SCIA at SemEval-2019 Task 3: Sentiment analysis in textual conversations using Deep Learning

Zinedine Rebiai [1]
zinedine.rebiai@epita.fr

Simon Andersen [1]
simon.andersen@epita.fr

Antoine Debrenne [1]
antoine.debrenne@epita.fr

Victor Lafargue [1]
victor.lafargue@epita.fr

[1] EPITA Graduate School of Computer Science, France

Abstract

In this paper we present our submission for SemEval-2019 Task 3: EmoContext. The task consisted of classifying a textual dialogue into one of four emotion classes: happy, sad, angry or others. Our approach tried to improve on multiple aspects, preprocessing with an emphasis on spell-checking and ensembling with four different models: Bi-directional contextual LSTM (**BC-LSTM**), categorical Bi-LSTM (**CAT-LSTM**), binary convolutional Bi-LSTM (**BIN-LSTM**) and Gated Recurrent Unit (**GRU**). On the leader-board, we submitted two systems that obtained a micro F1 score ($F1\mu$) of **0.711** and **0.712**. After the competition, we merged our two systems with ensembling, which achieved a $F1\mu$ of **0.7324** on the test dataset.

1 Introduction

Rapid progress in natural language processing with the rise of deep learning has brought increasing attention on tasks such as text classification and sentiment analysis. Most of the work in that field was made using social media due to the large amount of data available. The task addressed in this paper focuses on emotion detection within conversations from social media. The key point is that we need to take into account multiple speakers and capture a global emotion out of their conversation. It becomes a challenge when facing different users who each have a different way to express their emotions depending on their personalities. In a dialogue, users have an initial emotional state, and their mood will shift as the dialogue goes on. Therefore, the task of labelling a turn-based conversation with the right emotion is even more challenging.

State of the art approaches consist of using language models (Vaswani et al., 2017) to pre-train the model on the general NLP task of language modeling before fine-tuning on specific tasks like classification or translation. The language model approach used by **ULMFiT** (Howard and Ruder, 2018), **ELMO** (Peters et al., 2018) and **BERT** (Devlin et al., 2018) was especially successful for this kind of tasks. For the specific task of emotion classification in textual conversation, Gupta et al. (2017) achieved a $F1\mu$ score of 0.7134 on the same dataset, using an architecture based on **LSTM**. For sentiment analysis, other successful approaches also used **Bi-LSTM** (Baziotis et al., 2017b) as well as transfer learning (Daval-Frerot et al., 2018).

In this paper, we present two sub-systems that are composed of four deep-learning models (using **Bi-LSTM**, **GRU** and **CNN**). Those two sub-systems competed independently at SemEval-2019 Task 3 (Chatterjee et al., 2019). After the final evaluation, we merged both sub-systems, taking advantage of ensemble learning. The rest of the paper is organized as follows. Part 2 gives an overview of our approach. Our preprocessing methods, the description of our models, and our ensembling approach are all described in Part 3. Part 4 shows the obtained results and in Part 5, we give a conclusion with remarks for future works.

2 Overview

Our four models are: **CAT-LSTM**, **BIN-LSTM**, **BC-LSTM** and **GRU**. We decided to use four different model architecture, two different preprocessing and two different word embeddings. We built very diverse models in order to maximize the effect of ensembling on our system. **CAT-LSTM** and **BIN-LSTM** share the exact same preprocessing and embeddings, while **BC-LSTM** and **GRU** use the same embeddings but slightly different preprocessing methods.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 297–301
Minneapolis, Minnesota, USA, June 6–7, 2019.

3 Proposed System

3.1 Text Preprocessing

We used two different preprocessing methods. However, both preprocessing share the same normalization (all words are lower cases and numbers, links, emails and dates were replaced by special tags). String emoticons are transformed into Emojis before tokenization (':)' becomes 😀).

3.1.1 CAT-LSTM and BIN-LSTM

Here, the preprocessing used was motivated by the fact that the dataset comes from social media, meaning the writing style contains improper use of grammar, misspellings, emoticons and slang. Because of that we used the *ekphrasis*[1] (Baziotis et al., 2017a) library which was made specifically for preprocessing text from social networks. This tool performs tokenization, word normalization, word segmentation and spell correction. Here, we didn't take into account the fact that our inputs are turned based and instead we added a special token `<eos>` in-between each turn of the dialogues which we then concatenated together. We also improved **spellchecking** with this method. Indeed, we realized that 8.8% of our vocabulary consisted of words that weren't part of our word embeddings because they were misspelled, even after preprocessing with *ekphrasis*. Obvious spelling errors like *angru* instead of *angry* were still present. In order to solve this problem we used a spellchecking library named *autocorrect*[2] after preprocessing with *ekphrasis* which decreased to 3.4% the number of unknown words from our vocabulary.

3.1.2 BC-LSTM and GRU

Here, for the normalization, specials tags (numbers, links, emails and dates) were removed. We did the **spellchecking** ourselves with the most common mistakes (e.g: *waht* becomes *what*). This makes the conversation cleaner and easier to understand while leaving a part of natural since most of the words are not corrected (e.g: *nooo* stay *nooo* instead of just *no* since it has a stronger meaning). Notice that for our **GRU** model, we concatenated the three input turns to make it simpler for the **GRU** to process but for our **BC-LSTM**, we made separate layers to process each turns.

3.2 Pre-Trained Word Embeddings

Word embeddings are dense vectors representing semantic meaning for each word of the vocabulary. We used pre-trained word embeddings to initialize the weights of our embedding layers. **CAT-LSTM** and **BIN-LSTM** used *Datastories* embeddings [3], while **BC-LSTM** and **GRU** used our own pre-trained word embeddings.

3.2.1 CAT-LSTM and BIN-LSTM

The weights we used for the embedding matrix of these models were the same as Baziotis et al. (2017b), pre-trained on 330 millions of english tweets messages posted from 12/2012 to 07/2016 with GloVe.

3.2.2 BC-LSTM and GRU

For these models, each word is represented by a vector of 312 dimensions which are obtained by the concatenation of the following features :

- *Word2Vec* (Mikolov et al., 2013) - We created our vector representations of words using *Word2Vec* networks trained with skip-gram and negative sampling on 30 millions of english tweets messages posted from 01/2017 to 06/2017. This word embeddings are 300 dimensional.

- *Affect Intensity Lexicons* (Mohammad and Turney, 2013) - 6,000 entries for four basic emotions: anger, fear, joy, and sadness. Considering fear as other, it adds 4 dimensions

- *Emolex* (Novak et al., 2015) - The NRC Emolex is a list of words and their associations with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust). We transformed fear, anticipation, trust, surprise and disgust into the class 'others' with a vector of 4 dimensions (joy, anger, sad, others)

- *Emoji Emotion* [4] - List of emoji rated by polarity. The polarity was hand classified (by one person) based on the names of these emoji. The contained emoji are the faces defined by Unicode

[1] https://github.com/cbaziotis/ekphrasis

[2] https://github.com/phatpiglet/autocorrect

[3] https://github.com/cbaziotis/datastories-semeval2017-task4

[4] https://github.com/words/emoji-emotion

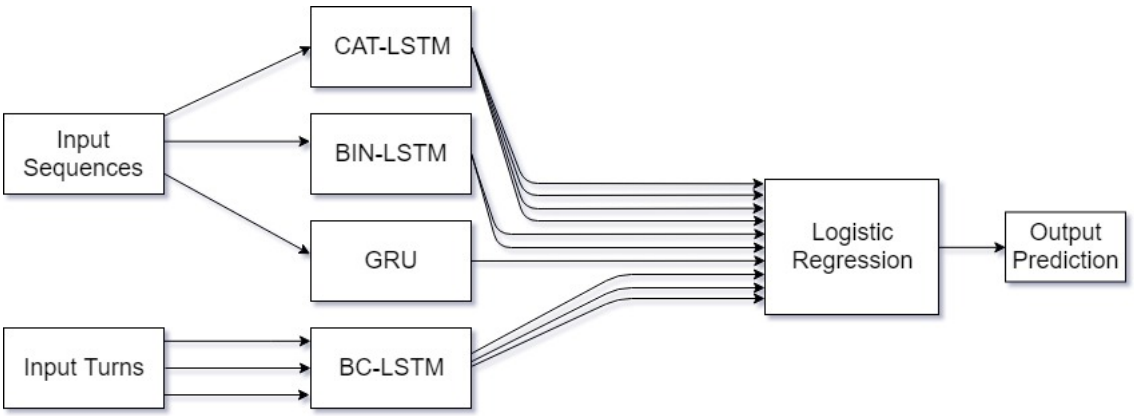

Figure 1: Ensembling method.

3.3 Models Description

We used Bi-directional Long Short-Term Memory networks (*B-LSTM*) in every model except the **GRU**. Every model used *Adam* optimizer and *crossentropy* [5] as the loss function.

3.3.1 CAT-LSTM Model

The core of the network is composed of two sets of *B-LSTM* as in Baziotis et al. (2017a). The input layer is composed of an embedding layer of size 300, followed by a dropout layer (0.4) directly after the embedding layer to help regularizing by showing slightly different sequences every epochs. Each *B-LSTM* layer consist of 150 units with recurrent dropout (0.5) and regular dropout (0.5). *B-LSTM* layers reads each sequence two times in different order, forward (from left to right) and backward (from right to left) which helps to capture the context of the sentence. The output layer is a dense layer followed by a softmax.

3.3.2 BIN-LSTM Model

With the previous model (**CAT-LSTM**), we realized most of our errors came from confusion between the class 'others' and the rest (angry, happy and sad). Which might be because the training set is slightly unbalanced (15k others, 5.5k angry, 5.4k happy, 4.2k sad). Because of that we decided to train a model specifically on binary classification between the class 'others' and the rest. That way we could use ensembling to help our categorical model to differentiate between the two categories. For this binary model, we kept a similar architecture as **CAT-LSTM**. However, we added a convolution 1D and a maxpool before the first *B-LSTM* layer to train faster. Since our input is a one-dimensional sequence of words, it makes sense to use a one-dimensional convolution right after the embedding layer in order to capture meaningful context about our sequence while reducing its size. We kept most spatial information by using a kernel size of 5 so that we take every group of 5 adjacent words into account. After the convolution, we used a one-dimensional maxpool layer of size 5 in order to reduce the input size. With this new architecture we were able to train a second model more focused on separating 'others' class from the rest of the emotion classes while increasing our training speed by 80%.

3.3.3 BC-LSTM Model

We used the **BC-LSTM** architecture introduced in Poria et al. (2017). **BC-LSTM** (Bidirectional Contextual LSTM) is a model for context-dependent sentiment analysis and emotion recognition. For this model we treated each turn inputs in separate parallel layers before concatenating the results. Hence, we have 3 parallel embedding layers with our pre-trained embeddings, which then go through 3 parallel bidirectional LSTM layers of 300 units. Each of those *B-LSTM* have a Dropout of 0.5. After that, we stack each *B-LSTM* layer. We then have a fully-connected layer with 4 units, that will give us probabilities for each emotion class. This model was particularly useful to detect happy, sad, and others emotions so we only keep those 3 probabilities.

3.3.4 GRU Model

The **GRU** model (Cho et al., 2014) allowed us to discriminate the angry class. The first layer is made using our pre-trained embeddings to process

[5] We used *categorical crossentropy* except for the binary (**BIN-LSTM**) model which used *binary crossentropy*

the concatenated text input. We added a dropout of 0.5. Then, the output of our embedding goes through a *GRU* layer of 128 units. On top of our *GRU*, we added a fully-connected layer of 32 units with relu as the activation function. A Dropout of 0.2 was added after this layer.

3.4 Ensembling

As stated before, we trained each model individually, hence giving us multiple sets of predictions for each input sample. We used all of those probabilities to train a logistic regression. We stack all 10 predictions (4 from **CAT-LSTM**, 2 from **BIN-LSTM**, 1 from **GRU** and 3 from **BC-LSTM**) to a create a new training sample associated with the corresponding true label of the sample, as shown in Figure 1. This way, we take each of our models into account and the logistic regression takes care of weighting the importance of our models. Since our four models are very diverse, they all contribute to the final prediction.

4 Results and Analysis

When evaluating each group of model separately, we found that they were correct on different samples even if the $F1\mu$ score is almost the same. Table 1 illustrate the performances of our systems on the test set. We can see that the class 'happy' gave our models the most trouble. Which might be because it is the smallest class in the dataset. Ensembling had a little impact on this emotion compared to 'angry' and 'sad'. Our final system achieves an $F1\mu$ score of **0.7324**.

Emotion / Models	Angry	Happy	Sad	$F1\mu$
CAT-LSTM+ BIN-LSTM	0.719	0.678	0.734	0.711
GRU+ BC-LSTM	0.722	0.673	0.739	0.712
Model ensembling	0.744	0.689	0.766	**0.7324**

Table 1: $F1\mu$ score on the test set for each model.

Note that [**CAT-LSTM + BIN-LSTM**] and [**GRU + BC-LSTM**] were submitted independently for the final submission. However, both systems were combined in **model ensembling** after the competition ended which significantly improved our final score.

5 Conclusion

In this paper, we proposed to use ensemble learning for sentiment analysis in conversations (SemEval2019 Task 3). Using various neural networks structures such as **B-LSTM**, parallel **B-LSTM**, **GRU** and **CNN**, ensemble learning takes advantage of this diversity of approach to make a prediction for our emotions classes (angry, happy, sad or others). Each model was trained separately on the given corpus. Then, we trained a logistic regression with the probabilities given by our four deep learning models in order to make the final predictions for each conversations. Our ensembling system achieved a $F1\mu$ score of **0.7324** on the final testing set, after the competition ended.

Improvements could be made by gathering more models. A properly fine-tuned language model (for instance using ULMFiT) or LSTM with attention mechanism could improve our current system. Future work will consist of finding better ensembling methods and working with language models pre-trained on larger corpus of data.

Acknowledgments

We would like to thank Abdessalam Bouchekif and Anatole Moreau for interesting scientific discussions (Daval-Frerot et al., 2018).

References

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017a. 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, Vancouver, Canada. Association for Computational Linguistics.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017b. Datastories at semeval-2017 task 6: Siamese lstm with attention for humorous text comparison. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 381–386, Vancouver, Canada. 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.

Kyunghyun Cho, Bart van Merrienboer, aglar Gülehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Guillaume Daval-Frerot, Abdesselam Bouchekif, and Anatole Moreau. 2018. Epita at semeval-2018 task 1: Sentiment analysis using transfer learning approach. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 151–155. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805.

Umang Gupta, Ankush Chatterjee, Radhakrishnan Srikanth, and Puneet Agrawal. 2017. A sentiment-and-semantics-based approach for emotion detection in textual conversations. *CoRR*, abs/1707.06996.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. *arXiv e-prints*, page arXiv:1801.06146.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

Saif Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29:436–465.

Petra Kralj Novak, Jasmina Smailovic, Borut Sluban, and Igor Mozetic. 2015. Sentiment of emojis. In *PloS one*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-dependent sentiment analysis in user-generated videos. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762.