HATE SPEECH DETECTOR

Group 13

Nurefşan Müsevitoğlu - 21503062
Yasin Balcancı - 21501109
Semih Teker - 21300964
Nejdet Balkır Göka - 21400551
Muammer Tan - 21400967

1. Introduction & Background Information

Our aim on this project is to determine the intention of writing a tweet whether it contains a hate speech or not. Tweets are written in natural language and also they include shortcuts. Therefore, the models should understand the language used in each post and draw a conclusion from them. Additionally, since words could have different meanings depending on the context they are used, the model should understand the meaning of the posts not just by looking at each word but by analyzing it in the context.

We use a dataset used in a paper which includes 100,000 English tweets, 4,965 labeled as hateful, 53,851 labeled as normal, 27,154 labeled as abusive and 14,030 labeled as spam [1]. To use this database, we mailed the author and got the permission of the author.

As we stated in the proposal, we are going to try 6 different algorithms:

- Naive Bayes
- Logistic Regression
- Support Vector Machine (SVM)
- Random Forest
- Hybrid Class Semantic Classifier (HCSC)
- k-NN

to train and test the data and the results are going to be compared to see which algorithm works best for this type of situation.

2. What we have done so far

So far, we have preprocessed the tweets; lower the words, omit the stopwords, omit emoji characters, omit the usernames, omit the web addresses, omit the punctuations and finally omit the numericals.

Furthermore, we have tried the Naive Bayes Algorithm and Logistic Regression Algorithm for modelling our data. Since we exceed the RAM provided by colab when we use all the dataset in our algorithms, we randomly chose small portions of the dataset. The sizes used for models are given in the results.

2.1. Naive Bayes

First, we have tried the Naive Bayes Algorithm for modelling our data. We have tried different data sizes and different test sizes as given below in the results.

2.1.1. Results

Data size: 20,000 Test size: 30% Accuracy: 70.4%		Precision	Recall	F1 Score
	Macro	64%	66%	64%
	Micro	70.4%	70.4%	70.4%
	Weighted	74%	70%	71%

Data size: 20,000 Test size: 25% Accuracy: 70.1%		Precision	Recall	F1 Score
	Macro	63%	66%	64%
	Micro	70.1%	70.1%	70.1%
	Weighted	74%	70%	71%

Data size: 20,000 Test size: 20% Accuracy: 69.6%		Precision	Recall	F1 Score
	Macro	63%	66%	64%
	Micro	69.6%	69.6%	69.6%
	Weighted	74%	69%	71%

Data size: 20,000 Test size: 10% Accuracy: 70.9%		Precision	Recall	F1 Score
	Macro	65%	68%	65%
	Micro	70%	70%	70%
	Weighted	74%	70%	72%

Data size: 25,000 Test size: 30% Accuracy: 72.8%		Precision	Recall	F1 Score
	Macro	62%	66%	63%
	Micro	72%	72%	72%
	Weighted	77%	72%	74%

2.2. Logistic Regression

Second, we have tried the Logistic Regression Algorithm for modelling our data. We were able to try it only for more limited data sizes and training size since the runtime took so long and it was not very likely for us to get proper results in given time for the project. We used "sag" solver for Logistic Regression because it seemed to be fitting to our problem because of our data size and binomial classification. Results are given below.

2.2.1. Results

Data size: 20,000 Test size: 30% Accuracy: 88.3%		Precision	Recall	F1 Score
	Macro	88%	80%	83%
	Micro	88%	88%	88%
	Weighted	88%	88%	87%

Data size: 25,000 Test size: 30% Accuracy: 90.9%		Precision	Recall	F1 Score
	Macro	89%	81%	84%
	Micro	91%	91%	91%
	Weighted	91%	91%	90%

3. What remains to be done

So far, we have just implemented the the Naive Bayes and Logistic Regression and the rest of the techniques mentioned on proposal remains for the future work. We are going to

- prepare the dataset to be ready to use for each technique if necessary.
- implement the algorithms using related libraries
- compare the results of each algorithm
- and interpret the each algorithm's pros and cons

4. Division of work among the teammates

Nurefşan Müsevitoğlu
- Prepared the report

Yasin Balcancı
- Worked on implementing Gaussian Naive Bayes and Logistic Regression

Semih Teker
- Worked on implementing Gaussian Naive Bayes

Balkır Göka
- Worked on implementing Gaussian Naive Bayes

Muammer Tan
- Preprocessed the dataset and prepared the results

References

[1] Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. [Online] Available: https://arxiv.org/pdf/1802.00393.pdf Accessed: March 15, 2019.