INSIGHT-1 at SemEval-2016 Task 4: Convolutional Neural Networks for Sentiment Classification and Quantification

Sebastian Ruder[1 2] **Parsa Ghaffari**[2] **John G. Breslin**[1]

[1]Insight Centre for Data Analytics
National University of Ireland, Galway
firstname.lastname@insight-centre.org

[2]Aylien Ltd.
Dublin, Ireland
firstname@aylien.com

Abstract

This paper describes our deep learning-based approach to sentiment analysis in Twitter as part of SemEval-2016 Task 4. We use a convolutional neural network to determine sentiment and participate in all subtasks, i.e. two-point, three-point, and five-point scale sentiment classification and two-point and five-point scale sentiment quantification. We achieve competitive results for two-point scale sentiment classification and quantification, ranking fifth and a close fourth (third and second by alternative metrics) respectively despite using only pre-trained embeddings that contain no sentiment information. We achieve good performance on three-point scale sentiment classification, ranking eighth out of 35, while performing poorly on five-point scale sentiment classification and quantification. An error analysis reveals that this is due to low expressiveness of the model to capture negative sentiment as well as an inability to take into account ordinal information. We propose improvements in order to address these and other issues.

1 Introduction

Social media allows hundreds of millions of people to interact and engage with each other, while expressing their thoughts about the things that move them. Sentiment analysis (Pang and Lee, 2008) allows us to gain insights about opinions towards persons, objects, and events in the public eye and is used nowadays to gauge public opinion towards companies or products, to analyze customer satisfaction, and to detect trends.

Its immediacy allowed Twitter to become an important platform for expressing opinions and public discourse, while the accessibility of large quantities of data in turn made it the focal point of social media sentiment analysis research.

Recently, deep learning-based approaches have demonstrated remarkable results for text classification and sentiment analysis (Kim, 2014) and have performed well for phrase-level and message-level sentiment classification (Severyn and Moschitti, 2015).

Past SemEval competitions in Twitter sentiment analysis (Rosenthal et al., 2014; Rosenthal et al., 2015) have contributed to shape research in this field. SemEval-2016 Task 4 (Nakov et al., 2016) is no exception, as it introduces both quantification and five-point-scale classification tasks, neither of which have been tackled with deep learning-based approaches before.

We apply our deep learning-based model for sentiment analysis to all subtasks of SemEval-2016 Task 4: three-point scale message polarity classification (subtask A), two-point and five-point scale topic sentiment classification (subtasks B and C respectively), and two-point and five-point scale topic sentiment quantification (subtasks D and E respectively).

Our model achieves excellent results for subtasks B and D, ranks competitively for subtask A, while performing poorly for subtasks C and E. We perform an error analysis of our model to obtain a better understanding of strengths and weaknesses of a deep learning-based approach particularly for these new tasks and subsequently propose improvements.

Proceedings of SemEval-2016, pages 183–187,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

2 Related work

Deep-learning based approaches have recently dominated the state-of-the-art in sentiment analysis. Kim (2014) uses a one-layer convolutional neural network to achieve top performance on various sentiment analysis datasets, demonstrating the utility of pre-trained embeddings.

State-of-the-art models in Twitter sentiment analysis leverage large amounts of data accessible on Twitter to further enhance their embeddings by treating smileys as noisy labels (Go et al., 2009): Tang et al. (2014) learn sentiment-specific word embeddings from such distantly supervised data and use these as features for supervised classification, while Severyn and Moschitti (2015) use distantly supervised data to fine-tune the embeddings of a convolutional neural network.

In contrast, we observe distantly supervised data not to be as important for some tasks as long as sufficient training data is available.

3 Model

The model architecture we use is an extension of the CNN structure used by Collobert et al. (2011).

The model takes as input a text, which is padded to length n. We represent the text as a concatentation of its word embeddings $x_{1:n}$ where $x_i \in \mathbb{R}^k$ is the k-dimensional vector of the i-th word in the text.

The convolutional layer slides filters of different window sizes over the word embeddings. Each filter with weights $w \in \mathbb{R}^{hk}$ generates a new feature c_i for a window of h words according to the following operation:

$$c_i = f(w \cdot x_{i:i+h-1} + b) \tag{1}$$

Note that $b \in \mathbb{R}$ is a bias term and f is a non-linear function, ReLU (Nair and Hinton, 2010) in our case. The application of the filter over each possible window of h words or characters in the sentence produces the following feature map:

$$c = [c_1, c_2, ..., c_{n-h+1}] \tag{2}$$

Max-over-time pooling in turn condenses this feature vector to its most important feature by taking its maximum value and naturally deals with variable input lengths.

A final softmax layer takes the concatenation of the maximum values of the feature maps produced by all filters and outputs a probability distribution over all output classes.

4 Methodology

4.1 Datasets

For every subtask, the organizers provide a training, development, and development test set for training and tuning. We use the concatentation of the training and development test set for each subtask for training and use the development set for validation.

Additionally, the organizers make training and development data from SemEval-2013 and trial data from 2016 available that can be used for training and tuning for subtask A and subtasks B, C, D, and E respectively. We experiment with adding these datasets to the respective subtask. Interestingly, adding them slightly increases loss on the validation set, while providing a significant performance boost on past development test sets, which we view as a proxy for performance on the 2016 test set. For this reason, we include these datasets for training of all our models.

We notably do not select the model that achieves the lowest loss on the validation set, but choose the one that maximizes the F_1^{PN} score, i.e. the arithmetic mean of the F_1 of positive and negative tweets, which has historically been used to evaluate the SemEval message polarity classification subtask. We observe that the lowest loss does not necessarily lead to the lowest F_1^{PN}, as it does not include F_1 of neutral tweets.

4.2 Pre-processing

For pre-processing, we use a script adapted from the pre-processing script[1] used for training GloVe vectors (Pennington et al., 2014). Besides normalizing urls and mentions, we notably normalize happy and sad smileys, extract hashtags, and insert tags for repeated, elongated, and all caps characters.

4.3 Word embeddings

Past research (Kim, 2014; Severyn and Moschitti, 2015) found a good initialization of word embed-

[1] http://nlp.stanford.edu/projects/glove/preprocess-twitter.rb

dings to be crucial in training an accurate sentiment model.

We thus evaluate the following evaluation schemes: random initialization, initialization using pre-trained GloVe vectors, fine-tuning pre-trained embeddings on a distantly supervised corpus (Severyn and Moschitti, 2015), and fine-tuning pre-trained embeddings on 40k tweets with crowd-sourced Twitter annotations. Perhaps counter-intuitively, we find that fine-tuning embeddings on a distantly supervised or crowd-sourced corpus does not improve performance on past development test sets when including the additionally provided data for training. We hypothesize that additional training data facilitates learning of the underlying semantics, thereby reducing the need for sentiment-specific embeddings. Our scores partially echo this theory.

For this reason, we initialize our word embeddings simply with 200-dimensional GloVe vectors trained on 2B tweets. Word embeddings for unknown words are initialized randomly.

4.4 Hyperparameters and pre-processing

We tune hyperparameters over a wide range of values via random search on the validation set. We find that the following hyperparameters, which are similar to ones used by Kim (2014), yield the best performance across all subtasks: mini-batch size of 10, maximum sentence length of 50 tokens, word embedding size of 200 dimensions, dropout rate of 0.3, l_2 regularization of 0.01, filter lengths of 3, 4, and 5 with 100 filter maps each.

We train for 15 epochs using mini-batch stochastic gradient descent, the Adadelta update rule (Zeiler, 2012), and early stopping.

4.5 Task adaptation and quantification

To adapt our model to the different tasks, we simply adjust the number of output neurons to conform to the scale used in the task at hand (two-point scale in subtasks B and D, three-point scale in subtask A, five-point scale in subtasks C and E).

We perform a simple quantification for subtasks D and E by aggregating the classified tweets for each topic and reporting their distribution across sentiments. We would thus expect our results on subtasks B and D and results on subtasks C and E to be closely correlated.

Metric	Our score	Best score	Rank
F_1^{PN}	**0.593**	**0.633**	**8/34**
R^{PN}	0.616	0.670	12/34
Acc^{PN}	0.635	0.646	5/34

Table 1: Our score and rank for subtask A for each metric compared to the best team's score (results for official metric in bold).

	2013		**2014**			**2015**
	TW	**SMS**	**TW**	**TW /s**	**LJ**	**TW**
+	72.49	66.73	76.84	64.52	69.08	65.56
-	47.97	49.65	51.95	13.64	50.00	53.00
=	67.53	77.83	65.51	45.71	67.28	65.23

Table 2: F_1 scores of our model for positive, negative, and neutral tweets for each progress dataset of subtask A. TW: Tweet. /s: sarcasm. LJ: Live Journal. +: positive. -: negative. =: neutral.

5 Evaluation

We report results of our model in Tables 1 and 2 (subtask A), Table 3 (subtask B), Tables 5 and 6 (subtask C), Table 4 (subtask D), and Table 7 (subtask E). For some subtasks, the organizers make available alternative metrics. We observe that the choice of the scoring metric influences results considerably, with our system always placing higher if ranked by one of the alternative metrics.

Subtask A. We obtain competitive performance on subtask A in Table 1. Analysis of results on the progress test sets in Table 2 reveals that our system achieves competitive F_1 scores for positive and neutral tweets, but only low F_1 scores for negative tweets due to low recall. This is mirrored in Table 1, where we rank higher for accuracy than for recall. The scoring metric for subtask A, F_1^{POS} accentuates F_1 for positive and negative tweets, thereby ignoring our good performance on neutral tweets and leading to only mediocre ranks on the progress test sets for our system.

Subtasks B and D. We achieve a competitive fifth rank for subtask B by the official recall metric in Table 3. However, ranked by F_1 (as in subtask A), we place third – and second if ranked by accuracy. Similarly, for subtask D, we rank fourth (with a differential of 0.001 to the second rank) by KLD, but second and first if ranked by AE and RAE respectively. Jointly, these results demonstrate that classifi-

Metric	**Our score**	**Best score**	**Rank**
R	**0.767**	**0.797**	**5/19**
F_1	0.786	0.799	3/19
Acc	0.864	0.870	2/19

Table 3: Our score and rank for subtask B for each metric compared to the best team's score (results for official metric in bold).

Metric	**Our score**	**Best score**	**Rank**
KLD	**0.054**	**0.034**	**4/14**
AE	0.085	0.074	2/14
RAE	0.423	0.423	1/14

Table 4: Our score and rank for subtask D for each metric compared to the best team's score (results for official metric in bold).

cation performance is a good indicator for quantification without using any more sophisticated quantification methods. These results are in line with past research (Kim, 2014) showcasing that even a conceptually simple neural network-based approach can achieve excellent results given enough training data per class. These results also highlight that embeddings trained using distant supervision, which should be particularly helpful for this task as they are fine-tuned using the same classes, i.e. positive and negative, are not necessary given enough data.

Subtasks C and E. We achieve mediocre results for subtask C in Table 5, only ranking sixth – however, placing third by the alternative metric. Similarly, we only achieve an unsatisfactory eighth rank for subtask E in Table 7. An error analysis for subtask C in Table 6 reveals that the model is able to differentiate between neutral, positive, and very positive tweets with good accuracy. However, similarly to results in subtask A, we find that it lacks expressiveness for negative sentiment and completely fails to capture very negative tweets due to their low number in the training data. Additionally, it is unable to take into account sentiment order to reduce error for very positive and very negative tweets.

Metric	**Our score**	**Best score**	**Rank**
MAE^M	**1.006**	**0.719**	**6/11**
MAE^μ	0.607	0.580	3/11

Table 5: Our score and rank for subtask C for each metric compared to the best team's score (results for official metric in bold).

Sentiment	-2	-1	0	1	2
MAE^M	2.09	1.29	0.78	0.17	0.71

Table 6: Macro-averaged mean absolute error (MAE^M) of our model for each sentiment class for subtask C. Lower error is better.

Metric	**Our score**	**Best score**	**Rank**
EMD	**0.366**	**0.243**	**8/10**

Table 7: Our score and rank for subtask E compared to the best team's score.

5.1 Improvements

We propose different improvements to enable the model to better deal with some of the encountered challenges.

Negative sentiment. The easiest way to enable our model to better capture negative sentiment is to include more negative tweets in the training data. Additionally, using distantly supervised data for fine-tuning embeddings would likely have helped to mitigate this deficit. In order to allow the model to better differentiate between different sentiments on a five-point scale, it would be interesting to evaluate ways to create a more fine-grained distantly supervised corpus using e.g. a wider range of smileys and emoticons or certain hashtags indicating a high degree of elation or distress.

Ordinal classification. Instead of treating all classes as independent, we can enable the model to take into account ordinal information by simply modifying the labels as in (Cheng et al., 2008). A more sophisticated approach would organically integrate label-dependence into the network.

Quantification. Instead of deriving the topic-level sentiment distribution by predicting tweet-level sentiment, we can directly minimize the Kullback-Leibler divergence for each topic. If the feedback from optimizing this objective proves to be too indirect to provide sufficient signals, we can jointly optimize tweet-level as well as topic-level sentiment as in (Kotzias, 2015).

6 Conclusion

In this paper, we have presented our deep learning-based approach to Twitter sentiment analysis for two-point, three-point, and five-point scale sentiment classification and two-point and five-point scale sentiment quantification. We reviewed the different aspects we took into consideration in creating our model. We rank fifth and a close fourth (third and second by alternative metrics) on two-point scale classification and quantification despite using only pre-trained embeddings that contain no sentiment information. We analysed our weaker performance on three-point scale sentiment classification and five-point scale sentiment classification and quantification and found that the model lacks expressiveness to capture negative sentiment and is unable to take into account class order. Finally, we proposed improvements to resolve these deficits.

Acknowledgments

This project has emanated from research conducted with the financial support of the Irish Research Council (IRC) under Grant Number EBPPG/2014/30 and with Aylien Ltd. as Enterprise Partner. This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

Jianlin Cheng Jianlin Cheng, Zheng Wang Zheng Wang, and G. Pollastri. 2008. A neural network approach to ordinal regression. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter Sentiment Classification using Distant Supervision. *Processing*, 150(12):1–6.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Dimitrios Kotzias. 2015. From Group to Individual Labels using Deep Features.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning*, (3):807–814.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, San Diego, California. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, (SemEval):73–80.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, (SemEval):451–463.

Aliaksei Severyn and Alessandro Moschitti. 2015. UNITN : Training Deep Convolutional Neural Network for Twitter Sentiment Classification. (SemEval):464–469.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 1:1555–1565.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method.