Artificial Intelligence

ASSIGNMENT 2 – FINAL DELIVERY

Group 07

Carolina Rosemback Guilhermino, up201800171
José Eduardo Henriques, up201806372
Miguel Carreira Neves, up201608657

Project Specification

Natural Language Processing (NLP Problems): Detecting Offense

In an NLP Problem, the textual data should be processed and transformed into appropriate datasets. Then, an initial exploratory data analysis should be carried out, along with different pre-processing and feature engineering techniques. The employed machine learning algorithms should be tested and compared (performance during learning, confusion matrix, precision, recall, accuracy, F1 measure) and the time spent to train/test the models.

This project aims to identify how offensive a given text is, by attributing a score from (higher score => more offensive).

The test file contains 9000 labels and ratings from a balanced set of age groups from 18-70. The annotators also represented a variety of genders, political stances and income levels.

Related Work

- Splitting into train, dev and test sets: https://cs230.stanford.edu/blog/split/
- N-Grams: https://www.lexalytics.com/lexablog/context-analysis-nlp
- SMOTE: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

Dealing with unbalanced datasets: https://machinelearningmastery.com/data-sampling-methods-for-imbalanced-classification/

- “Method for Detecting and Rating Humor Based on Multi-Task Adversarial Training”: https://arxiv.org/pdf/2104.10336v1.pdf
- Learning curve: https://scikit-learn.org/stable/auto_examples/neural_networks/plot_mlp_training_curves.html
- Moodle contents from the UC

Tools and Algorithms Implemented

Machine learning algorithms:

- SVM
- Neural Networks (MLP)
- Logistic Regression
- Random Forest
- Decision Tree

Data Preprocessing

- Porter Stemmer
- Bag of Words
- N-grams (bi-grams and tri-grams)
- TF-IDF

Techniques Used

- Divided Data into train, test and dev sets in order to avoid overfitting to the test set.

- Removed Stop words from phrases.

- Due to our dataset being highly imbalanced (from 9000 rows, ~6400 were scored 0):
- → Rounded our labels (they were floats) and transformed labels 2-5 in 2, in order to simplify classification. The amount of examples of the higher scores was below 200.
- → Used SMOTE along with Random Under Sampler to balance our data to 3.000 examples of each

F1 average score and fit elapsed time

Results for each (pre-processor, classifier) pair prediction applied to the dev_set.

dev_set	LR		SVM		DT		RF		MLP	
	F1 avg (%)	Time (s)	F1 avg (%)	Time (s)	F1 avg (%)	Time (s)	F1 avg (%)	Time (s)	F1 avg (%)	Time (s)
BW	46	0.009	48.2	7.6	72.6	0.077	41.1	0.18	63.7	2.871
BW Bi	63.7	0.013	60.8	16.36	58.6	0.02	61.2	0.29	47.9	1.56
BW Tri	43.3	0.01	47.2	15.53	44.1	0.02	43.9	0.42	48.7	4.166
TF-IDF	80.6	0.006	83.7	134	74.5	0.08	**86**	0.367	**40**	26.97

MLP Learning curve

```
sgd with relu
- accuracy: 0.693968
- loss: 1.027615
training: adam with relu
- accuracy: 1.000000
- loss: 0.003189
training: sgd with tanh
- accuracy: 0.696667
- loss: 0.987114
training: adam with tanh
- accuracy: 1.000000
- loss: 0.003197
training: sgd with identity
- accuracy: 0.696667
- loss: 0.986929
training: adam with identity
- accuracy: 1.000000
- loss: 0.003184
training: sgd with logistic
- accuracy: 0.332698
- loss: 1.098803
training: adam with logistic
- accuracy: 1.000000
- loss: 0.014587
```