CIL 2018: Text Sentiment Classification

Pirmin Schmid, Philip Junker, Nikolas Göbel, Aryaman Fasciati
The Optimists
Department of Computer Science, ETH Zurich, Switzerland

Abstract—**In this work we consider the task of classifying tweets by sentiment. In particular we want to determine whether a given tweet, which has been stripped of emoticons, used to include a positive smiley ":)" or a negative smiley ":(". We present a stacked RNN model trained on GloVe word embeddings. Our model achieves a classification accuracy of TODO: final-accuracy.**

I. INTRODUCTION

The goal of this project is to build a sentiment classifier that predicts whether a tweet text used to include a positive smiley :) or a negative smiley :(, based on the remaining text.

Our first baseline uses random forests and achieved an accuracy of 72%. Our second baseline uses a Recurrent Neural Network (RNN) model with an accuracy of TODO: rnn-baseline-accuracy.

In a third model, we refined our second baseline, incorporating TODO: describe novel approach, in a RNN-based approach. This model achieved TODO: % accuracy.

II. RELATED WORK

Write about related work [1]

III. MODELS

A. First Baseline (B1)

Our first baseline uses a random forest model with unlimited max_depth and 20 estimators. We used the random forest implementation provided with the scikit-learn library [2]. The classifier is trained on tweet embeddings, derived by computing the mean over all embedding vectors of the words contained within it. This approach does not differentiate between permutations of similar words. It also assigns equal weight to every word. Words that are not in the vocabulary are ignored. Word embeddings are provided by the GloVe [3] project from Stanford, pre-trained on two billion tweets. In addition to that, tweets were pre-processed with a slightly adapted version of the preprocessor script provided by Stanford.

This model achieved an accuracy of 72%.

B. Second Baseline (B2)

For the second baseline, we trained a stacked, recurrent neural network. In contrast to the random forest baseline, tweet embeddings were not aggregated in any way from their words. Rather, the network was trained on matrices of dimension $word\_count_{tweet} \times 200$. Statistical analysis of the provided data in Figure 1 shows that the majority ($>98\%$) of tweets have less than 30 words. Tweets longer than that were ignored, whereas shorter tweets were padded.

In contrast to B1, this model does take word order into account and allows for different weights to be assigned to certain, sentiment-implying words, during training. Intuitively, we would expect the latter to result in significantly higher classification accuracy, compared to the random forest model. It is not as clear, whether taking word order into account during training is important for the task of sentiment classification and would thus dilute inputs in a sense.

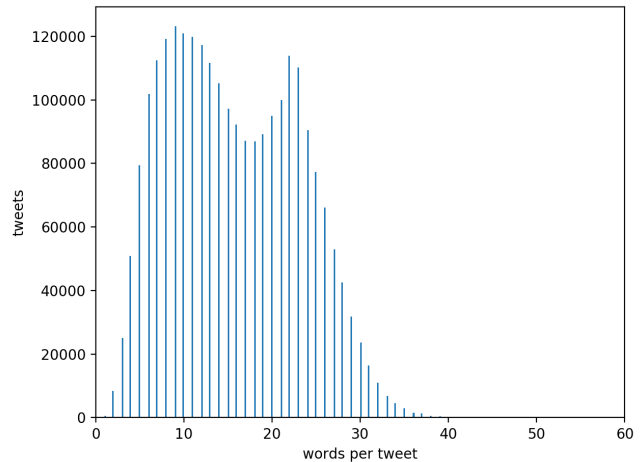This model achieved an accuracy of 85%.

Figure 1. Word Count Distribution

C. Our Model

For our final model, we expanded on the RNN approach taken in baseline B2. We introduced two additional convolution layers.

ACKNOWLEDGEMENTS

REFERENCES

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershel-vam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[3] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–43.