FOI DSS at SemEval-2018 Task 1: Combining LSTM States, Embeddings, and Lexical Features for Affect Analysis

Maja Karasalo **Mattias Nilsson** **Magnus Rosell** **Ulrika Wickenberg Bolin**
FOI - Swedish Defence Research Agency
{majkar, matnil, magros, ulrwic}@foi.se

Abstract

This paper describes the system used and results obtained for team FOI DSS at SemEval-2018 Task 1: Affect In Tweets. The team participated in all English language subtasks, with a method utilizing transfer learning from LSTM nets trained on large sentiment datasets combined with embeddings and lexical features. For four out of five subtasks, the system performed in the range of 92-95% of the winning systems, in terms of the competition metrics. Analysis of the results suggests that improved pre-processing and addition of more lexical features may further elevate performance.

1 Introduction

In the field of automatic emotion detection, many contributions consider the issue of detecting presence of emotions (Liu, 2012). The task of detecting intensity of emotion in a given text is less studied, but is relevant to many applications in fields such as e.g., brand management, public health, politics, and disaster handling (Mohammad, 2016). When developing prediction systems, access to suitably annotated data is critical. Most annotated emotion and affect datasets are categorical, but examples of sets annotated with intensity or degree of emotional content include EmoBank (Buechel and Hahn, 2017a,b), AFINN (Nielsen, 2011), the Pietro Facebook post set (Preoţiuc-Pietro et al., 2016), and the Warriner-Kuperman set (Warriner et al., 2013). For tweets, the Tweet Emotion Intensity Dataset (Mohammad and Bravo-Marquez, 2017) has recently been published, with more than 7000 tweets annotated with emotion category and intensity.

This paper describes methods used and results achieved with the FOI DSS contribution to the five subtasks for English tweets of SemEval 2018 Task 1: Affect in Tweets (Mohammad et al., 2018).

The paper is organized as follows. A description of Task 1 is provided in Section 2. Section 3 discusses the provided datasets. Section 4 describes the methods and system used to produce predictions of scores and labels for all subtasks. In Sections 5 and 6 results are presented and analyzed, and suggestions for improvements are outlined. Finally, concluding remarks are found in Section 7.

2 Task formulation

Task 1 consisted of five subtasks, all regarding estimation of the mental state of a tweeter, based on the tweeted text. Valence[1] intensity, as well as emotion, and emotion intensity classification, were covered. The subtasks are summarized below:

1. **Emotion intensity regression (EI-reg):** For a given tweet and emotion[2], determine the intensity of the emotion as a score $\in [0, 1]$.
2. **Emotion intensity, ordinal classification (EI-oc):** For a given tweet and emotion[2], classify the tweet into one of four ordinal classes of intensity.
3. **Valence regression (V-reg)**: For a given tweet, determine the intensity of valence as a score $\in [0, 1]$.
4. **Valence, ordinal classification (V-oc):** For a given tweet, classify it into one of seven ordinal classes corresponding to levels of positive and negative intensity.
5. **Multi-label emotion classification (E-c):** For a given tweet and eleven emotions[3], classify the tweet as neutral, or expressing one or more of the emotions.

[1] The intrinsic attractiveness (positive valence) or averseness (negative valence) of an event, object, or situation (Frijda, 1986).

[2] anger, joy, fear or sadness.

[3] anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise and trust.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 109–115
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

Subtask	Train	Val.	Test
EI anger	1701	388	1002
EI fear	2252	389	986
EI joy	1616	290	1105
EI sadness	1533	397	975
V	1181	449	937
E-c	6838	886	3259

Table 1: Number of tweets in the datasets for different subtasks. The sets for EI-reg and EI-oc were identical, as was also the case for V-reg and V-oc.

3 Data

The dataset made available for Task 1 was the AIT Dataset (Mohammad and Kiritchenko, 2018). For each subtask, labeled datasets for training and validation were released for the prediction system development phase. Intensity scores were roughly normally distributed, and ordinal classes were defined as intervals for the scores. Unlabeled test data was later released for the evaluation phase. Table 1 gives a brief overview of the data. Details on the data and annotation can be found in (Mohammad et al., 2018) and (Mohammad and Kiritchenko, 2018).

In addition to the test data, an unlabeled ”mystery” set of 16937 short texts was provided for the regression subtasks. The task organizers asked that participants in these subtasks use their existing systems to produce predictions for the mystery set as well, and the results were used to perform a bias analysis. This is further discussed in Section 5.4.

4 Method

Initially, the team focused on Subtask 4 (V-reg). Several different approaches were explored, and evaluated using the official competition metric, the Pearson Correlation Coefficient (PCC) with gold ratings. The combination of methods found to have the best performance on the V-reg task was chosen. The approach is described in Sections 4.1 - 4.3. Contributions to Subtasks 1, 2, 3 and 5 were constructed by altering the final stage model to fit each task, and tuning the hyperparameters for best performance.

4.1 Pre-processing

We performed some rudimentary pre-processing of the tweets prior to feature extraction. Following the findings reported in (Zhao, 2015) we expanded negations such as ”can't” and ”n't” etc., into ”cannot” and ”not”. The hashtag character # was also removed and we replaced user names and links with ”usr” and ”http://url”, respectively. We finally mapped unicoded emoticons into their associated emoticon text description [4].

4.2 Feature Extraction

The small amount of labeled data prevented us from automatically discovering optimal features for the different tasks. Instead, we utilized transfer learning techniques (i.e., reusing a model trained on a different but related task where more data is available) and classical natural language processing features. Three different methods were used to extract features from the tweet sets; two using variants of Long Short-Term Memory (LSTM) nets obtained by training on large sentiment datasets and extracting the internal model states, and one utilizing the Weka Affective Tweets package. The feature vectors from each of the methods described below were then concatenated to form one 5265 dimensional feature vector for each tweet.

4.2.1 Sentiment Neuron

In (Radford et al., 2017), the authors consider the problem of predicting the next character in a text given the preceding characters. More specifically, they predict next byte (each UTF-8 encoded character constitutes one to four bytes) from the previous bytes using a single layer multiplicative LSTM (Krause et al., 2016) with 4096 states. The model was trained using 82 million Amazon product reviews amounting to 38 billion bytes of training data. The authors show state-of-the-art (or close to state-of-the-art) sentiment classification performance on four different datasets when training a logistic regression classifier with the model's states as feature vector. Because of the reported strong predictive quality of the model's state we used that as one of the feature vectors for our method. We used the authors code for feature extraction available on github [5].

4.2.2 Bidirectional-LSTM

Tweets can often be quite different from typical text seen in novels, news, or product reviews.

[4] https://apps.timwhitlock.info/emoji/tables/unicode#block-6a-additional-emoticons

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

The short messages commonly contain intentional misspelling to express affects (e.g., happppppyyy), hashtags (e.g., #love), and emoticons (e.g., :-)). One option to capture the specific characteristics of tweets would be to fine-tune the sentiment neuron model described in the previous section using twitter data. We did not explore this direction in this work. Instead, in an attempt to directly capture affects, we trained (from scratch) a bidirectional LSTM on a sentiment labeled (two classes; positive and negative sentiment) twitter dataset [6]. The dataset contains 1.5 million tweets and we used 90% for training and 10% for validation. We used a bidirectional LSTM with 512 states in each direction (1024 in total). The input characters were first mapped to integers and subsequently fed to the embedding front-end (where an integer to a dense 64 dimensional embedding is learned) of the bidirectional LSTM. A dropout of 50% was used during the training for the sentiment prediction. The model achieves approximately 85% classification accuracy on the validation set. Similar to the sentiment neuron's multiplicative LSTM we use the bidirectional LSTM's state as a feature vector.

4.2.3 Weka Affective Tweets filters

A combination of tweet-level filters from the Weka Affective Tweets package (Mohammad and Bravo-Marquez, 2017) was used as the third part of the feature extraction method. These filters produce embeddings and lexical features, e.g. counts of positive and negative sentiment words, from systems such as the NRC-Canada System[7].

To evaluate contributions from different filters, the final stage model (Section 4.3) was run using their resulting feature vectors for the V-reg dataset as input. For combinations of filters, the resulting feature vectors were concatenated and run through the final stage model. Details of this evaluation can be found in Section 5.1

4.3 Final stage

For each of the different subtasks we trained a fully connected neural network with two hidden layers mapping the input feature vector (i.e., the concatenation of the feature vectors described in Sections 4.2.1 - 4.2.3) to the target value, class, or classes. The activation function for the two hidden layers was $tanh$ and the activation functions for the output layer were set to linear, softmax, and sigmoid for the V-reg/EI-reg, V-oc/EI-oc, and E-c subtasks, respectively.

The Adam optimizer (Kingma and Ba, 2015) was used for the classification subtasks with categorical cross-entropy loss for the V-oc/EI-oc subtasks and binary cross-entropy loss for the E-c subtask. For the regression subtasks of V-reg and EI-reg we used mean squared error as loss function and the ADADELTA optimizer (Zeiler, 2012). However, the performance difference between Adam and ADADELTA was minor in our regression subtasks.

We used L2-regularization on the parameters of the hidden layers. For each subtask, the hyper-parameters (i.e., the penalty and the layer sizes) of the neural network were found by a grid search evaluating the PCC (or the Jaccard similarity score for E-c subtask) on the validation data.

The hyper-parameter search range was $[0.001, 0.05]$ for the penalty and $[5, 80]$ for the two layer sizes. Many configurations with quite different hyper-parameter values resulted in very similar scores. E.g., for the V-reg subtask the configurations[8] (0.03,10,15), (0.03,15,40), and (0.0096,70,35) all resulted in PCCs in the range 0.841-0.846.

5 Results

In this section we present a performance analysis of the set of features used, as well as results on the different subtasks.

5.1 Feature evaluation

To assess the quality of the feature vectors described in Section 4.2 we computed the PCC on the V-reg subtask using the validation data. For each set of features listed in Table 2 we performed a hyper-parameter search to find the parameters of the final stage model maximizing the PCC (cf. Section 4.3).

As seen in Table 2 the features provided by the Weka Affective Tweets package have the strongest individual predictive power. From the Weka filters, the feature combination chosen to be included in the combined method was $\mathrm{W_E} + \mathrm{W_{SS}} + \mathrm{W_L}$, which produced the highest PCC during evaluation.

[6] http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/

[7] http://saifmohammad.com/WebPages/NRC-Canada-Sentiment.htm

[8] configuration = (penalty, layer 1 size, layer 2 size)

Features	PCC
Weka	
TweetToEmbeddings (W_E)[9]	0.665
TweetToEmbeddings 400 (W_{E400}) [10]	0.702
TweetToSentiStrength (W_{SS})	0.675
TweetToLexicon (W_L)	0.790
TweetToInputLexicon (W_{IL})	0.687
W_E + W_{SS} + W_L	**0.800**
W_E + W_{SS} + W_L+ W_{IL}	0.797
W_{E400} + W_{SS} + W_L	0.795
Sentiment Neuron (SN)	0.767
Bi-LSTM	0.738
SN + Bi-LSTM	0.818
Bi-LSTM + W_E + W_{SS} + W_L	0.820
SN + W_E + W_{SS} + W_L	0.838
SN + Bi-LSTM + W_E + W_{SS} + W_L	**0.846**

Table 2: **V-reg validation set:** PCC of valence intensity score predictions with gold scores for the different feature vector combinations.

Although the sentiment neuron is not trained on Twitter specific data it still shows good performance. The bidirectional LSTM has the weakest performance but still has a positive impact on the final score.

5.2 Results on validation and test data

The official competition metric was PCC for Subtasks 1-4, but as Subtask 5 was a multi-label classification task, the metric used was multi label accuracy, or Jaccard similarity score. The PCC/Jaccard similarity score for validation and test data for the FOI DSS system is presented in Table 3. For the regression tasks, the system's performance on the test data is close to the validation data results. For the classification tasks, the gap between validation and test scores is somewhat larger, indicating that the model may be biased for the validation data.

The team's ranking in different subtasks varied from 6 (out of 46 and 35 teams, respectively) for EI-reg and V-oc, to 11 of 37 for EI-oc. For Subtasks 1,3,4, and 5 the scores of our system was in the range of 92-95 % of the winning result on each subtask. The weakest performance was observed on Subtask 2 (EI-oc), with a PCC corresponding to 84 % of winning PCC. Figure 1 shows results of the FOI DSS system compared to mean, median and max competition results for test data on all English subtasks.

Subtask	Validation	Test	Baseline (Test)
EI-reg	0.739	0.737	0.520
EI-oc	0.636	0.590	0.394
V-reg	0.846	0.831	0.585
V-oc	0.818	0.777	0.509
E-c	0.554	0.544	0.442

Table 3: PCC/Jaccard similarity score on validation and test data for the FOI DSS system for all English subtasks of Task 1. The performance of the organizers' SVM unigrams baseline model on the test data is provided for comparison.

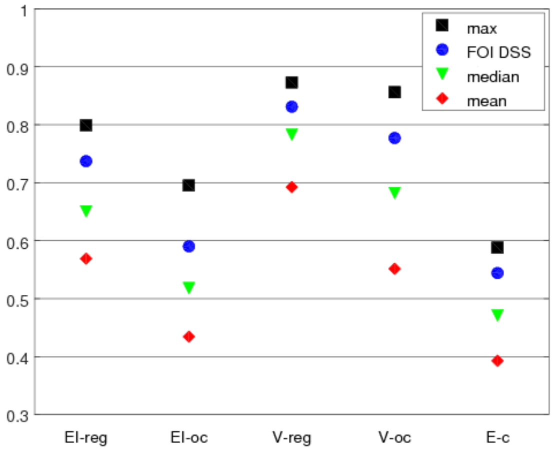

Figure 1: PCC/Jaccard similarity score of test data score and label predictions with gold scores and labels for all English subtasks. FOI DSS results compared to mean, median and max results for all participating teams.

5.3 Error analysis on the V-reg subtask

As already mentioned, our method achieved a PCC of 0.831 for the V-reg subtask on the test data. Figure 2 shows the corresponding scatter plot of the estimated and gold valence. To get some insight into potential future improvements of our system it is of interest to do analysis of tweets having poor valence estimates.

Some of the tweets from the validation and test datasets with large absolute error between the estimated and gold valence are listed in Table 4. For the first validation set tweet our method predicted a fairly low valence whereas the gold score is fairly high. A possible explanation could be that our system has problems with the constructions

[9] The TweetToEmbeddingsFeatureVector filter using embeddings trained from the small default corpus, yielding a 100-dimensional feature vector. https://affectivetweets.cms.waikato.ac.nz. .

[10] The TweetToEmbeddingsFeatureVector filter using embeddings trained from the 10 million tweets of the Edinburgh corpus (Petrović et al., 2010), yielding a 400-dimensional feature vector.

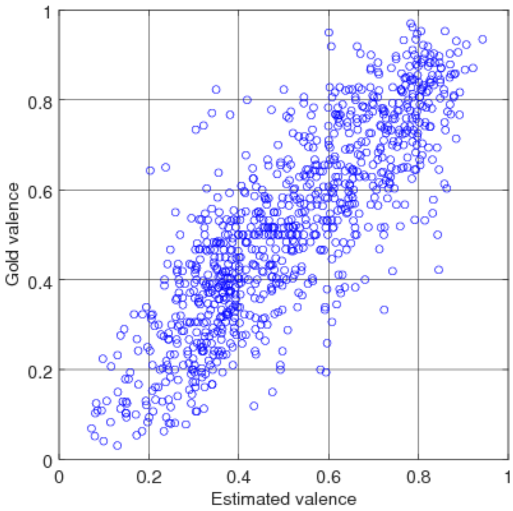

Figure 2: Scatter plot showing the estimated versus gold valence for the V-reg test dataset.

and concatenations such as B4, Thankful4all, and ImWakingUpHappyNot. Especially not properly splitting the last concatenation leaves the end of the tweet ”dreading the day” which should result in a low valence.

The emoticon of the second validation tweet, \xF0\x9F\x98\xA4, is interesting. It depicts a face with steam coming out of the nostrils, which clearly signals anger, but the mapping[4] we used describes it (wrongly according to us) as ”face with look of triumph”. In the third tweet we failed to map the emoticon \xF0\x9F\xA4\xA3 to words. The emoticon shows some sort of laughing creature.

The top two test set tweets in Table 4 had the largest prediction errors for this set. They were both predicted to have a lower valence than the gold score. Interestingly, they both contain the hashtag #blessed and include constructions using the word *not*, where, if the negation is missed, the sentiment of the tweet would change from positive to negative. Possibly, our method has trouble correctly interpreting the negation and also has failed to award enough importance to the positive sentiment word *blessed*. Since our pre-processing involved removing hashtag character #, added intensity expressed this way will not be captured.

Finally, the third test set tweet had a high predicted valence but a low gold score. This text contains both negative words and phrases such as *nervous* and *I could puke*, but also expresses laughter. It would seem our method has deemed the latter a marker of high valence, while a human reader would probably interpret it as a nervous laughter, thus low valence, considering the context provided by the tweet as a whole.

5.4 Mystery dataset and bias analysis

An analysis for inapproperiate gender and race bias in scoring and classifications was performed by the task organizers for the ”mystery” dataset (Section 3). For most teams, the bias was small (below 3%) but statistically significant, in part likely due to biases in the AIT dataset. For the FOI DSS system, the biases were below average for EI-joy, EI-sadness and valence, and 1% or less for all datasets except gender bias for EI-fear (2.3%). Biases in the datasets used to train our LSTM models as well as in the lexicons used to extract lexical features may have contributed to biases in scoring and classification.

6 Discussion

Designing high performance regression and classification algorithms using only a small amount of labeled data is always a challenge. The variability in tweets is enormous, and thus, there is a major risk of over-fitting when designing and tuning the algorithms on the the very limited labeled datasets provided for the competition. We used transfer learning and classical NLP features to alleviate the problem. We believe further improvements can be made by reducing the noise of the dataset, features, and final prediction. In the following, we discuss some of these ideas.

6.1 Pre-processing extensions

The error analysis in Section 5.3 indicates that the performance of our method could be improved by extending and refining the pre-processing. Splitting concatenations into separate words and addressing some common abbreviations would be one extension. Adjusting the emoticon lookup tables would be another.

6.2 Weka filter combinations: robustness

The combination of Weka Affective Tweets filters used in the FOI DSS system, $W_E + W_{SS} + W_L$, achieved the highest PCC during evaluation (Section 5.1). However, as results for neural networks are hard to reproduce, it should be examined what combinations of filters on average perform better. Initial findings from two such evaluations conducted after the end of the competition are reported in this section:

Dataset	Tweet	Pred.	Gold
Validation	B4 I couldnt get out of bed or look in mirror Thankful4all the support I have recieved here ImWakingUpHappyNot dreading theday	0.303	0.734
	3 and a half hour more \xF0\x9F\x98\xA4 #EXO	0.603	0.250
	@TheEllenShow I follow you bc your TV show keeps me laughing \xF0\x9F\xA4\xA3. When you #startle your guest sitting on that couch...booo...	0.461	0.783
Test	i'll have my own apartment and not have to sneak alcohol into my dorm room or worry about being loud #blessed	0.349	0.823
	mum got out of a rlly bad car crash completely not injured and i found a rlly sentimental piece of jewellery i thought i'd lost #blessed	0.203	0.643
	I'm so nervous I could puke + my body temp is rising ha ha ha ha ha	0.845	0.422

Table 4: Tweets with large prediction errors for the valence validation and test sets.

1. **W_{E400}** : When used on their own, the W_{E400} filter, which utilizes a much larger corpus[10], outperforms W_E (Table 2). Therefore it is of interest to compare performance of the two filters combined with W_{SS} and W_L.
2. **W_{IL}:** Using its default lexicon[11], W_{IL} produces 4-dimensional feature vectors. We wanted to investigate whether contributions from W_{IL} on average increases performance.

The different vector combinations were input to the final stage model (Section 4.3) for 486 different hyper-parameter configurations, and the resulting PCC scores were compared. For 59% of the configurations, W_E + W_{SS} +W_L still performed better than W_{E400} + W_{SS} + W_L. It would therefore seem that the loss of features captured by the larger W_{E400} vector is compensated for when combining the smaller W_E vector with W_{SS} + W_L.

However, W_E + W_{SS} +W_L +W_{IL} outperformed W_E + W_{SS} +W_L for 67% of the configurations. We may therefore conclude that including the W_{IL} filter would result in an overall more robust system.

6.3 Final stage: robustness

The purpose of the validation data is to measure generalization of the method. However, given the small dataset size there is as well an imminent risk of over-fitting against the validation data when searching for the optimal hyper-parameters. The latter might be the reason for the performance gaps between validation and test PCCs for the EI-oc and V-oc subtasks in particular. Also, even when using the same hyper-parameter settings, the performance (in terms of PCC/Jaccard similarity score) of the final stage varies depending on the random initialization of the network parameters. Constructing an ensemble estimate, using multiple final stage models for each subtask, could perhaps be beneficial for the performance on the test set.

7 Conclusions

This paper presents the method and results for the FOI DSS contribution to SemEval-2018 Task 1. A major challenge with this task was the small amount of available labeled data. We utilized techniques such as transfer learning as well as classical NLP features. Our system used features from Weka Affective Tweets combined with two LSTM-state vectors. Fully connected neural networks with two hidden layers were used to map the features into the target outputs for each of the subtasks. For subtasks EI-reg, V-reg, V-oc, and E-c the PCC/Jaccard similarity score of our system was in the range of 92-95 % of the winning result. The weakest performance was observed on subtask EI-oc. Initial error- and robustness analysis indicates that performance might be enhanced by improved pre-processing of the tweets, and by including more lexical features. The difference between our results on validation and test data was larger for the emotion intensity classification subtasks than for the regression and emotion classification subtasks, which would be interesting to investigate further.

Acknowledgments

We would like to thank Magnus Sahlgren for valuable input on the algorithmic design.

[11] The NRC-AffectIntensity lexicon (Mohammad, 2017).

References

Sven Buechel and Udo Hahn. 2017a. Emobank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 578–585.

Sven Buechel and Udo Hahn. 2017b. Readers vs. writers vs. texts: Coping with different perspectives of text understanding in emotion annotation. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 1–12.

Nico H Frijda. 1986. *The emotions*. Cambridge University Press.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference for Learning Representations ICLR*.

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

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Saif M Mohammad. 2016. Sentiment analysis: detecting valence, emotions, and other affectual states from text. *Emotion Measurement*, pages 201–237.

Saif M. Mohammad. 2017. Word affect intensities. *CoRR*, abs/1704.08798.

Saif M Mohammad and Felipe Bravo-Marquez. 2017. Emotion intensities in tweets. *arXiv preprint arXiv:1708.03696*.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Saif M. Mohammad and Svetlana Kiritchenko. 2018. Understanding emotions: A dataset of tweets to study interactions between affect categories. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference*, Miyazaki, Japan.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. The edinburgh twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 25–26.

Daniel Preoţiuc-Pietro, H Andrew Schwartz, Gregory Park, Johannes Eichstaedt, Margaret Kern, Lyle Ungar, and Elisabeth Shulman. 2016. Modelling valence and arousal in facebook posts. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 9–15.

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

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Jianqiang Zhao. 2015. Pre-processing boosting twitter sentiment analysis? In *IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 748–753.