CECL at SemEval-2019 Task 3: Using Surface Learning for Detecting Emotion in Textual Conversations

Yves Bestgen
Centre for English Corpus Linguistics
Université catholique de Louvain
Place Cardinal Mercier, 10 1348 Louvain-la-Neuve Belgium
yves.bestgen@uclouvain.be

Abstract

This paper describes the system developed by the Centre for English Corpus Linguistics for the SemEval-2019 Task 3: EmoContext. It aimed at classifying the emotion of a user utterance in a textual conversation as happy, sad, angry or other. It is based on a large number of feature types, mainly unigrams and bigrams, which were extracted by a SAS program. The usefulness of the different feature types was evaluated by means of Monte-Carlo resampling tests. As this system does not rest on any deep learning component, which is currently considered as the state-of-the-art approach, it can be seen as a possible point of comparison for such kind of systems.

1 Introduction

This paper presents the participation of the Centre for English Corpus Linguistics (CECL) in the SemEval-2019 Task 3 "EmoContext: Contextual Emotion Detection in Text". The objective of the task is to classify the emotion of a user utterance in a textual conversation with a bot as happy, sad, angry or other. Contextual information is provided by means of the two previous utterances, the one just before, which was produced by the bot, and the first one of the triplet produced by the same person as the third. This task can be seen as a difficult problem in absence of any non-written information (Gupta et al., 2017).

Because of its recent and important development for the analysis of big data and especially for natural language processing, deep learning has become the preferred procedure to take up this kind of challenge (Chatterjee et al., 2019a). However, detecting emotion in texts is a well-established field of research for which unsupervised approaches have been proposed in content analysis (e.g. Anderson and McMaster, 1982; Bestgen, 1994) as well as less recent supervised approaches such as SVM (Chatterjee et al., 2019a; Pang and Lee, 2008). SVM has long been considered as the state-of-the-art in text categorization (Joachims, 2002). Therefore, it seemed interesting to get an idea of its effectiveness for the present task in comparison to deep learning approaches that should be used by many participants in this challenge. Trying to determine what level of performance can be achieved with a surface learning system was thus the main focus of this study. It should be noted that the developed system does not rely on complementary training data, nor on lexical emotional norms produced manually or automatically (Bestgen and Vincze, 2012), nor on semantic knowledge extracted from large databases or large corpora (Miller et al., 1990). However, several attempts have been made to increase its effectiveness by adding more complex features to the usual token n-grams.

The remainder of this report describes the datasets made available for this challenge, the systems developed, and the results obtained as well as the analyses performed to get a better idea of the factors that affect the system performance as well as the usefulness of the various types of features used.

2 Data

The challenge organizers divided the materials into three datasets (see Chatterjee et al. (2019b) for details):

- The learning set (Learn) that contained 30160 instances of which approximately 16.7% of the three emotion categories and the remaining 50% of Other,
- The development set (Dev) that contained 2755 instances of which approximately 5% of the emotion categories and the remaining 85% of Other

- The test set (Test) that contained twice as many instances as the Dev set and the same proportion of the four categories as in that set.

The true labels for the Learn set were available from the beginning of the training phase, those for the Dev set on December 10 and those for the Test set after the end of the challenge. Between the beginning of the development phase (August 21, 2018) and the beginning of the test phase (January 18, 2019), it was possible to use Codalab to evaluate the systems on the Dev set.

3 Systems

This section describes the systems developed to maximize the approach effectiveness. The feature extraction was performed by means of a custom SAS program running in SAS University (freely available for research at http://www.sas.com/en_us/software/university-edition.html). The predictive models used during the development phase were built on the Learn set and evaluated on the Dev set (see section 4 for a justification) by means of the L1-regularized L2-loss Support Vector Machine for classification (L1-L2-SVM) available in the LIBLINEAR package (-s 5, Fan et al., 2008).

3.1 Base System

The main part of the system consists of tokens and tokens n-grams present in the three utterances of an instance to be categorized. First, the utterances are processed by a Perl script that group the most frequent contracted forms with their corresponding full forms (e.g. I'm $>$ I am). Then, an ad-hoc tokenizer splits each sequence of characters separated by a space in tokens composed of (lowercased) letters, numbers, punctuation marks or emojis, trying not to cut the potential emoticons such as :-) and :D. In the case of tokens composed exclusively of alphabetic characters, the encoder detects the presence of the same character at least three times at the end of the token (e.g. ahhh) and keeps only one occurrence.

The features for the SVM were generated from these tokens. These are mainly unigrams and bigrams of tokens, present in each utterance, to which a tag is added to keep trace of the utterance it comes from. The tokens, bigrams, and trigrams from the first and the last utterances, those produced by the human participant, are also outputted without being distinguished by a tag so that their frequencies add up. The emojis of the first and third utterances are also processed specifically. Each token composed of several emojis is not only produced as it is, but the different emojis that compose it are also outputted separately.

Finally, a feature frequency dictionary based on all the available materials, thus also on the Test set, is produced in order to be able to use a minimum frequency threshold (set in all the analyses reported here at 2). This dictionary is also used to number the features in order to put them in LIBLINEAR format. The weight of each feature is equal to the logarithm in base 10 of its frequency in the instance to which one is added.

3.2 Extended System

The basic system was extended during the development period with a series of meta-features that seemed to improve its predictive power. They consist in a dozen simple global statistics computed on each of the utterances: number of tokens, number of characters without the spaces, number of capital letters and a potential index of emotional intensity based on the number of characters repeated at least three times, the number of emoticons and of emojis (whatever their meaning). These statistics were divided by the maximum values observed in the dataset.

Another addition was made specifically for the instances whose third utterance contained only the "yes" token: the content of the first utterance was copied to this third one.

3.3 Parameters

The regularization meta-parameter C of the SVM was optimized using a grid search. Although LIBLINEAR has a built-in program for optimizing C, we used our optimization program to have more flexibility in choosing the values to test. To take into account the differences between the frequencies of the four categories in the Learn and the other two datasets, the LIBLINEAR -wi parameter, which allows modulating the C parameter according to the category, was used. The values were set using a heuristic approach.

4 Analyses and Results

All the results reported below are expressed in terms of the challenge metric, which is the microaveraged F1 score (F1μ) for the three emotion classes (see Chatterjee et al. (2019b) for details). A first analysis was aimed to determine whether it was possible to extract several development datasets from the Learn set that produced similar performances to those obtained by using the Dev set provided by the organizers so as to limit the risk of overfitting when only one evaluation dataset is used. To this end, ten development samples were independently extracted from the Learn set in such way that they contained exactly the same number of instances in each of the four categories as in the official Dev set[1]. The results showed that the performances on these ten samples were systematically much higher than those obtained on the Dev set (0.78 vs. 0.69 for the version of the system available at that time). It would thus seem that the Dev set has specificities that distinguish it quite clearly from the Learn set and therefore all the developments have been made on the Dev set, accepting the risk of overfitting.

4.1 Official Performance on the Test Set

During the test phase, teams were allowed to submit 30 trials during 9 days. The base system (System 1) was first submitted to Codalab and obtained a F1μ of 0.721[2]. The remaining 29 trials were used to try improving performance. Three tracks were followed:

- First, I tried to optimize the number of instances assigned to each category in order to obtain better recall and precision values. To this end, I used the L1-regularized logistic regression (L1-LR) available in LIBLINEAR (-s 6) which is for the present datasets a little less effective than the L1-L2-SVM used previously but is able to estimate the probabilities of each instance belonging to each class. The most effective approach found was to classify into a category the same number of instances as the one actually presents in the Test set (the needed counts were obtained during the test phase by means of the procedure described in Note 1). This system 2 obtained a F1μ of 0.726.

- Then, the predictions of these two first systems were combined to try maximizing the F1s in each category because it was observed that System 1 was less accurate than System 2 for the Angry and Sad categories, but more accurate for the Happy category. Since the prediction differences between the two systems were systematically changes from one emotion category to the Other category or vice versa, but never from one emotion category to another one, it was easy to combine the two systems by taking the Happy predictions of System 1 and the Sad and Angry predictions of System 2 and placing the remaining instances in the Other category. This System 3 reached a F1μ of 0.73.

- Finally, System 3 was combined with a model based only on the third utterance. This model was used to move to the Other category instances assigned to one of the three emotional categories for which the probability of membership (provided by the logistic regression) was the lowest. A series of tests carried out on Codalab to optimize the decision rules made it possible to reach the final system performance of 0.736 (F1_Angry=0.7331, F1_Happy=0.7035, F1_Sad=0.7746).

4.2 Factors that Affect the System Performance

The remainder of this report analyzes the impact of several factors, including the different types of features, to the system performance. All these analyses were conducted using various versions of System 1. They aimed at categorizing the Dev set using the Learn set to build the model and the Test set using the Learn and Dev sets to build the model. In these analyses, the L1-L2 SVM was used with the parameters considered as optimal for each evaluation dataset. In order to determine if the observed differences were statistically significant, two Monte-Carlo resampling tests (Howell, 2008, Chap. 18) were used, a test for related samples to compare two different models on the same evaluation set and a test for independent samples to compare the performance on the Dev and Test sets.

A first analysis showed that using only the Learn set for predicting the Test set instead of the concatenation of the Learn and Dev sets hurt the performance (0.716 vs. 0.721), but the dif-

[1]As these analyses were performed before the labels for the Dev set had been released, I used the number of instances assigned to each category in a submission and the precision and recall for each category outputted by CodaLab for that submission to calculate the actual category frequencies.

[2]SVM parameters for this system were as following: $C = 0.67$, $w\_angry = 0.275$, $w\_happy = 0.31$ and $w\_sad = 0.275$.

	Dev Set			Test Set		
System description	**F1μ**	**Diff.**	**p**	**F1μ**	**Diff.**	**p**
Full (System 1)	0.762			0.721		
Without Meta-Features	0.752	-0.010	0.018	0.719	-0.002	0.715
Without 3grams	0.752	-0.010	0.007	0.719	-0.002	0.621
Without 2grams and 3grams	0.735	-0.027	0.002	0.710	-0.011	0.107
Without Utterance 2	0.751	0.012	-0.041	0.726	+0.005	0.193
Without Emojis Processing	0.746	-0.016	0.010	0.709	-0.012	0.006
Without Repeated Letters Processing	0.760	-0.002	0.508	0.722	+0.001	0.411

Table 1: F1μ, difference from the full system, and p-value for the ablation approach.

ference was far from being statistically significant ($p > 0.30$).

A second analysis showed that the system was more efficient on the Dev set (0.762) than on the Test set (0.721), a statistically significant difference ($p = 0.043$). Providing that the Dev and Test sets were randomly extracted from the same sample, this result suggests that the system suffers from overfitting since it was developed on the basis of the Dev set.

The ablation approach was then used to assess the independent contribution of each type of features to the overall performance. It consists in removing some sets of features of the model and re-evaluating it. As Table 1 shows, the impact of removing feature types was often quite different on the two evaluation sets. Suppressing most of the feature types for the Dev set produced a sizable decrease in performance while a very small decrease or even an increase in performance was observed for the Test set. Significance tests, which compare the ablated systems to the full one, confirm this analysis. Table 1 also shows that the specific treatment of emojis seemed particularly useful for both datasets.

A final analysis consisted in constructing a system from which all feature types in Table 1 were simultaneously removed except the specific treatment of emojis. This system obtained a F1μ of 0.743 on the Dev set, a decrease of 0.019, and a F1μ of 0.731 on the Test set, an increase of 0.01.

5 Conclusion

The main goal of this study was to try to determine what level of performance could be reached in the EmoContext task by employing a non-deep learning approach. The achieved F1μ was between 0.721 and 0.736, ranking the system approximately in the first quarter of the 165 teams who submitted a prediction for the Test set, yet far enough from the top teams who achieved a F1μ of 0.79. Looking at the other system description papers should allow to find out whether other teams used a surface learning approach while achieving better performance.

To conclude, it seems useful to underline an observation reported above that could have greatly affected the performance of the system, but also of other systems. The analyses carried out showed that the Dev set provided by the organizers gave rise to significantly lower performances than those obtained by using evaluation datasets from the Learn set as similar as possible to the actual Dev set. This led me to develop the system on the basis of the Dev set provided by the organizers and the consequence was a statistically significant overfit on the Dev set when compared to the Test set. In the absence of information about how the three datasets were created, it is not possible to comment on the origin of these differences.

Acknowledgments

The author wishes to thank the organizers of the EmoContext task for putting together this valuable event. The author is a Research Associate of the Fonds de la Recherche Scientifique - FNRS.

References

C. W. Anderson and G. E. McMaster. 1982. Computer assisted modeling of affective tone in written documents. *Computers and the Humanities*, 16(1):1–9.

Yves Bestgen. 1994. Can emotional valence in stories be determined from words? *Cognition and Emotion*, 8(1):21–36.

Yves Bestgen and Nadja Vincze. 2012. Checking and bootstrapping lexical norms by means of word similarity indexes. *Behavior Research Methods*, 44(4):998–1006.

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 International Workshop on Semantic Evaluation (SemEval-2019)*, Minneapolis, USA.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

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*.

David Howell. 2008. *Méthodes statistiques en sciences humaines*. De Boeck Université, Bruxelles.

Thorsten Joachims. 2002. *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer/Springer.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An On-line Lexical Database*. *International Journal of Lexicography*, 3(4):235–244.

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