SeerNet at SemEval-2018 Task 1: Domain Adaptation for Affect in Tweets

Venkatesh Duppada, Royal Jain, Sushant Hiray
Seernet Technologies, LLC
{venkatesh.duppada, royal.jain, sushant.hiray}@seernet.io

Abstract

The paper describes the best performing system for the SemEval-2018 Affect in Tweets (English) sub-tasks. The system focuses on the ordinal classification and regression sub-tasks for valence and emotion. For ordinal classification valence is classified into 7 different classes ranging from -3 to 3 whereas emotion is classified into 4 different classes 0 to 3 separately for each emotion namely anger, fear, joy and sadness. The regression sub-tasks estimate the intensity of valence and each emotion. The system performs domain adaptation of 4 different models and creates an ensemble to give the final prediction. The proposed system achieved 1st position out of 75 teams which participated in the fore-mentioned subtasks. We outperform the baseline model by margins ranging from 49.2% to 76.4%, thus, pushing the state-of-the-art significantly.

1 Introduction

Twitter is one of the most popular micro-blogging platforms that has attracted over 300M daily users[1] with over 500M [2] tweets sent every day. Tweet data has attracted NLP researchers because of the ease of access to large data-source of people expressing themselves online. Tweets are micro-texts comprising of emoticons, hashtags as well as location data, making them feature rich for performing various kinds of analysis. Tweets provide an interesting challenge as users tend to write grammatically incorrect and use informal and slang words.

In domain of natural language processing, emotion recognition is the task of associating words, phrases or documents with emotions from predefined using psychological models. The classification of emotions has mainly been researched from two fundamental viewpoints. (Ekman, 1992) and (Plutchik, 2001) proposed that emotions are discrete with each emotion being a distinct entity. On the contrary, (Mehrabian, 1980) and (Russell, 1980) propose that emotions can be categorized into dimensional groupings.

Affect in Tweets (Mohammad et al., 2018) - shared task in SemEval-2018 focuses on extracting affect from tweets confirming to both variants of the emotion models, extracting valence (dimensional) and emotion (discrete). Previous version of the task (Mohammad and Bravo-Marquez, 2017) focused on estimating the emotion intensity in tweets. We participated in 4 sub-tasks of Affect in Tweets, all dealing with English tweets. The sub-tasks were: EI-oc: Ordinal classification of emotion intensity of 4 different emotions (anger, joy, sadness, fear), EI-reg: to determine the intensity of emotions (anger, joy, sadness, fear) into a real-valued scale of 0-1, V-oc: Ordinal classification of valence into one of 7 ordinal classes [-3, 3], V-reg: determine the intensity of valence on the scale of 0-1.

Prior work in extracting Valence, Arousal, Dominance (VAD) from text primarily relied on using and extending lexicons (Bestgen and Vincze, 2012) (Turney et al., 2011). Recent advancements in deep learning have been applied in detecting sentiments from tweets (Tang et al., 2014), (Liu et al., 2012), (Mohammad et al., 2013).

In this work, we use various state-of-the-art machine learning models and perform domain adaptation (Pan and Yang, 2010) from their source task to the target task. We use multi-view ensemble learning technique (Kumar and Minz, 2016) to produce the optimal feature-set partitioning for the classifier. Finally, results from multiple such classifiers are stacked together to create an ensemble (Polikar, 2012).

[1] https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

[2] http://www.internetlivestats.com/twitter-statistics/

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 18–23
New Orleans, Louisiana, June 5–6, 2018.

In this paper, we describe our approach and experiments to solve this problem. The rest of the paper is laid out as follows: Section 2 describes the system architecture, Section 3 reports results and inference from different experiments. Finally we conclude in Section 4 along with a discussion about future work.

2 System Description

2.1 Pipeline

Figure 1 details the System Architecture. We now describe how all the different modules are tied together. The input raw tweet is pre-processed as described in Section 2.2. The processed tweet is passed through all the feature extractors described in Section 2.3. At the end of this step, we extract 5 different feature vectors corresponding to each tweet. Each feature vector is passed through the model zoo where classifiers with different hyper parameters are tuned. The models are described in Section 2.4. For each vector, the results of top-2 performing models (based on cross-validation) are retained. At the end of this step, we've 10 different results corresponding to each tweet. All these results are ensembled together via stacking as described in Section 2.4.3. Finally, the output from the ensembler is the output returned by the system.

2.2 Pre-processing

The pre-processing step modifies the raw tweets to prepare for feature extraction. Tweets are pre-processed using **tweettokenize** [3] tool. Twitter specific keywords are replaced with tokens, namely, `USERNAME`, `PHONENUMBER`, `URLs`, `timestamps`. All characters are converted to lowercase. A contiguous sequence of emojis is first split into individual emojis. We then replace an emoji with its description. The descriptions were scraped from EmojiPedia[4].

2.3 Feature Extraction

As mentioned in Section 1, we perform transfer learning from various state-of-the-art deep learning techniques. We will go through the following sub-sections to understand these models in detail.

2.3.1 DeepMoji

DeepMoji (Felbo et al., 2017) performs distant supervision on a very large dataset (1246 million tweets) comprising of noisy labels (emojis). DeepMoji was able to obtain state-of-the-art results in various downstream tasks using transfer learning. This makes it an ideal candidate for domain adaptation into related target tasks. We extract 2 different feature sets by extracting the embeddings from the softmax and the attention layer from the pre-trained DeepMoji model. The vector from softmax layer is of dimension 64 and the vector from attention layer is of dimension 2304.

2.3.2 Skip-Thought Vectors

Skip-Thought vectors (Kiros et al., 2015) is an off-the-shelf encoder that can produce highly generic sentence representations. Since tweets are restricted by character limit, skip-thought vectors can create a good semantic representation. This representation is then passed to the classifier. The representation is of dimension 4800.

2.3.3 Unsupervised Sentiment Neuron

(Radford et al., 2017) developed an unsupervised system which learned an excellent representation of sentiment. The original model was trained to generate amazon reviews, this makes the sentiment neuron an ideal candidate for transfer learning. The representation extracted from Sentiment Neuron is of size 4096.

2.3.4 EmoInt

Apart from all the pre-trained embeddings, we choose to also include various lexical features bundled through the EmoInt package [5] (Duppada and Hiray, 2017) The lexical features include AFINN (Nielsen, 2011), NRC Affect Intensities (Mohammad, 2017), NRC-Word-Affect Emotion Lexicon (Mohammad and Turney, 2010), NRC Hashtag Sentiment Lexicon and Sentiment140 Lexicon (Mohammad et al., 2013). The final feature vector is the concatenation of all the individual features. This feature vector is of size (141, 1).

This gives us five different feature vector variants. All of these feature vectors are passed individually to the underlying models. The pipeline is explained in detail in Section 2.1

2.4 Machine Learning Models

We participated in 4 sub-tasks, namely, EI-oc, EI-reg, V-oc, V-reg. Two of the sub-tasks are ordinal classification and the remaining two are regressions. We describe our approach for building ML

[3] https://github.com/jaredks/tweetokenize
[4] https://emojipedia.org/
[5] https://github.com/SEERNET/EmoInt

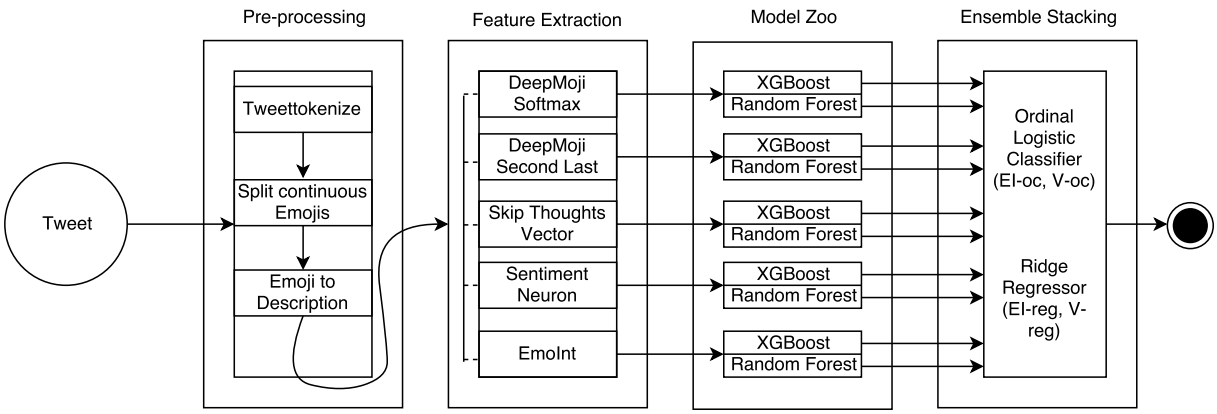

Figure 1: System Architecture.

models for both the variants in the upcoming sections.

2.4.1 Ordinal Classification

We participated in the emotion intensity ordinal classification where the task was to predict the intensity of emotions from the categories anger, fear, joy, and, sadness. Separate datasets were provided for each emotion class. The goal of the subtask of valence ordinal classification was to classify the tweet into one of 7 ordinal classes [-3, 3]. We experimented with XG Boost Classifier, Random Forest Classifier of sklearn (Pedregosa et al., 2011).

2.4.2 Regression

For the regression tasks (E-reg, V-reg), the goal was to predict the intensity on a scale of 0-1. We experimented with XG Boost Regressor, Random Forest Regressor of sklearn (Pedregosa et al., 2011).

The hyper-parameters of each model were tuned separately for each sub-task. The top-2 best models corresponding to each feature vector type were chosen after performing 7-fold cross-validation.

2.4.3 Stacking

Once we get the results from all the classifiers/regressors for a given tweet, we use stacking ensemble technique to combine the results. In this case, we pass the results from the models to a meta classifier/regressor as input. The output of this meta model is treated as the final output of the system.

We observed that using ordinal regressors gave us better performance than using classifiers which treat each output class as disjoint. Ordinal Regression is a family of statistical learning methods where the output variable is discrete and ordered. We use the ordinal logistic classification with squared error (Rennie and Srebro, 2005) from the python library Mord. [6] (Rennie and Srebro, 2005)

In case of regression sub-tasks we observed the best cross validation results with Ridge Regression. Hence, we chose Ridge Regression as the meta regressor.

Task	Baseline	2nd best	Our Results
EI-reg	0.520	0.776	**0.799**
EI-oc	0.394	0.659	**0.695**
V-reg	0.585	0.861	**0.873**
V-oc	0.509	0.833	**0.836**

Table 1: Primary metrics across various sub-tasks.

3 Results and Analysis

3.1 Task Results

The metrics used for ranking various systems are discussed in this section.

3.1.1 Primary Metrics

Pearson correlation with gold labels was used as a primary metric for ranking the systems. For EI-reg and EI-oc tasks Pearson correlation is macro-averaged (MA Pearson) over the four emotion categories.

Table 1 describes the results based on primary metrics for various sub-tasks in English language. Our system achieved the best performance in each of the four sub-tasks. We have also included the results of the baseline and second best performing system for comparison. As we can observe,

[6] https://github.com/fabianp/mord

Task	Pearson (SE)	Kappa	Kappa (SE)
V-oc	0.884 (1)	0.831 (1)	0.873 (1)
EI-oc	0.547 (1)	0.669 (1)	0.503 (1)

Table 2: Secondary metrics for ordinal classification sub-tasks. System rank is mentioned in the brackets.

Task	Pearson (gold in 0.5-1)
V-reg	0.697 (1)
EI-reg	0.638 (1)

Table 3: Secondary metrics for regression sub-tasks. System rank is mentioned in brackets.

our system vastly outperforms the baseline and is a significant improvement over the second best system, especially, in the emotion sub-tasks.

3.1.2 Secondary Metrics

The competition also uses some secondary metrics to provide a different perspective on the results. Pearson correlation for a subset of the test set that includes only those tweets with intensity score greater or equal to 0.5 is used as the secondary metric for the regression tasks. For ordinal classification tasks following secondary metrics were used:

- Pearson correlation for a subset of the test set that includes only those tweets with intensity classes low X, moderate X, or high X (where X is an emotion). The organizers refer to this set of tweets as the some-emotion subset (SE).
- Weighted quadratic kappa on the full test set
- Weighted quadratic kappa on the some-emotion subset of the test set

The results for secondary metrics are listed in Table 2 and 3. We have also included the ranking in brackets along with the score. We see that our system achieves the top rank according to all the secondary metrics, thus, proving its robustness.

3.2 Feature Importance

The performance of the system is highly dependent on the discriminative ability of the tweet representation generated by the featurizers. We measure the predictive power for each of the featurizer used by calculating the pearson correlation of the system using only that featurizer. We describe the results for each sub task separately in tables 4-7.

Feature Set	Pearson
Deepmoji (softmax layer)	0.808
Deepmoji (attention layer)	0.843
EmoInt	0.823
Unsupervised sentiment Neuron	0.714
Skip-Thought Vectors	0.777
Combined	**0.873**

Table 4: Pearson Correlation for V-reg task. Best results are highlighted in bold.

Feature Set	Pearson
Deepmoji (softmax layer)	0.780
Deepmoji (attention layer)	0.813
EmoInt	0.785
Unsupervised sentiment Neuron	0.685
Skip-Thought Vectors	0.748
Combined	**0.836**

Table 5: Pearson Correlation for V-oc task. Best results are highlighted in bold.

Feature Set	Pearson
Deepmoji (softmax layer)	0.703
Deepmoji (attention layer)	0.756
EmoInt	0.694
Unsupervised sentiment Neuron	0.548
Skip-Thought Vectors	0.656
Combined	**0.799**

Table 6: Macro-Averaged Pearson Correlation for EI-reg task. Best results are highlighted in bold.

Feature Set	Pearson
Deepmoji softmax layer	0.611
Deepmoji attention layer	0.664
EmoInt	0.596
Unsupervised sentiment Neuron	0.445
Skip-Thought Vectors	0.557
Combined	**0.695**

Table 7: Macro-Averaged Pearson Correlation for EI-oc task. Best results are highlighted in bold.

We observe that deepmoji featurizer is the most powerful featurizer of all the ones that we've used. Also, we can see that stacking ensembles of models trained on the outputs of multiple featurizers gives a significant improvement in performance.

3.3 System Limitations

We analyze the data points where our model's prediction is far from the ground truth. We observed some limitations of the system, such as, sometimes understanding a tweet's requires contextual knowledge about the world. Such examples can be very confusing for the model. We use deepmoji pre-trained model which uses emojis as proxy for labels, however partly due to the nature of twitter conversations same emojis can be used for multiple emotions, for example, joy emojis can be sometimes used to express joy, sometimes for sarcasm or for insulting someone. One such example is *'Your club is a laughing stock'*. Such cases are sometimes incorrectly predicted by our system.

4 Future Work & Conclusion

The paper studies the effectiveness of various representations of tweets and proposes ways to combine them to obtain state-of-the-art results. We also show that stacking ensemble of various classifiers learnt using different representations can vastly improve the robustness of the system.

Further improvements can be made in the preprocessing stage. Instead of discarding various tokens such as punctuation's, incorrectly spelled words, etc, we can utilize the information by learning their semantic representations. Also, we can improve the system performance by employing multi-task learning techniques as various emotions are not independent of each other and information about one emotion can aid in predicting the other. Furthermore, more robust techniques can be employed for distant supervision which are less prone to noisy labels to get better quality training data.

References

Yves Bestgen and Nadja Vincze. 2012. Checking and bootstrapping lexical norms by means of word similarity indexes. *Behavior research methods*, 44(4):998–1006.

Venkatesh Duppada and Sushant Hiray. 2017. Seernet at emoint-2017: Tweet emotion intensity estimator. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 205–211.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Vipin Kumar and Sonajharia Minz. 2016. Multi-view ensemble learning: an optimal feature set partitioning for high-dimensional data classification. *Knowledge and Information Systems*, 49(1):1–59.

Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. 2012. Emoticon smoothed language models for twitter sentiment analysis. In *Aaai*.

Albert Mehrabian. 1980. Basic dimensions for a general psychological theory implications for personality, social, environmental, and developmental studies.

Saif M Mohammad. 2017. Word affect intensities. *arXiv preprint arXiv:1704.08798*.

Saif M Mohammad and Felipe Bravo-Marquez. 2017. Wassa-2017 shared task on emotion intensity. *arXiv preprint arXiv:1708.03700*.

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, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Saif M Mohammad and Peter D Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics.

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

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Robert Plutchik. 2001. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American scientist*, 89(4):344–350.

Robi Polikar. 2012. Ensemble learning. In *Ensemble machine learning*, pages 1–34. Springer.

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

Jason DM Rennie and Nathan Srebro. 2005. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*, pages 180–186. Kluwer Norwell, MA.

James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565.

Peter D Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 680–690. Association for Computational Linguistics.