SWAP at SemEval-2019 Task 3: Emotion detection in conversations through Tweets, CNN and LSTM deep neural networks

Marco Polignano
University of Bari A. MORO
Dept. Computer Science
E.Orabona 4, Italy
marco.polignano@uniba.it

Marco de Gemmis
University of Bari A. MORO
Dept. Computer Science
E.Orabona 4, Italy
marco.degemmis@uniba.it

Giovanni Semeraro
University of Bari A. MORO
Dept. Computer Science
E.Orabona 4, Italy
giovanni.semeraro@uniba.it

Abstract

Emotion detection from user-generated contents is growing in importance in the area of natural language processing. The approach we proposed for the EmoContext task is based on the combination of a CNN and an LSTM using a concatenation of word embeddings. A stack of convolutional neural networks (CNN) is used for capturing the hierarchical hidden relations among embedding features. Meanwhile, a long short-term memory network (LSTM) is used for capturing information shared among words of the sentence. Each conversation has been formalized as a list of word embeddings, in particular during experimental runs pre-trained Glove and Google word embeddings have been evaluated. Surface lexical features have been also considered, but they have been demonstrated to be not usefully for the classification in this specific task. The final system configuration achieved a micro F1 score of 0.7089. The python code of the system is fully available at https://github.com/marcopoli/EmoContext2019.

1 Introduction

The task of emotion detection from a text is growing in importance as a consequence of a large number of possible applications in personalized systems. This task can be considered as part of the sentiment analysis process also if it differs about the information collected. Sentiment Analysis aims to detect the polarity (positive, negative or neutral) about a topic of discussion or a specific aspect. On the contrary, Emotion Detection aims to associate an emotional label to textual content to explicitly understand what is the emotional state of the user while writing it. The final user behaviors are strongly influenced by the emotional state which she is in. Following the studies of Ekman (Ekman et al., 1987), Plutchik (Plutchik, 1990), Parrot (Parrott and Sabini, 1990), and Frijda (Frijda and Mesquita, 1994) some emotions can be considered "basics" and consequently more important than others during everyday decisions. Their identification is, therefore, one crucial aspect for applications in commerce, public health, disaster management, and trend analysis (consumer understanding). In the research area of emotion detection and sentiment analysis, many challenges are organized every for overcoming the state-of-the-art results. *SemEval* [1] is one of the most famous among them and it provides a large amount of data every year useful for supporting the research about the topic and commonly considered as state-of-the-art. Recently the best results are obtained by machine learning approaches (Colneriĉ and Demsar, 2018) based on recurrent neural networks (long short-term memory network) (Li and Qian, 2016; Wöllmer et al., 2010). These algorithms have quickly become the standard approach for solving the Emotion detection task placing great emphasis on the strategies used for formalizing the training data (Levy et al., 2015; Goldberg and Levy, 2014) and for optimizing hyper-parameters of the algorithms (Vilalta and Drissi, 2002).

2 Background and Related Work

Machine learning, and more recently deep learning algorithms, have been demonstrated to be the best option when approaching classification tasks of contents in natural language (Collobert and Weston, 2008). Example of state-of-the-art results have been achieved for hate speech detection (Zhang et al., 2018), part-of-speech tagging (Blevins et al., 2018) and name entity recognition (Chiu and Nichols, 2016; Chen et al., 2018).

Typical emotion detection systems work mostly with features directly extracted from text (Kao

[1] http://alt.qcri.org/semeval2019/

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 324–329
Minneapolis, Minnesota, USA, June 6–7, 2019.

et al., 2009). A simple vector-space strategy can often be sufficient for resolving easier tasks, but it suffers from sparsity and lack of generalization. In (Bengio et al., 2003) the author exposes the concept of word embedding summarized as a "learned distributed feature vector to represent similarity between words". This concept has been exploited by Mikolov (Mikolov et al., 2013b) through word2vec, a tool for implementing work embeddings through two standard approaches: skip-gram (Guthrie et al., 2006) and CBOW (Mikolov et al., 2013a). An alternative word embedding representation is described in (Pennington et al., 2014) as Glove trained on global word-word co-occurrence counts and able to use statistics for producing a word vector space with meaningful sub-structure. However, the use of word embeddings enriched with surface lexical features is common in sentiment classification algorithms. The relevance of these features is supported by Mohammad et al. (Mohammad et al., 2013) that produced the top ranked system at SemEval-2013 and SemEval-2014 for sentiment classification of Tweets using emotional lexicons. Moreover, word and character n-grams, number of URL, mentions, hashtags, punctuations, word and document lengths, capitalization, and more are often used for improving the classification performances (Shojaee et al., 2013). A support for a correct classification is also provided by lexical resources used for look up the sentiment of words in sentences. Linguistic features include syntactic information such as Part of Speech (PoS) which can provide relevant information for formalizing the syntactical form of the sentence. These aspects have been considered in our final classification system in order to provide a robust and updated tool for emotion detection from Tweets.

3 The EmoContext task at SemEval 2019

The EmoContext task at SemEval 2019 (Chatterjee et al., 2019) [2] aims to understand the emotion of the last turn expressed by a short dialog composed of three turns extracted from social media. The training set is composed of 30k records annotated with three main emotions: Happy, Sad, Angry and the 'other' class that includes all other not annotated emotions following a data distribution of respectively 5k, 5k, 5k, 15k. The test set is composed by 5509 records, 2,95% of the total 'Happy' , 2,68% about 'Sad', and 3,15% of 'Angry' records. The tuning of the systems has been performed over a "dev set" composed by 2755 records with a class distribution similar to the one of the test set. Evaluation has been performed by calculating micro-averaged F1 score ($\mu F1$) for the three emotion classes, i.e. Happy, Sad and Angry.

[2] https://www.humanizing-ai.com/emocontext.html

4 Classification model

The model of emotion understanding applied in this study is based on the synergy between two deep learning classification approaches: the convolutional neural networks (LeCun et al., 1989) (CNN) and the long-short-term memory networks (LSTM) (Hochreiter and Schmidhuber, 1997).

The conjunct use of a CNN and an LSTM has been demonstrated to be very efficient with textual data (Chiu and Nichols, 2016; Ordóñez and Roggen, 2016). Fig. 1 shows the complete stack of the classification model for emotion understanding. Data are provided as the input of the model through a word embedding layer. Each n-gram of the record has been mapped into a k-dimensional word embedding vector. The dimension of the word embedding is different for each strategy of encoding evaluated, and the length of the record has been truncated at max 50 tokens. Words not found in the embedding dictionary have been encoded using a randomly selected word. The output of the previous layer has been provided to a 1D convolution layer with 200 filters and a kernel of size 3x3. The activation function used is the rectified linear unit function ('ReLU') (Nair and Hinton, 2010). The output has been downsampled by a max pooling layer using a pool size of 4 along the number of tokens. The output of dimension 12x200 has been passed as input of a Bidirectional LSTM layer based, as for the CNN, on the ReLU activation function. The difference with a classic LSTM layer is the ability to find correlation among words in both the directions. In order to 'flatten' the results, we used a max pooling strategy for considering only the highest value obtained for each slot and each direction. The resultant 1x400 vector has been provided to a dense layer without activation function with the purpose to reduce the dimensionality of the vector obtained. Finally, another dense layer with a soft-max activation function has been applied for estimating the probability distribution of each of the four classes of the dataset. The model has

Layer (type)	Output Shape
embedding_1 (Embedding)	(None, 50, 600)
conv1d_1 (Conv1D)	(None, 48, 200)
max_pooling1d_1 (MaxPooling1D)	(None, 12, 200)
bidirectional_1 (Bidirectional)	(None, 12, 400)
global_max_pooling1d_1 (GlobalMaxPooling1D)	(None, 400)
dense_1 (Dense)	(None, 200)
dropout_1 (Dropout)	(None, 200)
dense_2 (Dense)	(None, 4)

Total params: 1,112,804
Trainable params: 1,112,804
Non-trainable params: 0

Figure 1: Model of emotion understanding using CNN and Bidirectional LSTM.

been trained using the categorical cross entropy loss function (Goodfellow et al., 2016) and Adam optimizer (Kingma and Ba, 2014).

5 Data processing

Each discussion in the dataset is provided as a set of three consecutive turns. We consider the dialog as a single textual content obtained concatenating the three turns into a single textual entity. Textual data have been processed for obtaining surface lexical features over the whole record. In particular, we calculate the following:

- **Statistics (RStat)**: number of tokens and characters; percent of uppercase characters and special tokens such as numbers, email, money, phone numbers, date and time, emoticons, stopwords, names, verbs, adverbs, pronoun; percent of punctuations including white spaces, exclamation points and word in a common words English dictionary [3];
- **Sentiment (RSent)**: the polarity of the record obtained through Stanford CoreNLP [4] and the percent of positive/negative words analyzed by TextBlob [5];

The textual record has been normalized before their transformation into word embeddings. We performed the correction of misspellings and the stripping of repeated characters using the Ekphrasis[6] python library. The record has been consequently tokenized using the TweetTokenizer of the "nltk" suite [7] and when required for the word embedding lookup, they have been transformed into lower case. For each token we calculate, other extra features:

- **Statistics (TStat)**: percentage upper case characters; percentage repeated characters, before the text normalization;
- **Sentiment (TSent)**: sentiment of the token obtained using TextBlob;
- **Sentiment (TLex)**: part of speech; name entity label; is exclamation mark; is question mark; is a stopword; is in a dictionary of common English Words;

The transformation of each token in a word embedding has been performed using the following pre-trained resources:

- **Google word embeddings (GoEmb)**[8]: 300 dimensionality word2vec vectors, case sensitive, composed by a vocabulary of 3 millions words and phrases that they trained on roughly 100 billion words from a Google News dataset;
- **Glove (GLEmb)**:[9]: 300 dimensionality vectors, composed by a vocabulary of 2.2 millions words case sensitive trained on data crawled from generic web pages;
- **Sentiment140 positive (SentPosEmb) and negative (SentNegEmb)**: word embeddings created over the tweets annotated in the Sentiment140 dataset [10]. We used a word2vec skip-gram strategy over a window of 5 positions, 30 epochs and considering only words counted at least five times in the dataset. We produced two word embeddings (one for positive tweets and one for negative) of 100 dimensionality vectors each case sensitive;
- **Generic Tweets (GTEmb)**: word embeddings created over 1.1 million of generic tweets in English language. As previously, we used the skip-gram strategy over a window of 5 positions, 30 epochs min word count of 5 for obtained 300 dimensionality vectors case sensitive.

[3] https://github.com/cbaziotis/ekphrasis
[4] https://stanfordnlp.github.io/CoreNLP/
[5] https://textblob.readthedocs.io/en/dev/
[6] https://github.com/cbaziotis/ekphrasis
[7] https://www.nltk.org
[8] http://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python/
[9] https://nlp.stanford.edu/projects/glove/
[10] https://www.kaggle.com/kazanova/sentiment140

	Dimensions	***Accuracy***	***Precision***	***Recall***	***μ F1***
GoEmb	300	0.87005	0.63309	0.53441	0.57958
GoEmb + SentEmb	500	0.87150	0.73141	0.53415	0.61740
GoEmb + GTEmb	**600**	**0.91742**	0.71223	**0.71016**	**0.71119**
GeEmb + SentEmb + GTEmb	800	0.91070	0.72661	0.68397	0.70465
GLEmb	300	0.87005	0.69304	0.52260	0.59587
GLEmb + SentEmb	500	0.86787	0.69784	0.51871	0.59509
GLEmb + GTEmb	600	0.86896	**0.81055**	0.53228	0.64258
GLEmb + SentEmb + GTEmb	800	0.88094	0.77697	0.56055	0.65125

Table 1: Results obtained by different formalization of records through word emebeddings.

	Dim.	**Accuracy**	**Precision**	**Recall**	**μF1**	**diff. μ F1**
GoEmb + GTEmb	***600***	***0.91742***	***0.71223***	***0.71016***	***0.71119***	**-**
all_Lex_features	638	0.85562	0.76627	0.59110	0.66738	*-0.0438*
- RStat	617	0.89574	0.71411	0.66123	0.68665	*-0.0245*
- Rsent	632	0.86214	0.73456	0.61756	0.67099	*- 0.0401*
-TStat	636	0.85146	0.77134	0.56713	0.65365	*- 0.0573*
-TSent	636	0.85214	0.74840	0.59232	0.66131	*- 0.0498*
-TLex	631	0.86467	0.78254	0.58713	0.67089	*- 0.0402*

Table 2: Results obtained by different formalization of records through word emebeddings.

6 Experiments, discussion and results

We began to configure the proposed model pointing attention on the strategy to formalize records. We decided to train our model for 10 epochs for each run using a batches size equal to 64 on the *train* dataset and validating the model on the *dev* dataset. For each run, we vary the word embedding formalization. In Tab. 1 are shown the results that allow us to observe how the concatenation of Google pre-trained word embeddings (GoEmb) and the words embeddings obtained by general tweets (GTEmb) is the most promising for the classification task in term of micro F1. It is also important to note that the value of precision obtained by the concatenation of Glove pre-trained word embeddings (GLEmb) and the GTEmb set is the higher obtained but very unbalanced with the recall. This is a clear index of the instability of the model. The second step performed in this tuning phase has been the inclusion of surface lexical features about the records and every single token. In order to understand the influence of each set of lexical features on the final micro F1 score, we performed an ablation test. The results in Tab. 2 demonstrate that lexical features, in this specific classification task and dataset do not contribute positively to the final performances of the model. As a consequence of this observation, we decided to do not use them in our model.

Following the goal to make the model robust, we decided to train it for its final configuration also on data which comes from the dev set about the classes Happy, Sad and Angry. Then we trained the model again for 10 times on 100 epochs, with a batch size of 64 using GoEmb + GTEmb for data embeddings with a validation set of 20% of training data and an early stop when the micro F1 of the validation would overcome 0.75. We obtained three final models with micro F1 respectively of 0.7714, 0.8078 and 0.78163. We used these final models to classify the test set adopting a majority vote algorithm of the predictions. This strategy has allowed us to reach a final evaluation score of 0.7089 in the final task leader-board.

7 Conclusion

In this work, we proposed a robust emotion detection classifier based on the synergy of a CNN and an LSTM deep learning algorithm. The model has been evaluated with different data formalization and configurations for finding the one which better fits the data provided for the EmoContext task at SemEval-2019. Future work will include the evaluation of other model shapes and deep learning algorithms in order to increase the final performances of the system. The source code is available at `https://github.com/marcopoli/EmoContext2019`.

8 Acknowledgment

This research has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement N. 691071.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep rnns encode soft hierarchical syntax. *arXiv preprint arXiv:1805.04218*.

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.

Guangyu Chen, Tao Liu, Deyuan Zhang, Bo Yu, and Baoxun Wang. 2018. Complex named entity recognition via deep multi-task learning from scratch. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 221–233. Springer.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Niko Colneriĉ and Janez Demsar. 2018. Emotion recognition on twitter: Comparative study and training a unison model. *IEEE Transactions on Affective Computing*.

Paul Ekman, Wallace V Friesen, Maureen O'Sullivan, Anthony Chan, Irene Diacoyanni-Tarlatzis, Karl Heider, Rainer Krause, William Ayhan LeCompte, Tom Pitcairn, Pio E Ricci-Bitti, et al. 1987. Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of personality and social psychology*, 53(4):712.

Nico H Frijda and Batja Mesquita. 1994. The social roles and functions of emotions.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT press Cambridge.

David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, and Yorick Wilks. 2006. A closer look at skip-gram modelling. In *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)*, pages 1–4.

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

Edward Chao-Chun Kao, Chun-Chieh Liu, Ting-Hao Yang, Chang-Tai Hsieh, and Von-Wun Soo. 2009. Towards text-based emotion detection a survey and possible improvements. In *Information Management and Engineering, 2009. ICIME'09. International Conference on*, pages 70–74. IEEE.

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

Yann LeCun et al. 1989. Generalization and network design strategies. *Connectionism in perspective*, pages 143–155.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Dan Li and Jiang Qian. 2016. Text sentiment analysis based on long short-term memory. In *Computer Communication and the Internet (ICCCI), 2016 IEEE International Conference on*, pages 471–475. IEEE.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115.

W Gerrod Parrott and John Sabini. 1990. Mood and memory under natural conditions: Evidence for mood incongruent recall. *Journal of personality and Social Psychology*, 59(2):321.

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.

Robert Plutchik. 1990. Emotions and psychotherapy: A psychoevolutionary perspective. In *Emotion, psychopathology, and psychotherapy*, pages 3–41. Elsevier.

Somayeh Shojaee, Masrah Azrifah Azmi Murad, Azreen Bin Azman, Nurfadhlina Mohd Sharef, and Samaneh Nadali. 2013. Detecting deceptive reviews using lexical and syntactic features. In *Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference on*, pages 53–58. IEEE.

Ricardo Vilalta and Youssef Drissi. 2002. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95.

Martin Wöllmer, Angeliki Metallinou, Florian Eyben, Björn Schuller, and Shrikanth Narayanan. 2010. Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional lstm modeling. In *Proc. INTERSPEECH 2010, Makuhari, Japan*, pages 2362–2365.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European Semantic Web Conference*, pages 745–760. Springer.