Emotion Detection using RNNs

Eduard-Gabriel Poesina and Florin Brad*
University of Bucharest, Romania
*Bitdefender, Romania

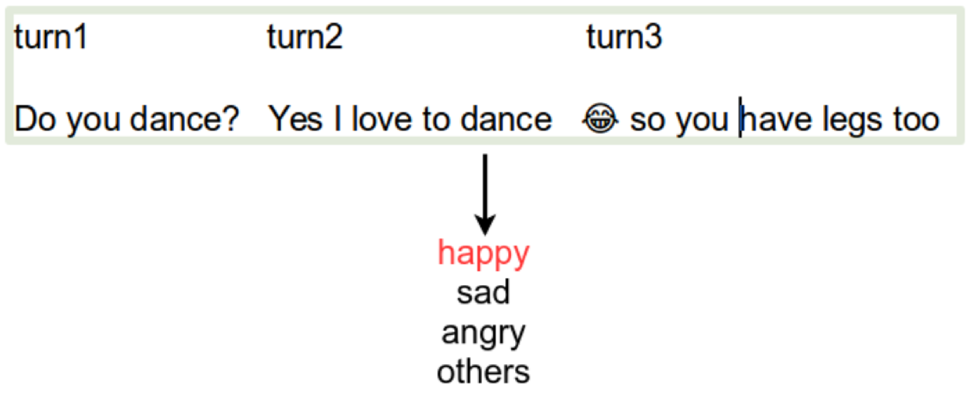
eduardgabriel.poe@gmail.com,fbrad@bitdefender.com

UNIVERSITY OF BUCHAREST
VIRTUTE ET SAPIENTIA

1. Introduction

Motivation: Detecting emotions from text only, without acoustic or facial clues is challenging. Moreover, the rise of internet slang and emoji add to the task complexity.

Proposal: As such, we propose as our current target to make machines able to understand human emotions. By decoding the most basic feelings(anger, sadness, happiness) we will be able to lower the gap between humans and machines. We will achieve our goal by analyzing messages between individuals and outputting the most probable emotion extracted from the texts.



2. Dataset and Preprocessing

Dataset Description: We use the EmoContext dataset [1], provided for the humanizing-AI workshop at ICJAI 2019. It contains approximately 30000 training samples, 2700 validation samples and 5500 test samples. Each sample consists of three messages exchanged by two talking individuals and a label which describes the sentiment which needs to be extracted from their discussion.The messages contain are composed of UTF-8 characters and the possible labels for the samples are: "angry","sad","happy" and "others".

Data preprocessing: To prepare our dataset for a RNN we padded each sample with a <pad> token until the desired length was achieved. Inputs that were too long were cropped by dropping words based on each reply length so that we can maintain a balance between the turns.As to increase our network's performance we decided to parse the input data by getting rid of non-ascii characters, transforming them to the most appropriate ascii symbol, transforming emojis into special words, transforming word contraptions into separate words and correcting grammatical mistakes.To furtherly assist our network we separated each turn with a special <end> token.

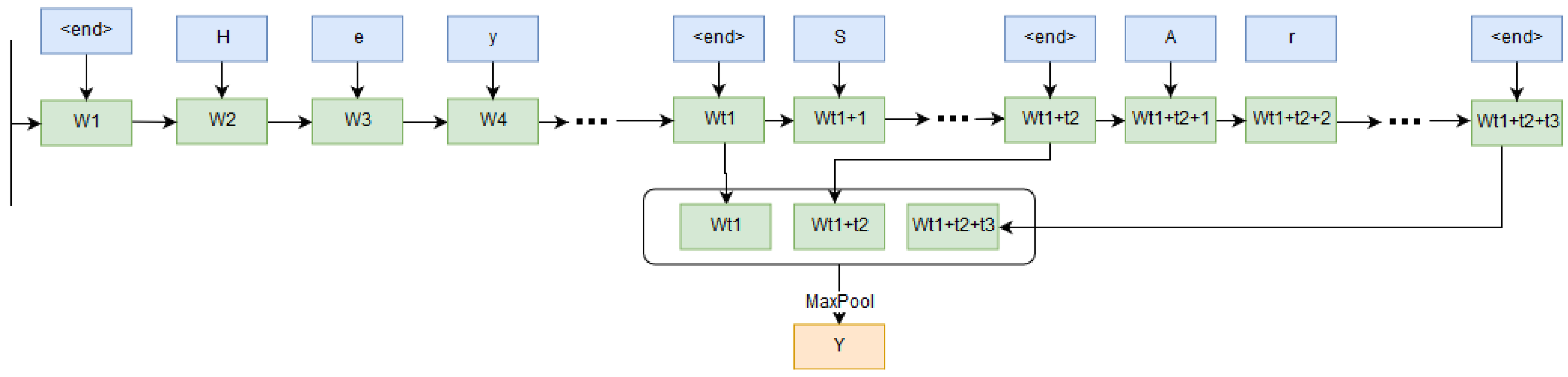
3. Models

Baseline: We train a feedforward network with one hidden layer, where the input is the sum of GloVe word embeddings [2] associated to words in the input sequence.

RNN Approaches: We use an LSTM cell in all of our recurrent models. We try character-level and word-level variants, as well as a combination of the two.

Last Hidden State: Our first type of model uses the last hidden state in order to make predictions (LAST HIDDEN in the table).

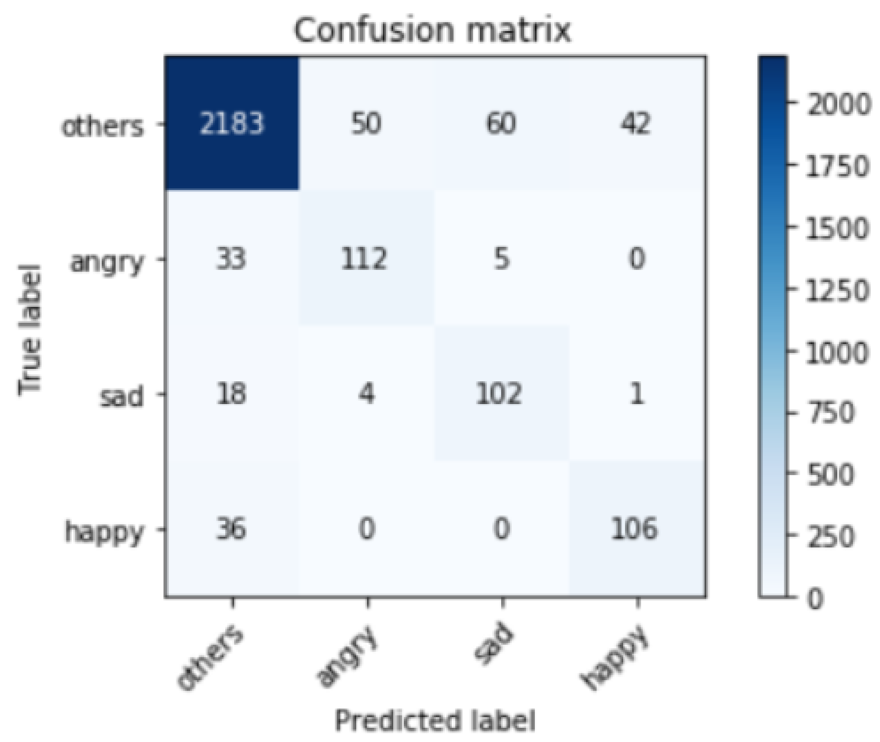
3 Hidden States: The RNN might "forget" what was learned in the early time-series, so we also use the earlier hidden states, corresponding to the <end> tokens of the first and second sentence. We concatenate the 3 hidden states and use max-pooling to obtain one single hidden state, used for prediction.



BiLSTM Over All Hidden States: Left-to-right LSTM misses context on the right side, so we also trained a bidirectional LSTM. We apply max-pooling over all the hidden states and concatenate the resulting vectors from both directions [3].

4. Results

Architecture	Input type	Accuracy percentage	F1 others	avg(F1) emotions	F1 emotions
FEED-FORWARD	words	86%	0.92	0.43	-
LSTM LAST HIDDEN	chars	90%	0.94	0.70	-
LSTM 3 HIDDEN MAXPOOL	chars	88%	0.93	0.65	-
BILSTM ALL HIDDEN MAXPOOL	chars	**92%**	**0.96**	**0.74**	-
BILSTM ALL HIDDEN MAXPOOL	words	91%	0.95	0.71	-
BILSTM ALL HIDDEN MAXPOOL	chars&words	91%	0.95	0.72	-
LSTM OTHVSEMO ALL HIDDEN	chars	90%	0.94	-	0.71
LSTM OTHVSEMO ALL HIDDEN	words	90%	0.93	-	0.68
LSTM OTHVSEMO 3 HIDDEN	chars	90%	0.94	-	0.72
BILSTM OTHVSEMO ALL HIDDEN	chars	88%	0.93	-	0.67



The confusion matrix of BiLSTM applied on both word and characters embeddings with maxpooling over all hidden states.

5. Conclusion

We have provided several recurrent neural network baselines which take into account multiple levels of information. Data parsing has influenced the performance of each model, improving our results: an increase from 75% to 82% accuracy on our feed-forward baseline model.

The model with the strongest prediction ability was: BiLSTM with MaxPool over all the hidden states, which took into account only characters. No increase in performance was observed when swapping to word-level embeddings nor when combining the two approaches. Selecting only the final states which correspond to the endings of sentences did not increase the results of our network. Our OneVsOthers approach F1 score failed to achieve a better result than our previous attempts. In respect to the accuracy level that have been achieved, we consider that it is sufficient given the increased difficulty of our dataset.

Taking into account the results of our architectures we can conclude that the recursive neural network approach can offer satisfying results in the task of emotion detection given the difficulty of the dataset.

6. References

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

[2] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. 2014.

[3] Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network. *CoRR*, abs/1508.01006, 2015.