GradAscent at EmoInt-2017: Character- and Word-Level Recurrent Neural Network Models for Tweet Emotion Intensity Detection

Egor Lakomkin[*], **Chandrakant Bothe**[*] and **Stefan Wermter**
Knowledge Technology, Department of Informatics,
University of Hamburg,
Vogt-Koelln Str. 30, 22527 Hamburg, Germany
knowledge-technology.info
{lakomkin, bothe, wermter}@informatik.uni-hamburg.de

Abstract

The WASSA 2017 EmoInt shared task has the goal to predict emotion intensity values of tweet messages. Given the text of a tweet and its emotion category (anger, joy, fear, and sadness), the participants were asked to build a system that assigns emotion intensity values. Emotion intensity estimation is a challenging problem given the short length of the tweets, the noisy structure of the text and the lack of annotated data. To solve this problem, we developed an ensemble of two neural models, processing input on the character. and word-level with a lexicon-driven system. The correlation scores across all four emotions are averaged to determine the bottom-line competition metric, and our system ranks place forth in full intensity range and third in 0.5-1 range of intensity among 23 systems at the time of writing (June 2017).

1 Introduction

Sentiment analysis of a text reveals information on the degree of positiveness or negativeness of the opinion expressed by the writer. Such information can be useful for providing better services for users (Kang and Park, 2014) or preventing potentially dangerous situations (O'Dea et al., 2015). Traditionally the most popular way of sentiment representation is either binary (positive, negative) or multi-class (for example 5 classes: very negative, negative, neutral, positive, very positive). While being simple, such a scheme looses interpretability and a continuous intensity scale might be preferred. Twitter sentiment and emotion intensity detection are still challenging tasks and remain active areas of research. These difficulties have several reasons: extensive usage of hashtags, slang, abbreviations, and emoticons. Also, tweets are usually typed on mobile devices which can lead to a substantial amount of typos. As traditional NLP tools are usually trained on datasets containing clean text, which makes it difficult to use them for tweet analysis.

Existing approaches for modeling emotion intensity rely heavily on manually constructed lexicons, which contain information about intensity weights for each available word (Mohammad and Bravo-Marquez, 2017a; Neviarouskaya et al., 2007). The intensity for the whole sentence can be inferred by combining individual scores of words. While being easily interpretable, such models have several limitations. Ignoring word order and compositionality of the language is the first issue, which is critical for modeling sequences. Constructing such lexicons is a labour-intensive process, which needs to be carried out continuously due to the constant development of language. Data-driven approaches like deep neural networks can overcome such limitations, and they have been behind many recent advances in text processing tasks, such as language modeling, machine translation, POS tagging, and classification (Irsoy and Cardie, 2014; Socher et al., 2013). The appealing property of such models is their ability to combine feature extraction and classification stages given a sufficient amount of training data.

In this paper, we augment traditional lexicon-based models with two neural network-based models: one with character and one with word input. Character-level deep neural networks recently showed outstanding results on text understanding tasks such as machine translation (Kalchbrenner et al., 2016) and text classification (Zhang et al., 2015). In a domain-specific task such as predict-

[*]equal contribution

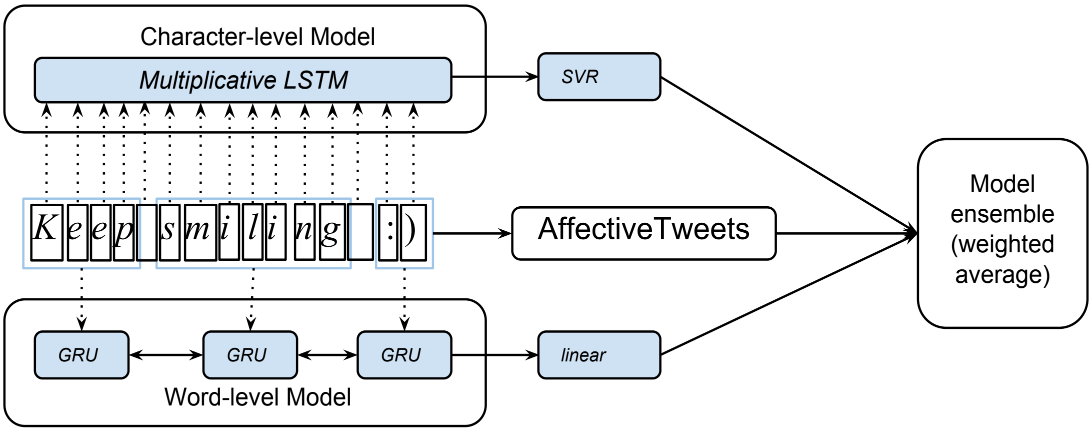

Figure 1: Overall model architecture. It combines a lexicon-based AffectiveTweets model with two neural models: a character and a word-level model via averaging scores with weights tuned on the provided validation set.

ing the emotion intensity of tweets, a character-level model can theoretically capture the notion of hashtags, emoticons, or character repetitions, which all are unique to social media. The intuition is that a character-level model captures common writing patterns such as punctuations and signaling characters. A word-level recurrent neural model can incorporate the order of information using distributed representations of words trained on a large amount of text.

Our final model is a weighted average of the scores provided by the baseline, our character- and word-level model. Our ensemble model achieved forth position in the 0-1 emotion intensity range task and third position in the 0.5-1.0 range task on the public leaderboard (GradAscent team) on CodaLab[1] at the time of writing this paper (June 2017).

2 Approach

Our system is an ensemble of the provided baseline system and two neural network-based models; processing character and word input respectively. Combining the word and character representations we can deal with noisiness of the tweet messages as well as capturing the semantics of the text by using distributed word representations.

2.1 Data pre-processing

We perform only a few preprocessing steps, like striping URLs, user mentions (@username) and leave only the following characters: `a-zA-Z@-!:(),;?.#'0-9*`. We always convert a message to lowercase before feeding it to the models.

Table 1: WASSA 2017 Emotion Intensity Shared task dataset statistics.

Split	Joy	Anger	Fear	Sadness	Sum
Train	823	856	1147	786	3612
Dev	78	83	109	73	343
Test	714	760	995	673	3142

2.2 Baseline model

The baseline system is a WEKA-based model called AffectiveTweets (Mohammad and Bravo-Marquez, 2017a). This system combines features derived from several lexicons like MPQA, Bing Liu, Afinn, Sentiment 140, NRC Hashtag sentiment lexicon, NRC-10 Expanded, NRC Hashtag Emotion Association, and SentiWordNet with traditional NLP features like word- and character n-grams, POS tags, and processing of negations. In addition to those features, AffectiveTweets incorporates Brown clusters (Brown et al., 1992) trained on ~53 million tweets[2], combining them with averaged and concatenated first k word embeddings of the tweet. Finally, a support Vector Machine model is used as a regression model for predicting emotion intensity values.

[1] https://competitions.codalab.org/competitions/16380

[2] http://www.cs.cmu.edu/~ark/TweetNLP/

2.3 Character-level RNN model

We extracted character-level sentence representations by encoding the whole tweet text with the pre-trained recurrent neural network model[3]. This model contains a single multiplicative LSTM (Krause et al., 2016) layer with 4,096 hidden units, trained on ∼80 million Amazon product reviews as a character-based language model (Radford et al., 2017). We extracted the hidden vector corresponding to the last character of a tweet and also averaged the representations of all hidden vectors. Concatenation of the two vectors is used as a tweet representation. In our experiments, we observed that adding averaged character representations improves the overall performance, especially when evaluating high-intensity tweets.

In addition to the pre-trained character-level language model, we investigate a model trained specifically for tweets. Our observation was that the tweets have a different language structure than product reviews, which might affect the transferability of features between domains. For instance, the extensive use of emoticons, character repetition, and hashtags, which are common for tweet messages, however, significantly different from product reviews which are often longer and grammatically correct.

We trained the character-based language model on the Sentiment 140 corpus comprised of 1.6 million tweets (Go et al., 2009). A single-layer LSTM (Hochreiter and Schmidhuber, 1997) with 1024 hidden units was trained with Adam optimizer (Kingma and Ba, 2014) with 0.0005 learning rate and clipping gradients at norm 1. We used the Support Vector Regressor (SVR) algorithm to classify tweets represented as a fixed-length vector with a character-based recurrent neural network. Results of different setups are reported in Table 2.

2.4 Word-level model

We used distributed representations to model the words in a tweet. We carried out several experiments where we used random initialization for word embeddings and two pre-trained versions of GloVe embeddings (Pennington et al., 2014) trained on Wikipedia and Twitter[4], to test if Twitter specific word representations are more suitable to solve the problem. Out-of-vocabulary words were replaced with a special word 'OOV' and initialized as a random vector, which was tuned during the training. We used a 50-dimensional embedding representation in all our experiments.

Table 2: Effect of different character-level recurrent neural network representations: last cell vector of the pre-trained model (PT, last) and Twitter-specific character LM (Twit, last). Also, in addition, we tested a concatenation of the last cell vector with the average of all cell vectors for the pre-trained model (PT, last+avg) and Twitter model (Twit, last+avg). Results are reported on the test set, where avg_p corresponds to Pearson coefficient and avg_s to Spearman.

Range	**(0.0-1.0)**		**(0.5-1.0)**	
Model	**avg_p**	**avg_s**	**avg_p**	**avg_s**
PT, last	0.470	0.468	0.412	0.404
PT, last+avg	**0.474**	**0.472**	**0.419**	**0.413**
Twit, last	0.312	0.307	0.296	0.288
Twit, last+avg	0.319	0.310	0.298	0.301

Table 3: Effect of different word embedding initializations for the word-level model: randomly initialized, pre-trained GloVe embeddings on Twitter and Wikipedia.

Range	**(0.0-1.0)**		**(0.5-1.0)**	
Model	**avg_p**	**avg_s**	**avg_p**	**avg_s**
Random emb.	0.291	0.276	0.250	0.227
GloVe (Twitter)	0.300	0.293	0.231	0.220
GloVe (Wiki)	**0.326**	**0.323**	**0.259**	**0.252**

A bidirectional gated recurrent unit (GRU) network (Chung et al., 2014) with a 32-dimension cell size was used for modeling the tweet as a hidden memory vector. The vector corresponding to the last word was fed to a dense layer with 1 neuron predicting emotion intensity. We used GRUs as they tackle the common vanishing gradient problem of RNNs during the training and they contain fewer parameters than LSTM units. The word-level model is trained on the given EmoInt corpus with Adam optimizer using different embedding setups, the results are presented in Table 3.

3 Experiment

The dataset for the WASSA-2017 competition (Mohammad and Bravo-Marquez, 2017b) is comprised of 7097 annotated tweets, classified

[3] https://github.com/openai/generating-reviews-discovering-sentiment

[4] https://nlp.stanford.edu/projects/glove/

Table 4: Pearson and Spearman correlation coefficients of baseline, character and word-level models and its ensemble for fear, anger, joy and sadness emotions and also average values. Results are calculated on the provided test set labels.

Model	avg_p	avg_s	anger_p	anger_s	fear_p	fear_s	joy_p	joy_s	sad_p	sad_s
Test set results (Intensity range: 0-1)										
Baseline	0.655	0.652	0.631	0.623	0.631	0.622	0.645	0.654	0.712	0.711
Char_LM	0.474	0.472	0.415	0.400	0.575	0.551	0.278	0.299	0.629	0.638
Word_Level	0.326	0.323	0.253	0.258	0.337	0.332	0.201	0.194	0.435	0.395
Char_LM + Word_Level	0.659	0.656	0.580	0.572	0.658	0.638	0.708	0.714	0.688	0.701
Baseline + Char_LM + Word_Level	**0.721**	**0.717**	**0.678**	**0.665**	**0.698**	**0.686**	**0.744**	**0.750**	**0.763**	**0.767**
Test set results (Intensity range: 0.5-1)										
Baseline	0.475	0.449	0.495	0.464	0.476	0.432	0.370	0.363	0.558	0.537
Char_LM	0.419	0.413	0.316	0.327	0.488	0.435	0.416	0.423	0.457	0.467
Word_Level	0.259	0.252	0.237	0.257	0.220	0.226	0.211	0.201	0.451	0.408
Char_LM + Word_Level	0.471	0.467	0.389	0.406	0.488	0.435	0.536	0.547	0.470	0.481
Baseline + Char_LM + Word_Level	**0.562**	**0.543**	**0.565**	**0.545**	**0.531**	**0.494**	**0.528**	**0.531**	**0.624**	**0.601**

into 4 categories: joy, anger, fear, and sadness (dataset statistics are presented in Table 1). For each annotated tweet there is an ID, full text, emotion category, and emotion intensity value. Emotion intensity is a real value in the range from 0 to 1, where higher value correspond to a higher intensity of the emotion conveyed. A sample from the EmoInt corpus:

```
30112 LOVE LOVE LOVE #smile
#fun #relaxationiskey joy 0.740,
```

where 30112 is the ID of a tweet, which is labeled as ”*joy*” with an intensity of 0.740.

3.1 Ensembling of the models

Ensembling of several models is a widely used method to improve the performance of the overall system by combining predictions of several classifiers. Several ensembling techniques have been proposed recently: mixing experts (Jacobs et al., 1991), model stacking, bagging and boosting (Breiman, 1996) and a simple weighted average of the scores of individual models, which we used in this work. The main reason for our choice was the limited size of the training data, and using more complex approach like stacking could lead to overfitting. In this work, we output emotion intensity values as a linear combination of individual predictions of three systems: baseline, character and word-level models.

$$\begin{aligned} emotion_{intensity} &= w_b * baseline_{emotion} \\ &+ w_w * w\_rnn_{emotion} + w_c * c\_rnn_{emotion}, \\ &w_b + w_w + w_c = 1 \quad (1) \end{aligned}$$

where $baseline_{emotion}$, $w\_rnn_{emotion}$ and $c\_rnn_{emotion}$ are intensities of the baseline, character and word-level models correspondingly for the emotion (joy, anger, fear or sadness). Ensembling coefficients w_b, w_c and w_w were tuned on the development set to maximize the average Pearson correlation coefficient using grid-search.

4 Results & Conclusion

We report Pearson and Spearman correlation for each emotion class on the provided test data, shown in Table 4. The correlation rank coefficients assess how relevant and similar the two sets of ranking are. The character and word-level neural models achieve lower correlation values than the baseline, which is an indicator that models containing much of external knowledge perform better than end-to-end models on the tasks with a handful amount of samples; however, they bring additional value to the ensemble. Pearson and Spearman correlation coefficients are improved by 0.066 and 0.065 for the intensities in the full range of 0-1, achieving #4 position on the leaderboard. Additionally, the systems were evaluated on the sample with moderate or high emotional intensities with values from 0.5 to 1. Our ensemble model places rank #4 and shows 0.087 ($\sim$ 18.5% relative) improvement on both correlation coefficients.

Surprisingly, tweet representations obtained with the character-level model show competitive or even better results for fear and joy emotion categories for samples with high-intensity emotions, and overall the Char_LM model shows similar results to the AffectiveTweet baseline model. Given the fact that the Char_LM model did not have any external knowledge or supervision other than the provided data, this demonstrates the effectiveness of the character-level modeling of noisy and short texts.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 642667 (SECURE). We would like to thank Dr. Cornelius Weber and Dr. Sven Magg for their helpful comments and suggestions.

References

Leo Breiman. 1996. Bagging predictors. *Machine learning* 24(2):123–140.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18(4):467–479.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv: 1412.3555v1* pages 1–9.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1(12).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Ozan Irsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *the Proceedings of the Conference on EMLNP*. pages 720–728.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation* 3(1):79–87.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv:1610.10099* .

Daekook Kang and Yongtae Park. 2014. Review-based measurement of customer satisfaction in mobile service: Sentiment analysis and vikor approach. *Expert Systems with Applications* 41(4):1041–1050.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Ben Krause, Liang Lu, Iain Murray, and Steve Renals. 2016. Multiplicative lstm for sequence modelling. *arXiv:1609.07959* .

Saif M. Mohammad and Felipe Bravo-Marquez. 2017a. Emotion intensities in tweets. In *Proceedings of the Sixth Joint Conference on Lexical and Computational Semantics (*Sem)*. Vancouver, Canada.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017b. WASSA-2017 shared task on emotion intensity. In *Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*. Copenhagen, Denmark.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2007. Textual affect sensing for sociable and expressive online communication. *Affective Computing and Intelligent Interaction* pages 218–229.

Bridianne O'Dea, Stephen Wan, Philip J Batterham, Alison L Calear, Cecile Paris, and Helen Christensen. 2015. Detecting suicidality on twitter. *Internet Interventions* 2(2):183–188.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *the Proceedings of the Conference on EMLNP*. pages 1532–1543.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv: 1704.01444* .

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *the Proceedings of the Conference on EMLNP*. volume 1631, pages 1631–1642.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *arXiv: 1509.01626* .