ELiRF-UPV at SemEval-2019 Task 3: Snapshot Ensemble of Hierarchical Convolutional Neural Networks for Contextual Emotion Detection

José-Ángel González, Lluís-F. Hurtado, Ferran Pla
Departament de Sistemas Informàtics i Computació
Universitat Politècnica de València.
Camí de Vera, sn
46022, València
{jogonba2, lhurtado, fpla}@dsic.upv.es

Abstract

This paper describes the approach developed by the ELiRF-UPV team at SemEval 2019 Task 3: Contextual Emotion Detection in Text. We have developed a Snapshot Ensemble of 1D Hierarchical Convolutional Neural Networks to extract features from 3-turn conversations in order to perform contextual emotion detection in text. This Snapshot Ensemble is obtained by averaging the models selected by a Genetic Algorithm that optimizes the evaluation measure. The proposed ensemble obtains better results than a single model and it obtains competitive and promising results on Contextual Emotion Detection in Text.

1 Introduction

Emotion Detection problem arises in the context of conversational interactions, among two or more agents, when one agent is interested in knowing the emotional state of other agent involved in the conversation. The detection of emotions is a difficult task when the content is expressed by using only text, due to the lack of facial and hand gesture expressions, voice modulations, etc. Moreover, the task becomes more complex if the detection of emotions is applied only on a short piece of text without including context. This is because the context can act as an emotion modifier of a given turn in the conversation.

Although, researchers mainly focus on emotion detection on text in absence of context (Mohammad et al., 2018) (Klinger et al., 2018), tipically extracted from social media, recently, there are few works that approach the emotion detection in conversations by using context information (Hazarika et al., 2018b) (Majumder et al., 2018) (Hazarika et al., 2018a). These contextual systems work on long conversations where different users are involved and they use multimodal data, specifically, text, audio and video in order to address the emotion detection problem on large multi-party conversations.

In this work, we present an approach to the Semeval 2019 Task 3: Contextual Emotion Detection in Text (Chatterjee et al., 2019). This task is a simplification of the text emotion detection problem on conversations where each conversation have only three utterances. Only two different users are involved in each conversation, where the first and third turn corresponds to the first user and the second turn corresponds to the second user. The goal of this tasks is to predict the emotion of the third turn. We propose a Snapshot Ensemble (SE) of 1D Hierarchical Convolutional Neural Networks (HCNN) trained to extract useful information from 3-turn conversations. Our system was designed following some ideas of (Morris and Keltner, 2000) and (Majumder et al., 2018). Concretely, we consider the inter-turn and self-turn dependencies (Morris and Keltner, 2000) along with the context given by the preceding utterances (Majumder et al., 2018) to determine the emotion of a given turn.

2 System Description

2.1 Preprocessing

For the tokenization process, our system used TweetTokenizer from NLTK (Loper and Bird, 2002). In addition, we performed some other actions. All the text was transformed to lowercase. Multiple spaces were converted to a single space. Urls were replaced by the tag ”url”. We transformed multiple instances of punctuation marks in a single one (e.g., ”???” → ”?”). In order to extract semantic representations of the unicode emojis, they are replaced by their description using the Common Locale Data Repository (CLDR) Short Name (e.g., 🤩 → ”grinning face with star eyes”). Moreover, non relevant and common words are

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 195–199
Minneapolis, Minnesota, USA, June 6–7, 2019.

removed from these descriptions ("grinning face with star eyes" $\rightarrow$ "grinning star eyes").

2.2 Word Embeddings

It is well known that word embeddings (WE) learned from the same domain of a downstream task usually lead to obtain better results than those obtained using general domain WE. Due to the fact that we did not have sentences of the task to learn word embeddings from them, we used embeddings learned from Twitter posts because we considered that the characteristics of tweets are similar to the task language. Both of them have a noisy nature and they share common features of the internet language (slang, letter homophones, onomatopoeic spelling, emojis, lexical errors, etc.). Therefore, we used 400-dimensional WE obtained from a skip-gram model trained with 400 million tweets gathered from 1/3/2013 to 28/2/2014 (Godin et al., 2015).

2.3 Hierarchical Convolutional Neural Networks

We considered several characteristics of the task in order to design our system. First, the utterances are short and there are many short-term dependencies among these words. Therefore, we propose to use 1D Convolutional Neural Networks (Kim, 2014) (CNN) to extract a rich semantic representation of each utterance. Second, the conversations are composed only by 3 utterances, for that reason, it is not required to uses models with high capacity to learn long contexts. Thus, we propose to use another CNN on top of the first CNN that extracts sentence representations, in order to obtain representation of conversations. We called this approach Hierarchical Convolutional Neural Networks (HCNN) following the work of (Yang et al., 2016).

As input to the model, each utterance j (composed by a maximum of N words) in a conversation i is arranged in a matrix $M_j \in \mathbb{R}^{N \times d}$, where each row corresponds with a word in the utterance j, represented by using d-dimensional WE. As each conversation is a sequence of three utterances, these conversations are arranged in a 3-dimensional matrix where each channel j is the representation of the utterance j in the conversation, i.e. for the conversation i, $M_i \in \mathbb{R}^{3 \times N \times d}$. On all the matrices of M_i, 1D Dropout (Srivastava et al., 2014) was used to augment the dataset, by deleting words of each utterance with $p = 0.3$.

Given the representation of the conversation i, M_i, for each utterance independently, a CNN with kernels of different sizes is applied in order to obtain a composition of word embeddings that can extract semantic/emotional properties from each utterance. At this first level, we use $f_1 = 256$ kernels of sizes $\{2, 4, 6\}$ and their weights are shared among the three channels. From that, for each utterance, three new matrices are obtained. These matrices capture relevant features for each kernel size and utterance. These features are pooled into a vector by using 1D Global Max Pooling (GMP).

The resulting three vectors from the previous level were concatenated as rows to obtain a matrix representation of the conversation i composed by the CNN map of its sentences, $Wi \in \mathbb{R}^{3 \times f_1}$. We considered that conversation features could be relevant for the task. At this level, in order to extract these relevant features and following the ideas in (Morris and Keltner, 2000) (Majumder et al., 2018), the system is intended to take into account the context and potentially the emotions given by preceding utterances to determine the emotion expressed by the last utterance. To do this, a CNN with $f_2 = 256$ kernels of sizes $\{1, 2, 3\}$ were used. The size of the filters is crucial to understand what features the system is capable to learn.

Concretely, 3-size kernels: semantic/emotional features over all the contexts (full conversation); 2-size kernels: inter-turn features and semantic/emotional features of preceding and later utterances given a context of two utterances; 1-size kernels: self-turn features and semantic/emotional features of each utterance independently.

On the output maps of this second CNN[1], GMP is used in order to extract the most relevant features from each dimension and the resulting vectors are concatenated. Later, a fully connected layer L_1 with 512 neurons is used to fuse the concatenated vectors. Finally, to obtain a probability distribution over $\mathbb{C}$ classes ($\{$happy, sad, angry, others$\}$) we use a softmax fully connected layer L_2. Figure 1 shows the proposed model architecture.

2.4 Snapshot Ensemble

Generally, ensemble models outperform single models in similar tasks (Duppada et al., 2018) (Rozental et al., 2018). Therefore, we decided

[1] After all the CNN layers (at two levels), BatchNormalization, LeakyReLU and Dropout are applied

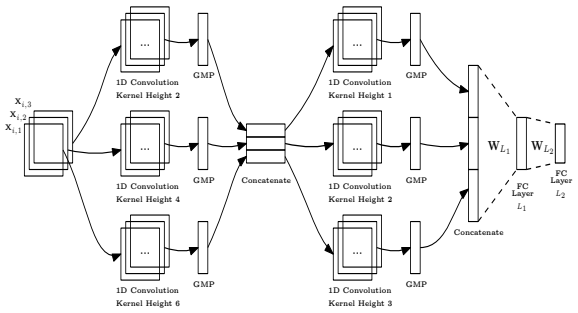

Figure 1: Hierarchical Convolutional Neural Networks.

to use ensemble methods instead of trying different architectures. We used the ideas of Snapshot Ensemble (SE) (Huang et al., 2017) to combine HCNN trained until reaching good and diverse local minima by using SGD and a cosine learning rate with $T = 24$ training iterations, $M = 6$ learning cycles, and initial learning rate $alpha = 0.4$.

From this training method, we took 24 snapshots (one for each training iteration). From the set of snapshots $S = \{s_i \ / \ 1 \leq i \leq 24 \wedge s_i : \mathbb{R}^{3\times N\times d} \rightarrow \mathbb{R}^{\mathbb{C}}\}$, we generate 4 different systems:

1. Best snapshot of all iterations

$$f_1 = \underset{s_i}{\text{argmax}} \ \mu F_1(s_i(x), y) \qquad (1)$$

2. Average of all snapshots

$$f_2 = \frac{1}{|S|} \sum_{s_i \in S} s_i(x) \qquad (2)$$

3. Average of best snapshot at each learning cycle

$$f_3 = \frac{M}{T} \sum_{i=0}^{\frac{T}{M}-1} \underset{s_i \in S\left[\frac{T}{M}\cdot i, \frac{T}{M}\cdot(i+1)\right]}{\text{argmax}} \mu F_1(s_i(x), y) \qquad (3)$$

4. Average of genetic selected snapshots

$$f_4 = \frac{1}{|g(S)|} \sum_{s_i \in g(S)} g(S)_i \ s_i(x) \qquad (4)$$

where x and y are the input and the target, respectively, and $g(S)_i$ is the decision of a genetic algorithm to include the snapshot s_i in the ensemble. We used this method in order to discretely select ($g(S)_i \in \{0, 1\}$) what snapshots are well-suited for the final averaging ensemble which tries to optimize μF_1. The genetic algorithm (Mitchell, 1998) starts with a population of 400 individuals, they are crossed by using two point crossover, mutated with flip bit and selected by using tournament selection during 100 generations. Moreover, this algorithm addresses a multi-objective problem, it must to reach combinations of snapshots whose averaged predictions yield to high values of μF_1 while minimizing the number of models in the ensemble (the final genetic ensemble is composed by 6 system, i.e. as many systems as learning cycles) These decisions were taken in order to reduce the overfiting risk during the learning of the ensemble i.e. we prioritize simpler ensembles which are composed by discretely selected snapshots.

3 Analysis of Results

In order to evaluate different configurations of our system we used the development set given by the task organizers. On this development set, ablation analysis on single HCNN was carried out in order to observe if the input Dropout and the incorporation of L_1 layer yield to better results (the capacity of HCNN must be greater when including both techniques). The results of this ablation analysis are shown in Table 1.

System	$\boldsymbol{\mu P}$	$\boldsymbol{\mu R}$	$\boldsymbol{\mu F_1}$
Vanilla	70.65	75.06	72.79
Dropout	71.78	**76.26**	73.95
$\boldsymbol{L_1}$	72.36	75.23	73.84
Dropout + $\boldsymbol{L_1}$	**75.42**	75.78	**75.60**

Table 1: Ablation analysis of input Dropout and L_1 layer on HCNN (development set)

Vanilla system is a single HCNN without input Dropout neither the L_1 layer. It can be observed that, the systems with Dropout and L_1 outperformed the **Vanilla** version of HCNN in terms of μP, μR and μF_1. In terms of μP, the systems which incorporate L_1 achieved better results. However, although **Dropout +** $\boldsymbol{L_1}$ obtained the best improvement in terms of $\boldsymbol{\mu P}$, the highest $\boldsymbol{\mu R}$ was obtained using only **Dropout**. This could indicate that data augmentation could be useful to increase the $\boldsymbol{\mu R}$ but it is required more network capacity to handle this augmentation in order to increase also the $\boldsymbol{\mu P}$.

These results were obtained by using a single HCNN with *adam* as update rule (Kingma and Ba, 2014) with default learning rate. However, the SE

training mode with Vanilla SGD and cosine learning rate, along with the proposed ensemble generation, allows the **Dropout + L_1** system to reach better results (Table 2).

Ensemble	μP	μR	μF_1
Best snapshot (single)	74.82	78.41	76.58
Average All	74.38	77.93	76.18
Best per Cycle	75.29	77.45	76.35
Genetic Average	**75.73**	**80.09**	**77.85**

Table 2: Results on development set with several SE of HCNN.

In this case, the best single model (**Best snapshot**) obtained in the SE training mode, provided higher μR than **Dropout + L_1** at the expense of a reduction in μP. This improvement of 3 points of μR yields also an increase of the μF_1 measure.

Among the ensembles, only **Genetic Average** improves the **Best snapshot** and **Dropout + L_1** systems in all the metrics. This is due to a big increase in μR. This suggests that it is possible to improve the μF_1 results by balancing μP and μR.

The other ensembles obtain lower results in terms of μR and μF_1 than **Best snapshot**, which is a single model. Moreover, all SE (including **Best snapshot**) except **Genetic Average** are less accurate (lower μP) than **Dropout + L_1**. However, all of them improved considerably the μR.

Due to the SE HCNN models generally outperformed the best single model **Dropout + L_1** in terms of μF_1 on the development set, we submitted all these systems to be evaluated on the test set. The results are shown in Table 3. It can be seen that the best system is **Genetic Average**, the same behavior observed on the development set. Although **Best snapshot** is more accurate than the ensembles (higher μP), two of the three ensembles yields better results μF_1. Moreover, a big degradation in the results are observed, all systems goes from 77 μF_1 on the development set, to 74 μF_1 on the test set.

System	μP	μR	μF_1
Best snapshot (single)	**75.69**	72.60	74.11
Average All	73.15	75.00	74.07
Best per Cycle	73.27	75.12	74.18
Genetic Average	73.43	**75.72**	**74.56**

Table 3: Results on test set with several SE of HCNN.

Table 4 shows the results of our best system (**Genetic Average**) at class level. The worse classified classes in terms of F_1 were Angry and Happy.

Class	P	R	F_1
Angry	68.73	78.19	73.16
Happy	75.19	69.37	72.16
Sad	77.82	80.00	78.90

Table 4: Results at class level of **Genetic Average** on test set.

4 Conclusion and Future Work

In this paper, we have presented Snapshot Ensembles of Hierarchical Convolutional Neural Networks to address the Semeval 2019 Task 3: Contextual Emotion Detection in Text. Our system is based on the use of a Genetic Algorithm in order to ensemble different snapshots of the same model. This ensemble outperformed single models and also classical snapshot ensembles, obtaining competitive results in the addressed task.

Due to the fact that in the proposed system, the semantic and emotional information is only provided by the representation of the words and the utterances, as future work we plan to study different word and sentence embeddings. It would be also interesting to incorporate other emotional or sentiment features such as: Sentiment Unit (Radford et al., 2017), DeepMoji (Felbo et al., 2017), Sentiment Specific WE (Tang et al., 2014); or polarity lexicons. Moreover, we are also interested in work with more powerful word embeddings such as BERT (Devlin et al., 2018) in order to incorporate a richer semantic word representation.

Acknowledgments

This work has been partially supported by the Spanish MINECO and FEDER founds under project AMIC (TIN2017-85854-C4-2-R) and the GiSPRO project (PROMETEU/2018/176). Work of José-Ángel González is also financed by Universitat Politècnica de València under grant PAID-01-17.

References

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.

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*.

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*, pages 18–23. Association for Computational Linguistics.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625. Association for Computational Linguistics.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab $@$ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153. Association for Computational Linguistics.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018a. Icon: Interactive conversational memory network for multimodal emotion detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2594–2604. Association for Computational Linguistics.

Devamanyu Hazarika, Soujanya Poria, Amir Zadeh, Erik Cambria, Louis-Philippe Morency, and Roger Zimmermann. 2018b. Conversational memory network for emotion recognition in dyadic dialogue videos. In *NAACL-HLT*.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. 2017. Snapshot ensembles: Train 1, get m for free. *CoRR*, abs/1704.00109.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Roman Klinger, Orphee De Clercq, Saif Mohammad, and Alexandra Balahur. 2018. Iest: Wassa-2018 implicit emotions shared task. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 31–42. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander F. Gelbukh, and Erik Cambria. 2018. Dialoguernn: An attentive rnn for emotion detection in conversations. *CoRR*, abs/1811.00405.

Melanie Mitchell. 1998. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, USA.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17. Association for Computational Linguistics.

Michael W. Morris and Dacher Keltner. 2000. How emotions work: The social functions of emotional expression in negotiations. *Research in Organizational Behavior*, 22:1 – 50.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444.

Alon Rozental, Daniel Fleischer, and Zohar Kelrich. 2018. Amobee at iest 2018: Transfer learning from language models. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 43–49. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

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. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*.