Neeraj Vashistha
190573735

Problem Statement

The aim of this assignment is to classify gender and character based on dialogue spoken. The dataset consists of 10113 instances of dialogues spoken by 18 characters. Our aim is to predict whether the dialogue is spoken by male or female and by which character.

Objective

The purpose of this assignment is to build a machine learning model which can learn from the given data and provide inference on gender and character classification. The given data is in textual form, in order to achieve the objective, we need to transform this textual raw data to some form of information so that the machine learning model can inference knowledge out of it.

Procedures Implemented

To understand data better and to build a good classifier below steps have been implemented.

1. Data Cleaning and Normalization
2. Exploratory Data Analysis
3. Feature Engineering
4. Model Selection
5. Evaluation and Optimisation

Data Cleaning and Normalization

The dataset comprises of 10113 training row of dialogue spoken by 18 characters of television soap opera EastEnders (2008). The dataset is present in CSV format with three columns text, gender, and character. The testing dataset has 1124 rows with 3 columns.

Dialogues have an average length of 7 with median centered around 5. The sentences are incomplete with a varying degree of abstractness.

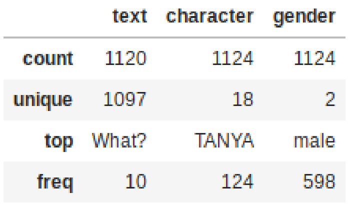
Due to lack of sufficient data, a mild pre-processing approach is applied. In pre-processing stage we create two columns 'text_norm' and 'token_text_norm', in the first column a simple token normalization is done with tokenization such that sentences are transformed as seen below. While in second column tokens are filtered through NLTk's stop word and then lemmatized and returned.

It's a great day. -> It 's a great day .

Exploratory Data Analysis

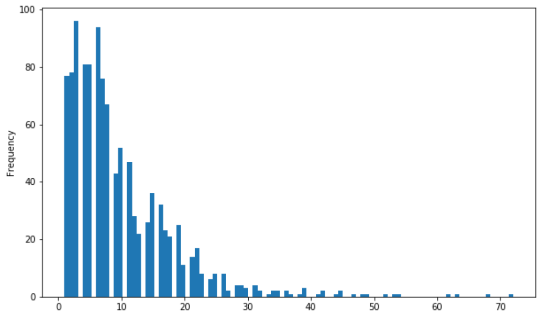
Here we understand how the data is presented to us, to better evaluate and grasp the domain knowledge of the data.

The below table shows the brief description of the training data, number of characters, top characters, total male counts (as these are highest – 598)



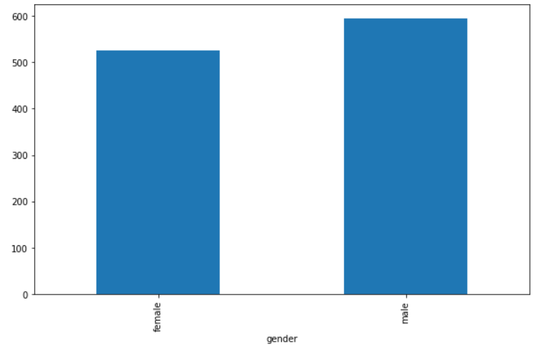
	text	character	gender
count	1120	1124	1124
unique	1097	18	2
top	What?	TANYA	male
freq	10	124	598

Distribution of dialogue word count



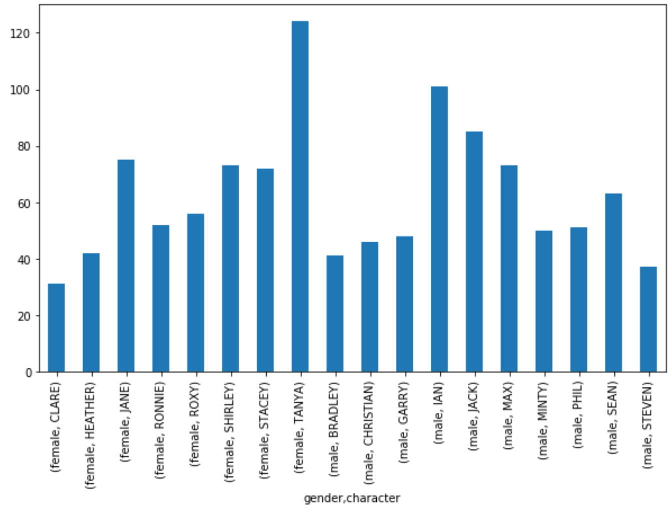
We see that most of the sentences are have length between 0 to 10

Distribution of Gender



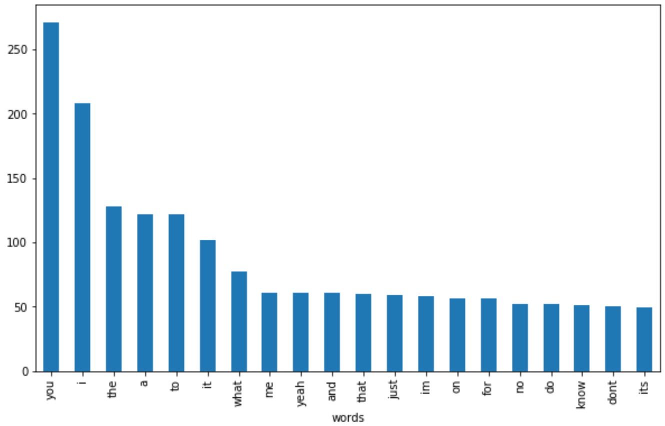
We see that the amount of dialogue for each of gender is almost equally divided.

Distribution of Character



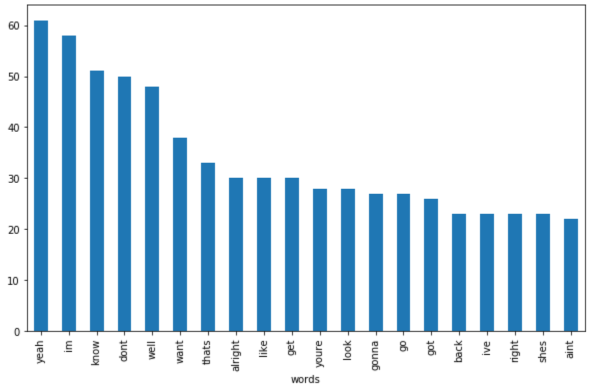
From above we can understand that an average of 60 dialogue is spoken by each character.

Distribution of Most common unigrams



We can see there are several unwanted tokens, which do not have any meaning, and can be removed.

Distribution of Most common unigrams with stop-word removal



Now when we have removed the stop words, we are getting several good words which are actually seen in spoken dialogue.

Feature Engineering

We are working on Dialogue data set, and we have observed that there is a very small amount of textual data available in the spoken dialogue (the length of each dialogue is small). But if we can determine the type of statement that a person is speaking, i.e. a question-type statement or an answer, it can be considered as one of the features that can be useful. Thus, we look in Dialogue Act Tagging of each sentence. Using CRF tagging technique I have built a Dialogue Act CRF tagger which was trained on Switchboard corpus of telephonic conversations. A decent accuracy of 74% is attained.

Other features such as number of words, misspelled word, lexical counts are used but they donot contribute in bringing a good classifier.

Since one part of the problem is very specific to gender classification, Michael Tran et al, in his paper on gender classification describe F-measure which is, a unitary measure used to measure a text's relative contextuality (implicitness), as opposed to its formality (explicitness). Contextuality and formality can be captured by certain parts of speech. A lower score of F-measure indicates contextuality, which means a greater relative use of pronouns, verbs, adverbs, and interjections. A higher F-measure indicates formality, which means a greater relative use of nouns, adjectives, prepositions, and articles. F-measure is defined by the following formula:

$$F = 0.5 * [(freq.noun + freq.adj + freq.prep + freq.art) - (freq.pron + freq.verb + freq.adv + freq.int) + 100]$$

Another important feature is the gender preferential words, for example women use adjectives which are more expressive such as terribly, awfully, sorry.

Word2Vec can be used as one of the features if we combine the vector by averaging or some other operation but this would cause unnecessary overhead with less payoff, one way to use the essence of word2vec is to define a word feature dictionary which contain words belonging to similar groups and count the number of words occurring in dialogue.

POS tagging and chunking have been used as one feature, as certain sequence of sentences are structurally similar but have different meaning, and we would like to see how this change with gender and different character.

Thus, my feature set consists of

1. POS Tagged word of form TOKEN_POS
2. POS Tokens
3. Dialogue Act Features
4. F-measure
5. Gender Preferential word count
6. Similar Word Group factors count
7. Lexical counts

8. Misspelled word length
9. Total Word length

Model Selection

For selecting models, I have exploited sklearn's rich features. To build a model in sklearn, we can create feature pipeline, use different vectorizers on our feature set and select best parameters and fit different models.

The most important thing which has really improved my initial accuracy for gender classification of 55% is to correctly use different vectorizers available in sklearn.

POS Tokens and POS Tagged words should use CountVectoriser or TfIDFVectoriser.

First seven highly occurring Dialogue Act Features were first label encoded and then these were either transformed using TfIDF or used as it is. Same technique is used for Gender Preferential word count and Similar Word Group count.

For F-measure, word count, lexical Counts and others minmax scaling or actual values are used in feature building pipeline.

I have used 3 different types of classifiers Multinomial Naïve Bayes (NB), Support Vector Machine (SVM) and Support Vector Machine Regression (SVMR) for Gender Classification and NB and SVM for Character Classification.

In order to evaluate best parameters for a model, I have used Grid Search with Cross Validation (CV) set 5 to search best parameters. Due to the limited time, feature selection and dimension reduction operation is not completed and hence is not presented in the current submission.

I have tried several deep learning models too, but they fail to generalize well on the test data set, for this reason I have not included them in the report.

Evaluation and Optimisation

Below is the result of models trained on training dataset and tested on test for Gender Classification, the best accuracy was achieved for SVM (**59.96%**). All the models generate around 40% miss-classification error for Gender classification.

```
### Naive Bayes ###
              precision    recall  f1-score   support

      Female       0.55      0.65      0.60       526
        Male       0.64      0.54      0.59       598
confusion matrix:
[[340 186]
[273 325]]
Naive Bayes Accuracy: 59.16%
Mis-classification error for Naive Bayes : 40.84%
### SVM ###
              precision    recall  f1-score   support
      Female       0.57      0.61      0.59       526
        Male       0.63      0.59      0.61       598
confusion matrix:
[[323 203]
[247 351]]
```

SVM Accuracy: 59.96%

Mis-classification error for SVM : 40.04%

Other results are present in MainClassification.py

For Character Classification, SVM DISCRETE (here discrete means, that some of the features were used as they exist without TfIDF Transformation) achieved **23.13%** accuracy although it can be seen as a very poor performance of model, but this result can be due to various contributing factors such as absence of dominating features for particular character.

```
              precision    recall  f1-score   support

     BRADLEY       0.33      0.07      0.12        41
   CHRISTIAN       0.18      0.13      0.15        46
       CLARE       0.12      0.16      0.14        31
       GARRY       0.19      0.08      0.12        48
     HEATHER       0.26      0.33      0.29        42
         IAN       0.18      0.23      0.20       101
        JACK       0.20      0.12      0.15        85
        JANE       0.38      0.29      0.33        76
         MAX       0.30      0.34      0.32        73
       MINTY       0.33      0.22      0.26        51
        PHIL       0.23      0.21      0.22        53
      RONNIE       0.26      0.25      0.25        52
        ROXY       0.22      0.11      0.14        56
        SEAN       0.09      0.05      0.06        63
     SHIRLEY       0.23      0.18      0.20        73
      STACEY       0.17      0.18      0.17        72
      STEVEN       0.18      0.08      0.11        37
       TANYA       0.25      0.60      0.35       124
confusion matrix:
[[ 3  2  2  2  0  6  0  2  0  0  0  1  0  2  3  2  1 15]
[ 0  6  1  1  2 12  3  4  6  0  1  1  0  0  0  1  0  8]
[ 0  1  5  0  0  1  1  0  3  0  1  0  1  0  5  6  0  7]
[ 0  0  1  4  5  5  2  3  4  1  4  1  1  1  5  2  1  8]
[ 0  0  2  1 14  2  1  2  2  4  1  1  0  1  2  4  0  5]
[ 0  6  4  2  4 23  1  5  1  3  5  5  0  4  3  7  2 26]
[ 0  3  1  3  1  7 10  2  8  2  4  8  4  4  2  9  2 15]
[ 0  2  0  0  2 12  3 22  4  0  4  2  0  2  1  4  0 18]
[ 0  1  4  0  1 10  2  0 25  3  2  0  1  3  1  3  0 17]
[ 0  1  0  2  7  1  1  3  5 11  3  1  2  3  0  2  0  9]
[ 0  1  2  0  4  7  1  2  0  3 11  3  3  2  5  2  0  7]
[ 1  2  0  1  2  3  2  2  1  1  1 13  3  0  3  3  2 12]
[ 0  0  1  0  1  5  4  3  2  1  2  5  6  1  4  5  0 16]
[ 2  1  2  2  1  6  4  2 11  0  2  2  0  3  3  5  2 15]
[ 1  1  5  1  4  7  1  1  5  3  1  2  1  2 13  8  1 16]
[ 1  0  4  2  2  7  2  0  2  1  2  3  3  1  2 13  2 25]
[ 0  2  2  0  1  7  3  2  0  0  1  0  2  1  2  2  3  9]
[ 1  5  5  0  2 10  8  3  3  0  3  2  0  3  3  0  1 75]]
SVM DISCRETE Accuracy: 23.13%
```

Other results are present in MainClassification.py

Optimization of parameter selection using grid search is done so that the best parameter guides the model.

Error Analysis

Initially the TfIDF features on Cleaned Tokens were used to build a small model which was giving around 55% accuracy. Below 10 Fold Cross validation on Training sample with TfIDF as the only feature.

```
### Naive Bayes ###
Using TensorFlow backend.
0 - Cross Validation Accuracy: 0.4970 (+/- 0.01)
1 - Cross Validation Accuracy: 0.5068 (+/- 0.01)
2 - Cross Validation Accuracy: 0.5100 (+/- 0.01)
3 - Cross Validation Accuracy: 0.5255 (+/- 0.01)
4- Cross Validation Accuracy: 0.5365 (+/- 0.01)
5- Cross Validation Accuracy: 0.5348 (+/- 0.01)
6- Cross Validation Accuracy: 0.5478 (+/- 0.01)
7- Cross Validation Accuracy: 0.5477 (+/- 0.01)
8- Cross Validation Accuracy: 0.5435 (+/- 0.01)
9- Cross Validation Accuracy: 0.5571 (+/- 0.01)
```

From the above we can see that Naïve Bayes struggle to reach even 55%, which indicated that just TfIDF features were not enough, certain gender specific features needed to be engineered along with statement type features in order to better capture the data.

Conclusion

In this project we have implemented several NLP feature engineering techniques, such as building a CRF Tagger, using Sequential modelling techniques and combining other techniques like sematic and logical along with probabilistic learning. Few techniques such as counting number of bigrams and trigrams had adverse effect on performance of model while other techniques such as POS tagging to find patterns really improved the performance. The performance of the model depends upon the features we fit, better features better results.