Dataset effects on Twitter messages semantic classification

A.v

1 Introduction

In this report we discuss about a classification problem for the platform *twitter* messages and our task is to find the best model and dataset in order to predict a new set of labels for unseen data with a higher accuracy. To do so we have a vocabulary file and three datasets, each divided to three parts concluding: train, development, and test data sets which are in the Comma Separated Values (CSV) format selected by the holdout method. Each tweet in datasets is represented by its Identification (ID) and tweet information as well as three classes as our sentiments: negative, positive, and neutral. The first dataset uses a Bag of Words (BOW) and include the number of times a word is seen in a tweet. The second data set is known as the term frequency–otherwise, inverse document frequency or TF-IDF. In both datasets the most frequent and the least frequent word in each tweet is deleted so that we are left with clean data that is without repeated words and usernames with frequent word counts or TF-IDF amount, represented by word ID and the amounts. The third dataset used is called GloVe, which is a vector of sum of words in a tweet while each word is a one hundred dimension vector represented as a word embedding system. The datasets are derived from the resources published from Vadicamo et al. (2017) and Go et al. (2009). The question is what data set and method will result in a better outcome. Evaluation will be calculated through the development set while training is achieved through the train dataset.

2 Literature review

Among many similar studies about previous works on sentiment classification tasks, Go et al. (2009) used the idea of mapping emotions of tweets into two classes, negative and positive, without considering neutral messages to prevent mutual intervention in messages. Go et al. (2009) also mentioned that the frequency of misspelling and the use of slang as well as the variety of topic domains in twitter posts are different than other classifications and as Jiang et al. (2011) found, it is more difficult to classify tweet sentiments on the contrary to other sentiment tasks, because the length of tweets are short and limited to certain characters. Jiang et al. (2011) also provided a new approach as target dependent sentiments which lead to describing an expression and also considered word stems. Another similar study on classifying emotional sentiments of Internet Movie Data Base (IMDB) reviews was established by Park et al. (2020) and denotes that word embedded systems are not a good practice for sentiment classification and showed that, according to their study, TF-IDF had the best accuracy while word embedding had the lowest. The Park et al. (2020) study, just like Jiang et al. (2011), did not provide any class for neutral reviews.

3 Method

Besides benchmark models concluded in this study, a dummy baseline model was examined to evaluate whether our classifiers can beat the naïve baseline model or not and whether we actually require a method for learning in the first place? Implementation of these machine learning models use the *Python Scikit-Learn libraries* (Pedregosa, 2011). For evaluation, confusion matrices are obtained and precision and recall values are used.

3.1 Baseline

In order to evaluate our models after training them we need to see whether a dummy assumption for class predictions can be beaten by our machine learning algorithm or not. Since the data labels' distribution of positive and negative sentiments in the training set are very close to each other (See Figure 1), to evaluate with the highest precision, our labels are randomly assigned and this strategy is implemented with a hundred iterations over a uniform random classifier.

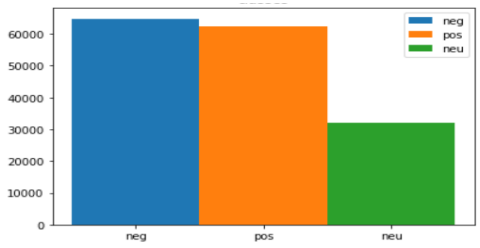

Figure 1- distribution of labels in training data sets

3.2 Naïve Bayes

Naïve Bayes (NB) is a simple classifier which works well on text categorization (Manning, 1999) and is based on probabilities of occurrences of features, where features are words in our case, based on which data set is selected. The basic assumption in NB is that features are conditionally independent from each other. Therefore, we use multinomial NB for our datasets.

3.3 Logistic Regression

Unlike 'Naïve Bayes', Logistic Regression (LR) makes no prior assumption about the relationships between features and will potentially perform better. LR is a type of maximum entropy model of Scikit-Learn and the default solver for LR implemented by Scikit-Learn is not usable for heavy datasets, and it is therefore better to use 'Newton-cg' (Curtis et al. 2019) solver to find the best parameters or weights.

3.4 Perceptron

Perceptron which is in fact the simplest shape of Multilayer Perceptron (MLP) has vectors of numeric data and output scalar values. Activation function is defined based on inputs for each data set on multi-class perceptions.

3.5 Multilayer perceptron

MLP is a fully connected network of perceptron, which is a type of neural network, and is useful for large datasets. By using this method it is possible to define activation function on normalized data.

4 Results

The results of the baseline and the three tables below for different data sets are based on the methods represented as confusion matrices. While for each Machine Learning (ML) method, rows are actual values and columns are the predictions.

According to these tables what we are interested in is true positive values in our confusion matrix which will allow us to increase the precision metric which will also reduce false positives or false alarms.

Baseline
33.3% accuracy

Table 1- Baseline dummy test accuracy for development datasets.

Bag of Words	Pos Neg Neu
	[5465 2083 368]

NB	[1916 5913 266] [434 292 3169]
LR	[5600 1855 461] [1989 5810 296] [363 196 3336]
Perceptron	[4819 2534 563] [2034 5683 378] [482 318 3095]
MLP	[5617 1848 451] [1972 5823 300] [380 187 3328]

Table 2- Confusion matrices for Bag of Words dataset.

TF-IDF	Pos Neg Neu
NB	[5456 2187 273] [1964 5940 191] [509 338 3048]
LR	[5586 1892 438] [1960 5854 281] [360 212 3323]
Perceptron	[4745 2674 497] [1970 5812 313] [439 428 3028]
MLP	[5550 1913 453] [1913 5897 285] [346 211 3338]

Table 3- Confusion matrices for TF-IDF dataset.

Glove	Pos Neg Neu
LR	[4858 2368 690] [2270 5495 330] [518 239 3138]
Perceptron	[3811 3334 771] [2108 5553 434] [943 306 2646]
MLP	[4858 2368 690] [2270 5495 330] [518 239 3138]

Table 4- Confusion matrices for Glove dataset.

5 Discussion / criteria analysis

Clearly, every trained model could beat our

baseline model. After the training process and evaluation of development datasets, in both BOW's and TFI-DF datasets, we can see that all our models are performing well and provide over 70 per cent correct predictions with all methods. The reason for that can be due to the feature selections in both BOW and TFI-DF datasets where all frequent and infrequent words have been deleted from consideration. Therefore, we have a fixed number of words or features, however, we are using only frequent words that are used in a sentence.

For instance, "*oh no! so sorry about your pets"*.

(oh:3083), (sorry:4054), (pets:3245)

"Oh", "Sorry", and "Pets" would be the only considered words, while the second value of tuples are the word ID's in our vocabulary.

The first issue that arises when looking at these datasets with different column sizes for each sentence or row is the issue of how we can manipulate the columns to fit our models. Hence, to overcome this matter the whole dataset must fit into a Sparse Matrix. By the definition of sparse matrix, unused words can be considered zero values. To do so, *Scikit-Learn library* provided us a vectorization function that accepts *Python* dictionaries and creates vectors in sparse spaces.

After preparation we can feed our models and train them for evaluation and comparison.

For the BOW dataset, the NB model worked fine, or smoothly, as expected. Regardless of data size we can see a 73 per cent accuracy on the development set. Logistic Regression and Perceptron performed the same, however, LR saw a faster performance time. The MLP classifier had the same performance with a slower run time, and it was fed by a Redefined Linear Unit (RELU) activation function with a single hidden layer of three nodes. After testing the MLP with different hidden layers and number of nodes the result didn't improve.

The TF-IDF dataset LR and Perceptron worked almost identically, however, both had the best prediction score among all the other models that were tested in this report.

It was the GloVe dataset that showed the worst performance in general. It was difficult for the GloVe data to achieve more than 70 per cent accuracy. While we know that GloVe data repre-sents 100 dimensions in a single vector for each word, a possible reasoning for this might be that the sum over all word vectors is calculated in-stead of averaging over each sentence. LR needed normalization since the vectors had negative numbers. Since we did not find any Gaussian normalization for GloVe vectors, we didn't consider training them in the first place.

On the other hand, for BOW dataset, MLP predicted a better result out of all predictions and that is because of the number of true positives which is higher in this solution.

Another thing we should consider about LR and specifically in Perceptron methods is that since we are calculating dot products and we have a great number of zeroes in our Sparse Matrix it can largely affect objective functions. While it appears less in the context of Naïve Bayes, due to its naïve assumption of sum over probabilities of independent features. Considering the tables 1 to 3 we can see that the model is evaluated as best when it has the highest true positives which means it predicted correctly and were less successful where false alarms occurred most.

In the end, this study showed similar results to Park et al. (2020), in that word embedding was not a good idea for these types of classification methods.

6 Conclusions

The aim of this report was to identify and drive the best ML classifier based on the dataset in order to predict tweet classified sentiments at an optimal level. Many Machine Learning algorithms and datasets were applied in order to achieve successful and optimal results or predictions including Naïve Bayes, Logistic Regression, Perceptron, and Multilayer Perceptron as a type of a neural network system. Overall, in case of performance and in order to get the best score prediction, MLP on BOW achieved the best precision value and lowest false errors along with the Logistic Regression on the TF-IDF data set. However, Perceptron and Naïve Bayes showed a close performance rate. The GloVe dataset didn't perform well in any of these assumptions, which can be due to the sizes of tweets, particularly those that are smaller in length, and non-conceptual tweets.

References

Curtis, F, E., Robinson, D, P., Royer, C., & Wright, S, J. (2019). Trust-Region Newton-CG with Strong Second-Order Com-

plexity Guarantees for Nonconvex Optimization. arXiv:1912.04365.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification Using Distant Supervision. CS224N Project Report. Stanford. 1(12).

Jiang, L., Yu, M., Zhou, M., & Liu, X. (2011). Target-dependent Twitter Sentiment Classification. *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference.* Portland, Oregon, USA. https://www.researchgate.net/publication/220874961_Target-dependent_Twitter_Sentiment_Classification

Manning, C, D., & Schutze, H. (1999). Foundations of statistical natural language processing. MIT Press. Cambridge, MA. https://nlp.stanford.edu/fsnlp/

Park, H., & Kim, K. (2020, August). Impact of Word Embedding Methods on Performance of Sentiment Analysis with Machine Learning Techniques. Journal of the Korea Society of Computer and Information. Vol. 25 (8) pp. 181-188. https://doi.org/10.9708/jksci.2020.25.08.181

Pedregosa, F., et al (2011). Scikit-learn: Machine Learning in Python. JMLR 12, pp. 2825-2830, 2011. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf?source=post_page-----------------------

Pennington, J., Socher, R., & Manning, D, C. (2014). GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/

Vadicamo, L., Carrara, F., Cimino, A., Cresci, S., DellOrletta, F., Falchi, F., & Tesconi, M. (2017). Crossmedia learning for image sentiment analysis in the wild. *In 2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 308-317