., 2017). Detailed explanations of the neural network's architecture are provided below.

Input Features Each word in each utterance is sent to the neural network by using their word embeddings, concatenated to their one-hot part-of-speech (POS) tag.

For word embeddings, we used a pretrained *ELMo* word embedder (Peters et al., 2018), which extracts the embeddings of each token in a textual context from their constituent characters. The suitability of *ELMo* for the current task lies in its ability to take into consideration the context of the tokens when generating the word embeddings, and also the handling of out-of-vocabulary words. We used pretrained *ELMo* embeddings of size 1024.

We followed the Penn Treebank tagset standard (Marcus et al., 1993) for assigning POS tags to each token. For this, we used *spaCy*[1] for tokenization and POS tagging of the data.

Recurrent Unit The system uses the bidirectional Gated Recurrent Unit (GRU) architecture, proposed by Cho et al. (2014), and stacks two layers of 25 bidirectional GRUs for the recurrent component of each utterance.

Attention Layer Equations 4, 5 and 6 show the mechanisms used in the attention layer. Using Equation 4, the weight matrix w is applied to the output of the GRU component at each time-step, y_t, which maps each output vector of the recurrent unit (with the size of 2×25 in our case) to a single value, ν_t. Then, using Equation 5, the weights, ω, are calculated. These will be used to calculate the output of the attention layer in Equation 6 (N represents the number of time-steps for each utterance, i.e. the length of the utterance).

$$\nu_t = y_t \times w \tag{4}$$

$$\omega = Softmax([\nu_1, \nu_2, \nu_3, \ldots, \nu_N]) \tag{5}$$

$$attn = \sum_{t'=1}^{N} \omega_{t'} y_{t'} \tag{6}$$

The Classifier The outputs of the attention layers from the three utterances are concatenated, and fed to a fully-connected feed-forward layer, which uses a softmax activation function at the end. The output of the classifier includes a vector of 4 reals, which represent the estimated probability for each of the 4 classes (*happy*, *sad*, *angry*, and *others*).

Optimization Technique Cross-entropy is used as the loss function and weights are applied to each class proportional to the inverse of number of their samples, in order to handle the unbalanced distribution of the four different classes over the data. The Adam optimizer (Kingma and Ba, 2014) was used and the learning rate was set to 10^{-4}. For computational reasons, minibatches of size 32 were used for training the neural network model, and different sequence sizes was handled by zero-padding.

Experiments showed that, when trained on the training dataset only, the neural network reaches its maximum performance on the development dataset (in our case, the micro-averaged F1 score for the three emotion classes) in approximately 7 epochs.

[1] https://spacy.io/

3.2 The SVM

The *scikit-learn* library (Pedregosa et al., 2011) was used for the SVM, which utilized a polynomial kernel with degree of 4. The γ parameter was set to the inverse of the number of features, which was 1/150 in our case (since the model uses 50 features from each utterance's attention layer, and there were 3 utterances for each sample), and set the penalty parameter C equal to 2.5. To train the SVM, we used the samples from both the training and the development datasets.

4 Results

We tested the system using two different configurations: 1) Using the neural network for feature extraction and classification (*NN*); and 2) Using the neural network for feature extraction and the SVM for classification (*NN+SVM*). We compared the two systems with the baseline system provided by the *EmoContext* organizers (Chatterjee et al., 2019a), which uses a neural network with LSTM units (Hochreiter and Schmidhuber, 1997) in the hidden layer and *GloVe* embeddings (Pennington et al., 2014). Table 1 shows the results from our two models in comparison to the baseline configuration.

System	*angry*	*happy*	*sad*	Micro
NN+SVM	**0.7130**	**0.6667**	**0.7443**	**0.7072**
NN	0.6206	0.6374	0.6800	0.6430
Baseline	N/A	N/A	N/A	0.5868

Table 1: F1 scores on the shared task test dataset for each emotion class and the micro-average. The final system (*NN+SVM*) is highlighted in bold.

The results in Table 1 show that, both our system configurations outperformed the baseline system, while the *NN+SVM* is significantly better than the others. We hypothesize that, given the same set of features, an SVM constitutes a stronger classifier due to its deterministic and more robust nature, and also due to its explicit design to optimize the margin size between different classes.

5 Discussion

Several interesting findings are worthy of discussion:

- The use of LSTMs in the neural network design instead of GRU yielded lower results with the development dataset. We believe that this is because the LSTM model was more prone to overfitting due to a higher number of parameters. Also, since most of the utterances were quite short (5.19 tokens on average), a GRU was enough to capture the necessary information.
- The use of POS tags alongside word embeddings did not help in improving the system performance, but it was helpful in stabilizing the output of the system (i.e. less performance fluctuations during training).
- For the SVM, an increase in the parameter C from its default value of 1 to 2.5 achieved slightly better results. We believe that this is due to the neural based features being quite representative of the final classes, and as a result, more penalty had to be assigned to errors during training as opposed to trying to achieve larger decision margins.

6 Conclusion

This paper presented the system that we developed for our participation to SemEval 2019, Task 3 (*EmoContext*). The task focused on detecting four classes of emotions, *happy*, *sad*, *angry*, and *others* in a dataset consisting of small dialogues between two people.

For this task, we developed a system that used pretrained *ELMo* word embeddings alongside POS tags as input features to a bidirectional GRU, followed by an attention layer and outputting a representation of a sample dialogue. This representation was, in turn, used as input to a final SVM classifier. Using this method, we could significantly outperform the baseline system, and achieved a micro-averaged F1 of 0.7072 for the three emotion classes on the test dataset.

Acknowledgments

We would like to express our gratitude to Parsa Bagherzadeh for insightful discussions throughout the course of our participation to *EmoContext*.

This work was financially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated

recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 718–728, Vancouver, Canada. Association for Computational Linguistics.

Pascal Ackermann, Christian Kohlschein, Jó Agila Bitsch, Klaus Wehrle, and Sabina Jeschke. 2016. Eeg-based automatic emotion recognition: Feature extraction, selection and classification methods. In *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6, Munich, Germany. IEEE.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Computing Research Repository*, arXiv:1409.0473.

R. E. Banchs. 2017. On the construction of more human-like chatbots: Affect and emotion analysis of movie dialogue data. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1364–1367, Kuala Lumpur, Malaysia. IEEE.

Lea Canales and Patricio Martínez-Barco. 2014. Emotion detection from text: A survey. In *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, pages 37–43, Quito, Ecuador. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Andre Cianflone, Yulan Feng, Jad Kabbara, and Jackie Chi Kit Cheung. 2018. Let's do it "again": A first computational approach to detecting adverbial presupposition triggers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 2747–2755, Melbourne, Australia. Association for Computational Linguistics.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Assel Davletcharova, Sherin Sugathan, Bibia Abraham, and Alex Pappachen James. 2015. Detection and analysis of emotion from speech signals. *Procedia Computer Science*, 58:91–96.

Bart Desmet and VéRonique Hoste. 2013. Emotion detection in suicide notes. *Expert Systems with Applications*, 40(16):6351–6358.

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

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computing Research Repository*, arXiv:1412.6980.

Jasy Suet Yan Liew and Howard R. Turtle. 2016. Exploring fine-grained emotion detection in tweets. In *Proceedings of the NAACL Student Research Workshop*, pages 73–80. Association for Computational Linguistics.

Chunling Ma, Helmut Prendinger, and Mitsuru Ishizuka. 2005. Emotion estimation and reasoning based on affective textual interaction. In *First International Conference on Affective Computing and Intelligent Interaction (ASCII 2005)*, pages 622–628, Beijing, China. Springer.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Compututational Linguistics*, 19(2):313–330.

Saif Mohammad. 2012. #emotional tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS 2017 Autodiff Workshop*, Long Beach, California, USA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

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 2014)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 2227–2237, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Tianrong Rao, Xiaoxu Li, Haimin Zhang, and Min Xu. 2019. Multi-level region-based convolutional neural network for image emotion classification. *Neurocomputing*, 333:429–439.

Martin D. Sykora, Thomas Jackson, Ann O'Brien, and Suzanne Elayan. 2013. Emotive ontology: Extracting fine-grained emotions from terse, informal messages. *IADIS International Journal on Computer Science and Information Systems*, 8(2):106–118.

Shabnam Tafreshi and Mona Diab. 2018. Emotion detection and classification in a multigenre corpus with joint multi-task deep learning. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 2905–2913, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5998–6008. Curran Associates, Inc., Long Beach, California, USA.

Svitlana Volkova and Yoram Bachrach. 2016. Inferring perceived demographics from user emotional tone and user-environment emotional contrast. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1567–1578, Berlin, Germany. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, xiaoyan zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 606–615, Austin, Texas, USA. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 1480–1489, San Diego, California, USA. Association for Computational Linguistics.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.