EmoSense at SemEval-2019 Task 3: Bidirectional LSTM Network for Contextual Emotion Detection in Textual Conversations

Sergey Smetanin
National Research University Higher School of Economics
Moscow, Russia
sismetanin@gmail.com

Abstract

In this paper, we describe a deep-learning system for emotion detection in textual conversations that participated in SemEval-2019 Task 3 "EmoContext". We designed a specific architecture of bidirectional LSTM which allows not only to learn semantic and sentiment feature representation, but also to capture user-specific conversation features. To fine-tune word embeddings using distant supervision we additionally collected a significant amount of emotional texts. The system achieved 72.59% micro-average F_1 score for emotion classes on the test dataset, thereby significantly outperforming the officially-released baseline. Word embeddings and the source code were released for the research community.

1 Introduction

Emotion detection has emerged as a challenging research problem that can make some valuable contribution not only in basic spheres like medicine, sociology and phycology but also in more innovative areas such as human-computer interaction. Nowadays, people increasingly communicate using text messages with dialogue systems, for which it is crucial to provide emotionally aware responses to users. The SemEval-2019 Task 3 "EmoContext" is focused on the contextual emotion detection in textual conversation. In EmoContext, given a textual user utterance along with 2 turns of context in a conversation, we must classify whether the emotion of the next user utterance is "happy", "sad", "angry" or "others" (4-point scale). For a detailed description see (Chatterjee et al., 2019).

In this paper, we present bidirectional LSTM for contextual emotion detection in textual conversations that participated in SemEval-2019 Task 3 "EmoContext". The proposed architecture aims to capture not only semantic and sentiment feature representation from the conversation turns, but also to capture user-specific conversation features. We avoided using traditional NLP features like sentiment lexicons and hand-crafted linguistic features by substituting them with word embeddings which were calculated automatically from the text corpora. Based on this paper, we make the following contributions[1] freely available for the research community:

- The source code of the deep-learning system for emotion detection.
- Word embeddings fine-tuned for emotional detection in short texts.

The rest of the article is organized as follows. Section 2 gives a brief overview of the related work. In section 3 we describe the proposed architecture of LSTM used in our system. Section 4 is focused on the texts pre-processing and training process. Section 5 lays emphasis on the different system architectures and approaches we have tried. In conclusion, the performance of our system and further ways of research are presented.

2 Related Work

In recent years deep learning techniques have captured the attention of researchers due to their ability to significantly outperform traditional methods in sentiment analysis task (Tang et al., 2015). This fact has also been confirmed by previous iterations of SemEval competition, where leading solutions used convolutional neural networks (CNN) and long short-term memory (LSTM) networks (Cliche, 2017; Baziotis et al., 2017, 2018) as well as transfer learning techniques (Duppada et al., 2018). However, limited research was focused

[1] https://github.com/sismetanin/emosense-semeval2019-task3-emocontext

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 210–214
Minneapolis, Minnesota, USA, June 6–7, 2019.

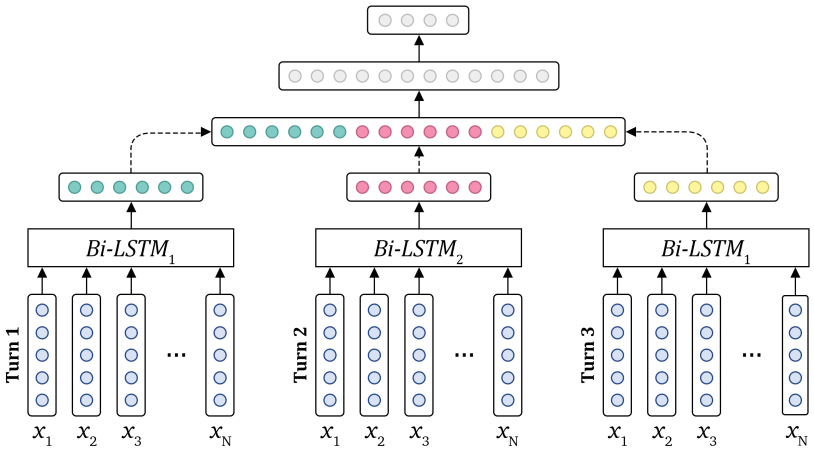

Figure 1: The architecture of a smaller version of the proposed architecture. LSTM unit for the first turn and for the third turn have shared weights.

on emotion identification in textual conversations. Since recurrent neural networks (RNNs) and their variations have been efficient in capturing sequential information, they have been successfully applied in emotion recognition systems (Poria et al., 2017; Gupta et al., 2017). Consequently, we draw our primary attention to the emotion classification in conversations using RRNs.

3 System Description

A recurrent neural network (RNN) is a family of artificial neural networks which is specialized in processing of sequential data. In contrast with traditional neural networks, RRNs are designed to deal with sequential data by sharing their internal weights processing the sequence. For this purpose, the computation graph of RRNs includes cycles, representing the influence of the previous information on the present one. As an extension of RNNs, Long Short-Term Memory networks (LSTMs) have been introduced in 1997 (Hochreiter and Schmidhuber, 1997). In LSTMs recurrent cells are connected in a special way in order to avoid vanishing and exploding gradient issues. Traditional LSTMs only preservs information from the past since they process the sequence only in one direction. Bidirectional LSTMs combine output from two hidden LSTM layers moving in opposite directions, where one moves forward through time, and another moves backwards through time, thereby enabling to capture information from both past and future states simultaneously (Schuster and Paliwal, 1997).

A high-level overview of our approach is provided in Figure 1. The proposed architecture of the neural network consists of the embedding unit and two bidirectional LSTM units $(dim = 64)$. The former LSTM unit is intended to analyze the utterance of the first user (i.e. the first turn and the third turn of the conversation), and the latter is intended to analyze the utterance of the second user (i.e. the second turn). These two units learn not only semantic and sentiment feature representation, but also how capture user-specific conversation features, which allows classifying emotions more accurately. At the first step, each user utterance is fed into corresponding bidirectional LSTM unit using pre-trained word embeddings. Next, these three feature maps are concatenated in a flatten feature vector and then passed to a fully connected hidden layer $(dim = 30)$, which analyzes interactions between obtained vectors. Finally, these features proceed through the output layer with the softmax activation function to predict a final class label. To reduce overfitting, regularization layers with Gaussian noise were added after the embedding layer, dropout layers (Srivastava et al., 2014) were added at each LSTM unit $(p = 0.2)$ and before the hidden fully connected layer $(p = 0.1)$.

4 Training

To train this model we had access to 30160 human-labelled tweets provided by task organizers, where about 5000 samples each from "angry", "sad", "happy" class and 15000 for "others" class (Table 1). Dev and test sets, which were also provided by organizers, in contrast with train set, have a real-life distribution, which is about 4% for each

Dataset	Happy	Sad	Angry	Others	Total
Train	14.07%	18.11%	18.26%	49.56%	30160
Dev	5.15%	4.54%	5.45%	84.86%	2755
Test	5.16%	4.54%	5.41%	84.90%	5509
Distant	33.3%	33.3%	33.3%	0%	900k

Table 1: Emotion class label distribution in datasets.

emotional class and the rest for the "others" class. Data provided by Microsoft.

In addition to this data, we collected 900k English tweets in order to create a distant dataset of 300k tweets for each emotion. To form the distant dataset, we based on the strategy of Go et al. (2009), under which we simply associate tweets with the presence of emotion-related words such as '#angry', '#annoyed', '#happy', '#sad, '#surprised', etc. The list of query terms was based on the query terms of SemEval-2018 AIT DISC (Duppada et al., 2018).

The key performance metric of EmoContext is a micro-average F_1 score for three emotion classes, i.e. 'sad', 'happy', and 'angry'. It is calculated as the harmonic mean of Precision and Recall.

4.1 Pre-processing

Before any training stage, texts were pre-processed by text pre-processing tool Ekphrasis (Baziotis et al., 2017). This tool helps to perform spell correction, word normalization and segmentation and allows to specify which tokens should be omitted, normalized or annotated with special tags. We used the following techniques for the pre-processing stage:

- URLs, emails, the date and time, usernames, percentage, currencies and numbers were replaced with the corresponding tags.
- Repeated, censored, elongated, and capitalized terms were annotated with the corresponding tags.
- Elongated words were automatically corrected based on built-in word statistics corpus.
- Hashtags and contractions unpacking (i.e. word segmentation) was performed based on built-in word statistics corpus.
- A manually created dictionary for replacing terms extracted from the text was used in order to reduce a variety of emotions.

In addition, Emphasis provides with the tokenizer which is able to identify most emojis, emoticons and complicated expressions such as censored, emphasized and elongated words as well as dates, times, currencies and acronyms.

4.2 Unsupervised Training

Word embeddings have become an essential part of any deep-learning approaches for NLP systems. To determine the most suitable vectors for emotions detection task, we try Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and FastText (Joulin et al., 2017) models as well as DataStories pre-trained word vectors (Baziotis et al., 2017). The key concept of Word2Vec is to locate words, which share common contexts in the training corpus, in close proximity in vector space. Both Word2Vec and Glove models learn geometrical encodings of words from their co-occurrence information, but essentially the former is a predictive model and the latter is a count-based model. In other words, while Word2Vec tries to predict a target word (CBOW architecture) or a context (Skip-gram architecture), i.e. to minimize the loss function, GloVe calculates word vectors doing dimensionality reduction on the co-occurrence counts matrix. FastText is very similar to Word2Vec except for the fact that it uses character n-grams in order to learn word vectors, so it's able to solve the out-of-vocabulary issue. For all techniques mentioned above, we used the default training prams provided by the authors. We train a simple LSTM model $(dim = 64)$ based on each of these embeddings and compare effectiveness using cross-validation. According to the result, DataStories pre-trained embeddings demonstrated the best average F_1 score.

4.3 Distant Pre-training

To enrich selected word embeddings with the emotional polarity of the words, we consider performing distant pre-training phrase by a fine-tuning of the embeddings on the automatically labelled distant dataset. The importance of using pre-training was demonstrated in (Deriu et al., 2017). We use the distant dataset to train the simple LSTM network to classify angry, sad and happy tweets. The embeddings layer was frozen for the first training epoch in order to avoid significant changes in the embeddings weights, and then it was unfrozen for the next 5 epochs. After the training stage, the fine-tuned embeddings was

System	Happy			Sad			Angry			Happy&Sad&Angry		
	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R
$Baseline$	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	58.61	n/a	n/a
$Random$	8.89	5.36	26.06	7.96	4.70	26.00	8.45	5.14	23.67	8.43	5.06	25.18
$LSTM_1$	67.07	59.07	77.68	76.95	71.57	83.36	71.18	61.75	84.67	71.37	63.31	81.89
$LSTM_2$	68.16	61.25	77.25	78.19	74.34	82.72	72.51	63.32	85.13	**72.58**	65.34	81.73
$LSTM_3$	67.33	60.70	75.77	75.23	69.60	82.00	70.06	59.08	86.27	70.58	62.34	81.41
$LSTM_s$	64.83	56.70	77.30	73.53	67.27	81.84	66.93	55.10	86.78	67.89	58.34	82.07
$LSTM_a$	66.50	58.71	77.00	75.64	71.40	80.71	69.88	59.27	85.56	70.30	62.15	81.19
$LSTM_w$	68.67	62.30	76.60	77.51	73.87	81.60	70.35	60.25	84.67	71.77	64.45	81.00

Table 2: Comparison of various models on dev dataset using micro-average Precision, Recall and F_1-score for emotional classes. $Baseline$ is an official baseline approach released by task organizers.

saved for the further training phases.

4.4 Supervised Training

At the final stage, the training dataset provided by SemEval-2019 was split into training and validation subsets. The validation subset was utilized as an unbiased accuracy evaluation of a model to fine-tune hyperparameters during training. The embedding layer was initialized with pre-trained word vectors from the previous distant training step. We use Adam optimizer (Kingma and Ba, 2014) with the initial learning rate of 0.001 and categorical cross-entropy as a loss function.

We train our network with frozen embeddings for the 15 epochs. We tried to unfrozen embeddings on the different epoch with the simultaneous reduction of learning rate but failed to get better results. It is probably connected with the size of the training dataset (Baziotis et al., 2017). The model was implemented using Keras with Tensorflow (Abadi et al., 2016) backend.

5 Experiments and Results

In the process of searching for optimal architecture, we experimented not only with the number of cells in layers, activation functions and regularization parameters but also with the architecture of the neural network. Let us take a closer look at the latter type of experiments. Comparison of various models presented in Table 2.

- $LSTM_1$ is a model with one bidirectional LSTM unit for all three conversation turns.
- $LSTM_2$ is a final model with two bidirectional LSTM units described in Section 2.
- $LSTM_3$ is a model with three bidirectional LSTM unit, where each unit is intended to analyze the corresponding conversation turn.
- $LSTM_w$ is $LSTM_2$ with an additional regularization based on class weights.
- $LSTM_s$ is $LSTM_2$ with an additional LSTM unit above concatenated layer.
- $LSTM_a$ is $LSTM_2$ with additional context-attention layer (Yang et al., 2016).

Since $LSTM_2$ demonstrated the best scores on the dev dataset, it was used in the final evaluation stage of the competition. On the final test dataset, it achieved 72.59% micro-average F_1 score for emotional classes. This is well above the official baseline released by task organizers, which was 58.68%.

6 Conclusion

In this paper, we presented the deep-learning system for emotion detection in textual conversations we used to compete in SemEval-2019 Task 3 "EmoContext" competition. Utilizing state-of-the-art approaches in the literature, we decided to use RNNs to detect emotions. We designed a specific architecture of LSTM which allows not only to learn semantic and sentiment feature representation, but also to capture user-specific conversation features. In this work, we didn't use any traditional NLP features such as sentiment lexicons or hand-crafted linguistic by substituting them with word embeddings which were calculated automatically from the text corpora with an advanced preprocessing stage.

Our approach achieved 72.59% micro-average F_1 score for emotion classes at the test dataset, thereby significantly outperform the officially-released baseline, namely larger in 14%. Further research will be focused on the advanced usage of techniques to handle imbalanced data. It also can be useful to consider the application of character-level language models.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pages 265–283, Berkeley, CA, USA. USENIX Association.

Christos Baziotis, Athanasiou Nikolaos, Pinelopi Papalampidi, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, and Alexandros Potamianos. 2018. Ntua-slp at semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive rnns. In *Proceedings of The 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 613–621. Association for Computational Linguistics.

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. Association for Computational Linguistics.

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

Mathieu Cliche. 2017. Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 573–580. Association for Computational Linguistics.

Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017. Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In *Proceedings of the 26th international conference on world wide web*, pages 1045–1052. International World Wide Web Conferences Steering Committee.

Venkatesh Duppada, Royal Jain, and Sushant Hiray. 2018. Seernet at semeval-2018 task 1: Domain adaptation for affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 18–23. Association for Computational Linguistics.

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*.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 2, pages 427–431. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2 of *NIPS'13*, pages 3111–3119, USA. Curran Associates Inc.

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. Association for Computational Linguistics.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 873–883. Association for Computational Linguistics.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Deep learning for sentiment analysis: Successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(6):292–303.

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*, pages 1480–1489. Association for Computational Linguistics.