SSN MLRG1 at SemEval-2018 Task 1: Emotion and Sentiment Intensity Detection Using Rule Based Feature Selection

Angel Deborah S, Rajalakshmi S, S Milton Rajendram, Mirnalinee T T
SSN College of Engineering
Chennai 603 110, Tamil Nadu, India
angeldeborahs@ssn.edu.in, rajalakshmis@ssn.edu.in,
miltonrs@ssn.edu.in, mirnalineett@ssn.edu.in

Abstract

The system developed by the SSN MLRG1 team for Semeval-2018 task 1 on affect in tweets uses rule based feature selection and one-hot encoding to generate the input feature vector. Multilayer Perceptron was used to build the model for emotion intensity ordinal classification, sentiment analysis ordinal classification and emotion classfication subtasks. Support Vector Regression was used to build the model for emotion intensity regression and sentiment intensity regression subtasks.

1 Introduction

Twitter is a huge microblogging service with more than 500 million tweets per day from different locations of the world and in different languages (Saif and Felipe, 2017). Tweets are often used to convey ones emotions, opinions towards products, and stance over issues (Nabil et al., 2016). Automatically detecting emotion intensities in tweets has several applications, including commerce (Jansen et al., 2009), crisis management (Verma et al., 2011), tracking brand and product perception, tracking support for issues and policies, and tracking public health and well-being (Chew and Eysenbach, 2010). The task is challenging because of the informal writing style, the semantic diversity of content as well as the "unconventional" grammar. These challenges in building a classification model and regression model can be handled by using proper approaches to feature generation and machine learning.

There are several machine learning techniques that can be used for sentiment intensity prediction or emotion intensity prediction. Some of the approaches inlclude Artificial Neural Network (ANN) (Sudipta et al., 2017), Random Forests, Support Vector Machine (SVM), Naive Bayes (NB) (Tabari et al., 2017), Multi-Kernel Gaussian Process (MKGP) (Angel Deborah et al., 2017a,b), AdaBoost Regressor (ABR), Bagging Regressor (BR)(Jiang et al., 2017) and Deep Learning (DL)(Pivovarova et al., 2017).

2 Multi-Layer Perceptron

A Multilayer Perceptron (MLP) as shown in Figure 1 is a feed-forward Neural Network model that maps input data sets onto appropriate output sets. An MLP has many layers of nodes in a directed graph, with each layer connected to the next layer. A neuron is a processing element with activation function (in the input layer the activation function is not applied). The output layer has as many nuerons as the number of class labels in the problem. Each connection has a weight assigned to it. Output of each neuron is calculated by applying the activation function on the weighted sum of the inputs. Linear, sigmoid, tanh, elu, softplus, softmax and relu are some of the commonly used activation functions. The supervised learning problem of the MLP can be solved with the back-propagation algorithm (Haykin, 1998). The algorithm consists of two steps. In the forward pass, the predicted outputs are calculated for the given inputs . In the backward pass, partial derivatives

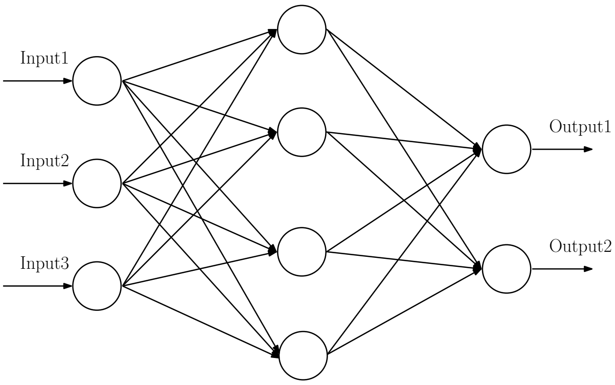

Figure 1: Multi-Layer Perceptron.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 324–328
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

of the cost function with respect to the weight parameters are propagated back through the network. The chain rule of differentiation gives very similar computational rules for the backward pass as the ones for the forward pass. The network weights can then be adapted using any gradient-based optimisation algorithm.

MLP was used in implementing for the following subtasks:

1. **EI-oc** (an emotion intensity ordinal classification task): Given a tweet and an emotion E, classify the tweet into one of four ordinal intensity classes of E that best represents the mental state of the tweeter.
2. **V-oc** (a sentiment analysis, ordinal classification task): Given a tweet, classify it into one of seven ordinal classes, corresponding to various levels of positive and negative sentiment intensity, that best represents the mental state of the tweeter.
3. **E-c** (an emotion classification task): Given a tweet, classify it as neutral or no emotion or as one, or more, of eleven given emotions that best represent the mental state of the tweeter.

3 Support Vector Regression

Support Vector Machines (SVM) are characterized by usage of kernels, absence of local minima, sparseness of the solution and capacity control obtained by acting on the margin or on number of support vectors, etc. As in classification, support vector regression (SVR) is characterized by the use of kernels, sparse solution, and Vapnik-Chervonenkis control of the margin and the number of support vectors. Although less popular than SVM, SVR has been proven to be an effective tool in real-value function estimation (Awad and Khanna, 2015). The idea of SVR is based on the computation of a linear regression function in a high dimensional feature space where the input data are mapped via a non-linear function. It contains all the main features that characterize maximum margin algorithm: a non-linear function is learned by a linear learning machine mapping into high dimensional kernel-induced feature space. The capacity of the system is controlled by parameters that do not depend on the dimensionality of feature space. Instead of minimizing the observed training error, Support Vector Regression (SVR) attempts to minimize the generalization error bound so as to achieve generalization performance.

SVR was used in implementing for the following subtasks:

1. **EI-reg** (an emotion intensity regression task): Given a tweet and an emotion E, determine the intensity of E that best represents the mental state of the tweeter, a real-valued score between 0 (least E) and 1 (most E).
2. **V-reg** (a sentiment intensity regression task): Given a tweet, determine the intensity of sentiment or valence (V) that best represents the mental state of the tweeter, a real-valued score between 0 (most negative) and 1 (most positive).

4 System Overview

The system consists of the following modules: data extraction, preprocessing, rule based feature selection, feature vector generation and building the model – Multilayer Perceptron model for clasification subtasks and Support Vector Regression for regression subtasks. The algorithm for preprocessing of the data is outlined below:

```
Algorithm: Data extraction and Preprocessing.
Input: Input dataset.
Output: Tokenized words and their parts of speech.
begin
  1. Separate labels and sentences
  2. Perform tokenization using word_tokenize, the function for tokenizing in the NLTK toolkit.
  3. Perform Parts of Speech tagging using pos_tag function from the NLTK toolkit.
  4. Return the tokenized words and their parts of speech as inputs to rule based feature selection.
end
```

The algorithm for rule based feature selection and feature vector generation is outlined below:

```
Algorithm: Rule based feature selection and feature vector generation.
Input: Tokenized words and their parts of speech.
Output: Feature vector.
begin
For each of the tokenized words, falling under one of the categories listed in Table 1, do the following steps.
  1. Lemmatize the word using WordNet Lemmatizer from the NLTK toolkit.
  2. Insert the lemmatized word into the dictio-
```

```
    nary.
  3. Represent each sentence as a feature vector
    using one-hot encoding by looking up the
    dictionary.
  4. Return the feature vector generated as the in-
    put to build the model.
end
```

Abbreviation	Parts of Speech
VB	verb, base form
VBZ	verb, 3rd person sing. present
VBP	verb, non 3rd sing. present
VBD	verb, past tense
VBG	verb, gerund/present participle
VBN	verb, past participle
JJ	adjective
JJR	adjective, comparative
JJS	adjective, superlative
RB	adverb, very
RBR	adverb, comparative
RBS	adverb, superlative
NN	noun, singular
NNP	proper noun, singular
NNS	noun, plural
NNPS	proper noun, plural

Table 1: Parts of speech categories.

The Multi-Layer Perceptron (MLP) model is built by the following algorithm.

```
Algorithm: Build a Multi-Layer Perceptron
model.
Input: Feature vectors and actual output labels.
Output: Learned MLP model.
begin
  1. Represent feature vectors as XTrain and ac-
    tual output label as YTrain.
  2. Build the initial classification model with two
    hidden layers and the output layer, using
    relu and softmax activation functions, re-
    spectively.
  3. Optimize the classification model using
    nadam optimizer of keras package
  4. Return the learned model.
end
```

The Support Vector Regression(SVR) model is bulit with the following algorithm.

```
Algorithm: Build a SVR model.
Input: Feature vectors and actual output labels.
Output: Learned SVR model.
begin
  1. Represent feature vectors as XTrain and ac-
    tual output label as YTrain.
  2. Build the initial classification model using
    appropriate kernel function.
  3. Optimize the model parameters using a
    heuristic approach.
  4. Return the learned model.
end
```

5 Performance Evaluation

We evaluated the system only for English language. The results obtained using MLP and SVR for the subtasks are tabulated in Table 2 to Table 6. From Table 2 which shows the Pearson scores obtained for SVR, we can infer that SVR predicts joy better compared to anger, fear and sadness. Similarly, from Table 3 which shows the Pearson scores obtained for MLP, we observe that MLP model predicts joy better compared to anger, fear and sadness. The Pearson score for valence intensity regression and sentiment intensity ordinal classification are given in Table 4 and 5 respectively.

anger	fear	joy	sadness	macro averaged
0.490	0.490	0.502	0.461	0.486

Table 2: Pearson score for EI-reg (emotion intensity regression) using SVR.

anger	fear	joy	sadness	macro averaged
0.365	0.363	0.390	0.383	0.375

Table 3: Pearson score for EI-oc (emotion intensity ordinal classification) using MLP.

Pearson (all instances)	Pearson (gold in 0.5-1)
0.582	0.424

Table 4: Pearson score for V-reg (valence intensity regression) using SVR.

Pearson (all classes)	Pearson (some-emotion)
0.427	0.479

Table 5: Pearson score for V-oc (Sentiment intensity ordinal classification) using MLP.

The Pearson score r is calculated using the Equation 1.

$$r = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (1)$$

Accuracy	Micro-avg F1	Macro-avg F1
0.468	0.595	0.476

Table 6: Pearson score for E-c (multi-label emotion class) using MLP.

where Y is actual output and y is predicted output. The accuracy, micro-averaged F score and macro-avearged F score for emotion classification are given in Table 6. The metrics are defined from Equations 2 to 5.

$$\text{Accuracy} = \frac{1}{|T|} \sum_{t \in T} \frac{|G_t \cap P_t|}{|G_t \cup P_t|} \tag{2}$$

where G_t is the set of the gold labels for tweet t, P_t is the set of the predicted labels for tweet t, and T is the set of tweets.

$$\text{Micro-avearge } F = \frac{2 \times \text{micro-}P \times \text{micro-}R}{\text{micro-}P \times \text{micro-}R} \tag{3}$$

where micro-P is micro-averaged precision and micro-R is micro-averaged recall

$$F = \frac{2 \times P_e \times R_e}{P_e \times R_e} \tag{4}$$

$$\text{Macro-average } F = \frac{1}{|E|} \sum_{e \in E} F_e \tag{5}$$

where P_e is precision, R_e is recall and E is the given set of eleven emotions.

6 Conclusion

We have presented the results of using MultiLayer Perceptron for emotion intensity ordinal classification, sentiment analysis ordinal classification and emotion classification. We built a basic MLP, which has an input layer, two hidden layers with 128 and 64 neurons, and an output layer with as many neurons as the number of class labels. We have used nadam optimizer with learning rate as 0.01. We have also presented the results of using Support Vector Regression for emotion intenisty and sentiment intensity regression. It is observed that both MLP and SVR predict joy more accurately when compared to anger, fear and sadness. We analyzed the feature vectors generated for various emotions. Feature vectors generated for joy helps to achieve better results than for other emotions. We used rule based feature selection and one-hot encoding to generate input feature vectors for buliding the models. The results obtained can be enhanced by using different feature selection approaches and incorporating sentiment lexicons.

References

S Angel Deborah, S Milton Rajendram, and T T Mirnalinee. 2017a. Ssn mlrg1 at semeval-2017 task 4: Sentiment analysis in twitter using multi-kernel gaussian process classifier. In *Proceedings the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 709–712. ACL,Vancouver, Canada.

S Angel Deborah, S Milton Rajendram, and T T Mirnalinee. 2017b. Ssn mlrg1 at semeval-2017 task 5: Fine-grained sentiment analysis using multiple kernel gaussian process regression model. In *Proceedings the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 823–826. ACL,Vancouver, Canada.

M Awad and R Khanna. 2015. *Support Vector Regression. In: Efficient Learning Machines*. Apress, Berkeley, CA.

C. Chew and G. Eysenbach. 2010. Pandemics in the age of twitter: content analysis of tweets during the 2009 h1n1 outbreak. *PloS ONE*, 5(11):1–13.

Simon Haykin. 1998. *Neural Networks – A Comprehensive Foundation*. Second Edition, Prentice-Hall, Englewood Cliffs, NJ.

B. J. Jansen, K. Sobel M. Zhang, and A. Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*, 60(11):2169–2188.

Mengxiao Jiang, Man Lan1, and Yuanbin Wu. 2017. Ecnu at semeval-2017 task 5: An ensemble of regression algorithms with effective features for fine-grained sentiment analysis in financial domain. In *Proceedings the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 888–893. ACL,Vancouver, Canada.

Mahmoud Nabil, Mohamed Aly, and Amir F. Atiya. 2016. Cufe at semeval-2016 task 4: A gated recurrent model for sentiment classification. In *Proceedings of SemEval-2016.ACL, Vancouver, Canada.*, pages 52–57.

Lidia Pivovarova, Llorenc Escoter, Arto Klami, and Roman Yangarber. 2017. Hcs at semeval-2017 task 5: Sentiment detection in business news using convolutional neural networks. In *Proceedings the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 842–846. ACL,Vancouver, Canada.

M. Mohammad Saif and Bravo-Marquez Felipe. 2017. Emotion intensities in tweets. In *Proceedings of sixth joint conference on lexical and computational semantics (*Sem). Vancouver, Canada.*, pages 65–77.

Kar Sudipta, Suraj Maharjan, and Thamar Solorio. 2017. Ritual-uh at semeval-2017 task 5: Sentiment analysis on financial data using neural networks. In *Proceedings the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 877–882. ACL,Vancouver, Canada.

Narges Tabari, Armin Seyeditabari, and Wlodek Zadrozny. 2017. Sentiheros at semeval-2017 task 5: An application of sentiment analysis on financial tweets. In *Proceedings the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 857–860. ACL,Vancouver, Canada.

S. Verma, S. Vieweg, W. J. Corvey, L. Palen, J. H. Martin, M. Palmer, A. Schram, and K. M. Anderson. 2011. Natural language processing to the rescue extracting situational awareness tweets during mass emergency. In *Proceedings of 5th International Conference on Web and Social Media*.