

Natural Language Processing

Assignment 03

Readme

by Yashraj and Rounak

Files in Folder:-

- 3a- Contains CSV, textfiles for data preprocessing, Twitter_sentiment.py and Twitter_sentiment.ipynb.
- 3b- Contains CSV, textfiles for data preprocessing, 3joy, and 3banger in both ipynb and py format.
- Outputs: Contain outputs in both txt and pdf format

How to run the code:-

- Open jupyter notebook, Select the file and run all the blocks.
- Can tune Smoothing, testing data, training data, and features.
- Just comment out or remove comments to tune the features in scoremat function in Twitter_sentiment and preprocess function in 3b's code.

For 3a:

Preprocessing: Make pandas data frame to access the test and train file. BOW is trained on all training sets but for making the model, we only took 1500 sentences due to computational limitations.

Methodology: Follows the given paper for applying all features and 1 extra feature. We can tune the features in scoremat function.


Naive Bayes:

Naive Bayes is implemented from scratch and results are given in outputs. We got pretty good accuracy only by naive Bayes which is 76 percent.

Results for Naive Bayes.....

accuracy = 76.220625

Aver. Precision = 0.7813764965217469



Aver. recall = 0.7626371341855975

Aver. f1s = 0.7582839123179834

For the second part, we predict our model by using SVC, MLP, Decision tree classifiers. Final results is dumped in output folder.

Stats For SVC Model

accuracy = 77.63333333333334

Aver. Precision = 0.7786009855741485

Aver. recall = 0.7764811333831565

Aver. f1s = 0.7759383237662073

Stats For Decision Tree Model

accuracy = 76.53625

Aver. Precision = 0.7656134392659152

Aver. recall = 0.7653090075005572

Aver. f1s = 0.7652807133685693

Stats For MLP Model

accuracy = 78.03520833333333

Aver. Precision = 0.780796450775221

Aver. recall = 0.7804164226793763

Aver. f1s = 0.7802899954593102

Findings: Data doesn't follow any grammatical rules, which makes Punctuation, POS tagging, negation(for the whole phase) features less irrelevant. SVC model's pickle file is very large and the model takes only 25 seconds to train so we skip that model.



For 3b.


Methodology: Follows the given paper for applying all features and 5 extra BONUS features. We can tune the features in the preprocess function. .

In this program we predict stats using ngrams, Using features, ngrams, and 25 features separately and all three models are predicted and applied on SVM, MLP, and decision tree. So there is a total of 9 models. 3 models for Ngrams, 3 models for Features and 3 models for ngrams + features. The program only takes 30 seconds to predict fit all models without using any pickle form which shows that model is very very fast.

FEATURES

- col1 Elongated words
- col2 number of hashtags (FOR BONUS)
- col3 number of capitals letter
- col4 number of tagged persons in a sentence (FOR BONUS)
- col5 for negations
- col6 count words having particular emotion(anger)
- col7 aggregate hashtag emotion value
- col8 Aggregate emotion score (Hashtags)
- col9 Emoticons score:
- col 10,11 Aggregate polarity scores:
- col 12 13 Aggregate polarity scores (Hashtags)
- col 14 15 16 17 Lexicon based Features:
- col 18 19 Punctuation feature (BONUS)
- col 20 negation feature (take the whole phrase) (BONUS)
- col 21 22 23 24 Vader
- col25 Only hashtags predicting nature for sentence (only for this training data) (FOR BONUS)

++ NGRAM (1,2) Features



Results: Output is given in the output folder. We got a good correlation in Ngram + Decision tree model. The model tells about MAE, MSE, RMSE, R squared, Pearson, and Spearman correlation. Since the whole program takes only 40 seconds we didn't use pickle.

Since the training data is very small. Model faces problem to predict more accurate answers.

Contributions: Both of us researched both topics.

Yashraj: Mainly codes for 3(b) and implements about 20 and 15 features in 3(a) and 3(b)

Rounak: Mainly codes for 3(a) and implements remaining features.