Amrita_student at SemEval-2018 Task 1: Distributed Representation of Social Media Text for Affects in Tweets

Nidhin A Unnithan, Shalini K., Barathi Ganesh H. B., M. Anand Kumar, K. P. Soman
Center for Computational Engineering and Networking (CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
nidhinkittu5470@gmail.com, shalinikholla@gmail.com,
bharathiganesh.hb@gmail.com, m_anandkumar@cb.amrita.edu,
kp_soman@amrita.edu

Abstract

In this paper we did an analysis of "Affects in Tweets" which was one of the task conducted by SemEval 2018. Task was to build a model which is able to do regression and classification of different emotions from the given tweets data set. We developed a base model for all the subtasks using distributed representation (Doc2Vec) and applied machine learning techniques for classification and regression. Distributed representation is an unsupervised algorithm which is capable of learning fixed length feature representation from variable length texts. Machine learning techniques used for regression is 'Linear Regression' while 'Random Forest Tree' is used for classification purpose. Empirical results obtained for all the subtasks by our model are shown in this paper.

1 Introduction

Most basic form of communication between humans is through language. Thus it can act as a medium of how we are feeling at any particular instance. For example, if we are angry at someone rather than just hitting him first we would express our feeling through our words. Thus from a conversion we can make out the different emotions a person is going through at that time. Apart from this social media texts can be used for determining the class of a person as described in (Ganesh H. B. et al., 2016b). In this work we are doing 2 ordinal classification, 1 classification and 2 regression of different emotions that people exhibits through tweets obtained from twitter (Mohammad and Kiritchenko, 2018; Bravo-Marquez et al., 2014; Mohammad et al., 2013) for three different languages namely Arabic, English and Spanish. The data set given has tweets from all the three languages for each subtask (Mohammad et al., 2018; Mohammad and Kiritchenko, 2018). There is a total of five subtask an emotion intensity regression task, an emotion intensity ordinal classification task, a sentiment intensity regression task, a sentiment analysis ordinal classification task and an emotion classification task.

We used distributed representation (Le and Mikolov, 2014; Ganesh H. B. et al., 2016a) to create feature vector which can be feed as input to machine learning algorithms for classification and regression. Bag-of-words is one of the most common method used to create fixed length feature vectors but the ordering and semantics of the words are ignored in this method. By using Doc2Vec, an unsupervised learning algorithm, we can create fixed length features from variable length data. Thus by using Doc2Vec we can preserve the ordering as well as the semantics of data. Another method for word representation is distributional representation (Ganesh H. B. et al., 2018) which is an extension of co-occurrence based representation and have the same disadvantages as co-occurrence based methods.

Once the feature vector is created it is pushed into machine learning algorithm for classification and regression. We have used Random Forest Tree for classification which is an ensemble learning method that creates a number of decision trees during training and gives an output class which appears most often. For regression we used Linear Regression which tries to fit a line between the actual and predicted values by minimizing the error sum of squares between them. The final model is obtained after doing hyper parameter tuning for Doc2Vec $size$ and $n\_estimator$, max_depth for Random Forest Tree which are fixed through a grid search method before pushing to machine learning algorithms.

Section 2 of this paper gives a brief introduction to corpus. Section 3 describes the theory of different methods used. Section 4 describes the method-

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 319–323
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

ology used. Section 5 covers result and discussion. Section 6 talks about our conclusion.

2 Corpus

The given corpus consists of tweets from three different languages for all five subtasks. The languages are English, Arabic and Spanish. Each language have training, development and test data set (Mohammad et al., 2018; Mohammad and Kiritchenko, 2018).

While building the model training data set was splitted into 80% for training and 20% for testing. Training and development data set consist of tweet id, tweet, affect dimension and intensity score while test data set has entries as none at intensity scores.

3 Methodology

3.1 Distributed Representation

Doc2Vec is an unsupervised learning algorithm which gives a fixed length vector representation of a variable length text. The text can be a sentence, paragraph or document. It is an extension of Word2Vec in which a vector representation of words are given inorder to predict a word given the vector representation of context words are given. Word2Vec is inspired because it can be used to predict the next word in a sentence given the context word vectors thus capturing the semantics of the sentence even though the word vectors are randomly initialized. Instead of word vector we will use document vector to predict next word given context from a document in Doc2Vec. In document vector every document is represented by a column of unique vector called document matrix and words are represented by unique vectors called word matrix. Next word in a context is predicted by the concatenation or averaging of document and word vectors.

In Doc2Vec the document vector is same for all context generated from same document but differs across documents. However word vector matrix is same for different document, i.e., the vector representation of same word across different document have the same vector representation.

3.2 Linear Regression

For regression tasks Linear Regression was used. Linear Regression tries to fit a line between the actual and predicted values by minimizing the error sum of squares between them. In a Linear Regression problem there will be one dependent variable and an independent variable. A regression tries to verify two objective, firstly whether a satisfactory prediction can be made by a set of predictor variables and secondly which all variables play an important role in predicting the outcome variable. The estimated regression outputs are used to explain the connection between independent and dependent variables.

3.3 Random Forest Tree

For classification problem we used Random Forest Tree. It is an ensemble learning method that creates a number of decision trees during training and gives an output class which appears most often. Advantage of Random Forest Tree is its ability to control over-fitting by taking an average of all the decision trees for prediction. If more than one algorithm of same or different kind are combined to classify an object such an algorithm is called ensemble algorithm. For example it may run a prediction on SVM, Naive Bayes and Decision Tree before taking the vote for classification of test object.

3.4 Experiment

The corpus was obtained from SemEval2018 website. Once the data was obtained the first process was to extract tweets from the data for all the languages. Once every thing was extracted from the document next step was to build a Doc2Vec model from the extracted tweets which will produce feature vectors which can be used as inputs for our machine learning techniques for regression and classification tasks. Gensim library was used to build the Doc2Vec model. Sklearn library was used for Random Forest Tree and Linear Regression.

Before fixing the Doc2Vec base model we did hyper parameter tuning for all subtasks in all languages. The parameters tuned for regression tasks was Doc2Vec $size$ and for classification were Doc2Vec $size$ and $n\_estimator$, max_depth for Random Forest Tree. $size$ of Doc2Vec means the dimensionality of the feature vector, i.e., in which dimension each document in a corpus is represented as. $n\_estimator$ of Random Forest Tree means the number of decision trees used in the forest, i.e., before taking vote of a class how many different algorithms are to be run. max_depth of Random Forest Tree gives the maximum depth of

Tasks	size	n_estimator	max_depth
Task 1	140	-	-
Task 2	250	40	17
Task 3	280	-	-
Task 4	820	30	12
Task 5	150	10	8

Table 1: Tuned parameters for English.

Tasks	size	n_estimator	max_depth
Task 1	20	-	-
Task 2	50	90	19
Task 3	110	-	-
Task 4	90	140	17
Task 5	80	1	18

Table 2: Tuned parameters for Arabic.

Tasks	size	n_estimator	max_depth
Task 1	190	-	-
Task 2	160	40	18
Task 3	120	-	-
Task 4	320	140	16
Task 5	180	10	11

Table 3: Tuned parameters for Spanish.

the tree in algorithm. We did a grid search method to find out the optimum parameter values for each subtasks.

For emotion intensity regression task (Task 1) and sentiment intensity regression task (Task 3) Doc2Vec $size$ was varied from 10 to 1000 with an increment of 10 in each iteration. For emotion intensity ordinal classification task (Task 2), sentiment analysis ordinal classification task (Task 4) and emotion classification task (Task 5) Doc2Vec $size$ was varied from 10 to 1000 with an increment of 10 in each iteration, $n\_estimator$ of Random Forest Tree was varied from 10 to 150 with an increment of 10 in each iteration and max_depth of Random Forest Tree was varied from 2 to 20 with an increment of 1 in each iteration. Variables used to estimate the ideal parameters for regression tasks were mean square error (MSE) and variance of Linear Regression algorithm. We selected those parameters that gave the least MSE value ans large variance value. Variables used to estimate the ideal parameters for classification tasks was accuracy of the Random Forest Tree algorithm. Once the parameters were fixed we build the model for each subtask and used it to predict the values for test data. Development data was used for hyper-parameter tuning while training data was used for building Doc2Vec model.

The ideal parameters obtained after hyper-parameter tuning for each subtask for English is consolidated in Table 1, Arabic is consolidated in Table 2 and Spanish is consolidated in Table 3. The control parameter values obtained for the optimum parameters which in turn are used to build the model is consolidated in Table 4 for task 1 Table 5 for task 3 Table 6 for task 2 Table 7 for task 4 Table 8 for task 5

Variable	English	Arabic	Spanish
MSE	0.03	0.03	0.04
Variance	0.03	0.03	0.08

Table 4: Control variable value for optimum parameters for Task 1.

Variable	English	Arabic	Spanish
Accuracy	0.4883	0.4039	0.4047

Table 5: Control variable value for optimum parameters for Task 2.

Variable	English	Arabic	Spanish
MSE	0.03	0.04	0.05
Variance	0.06	0.06	0.02

Table 6: Control variable value for optimum parameters for Task 3.

Variable	English	Arabic	Spanish
Accuracy	0.28	0.27	0.28

Table 7: Control variable value for optimum parameters for Task 4.

Variable	English	Arabic	Spanish
Accuracy	0.9525	0.9550	0.9401

Table 8: Control variable value for optimum parameters for Task 5.

4 Results and Discussion

The output of test data obtained by our model was compared with golden label available with SemEval2018 and the following results were obtained. The metric used for evaluation is macro average F-Score and Pearson correlation coefficient. In macro average method precision and re-

call on different sets of system is averaged. The harmonic mean of precision and recall will give us the F-Score. Such an obtained value is called macro F-Score. In Pearson correlation coefficient the linear correlation between two variables *X1* and *X2* is calculated. For emotion intensity regression task, emotion intensity ordinal classification task, sentiment intensity regression task and sentiment analysis ordinal classification task Pearson correlation coefficient is used as metric while for emotion classification task macro average F-Score is used as metric.

For emotion intensity regression task on English tweets our model obtained an accuracy of 20.0% when compared with the golden label under Pearson correlation coefficient. When compared for individual emotions we got an accuracy of 21.6%, 21.0%, 11.2%, 26.2% for anger, fear, joy and sadness respectively. On Arabic tweets our model obtained an accuracy of 22.1% when compared with the golden label under Pearson correlation coefficient. When compared for individual emotions we got an accuracy of -0.3%, 17.9%, 31.5%, 39.3% for anger, fear, joy and sadness respectively. On Spanish tweets our model obtained an accuracy of 21.8% when compared with the golden label under Pearson correlation coefficient. When compared for individual emotions we got an accuracy of 24.1%, 21.4%, 14.2%, 27.3% for anger, fear, joy and sadness respectively.

For emotion intensity ordinal classification task on English tweets our model obtained an accuracy of 3.7% when compared with the golden label under Pearson correlation coefficient. When compared for individual emotions we got an accuracy of 2.6%, -0.2%, 6.7%, 5.5% for anger, fear, joy and sadness respectively. On Arabic tweets our model obtained an accuracy of 13.8% when compared with the golden label under Pearson correlation coefficient. When compared for individual emotions we got an accuracy of -6.2%, 5.0%, 28.7%, 27.5% for anger, fear, joy and sadness respectively. On Spanish tweets our model obtained an accuracy of 2.5% when compared with the golden label under Pearson correlation coefficient. When compared for individual emotions we got an accuracy of 2.0%, -5.2%, 6.3%, 6.8% for anger, fear, joy and sadness respectively.

For sentiment intensity regression task on English tweets our model obtained an accuracy of 28.1% when compared with the golden label under Pearson correlation coefficient. On Arabic tweets our model obtained an accuracy of 47.0% when compared with the golden label under Pearson correlation coefficient. On Spanish tweets our model obtained an accuracy of 19.3% when compared with the golden label under Pearson correlation coefficient.

For sentiment analysis ordinal classification task on English tweets our model obtained an accuracy of 12.5% when compared with the golden label under Pearson correlation coefficient. On Arabic tweets our model obtained an accuracy of 38.3% when compared with the golden label under Pearson correlation coefficient. On Spanish tweets our model obtained an accuracy of 12.7% when compared with the golden label under Pearson correlation coefficient.

For emotion classification task on English tweets our model obtained an accuracy of 14.8% when compared with the golden label under macro average F-Score. On Arabic tweets our model obtained an accuracy of 25.0% when compared with the golden label under macro average F-Score. On Spanish tweets our model obtained an accuracy of 6.0% when compared with the golden label under macro average F-Score.

5 Conclusion

The task was to analyze the 'Affects of Tweets' from tweets comprising of different emotions from three different languages. We used distributed representation (Doc2Vec) for creating feature vector which was passed as the input to machine learning algorithm such as Linear Regression for regression tasks and Random Forest Tree for classification tasks. The model was fixed after doing hyperparameter tuning and the results obtained using the model on test data was evaluated using golden label by SemEval2018. The results obtained with the model after comparing with the golden label using some evaluation metric have been discussed in the paper.

References

Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete. 2014. Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems*, 69:86–99.

Barathi Ganesh H. B., M. Anand Kumar, and K. P. Soman. 2016a. Amrita_CEN at SemEval-2016 Task 1: Semantic relation from word embeddings in higher

dimension. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 706–711.

Barathi Ganesh H. B., M. Anand Kumar, and K. P. Soman. 2016b. Statistical semantics in context space : Amrita_CEN@Author Profiling. In *CEUR Workshop Proceedings, 1609*, pages 881–889.

Barathi Ganesh H. B., M. Anand Kumar, and K. P. Soman. 2018. From vector space models to vector space models of semantics. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10478 LNCS*, pages 50–60. Springer.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

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 (LREC-2018)*, Miyazaki, Japan.

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*.