Offensive Tweet Classification - A Comparative Study

Geethu Krishnan Mohan
Department of Mathematics, University of Essex

Abstract

This text classification project aims to compare and contrast the effects of advanced preprocessing (Uysal and Gunal, 2014) techniques using TF-IDF vectorization on three machine learning algorithms: Naive Bayes, Support Vector Machine, and Multi-layer Perceptron Model. The project takes an OLID dataset to carry out the techniques. The preprocessed data is then transformed into a TF-IDF matrix, and the data is used to train and test the three algorithms. The performance of each algorithm is evaluated and compared based on the preprocessing techniques used, using metrics such as accuracy and F1 score. The project's findings will shed light on the impact of preprocessing (Uysal and Gunal, 2014) techniques on text classification performance and aid in the selection of the best algorithm and preprocessing technique for a given text classification task.

1 Materials

The following is the list of resources for the project.

- Notebook Code File
- Google Drive Folder

2 Model Selection (Task 1)

The OLID dataset (Zampieri et al., 2019) is a well-known dataset for text classification of offensive speech (Warner and Hirschberg, 2012) (Davidson et al., 2017), which includes five different sub-tasks for classification. Regarding the choice of models for this classification task, analysing the data is very important.

2.1 Summary of Selected Models

The given OLID (Zampieri et al., 2019) dataset contain 12313 number of rows with each row have an id, tweet and label. After exploring the dataset further with respect to the response variable, it is observed that number of offensive language reported is twice as that of those not reported. Also, the data is clean and do not have missing values hence, it is ready for the preprocessing (Uysal and Gunal, 2014).

There are different algorithms which works well for text classification and have promising performances. Support vector machine (SVM) (Tong and Koller, 2001), multinomial Naive Base model (Abbas et al., 2019), multi-layer perceptron models and linear regression are few examples. Since the dataset have a very good distribution of both positive and negative responses, it will be good enough to go ahead with support vector machine model, multinomial Naive Base (MNB) model and multi-layer perceptron models (MLP).

2.2 Critical Discussion and Justification of Model Selection

As an experiment, the author selected three models above mentioned based on the dataset provided by researching about various positive and negative aspects. SVM is highly beneficial for high dimensional features where we can represent those in words or n-grams. In this problem, classification of two distinct classes always help to create a perfect hyper plane to separate the features and hence it must posses high accuracy. SVM also tries to fit properly and avoid over fitting by adjusting and maximizing the hyperplane.

Multi-layer perceptron neural model is another best choice to do text classification because of its ability to learn complex and non-linearly related data from the dataset. Another positive side of MLP is that it can have multiple hidden layers, which could implement those non-linear relationships. Since the project aim for an academic purpose the easiness to train and test makes this MLP a good choice to use and compare the results.

As a third model, it was decided to choose Multinomial Naive Base (Abbas et al., 2019) model to compare and contrast the performance. with re-

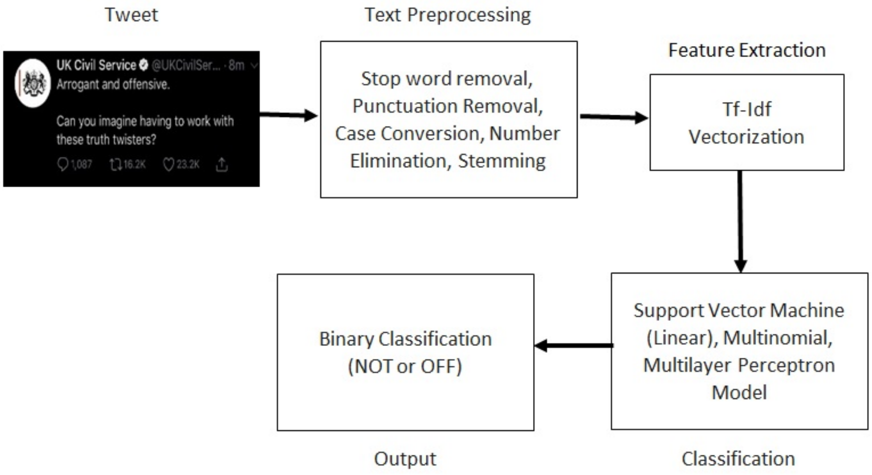
Tweet
UK Civil Service @UKCivilSer... · 8m
Arrogant and offensive.
Can you imagine having to work with these truth twisters?
1,087
16.2K
23.2K
Text Preprocessing
Stop word removal, Punctuation Removal, Case Conversion, Number Elimination, Stemming
Feature Extraction
Tf-Idf Vectorization
Support Vector Machine (Linear), Multinomial, Multilayer Perceptron Model
Classification
Binary Classification (NOT or OFF)
Output

Figure 1: The diagram explains a pipeline of process flow. The input dataset is cleaned and preprocessed before building the model. In the OLID (Zampieri et al., 2019) dataset, the input features are complete and do not have any missing values. Hence, it's ready for the preprocessing. The author performed various methods like stop word removal, punctuation removal, upper case to lower case conversion, Number elimination and stemming even before pass the data to the Tf-Idf vectorization. The vectorized data is pass to the classifier for each model building, such as SVM, MNB, MLP on various dataset.

spect to fast response time time without loosing its performance, MNB is a good pick to proceed with. This is also highly robust to irrelevant data in the input. Apart from these, this model is easy to implement and train with quick response.

3 Design and implementation of Classifiers (Task 2)

The source OLID(Zampieri et al., 2019) dataset with train data has a total of 12313 entries, out of which 33.31% are offensive and 66.69% are non-offensive. This dataset is likely used for training machine learning models to classify offensive and non-offensive content. The numbers are summarized in Table 1.

dataset	Total	% OFF	% NOT
Train %	12313	33.31	66.69
Valid	927	33.23	66.77
Test	860	27.90	72.10

Table 1: dataset Details

The validation dataset has 927 entries, with 33.23% being offensive and 66.77% being non-offensive. This dataset is likely used to fine-tune the model hyperparameters and to avoid overfitting.

The test dataset has 860 entries, with 27.90% being offensive and 72.10% being non-offensive. This dataset is used to evaluate the performance of the trained model on unseen data. The lower percentage of offensive content in the test set compared to the training set suggests that the model needs to generalize well to accurately classify offensive content in different scenarios.

3.1 Selection of Hyperparameter

Implementation of selected models is little challenging especially when it comes with hyper-parameter selection. The OLID (Zampieri et al., 2019) dataset is chosen to do the classification and while implementing SVM, hyper-parameter selections depends on various factors. The linear kernel will work well because the dataset has a clearly and linearly separable output variable called 'label'. The data can be plotted separately without much overlap, and over-fitting is reduced as well. Furthermore, linear kernels are simpler to construct and explain than other kernels such as radial basis functions, which define the boundaries in a more complex manner.

Implementing a neural network (Albawi et al., 2017) model is always recommended for better results because it learns the context and hidden layers work well for this job. It could also predict far better in unseen data(Ando and Zhang, 2005). The neurons, number of hidden layers, regularisation, activation function, learning rate, and batch size are the hyper-parameters for an MLP. The choice of those is always determined by the task at hand and the nature of the data. The number of hidden layers and neurons will be important factors in learning the flow of a text classification. The author chose 256 neurons with one input dense layer and one out layer with softmax activation function to ensure that the output is converted to 1 to ease the probability calculation. Three hidden layers with 256,128 and 64 neurons with relu activation function are being implemented. Dropout is enabled to add regularisation and ensure that the process does not overfit.

In the third model, MNB, the author opted some default hyperparameters like alpha 1.0, which is the maximum for making predictions using Laplace smoothing. The next parameter fit_prior=True which is good for calculating class of unseen data from the training data itself. Then class_prior is set to None, which indicates that the prior probabilities are calculated from training dataset. This model is also highly recommended with some promising performance.

3.2 Performance Evaluation on Selected Models

According to the results, the SVM model has a high F1 score of 0.7004 and an accuracy of 79.53% on the training dataset, indicating that the model can identify a good proportion of both true positives and true negatives. This is reflected in both classes' precision and recall scores, which are likely to be high.

The Multilayer Perceptron Model(Noriega, 2005), on the other hand, has a lower F1 score of 0.7374 on the test set but a relatively high accuracy of 80.93%. This implies that the model can correctly classify a large proportion of instances, but it may struggle to capture the nuances of the data as well as the SVM model. It is important to note that the precision and recall values may differ depending on the test set's specific class distribution.

The Multinomial Naive Bayes model appears to perform well on the training set, with an F1 score of 0.7667 and accuracy of 82.60%, but on the test set, its F1 score drops to 0.5737 and accuracy drops to 76.16%. Due to overfitting, the precision and recall scores for this model may be lower as well.

In short, while accuracy and F1 score are important metrics, precision and recall should also be considered when evaluating the performance of classification models. The SVM model and Multilayer Perceptron Model (Noriega, 2005) appear to be the best-performing models overall, with higher accuracy and F1 score the Multilayer Perceptron Model (Gardner and Dorling, 1998) is chosen to be the best model. The precise precision and recall values for each model depend on the specific class distribution in the dataset and can provide additional insights into the performance of the models.

Model	F1 Score
MLP	0.7374
Linear SVM	0.7004
MNB	0.5737

Table 2: Model Performance

In this section, you should add

4 Data Size Effect (Task 3)

The dataset contains a total of 12,313 entries, with 33.31% of the content being offensive (Waseem and Hovy, 2016) and 66.69% of the content being non-offensive. The dataset has been split into four parts for training, with the percentage of offensive and non-offensive content remaining fairly consistent across all four parts.

The distribution of offensive and non-offensive classes are in the ratio of 1:2. In the first part, which contains 25% of the dataset, there are 3,079 entries, with 33.91% of the content being offensive and 66.09% of the content being non-offensive. In the second part, which contains 50% of the dataset, there are 6,157 entries, with 32.73% of the content being offensive and 67.27% of the content being non-offensive. In the third part, which contains 75% of the dataset, there are 9,235 entries, with 33.11% of the content being offensive and 66.89% of the content being non-offensive. Finally, in the fourth part, which contains 100% of the dataset, there are 12,313 entries, with 33.31% of the content being offensive and 66.69% of the content being non-offensive. A tabulated version is displayed in Table 4. The distribution is not much imbalanced but still follow a regular pattern, which make the author to select the models more easier.

It is important to note that the definition of offensive content may vary depending on the context and the intended use of the dataset. Therefore, it is crucial to carefully examine the labeling process and the criteria used for identifying offensive content (Davidson et al., 2017) before using this dataset for any specific task.

Example %	GT	SVM(100%)	MLP(100%)
watching boomer getting news still parole always makes smile wentworth finaleuser treasure url	NOT	NOT	NOT
fuck time	OFF	OFF	OFF
user get feeling kissing user behind humiliate later 3	OFF	NOT	NOT
kznlt enjouji really prototype character brand nothing things love rolled one horrible package fed directly baby mes	NOT	OFF	OFF
subconscious	NOT	OFF	OFF
hiac damn matt hardy randy orton put one hell cell match woooo hope okay	NOT	OFF	OFF

Table 3: Comparing two Model's using 100% data: Sample Examples and model output using Model 1 & 2. GT (Ground Truth) is provided in the test.csv file.

dataset	Total	% OFF	% NOT
Train 25 %	3079	33.91	66.09
Train 50	6157	32.73	67.27
Train 75	9235	33.11	66.89
Train 100 %	12313	33.31	66.69

Table 4: dataset Details

4.1 Performance Evaluation on Different Size Data

The two models being compared are Support Vector Machines (SVM) and Multilayer Perceptron (MLP) (Gardner and Dorling, 1998), which are both popular machine learning algorithms used for classification tasks. The evaluation metrics used to compare the performance of the models are F1 score and accuracy. F1 score is a measure of the harmonic mean between precision and recall, which is a useful metric in cases where the dataset is imbalanced, like in this case, where the offensive content is much less frequent than non-offensive content (Davidson et al., 2017). Accuracy measures the proportion of correctly classified instances over the total number of instances.

Looking at the F1 scores and accuracy values for the different sizes of the dataset, it can be observed that the MLP model outperforms the SVM model in all cases. For example, when using the full dataset, the MLP model achieves an F1 score of 0.7374 with an accuracy of 80.93%, while the SVM model achieves an F1 score of 0.7004 with an accuracy of 79.53%. Similarly, for the 25% and 50% datasets, the MLP model has higher F1 scores and accuracy values than the SVM model.

Therefore, based on the evaluation metrics, the MLP model seems to be the better model for this classification task. However, it is important to note that other factors, such as model complexity, training time, and interpretability, may also need to be considered when choosing the best model for a specific use case. A pictorial representation of performance and f1 score for both test and validation data is plotted in figure 2 and 3.

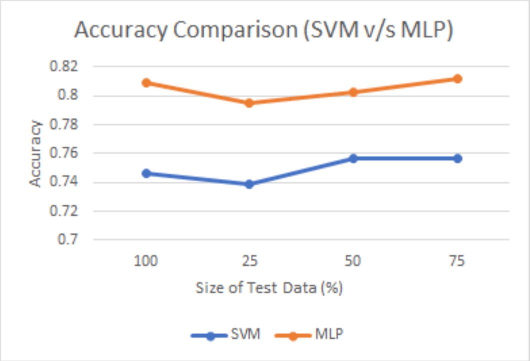

Figure 2: Comparision of accuracy among three models based on different data sizes.

You need to use Table 5 6 and 3 to compare the model's output and provide exciting insights. After observing the examples it's analyzed that model 1 with SVM could not predict the ground truth well for lesser data set but gradually picked up as the training data size increased. Hence, size of the data for training effects well in the processes. In model 2 as MLP, the same issues arises as in model 1. For less data the performance is very low and

Example %	GT	M1(25%)	M1(50%)	M1(75%)	M1(100%)
constoo place fed conservatives	NOT	OFF	NOT	NOT	NOT
dallaspolicewoman kills neighbor apartment went wrong seeofficer shot man apartment involved shooting suspect she devastatedsaid dallas police officer adds so sorry family url	NOT	OFF	NOT	NOT	NOT
user user user user user user user user user im shocked learn human traffickers guns probably illegal too ca needs gun control dont worry actual crime pic black guns worse	OFF	OFF	OFF	OFF	NOT
conservatives user clown url	OFF	OFF	OFF	OFF	OFF
user zuckerberg lies	NOT	OFF	OFF	OFF	NOT

Table 5: Comparing Model Size: Sample Examples and model output using Model 1 with different Data Size

Example %	GT	M2(25%)	M2(50%)	M2(75%)	M2(100%)
constoo place fed conservatives	NOT	OFF	NOT	NOT	NOT
gets bitches fortnitebattleroyale xboxshare user please ban cheating scum literally	OFF	NOT	OFF	OFF	OFF
invisible url	OFF	NOT	NOT	OFF	OFF
jeff sessions listen aclu antifa black lives matter want chicago shootings url	OFF	NOT	NOT	NOT	NOT
always bond dani minogue target pants	OFF	NOT	NOT	NOT	OFF

Table 6: Comparing Model Size: Sample Examples and model output using Model 2 with different Data Size

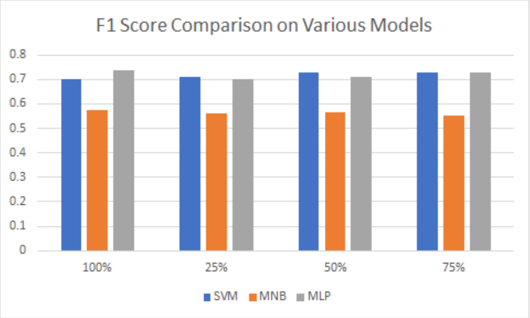

Figure 3: Comparision of validation F1 score on models based on different data sizes.

as the model trains with more data, the ground truth is keeping up. Also, for lengthy tweets, only higher training could get better accuracy in almost all cases. It is observed that dataset with more than 50% only could pick up good accuracy and performance.

5 Summary (Task 4)

The author implemented text classification of offensive languages on tweet using OLID (Zampieri et al., 2019) dataset. Out of three implemented models, SVM and MLP posses better accuracy and F1 score.

5.1 Discussion of work carried out

The data set received cleaned and preprocessed initially with stop word removal, stemming, punctuation removal etc. The model has been built with support vector machine classifier, multilayer perceptron model and multinomial naive base classifier. These models were trained, validated and tested on varied data sets of size 25%, 50%, 75% and 100%. In general accuracy of the models improved on size of the training data size. SVM and MLP outperformed the state of art of those classifiers in terms of accuracy and F1 score and selected as the best model for this classification problem. For 25% data, SVM outperformed all other classifier's because, SVM make the boundary larger between the classes and there fore less overfitting with more accurate result. The second model, MLP is a neural network implementaion which need take in account of many parameters and need more data to avoid overitting even with regularization.

5.2 Lessons Learned

Study of problem statement along with data set will give us a broad idea of how to select the classifier and solve the problem. Tn this problem, model selection was purely based on dataset nature and problem statement. Apart from this, adding additional preprocessing step like lemmatization, emoji removal etc will not showed considerable improvement in performance. Selection of vectorization is also played an important role by creating best matching vector for the classifier. Even if the dataset follows a 1:2 ratio for offensive (Waseem and Hovy, 2016) and non offensive classes, still the imbalance affected classification.

6 Conlusion

Finally, the SVM and MLP classifiers were used to differentiate between offensive and non-offensive text data in the OLID(Zampieri et al., 2019) dataset. F1-score and accuracy were used to assess the performance of both classifiers. The results demonstrated that the MLP classifier outperformed the SVM classifier in terms of F1-score and accuracy for both subtasks, which involved classifying tweets as offensive or not offensive. Measures to tackle class imbalance could not address in this solution.

References

Muhammad Abbas, K Ali Memon, A Aleem Jamali, Saleemullah Memon, and Anees Ahmed. 2019. Multinomial naive bayes classification model for sentiment analysis. *IJCSNS Int. J. Comput. Sci. Netw. Secur*, 19(3):62.

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Matt W Gardner and SR Dorling. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636.

Leonardo Noriega. 2005. Multilayer perceptron tutorial. *School of Computing. Staffordshire University*, 4:5.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Alper Kursat Uysal and Serkan Gunal. 2014. The impact of preprocessing on text classification. *Information processing & management*, 50(1):104–112.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.