Natural Language Processing (NLP)

FINAL PROJECT

Spyridon Roumpis

181004877

PROJECT: GENDER AND CHARACTER IDENTIFICATION IN EASTENDERS

1 Introduction

For this project we were provided with a training and a test set of data for the lines of the characters of the scripts of the television soap opera EastEnders scripts (2008). Both of the files have three columns – line, character, gender.

The objective of this project is to train a classification function which can go from the lines and predict the gender of the character (not using the character label) and the speaker of the lines (not using the gender label).

To evaluate our classifiers, we report the below metrics: Accuracy, Precision, Recall and F-Score on the test data.

2 Techniques Implemented

Text Classification is an automated process of classification of text into predefined categories, in our case classification of lines into gender or character.

The first step was to define the libraries we used and to add the Corpus. The data set was easily added as a pandas Data Frame.

Then we pre-processed our data, we transformed our raw data into a more understandable format for our models. Real-world data is often incomplete and/or lacking in certain behaviors or trends, and is likely to contain many errors. This will help in getting better results through the classification algorithms.

Below, I have explained the techniques that are performed in data pre-processing:

- **Remove Blank rows in Data, if any**: We do not want to deal with null values or having extra space if is not needed.
- **Change all the text to lower case:** It is necessary to convert the text to lower case as it is case sensitive
- **Word Tokenization**: This is a process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens.
- **Stopword Removal**: Stop words are a set of commonly used words in a language.The intuition behind using stop words is that, by removing low information words from text, we can focus on the important words instead.
- **Remove punctuation**: Removing punctuation from the text like ".?!" and also the symbols like "@#$"
- **Word Lemmatization**: The goal is to remove inflections and map a word to its root formAs the next step we tried to use as many as we can methods to provide different features to our classifiers.

Feature Selection

Firstly, we created a bag of words. To create the bag of words model, we had to create an array where the columns correspond to the most frequent words (500 in our case) in our dictionary and where rows correspond to the sentences/lines. The final step is to convert the sentences in our corpus into a vector representation, for each word in the most_freq dictionary if the word exists in the sentence add a 1 will for the word, else 0 will be added.

Secondly, after classifying with the above method we add an extra feature, which is the length of each sentences, at the end of the vector/matrix of our bag of words.

Furthmore, a TF-IDF approach was implemented. The bag of words model is that assigns equal value to the words (0 or 1), without taking in mind their importance so we tried this approach for improving the accuracy. The idea behind this is that the words that are more common in one line and less common in other lines should be given high weights. The TF-IDF value increases proportionally to how many times a word occurs in the line and is offset by the number of lines in the corpus that contain the word.

A different method is to create POS-tagging simply implies labelling words with their appropriate Part-Of-Speech (Noun, Verb, Adjective, Adverb, Pronoun, …)

After the different features' selection, we were able to classify our model. First, we spitted the training into 70% to train (training data) the classifiers and 30% (heldout data) to test it. After we achieved a good accuracy, we test the classifiers on the test set.

We wanted to have a variety of classifiers to evaluate our model, so for the purpose of this project we used the SVM, the Naive Bayes and the logistic regression.

4 Results

- **Gender**
  - Bag of words without the length of each sentence

```
SVM Accuracy Score ->  59.375
SVM Recall Score ->  59.375
SVM Precision Score ->  59.5591836734694
SVM F1 Score ->  59.34360893082273

Naive Bayes Accuracy Score ->  58.66071428571429
Naive Bayes Recall Score ->  58.66071428571429
Naive Bayes Precision Score ->  58.636134453781516
Naive Bayes F1 Score ->  58.64613481271659

Logistic Regression Accuracy Score ->  60.08928571428571
Logistic Regression Recall Score ->  60.08928571428571
Logistic Regression Precision Score ->  60.44285714285714
Logistic Regression F1 Score ->  60.07525405568298
```

  - Bag of words with the length of each sentence

```
SVM Accuracy Score ->  59.55357142857143
SVM Recall Score ->  59.55357142857143
SVM Precision Score ->  59.57833133253302
SVM F1 Score ->  59.51415892458254

Naive Bayes Accuracy Score ->  57.14285714285714
Naive Bayes Recall Score ->  57.14285714285714
Naive Bayes Precision Score ->  57.19151660664266
Naive Bayes F1 Score ->  57.16428028983505

Logistic Regression Accuracy Score ->  58.66071428571429
Logistic Regression Recall Score ->  58.66071428571429
Logistic Regression Precision Score ->  58.7764105642257
Logistic Regression F1 Score ->  58.623620950443936
```

- TF-IDF for gender

```
SVM Accuracy Score ->  58.48214285714286
SVM Recall Score ->  62.3574144486692
SVM Precision Score ->  55.12605042016807
SVM F1 Score ->  58.51917930419269

Naive Bayes Accuracy Score ->  58.30357142857143
Naive Bayes Recall Score ->  62.549019607843135
Naive Bayes Precision Score ->  53.61344537815126
Naive Bayes F1 Score ->  57.73755656108598

Logistic Regression Accuracy Score ->  57.67857142857142
Logistic Regression Recall Score ->  61.523809523809526
Logistic Regression Precision Score ->  54.285714285714285
Logistic Regression F1 Score ->  57.67857142857142
```

Overall, for classifying the gender the best accuracy is given when using the bag of words without the sentence length on logistic regression.

- **Character**
  - Bag of words without the length of each sentence

```
SVM Accuracy Score ->  17.67857142857143
SVM Recall Score ->  17.67857142857143
SVM Precision Score ->  28.117721984098743
SVM F1 Score ->  19.892292156924217

Naive Bayes Accuracy Score ->  21.071428571428573
Naive Bayes Recall Score ->  21.071428571428573
Naive Bayes Precision Score ->  36.65515740266564
Naive Bayes F1 Score ->  23.94890884521361

Logistic Regression Accuracy Score ->  21.160714285714285
Logistic Regression Recall Score ->  21.160714285714285
Logistic Regression Precision Score ->  33.711433099208236
Logistic Regression F1 Score ->  23.62150514635413
```

  - Bag of words with the length of each sentence

```
SVM Accuracy Score ->  15.982142857142856
SVM Recall Score ->  15.982142857142856
SVM Precision Score ->  23.680873002993398
SVM F1 Score ->  17.371303987706668

Naive Bayes Accuracy Score ->  20.80357142857143
Naive Bayes Recall Score ->  20.80357142857143
Naive Bayes Precision Score ->  36.34763094152309
Naive Bayes F1 Score ->  23.534547207991427

Logistic Regression Accuracy Score ->  20.0
Logistic Regression Recall Score ->  20.0
Logistic Regression Precision Score ->  30.186695932107536
Logistic Regression F1 Score ->  22.01468297737711
```

  - TF-IDF for character

```
SVM Accuracy Score ->  18.035714285714285
SVM Recall Score ->  18.035714285714285
SVM Precision Score ->  37.15930864961878
SVM F1 Score ->  20.779109784255777

Naive Bayes Accuracy Score ->  18.303571428571427
Naive Bayes Recall Score ->  18.303571428571427
Naive Bayes Precision Score ->  45.26814602161588
Naive Bayes F1 Score ->  21.538923333957705

Logistic Regression Accuracy Score ->  18.392857142857146
Logistic Regression Recall Score ->  18.392857142857146
Logistic Regression Precision Score ->  37.38115906030029
Logistic Regression F1 Score ->  21.577652226780206
```

Overall, for classifying the character the best accuracy is given when using the bag of words without the sentence length. It seems like logistic regression gives the best accuracy although when we did the cross validation it was overfitting.