Genre Identification on a sub-set of Gutenberg Corpus

*Semester Project for Advanced Topics in Machine Learning

1st Sesha Sai Kiran Bhavaraju
Fakultät für Informatik
Otto von Guericke University
Magdeburg, Germany
sesha.bhavaraju@ovgu.de

2nd Raj Rajeshwari Prasad
Fakultät für Informatik
Otto von Guericke University
Magdeburg, Germany
raj.prasad@st.ovgu.de

3rd Pawan Kumar
Fakultät für Informatik
Otto von Guericke University
Magdeburg, Germany
pawan.kumar@ovgu.de

4th Mariam Riaz
Fakultät für Informatik
Otto von Guericke University
Magdeburg, Germany
mariam.riaz@st.ovgu.de

5th Lakshmi Sindhu Kodali
Fakultät für Informatik
Otto von Guericke University
Magdeburg, Germany
lakshmi.kodali@st.ovgu.de

***Abstract*—This paper focuses on the classification of Genre in the Gutenberg Corpus. The corpus used here is of the 19th Century English Fiction created from Project Gutenberg. We have aimed this work at extracting interpret able domain specific features and their respective usages in various model scenarios. Project Gutenberg is a library of over 60,000 E-Books for public non-commercial use. We discuss the extraction of the said and so on, using public libraries and the classification on the feature vector. We also make points on how we handle feature representation, class imbalance, and model evaluation. The actual detection task consists of using the said feature rich space on 3 basic models namely, Artificial Neural Network, Support Vector Machine based Classifier and Naive Bayes Classifier. We then proceed to list the evaluation strategies and the various experiments performed, along with the steps taken to mitigate some of the issues that we faced. Finally some conclusions are drawn proceeding an examination.**

***Index Terms*—Genre Identification, Gutenberg Project, fiction**

I. Motivation and Problem Statement

The 19th Century English Book Category selected is a subset which consists of about 1079 books and 9 genres. Each instance in the data set is a book and the target label provided is a Genre label. The files are structured as HTML documents and parsing them would result in the meta structure being preserved in the form of HTML tags. Such an organization of data would lead to solving problems such as pruning the tags before any logical operation. We would also have to answer the question of numerating such domain specific features and addressing the notion of similarity between two texts. The effect of model selection and model evaluation approaches on the final classification will also be evaluated.

II. Dataset

The Data set provided to us was basically structured as follows, a ground truth containing the parameters Book_Name, book_id, guten_genre, Author_Name. The parameter "book_id" was then used as a primary key to a content directory containing 1079 HTML files representing the content of 1079 books. Author_Name follows the format Lastname — FirstName. Short stories or books with multiple authors are not present. 83 books which were present in the Content directory were not labelled in the ground truth. We eliminate these 83 Unlabelled Files and then use them as a special test set. A quick reference to the format in each HTML file is as follows,

```
<!DOCTYPE html>
    <html>
        <body>
            <p>Sample Paragraph 1</p>
            <p> Sample Paragraph 2</p>
            <p> Sample Paragraph n</p>
        </body>
    </html>
```

A. Class Distribution

We have 996 instances which are labelled, in total, and which are distributed across 9 classes as follows.

- Literary, 794
- Detective and Mystery, 111
- Sea and Adventure, 36
- Love and Romance, 18
- Western Stories, 18
- Ghost and Horror, 6
- Humorous and Wit and Satire, 6

- Christmas Stories, 5
- Allegories, 2

Further, for a better visualization of the classes, a bar chart is being shown , refer “Fig. 1” with labels on the horizontal axis, and the number of instances on the vertical axis.

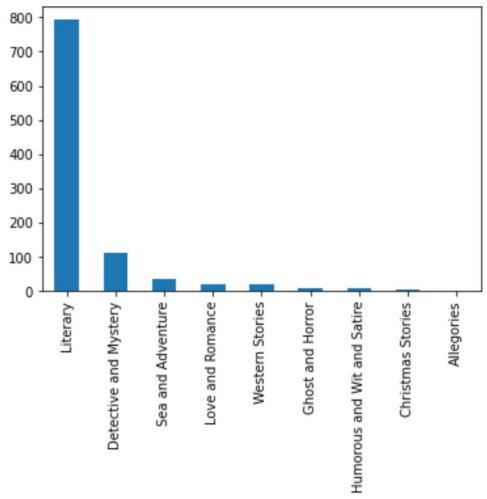

Fig. 1. Distribution Of Classes

B. Anomalies

These are some of the challenges that we need to address in the data set before we proceed further.

- We have found that some of the books had no content. These books were not considered.
- Some of the books had less than 40 Words which contained, zero to no meaningful information, i.e. all the words were either titles or some irrelevant aspects. These were left as is.
- The difference between the number of instances in the Most populous class (794/996) and the least populous class (2/996) is huge. We address this by multiple methods that will be discussed below.

III. Concept

The Data set provided to us consists of book data in HTML files. Upon research we have found that we could use certain domain specific features. The features used, along with the intuition have been mentioned below. The brief introduction of the models used and their brief summaries have also been listed below. We have also opted to use two representations of the data set. One a feature represented data set on book level and the other a feature represented data set on chunk level.

A. Chunking Approach

We have opted to perform a chunking procedure on each book which would then split it into multiple chunks.The strategy we have opted for is, a 5000 word limit in a single chunk to maintain that each represents the same amount of data approximately. For books with less than 5000 words, we have left them as is. This ensured also that each chunk had the same descriptive power. This posed a problem of maintaining Atomicity. Atomicity here meant we would have to ensure the model sees the complete book or sees none. At no point would there be a situation where parts of the same book are distributed among training and testing sets. We have tried best to ensure the said split by first splitting the books to train and test (80 : 20) and then performing chunking on the respective sets. This would not also mess up with the class distributions. Employing such chunking measures meant we would have more granular control over data while performing sampling techniques for instance.

B. Feature Intuition

Given a task of classifying “The Scarlet Letter” or “To Kill a Mockingbird”, the quality of distinction can be directly linked to the ability of interpreting and analyzing the semantics in literary texts. To achieve such an intuition, we need to make use of features which are domain specific, in our case, English Literature. We could then define literary styles, their definitions and establish their respective foundations on a numeric scale. Literary styles are techniques that writers use to create a special and precise effect in their writing, to convey information, or to help readers understand their writings on a conceptual level. More generally, it can vary from author to author and from genre to genre. Fiction writing in general, tends to place value on plot and an underlying theme, as a result, tending to be more popular with the masses. It has usually dense dimensional characters, a pleasing arc of tension, evocative language and thematic purpose. For instance, Romance, focuses on romantic love between two people and often has an overall positive sentiment. So, an overall assessment among the use of different kind of pronouns (personal, male, female), punctuation, prepositions, use of quotes, average sentence length, contributes significantly in distinguishing between different literary styles thereby distinguishing genres.

C. Formalization

The aforementioned domain specific literary features can be captured via these 22 semantic attributes. “Tab. I” [6] [7]

1) Paragraph Count: This feature defines the number of paragraphs in a book, which relates to our task because more number of paragraphs define a certain writing style. This feature is extracting the number of HTML tags in our text defined as follows.

```
<p>
```

This feature extraction requires us to not strip the book off any HTML tags in the beginning.

2) Female Pronoun Count: This feature deals with the female orientation in the given text. We are interested in this feature because some genres can be sensitive to a female oriented plot line. We extract this feature by searching through the text for identifying any general female pronoun. The list of female pronouns is exhaustive in the English language and

TABLE I
FEATURE LIST

Features
Paragraph Count
Female Pronoun Count
Male Pronoun Count
Personal Pronoun Count
Possessive Pronoun
Preposition Count
Coordinating Conjunction
Comma Count
Period Count
Colon Count
Semi Colon Count
Hyphen Count
Interjection Count
Punctuation and Subordinating Conjunction
Sentence Length
Quotes Count
Negative Sentiment
Positive Sentiment
Neutral Sentiment
Flesch Reading Score
Number of Characters
Type Token Ratio

is stored before hand. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification. Also the same genre books might then be related to each other based on the sentence complexity.

3) Male Pronoun Count: This feature deals with the male orientation in the given text. We are interested in this feature because some genres can be sensitive to a male oriented plot line. We extract this feature by searching through the text for identifying any general male pronoun. The list of male pronouns is exhaustive in the English language and is stored before hand. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification. Also the same genre books might then be related to each other based on the sentence complexity.

4) Personal Pronoun Count: We extract this feature by searching through the text for identifying any general personal pronoun. The list of personal pronouns is exhaustive in the English language and is stored before hand. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification. Also the same genre books might then be related to each other based on the sentence complexity.

5) Possessive Pronoun: We extract this feature by searching through the text for identifying any general possessive pronoun. The list of possessive pronouns is exhaustive in the English language and is stored before hand. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification. Also the same genre books might then be related to each other based on the sentence complexity.

6) Preposition Count: We extract this feature by searching through the text for identifying any general preposition. The list of prepositions is exhaustive in the English language and is stored before hand. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification. Also the same genre books might then be related to each other based on the sentence complexity.

7) Coordinating Conjunction: We extract this feature by searching through the text for identifying any general Coordinating Conjunction. The list of Coordinating Conjunctions is exhaustive in the English language and is stored before hand. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification. Also the same genre books might then be related to each other based on the sentence complexity.

8) Comma Count: We extract this feature by counting the number of commas in the given text. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification. Also the same genre books might then be related to each other based on the sentence complexity.

9) Period Count: We extract this feature by counting the number of periods in the given text. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification. Also the same genre books might then be related to each other based on the sentence complexity.

10) Colon Count: We extract this feature by counting the number of Colons in the given text. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification.

11) Semi Colon Count: We extract this feature by counting the number of Semi Colons in the given text. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification.

12) Hyphen Count: We extract this feature by counting the number of Hyphens in the given text. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification.

13) Interjection Count: We extract this feature by counting the number of ! characters in the given text.This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification.

14) Punctuation and Subordinating Conjunction: We extract this feature by searching through the text for identifying any general Punctuation or a Subordinating Conjunction. The

list of Punctuation and Subordinating Conjunctions is exhaustive in the English language and is stored before hand. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification.

15) Sentence Length: The sentence length is an important factor to analyse a literary text because longer sentences are difficult to interpret and shorter ones are not. The said, can be attributed to different writing style of an author who writes for Detective and Mystery in comparison to Christmas stories which are targeted for kids thus, helping in genre identification.

16) Quotes Count: We extract this feature by counting the number of " " in the given text. This feature contributes to the writing style of an author. The author can be directly related to books, i.e. writing books of the same genre and hence impacting our classification.

17) Sentiment: Sentiment can be loosely, is termed as the feeling that a piece of text conveys. This feature can be a vital sign in case of a genre identification. The assumption that we have made is that a novel based on the Detective and Mystery will have more parts of it which convey a negative sentiment when compared to a book on Humor, Wit and Satire which would most probably have more of its parts contributing to a positive sentiment. We have extracted sentiment for a given piece of text, using the NLTK library. The Specific method attributing to a 3 scale sentiment - Negative, Positive, Neutral, is analysed using the Vader Sentiment Analyser. VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is attuned to sentiments expressed in English text. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labelled according to their semantic orientation as either positive, negative or neutral.

18) Flesch Reading Score: In the Flesch reading-ease test, higher scores indicate material that is easier to read; lower numbers mark passages that are more difficult to read. The formula for the Flesch reading-ease score (FRES) test is as follows

```
206.835 - 84.6(tot. syllables/ tot. words)
- 1.015(tot. words/ tot. sentences)
```

The range of this score

$$[-\infty, 121.22]$$

We extract this feature using the library textstat. There is a api call to obtain the above ratio on a given piece of text. We reckon The score can be attributed to genre change under the assumption that a Romance novel is easier to read than a Detective and Mystery novel.

19) Number of Characters: This attribute is responsible for analysing the plot complexity. We assume that a plot is complex if it has more number of Characters and the relative dialog between them, and vice versa. Genre in texts might be sensitive to the plot complexity. For example a Romantic novel might be more focused on fewer characters, when compared to a genre like detective and mystery. We extract this feature using the label tagging function of the open source library Spacy. Spacy analyses a piece of text using their custom built language libraries to tag a word pertaining to their custom tags. One of them is a PERSON tag, which identifies proper nouns, disregarding places and objects. We exploit this functionality to identify unique character names.

20) Type Token Ratio: Type Token Ratio formally is defined as follows

$$TTR = UniqueWords/TotalWords$$

This feature is extremely useful to judge the lexical diversity of an author. Like mentioned above, an authors range of vocabulary varies attributing to the lexical diversity in a book of a certain genre. We extract this feature by splitting the input text on basis of a space and then counting the TTR ratio with respect to words.

D. Model

On the basis of the task definition and the data provided, we selected three classifiers - Support vector Machines, Multi Layer Perceptron, Naive Bayes Classifier. Further these classifiers were tuned and different models were evaluated.

1) Support Vector Machine: The support Vector Machines (SVM) are efficient classifiers for finding hyperplane in a N-dimensional space for the classification of data points with the help of Support Vectors. Support Vector Machine with the help of Kernel matrix plots the data set into high dimensional space for better separation of points.

Most of the text classification problems are linearly separable. For this reason, SVM are considered to perform better than other classifiers. Also, SVM classifiers are good in handling high dimensional data.

2) Multi Layer Perceptron: We used the neural network multilayer perceptron classifier with the help of sklearn library for training our model. MLP classifier is a feed forward neural network which performs its learning on the basis of back propagation. We used neural network as they are self-training Supervised models which are not severely affected by outliers and works well on text classification.

3) Naive Bayes Classifier: Naive Bayes classifiers are linear classifiers that are known for being simple yet very efficient. The probabilistic model of naive Bayes classifiers is based on Bayes' theorem, and the adjective naive comes from the assumption that the features in a dataset are mutually independent. In our experiment we were interested in finding the probability of a label given our extracted features, if we make this naive assumption that the extracted features are not co-related and contributes fully in defining a label, we can find a rough approximation of the generative model for each class, and then proceed with the Bayesian classification. In our model we have used Gaussian Naive Bayes classifier, which runs on the assumption that data from each label is drawn from a simple Gaussian distribution it means that in our task it assumes that the labels of each genre follows a different Gaussian distribution on the set of our features.

IV. Implementation

A. Specific Tools Used

- scikit-learn 0.23.0
- python 3.7
- spaCy v2.x
- spaCy "en-core-web-lg"
- nltk for python 3.7
- textstat 0.6.2
- imb-learn 0.50
- pandas 1.0.5
- numpy 1.9.2
- re 7.2
- codecs 7.8
- tensorflow 2.x

B. Tool Specific Assumptions

Using custom API for extracting features meant dealing with the APIs' constraints. We have assumed some of the restrictions with respect to our implementation. Most of APIs' these restrictions come with the black box nature of the S in general. We have listed our observations below

- Vader Sentiment analyser is attuned towards sentiments expressed in social media, but is also deemed compatible for general English texts on their documentation. This is assumed to be true.
- spacy.load() has a default maximum character length of 1000000 and is increased to 2000000 characters to accommodate the largest book which had 1900000 characters. The warning of memory leaks while parsing long texts is ignored.
- vader_lexicon from nltk is assumed to be updated for sentiment analysis
- sampling functions from imblearn are assumed to behave as expected and not contain duplicates while sampling.

C. Class Imbalance

The Idea for dealing with class Imbalance problem at hand we have implemented Under Sampling majority target class with 30% and creating synthetic data for the remaining minority target classes by oversampling and increasing the data by 30% We have used the API imblearn.under_sampling.RandomUnderSampler , which randomly picks and deletes the examples from the majority classes , we have kept the ratio for Majority class to be decreased by 30 % while instantiating the object of RandomOverSampler.

RandomOverSampler of the same API(imblearn.OverSampling) follows a naive approach of creating the exact duplicates of the instances to match/be comparable to the majority class therefore we considered a better approach by using Synthetic Minority Oversampling(SMOTE) which creates new instances using K nearest neighbor.

D. Model Implementations

1) Support Vector Machines: SVM models were built with K-Fold validation technique. For this, Sklearn's SVC and Cross validation library were used. The train and validation split performed is not used and whole training data is passed in the model as the K-Fold validation technique is being used for the purpose.

K-Fold Cross Validation Cross-Validation is an important technique to estimate the performance of the model on validation data which allows further parameter tunings. This is very helpful when our data is limited. The K-Fold validation splits the training data into K splits and holds one split for evaluation and rest for training. This process is repeated until every split is used as a validation set. Parameter Tuning Sklearn allows many parameter tuning options to improve model performance. For SVM, following parameters were tweaked,

- Kernel : Polynomial
- class weight : balanced/None
- scoring : f1 weighted

For KFold validation, following parameters were tweaked,

- cv : For deciding number of splits as 5

Three different data sets were passed to the models.

- Chunked data without synthetic data.
- Chunked data with synthetic data.
- Book level data.

2) Multi Layer Perceptron: MLP was built with the help of sklearn library with the following parameters:-

- Neurons per hidden layer: (100,1)
- Activation Function: relu
- Optimizer:Adam
- Number of epochs: 300

3) Naive Bayes Classifier: The Gaussian Naive Bayes Model has been applied to Book and chunk level data with the three approaches displayed in evaluation.

V. Evaluation

A. Model Selection

1) Train,Validation and Test Split: We estimated our model's performance on how well it predicts unobserved instances, therefore we divided our data and kept a specific part as testing data. The above approach was adopted using sklearn.modelselection library, the major optimization which was performed was to specify the split ratio. Since we had a class imbalance problem , any test size less than 0.2 was resulting less/no instances for minority target classes like allegories in test data.

Further, the similar approach was adopted with same test size for creating Validation data for comparing with our training data.

2) Performance Metrics: Performance metric is an important criterion for model selection. The following performance metric was taken into consideration, refer "Fig. V-A2"

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Fig. 2. Confusion Matrics

- Accuracy: It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = (TN + TP)/(TN + TP + FN + FP)$$

Accuracy in alone cannot be considered as a decisive criterion for model selection as it is highly biased towards the majority class of the dataset. In our case, 79.9% data belongs to the 'Literary Class'. Even if we predict all the data as of Class 'Literary', we would have an accuracy of 79.9%.

- Precision: It is a measure which determines the effect of False Positive Classification.

$$Precision = (TP)/(TP + FP)$$

- Recall: It is the ratio of True Positive and summation of Total Actual Positive.

$$Recall = (TP)/(TP + FN)$$

- F1-Score: F1 score is a better measure as it seeks a balance between Precision and Recall. It also handles the case of uneven class distribution.

$$f1score = (2{*}Precision{*}Recall)/(Precision{+}Recall)$$

B. Model Performance

1) Support Vector achines: In total 6 experiments were performed.(Chunk Data : CD, Synthetic Data: SD)

TABLE II
SVM EXPERIMENT RESULTS

Exp	Data Type	Class Weight	Folds	Mean f1 Score
1	CD ohne SD	None	5	0.74
2	CD ohne SD	Balanced	5	0.53
3	CD mit SD	None	5	0.59
4	CD mit SD	Balanced	5	0.59
5	Book Level Data	None	5	0.62

On the basis of above results, Model in experiment 1 was selected. Further, performance on Test data was recorded 0.69.

2) Multi Level Perceptron: Result of Multi Layer Perceptron classifier on Chunked Balanced Data with validation Data

3) Naive Bayes Classifier: Results of experiments:

Since after the train-test split, only single instance for Allegories was allocated in training set and the K nearest neighbor algorithm could not be implemented to generate synthetic data for Allegories using SMOTE. Therefore we considered the particular class as an outlier and removed.

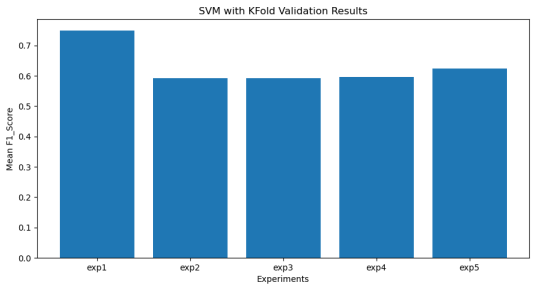

Fig. 3. SVM Experiment Results

	precision	recall	f1-score	support
Allegories	1.00	0.50	0.67	2
Christmas Stories	1.00	0.25	0.40	8
Detective and Mystery	0.50	0.41	0.45	259
Ghost and Horror	0.25	0.12	0.17	8
Humorous and Wit and Satire	0.33	0.05	0.08	22
Literary	0.87	0.95	0.91	2091
Love and Romance	1.00	0.11	0.20	36
Sea and Adventure	0.71	0.35	0.47	82
Western Stories	0.58	0.30	0.40	46
accuracy			0.84	2554
macro avg	0.69	0.34	0.42	2554
weighted avg	0.82	0.84	0.82	2554

Fig. 4. MLP Results.

	precision	recall	f1-score	support
Christmas Stories	0.00	0.00	0.00	0
Detective and Mystery	0.24	0.48	0.32	25
Ghost and Horror	0.00	0.00	0.00	2
Humorous and Wit and Satire	0.00	0.00	0.00	1
Literary	0.83	0.26	0.40	154
Love and Romance	0.04	0.25	0.06	4
Sea and Adventure	0.17	0.30	0.21	10
Western Stories	0.04	0.33	0.06	3
accuracy			0.29	199
macro avg	0.16	0.20	0.13	199
weighted avg	0.68	0.29	0.36	199

Fig. 5. Model performance on book level data with SMOTE to deal with class imbalance.

	precision	recall	f1-score	support
Allegories	0.00	0.00	0.00	1
Christmas Stories	0.00	0.00	0.00	0
Detective and Mystery	0.22	0.57	0.32	21
Ghost and Horror	0.00	0.00	0.00	0
Humorous and Wit and Satire	0.00	0.00	0.00	1
Literary	0.89	0.36	0.52	163
Love and Romance	0.20	0.33	0.25	3
Sea and Adventure	0.05	0.12	0.07	8
Western Stories	0.17	0.33	0.22	3
accuracy			0.37	200
macro avg	0.17	0.19	0.15	200
weighted avg	0.76	0.37	0.46	200

Fig. 6. Model performance on book level data with Random Under sampling in combination with SMOTE to deal with class imbalance.

	precision	recall	f1-score	support
Allegories	1.00	0.50	0.67	2
Christmas Stories	0.17	0.38	0.23	8
Detective and Mystery	0.29	0.56	0.38	259
Ghost and Horror	0.17	0.75	0.28	8
Humorous and Wit and Satire	0.14	0.32	0.19	22
Literary	0.91	0.67	0.77	2091
Love and Romance	0.18	0.42	0.25	36
Sea and Adventure	0.13	0.40	0.20	82
Western Stories	0.20	0.35	0.26	46
accuracy			0.64	2554
macro avg	0.36	0.48	0.36	2554
weighted avg	0.79	0.64	0.69	2554

accuracy: 0.63821

Fig. 7. Model performance on chunk level data with Random Under sampling in combination with SMOTE to deal with class imbalance.

VI. Conclusion

In this project we tried to evaluate the performance of three classifiers; Naive Bayes, Support vector Machines and Multi layer perceptron in classifying genre of the given books having different literary text styles. We used different techniques to extract domain specific features that quantify the concept of "plot" from a book, those features allowed us to compare the given books on a semantic and syntactic level, which makes our model more interpret able as compared to just using a Bag of words or word2vec representation of a text. We also used different evaluating metrics for our classifiers. Though all our models were successfully trained and tested, but not all of them performed well on the given task the best performance was achieved from Multi layer perceptron. We have the intuition that the results of our classifier can be greatly improved if we have a more balanced data set. Semi Supervised learning can also be an approach to further get a more balanced data set. Our project can give a good foundation for the future work on the pre-processing and domain specific feature extraction of the given corpus.

References

[1] Prasanna, P. & Rao, Dr. (2018) "Text classification using artificial neural networks" International Journal of Engineering and Technology(UAE).

[2] Basu, Atreya & Watters, Carolyn & Author, Michael. (2003). Support Vector Machines for Text Categorization.. 103. 10.1109HICSS.2003.1174243.

[3] Joachims, Thorsten. (1998). Text Categorization with Support Vector Machines. Proc. European Conf. Machine Learning (ECML'98). 10.17877/DE290R-5097.

[4] Raschka, Sebastian. (2014). Naive Bayes and Text Classification I Introduction and Theory. 10.13140/2.1.2018.3049.

[5] Liguo, Duan Peng, Di & Aiping, Li. (2014). A New Naive Bayes Text Classification Algorithm. TELKOMNIKA Indonesian Journal of Electrical Engineering. 12. 10.11591/telkomnika.v12i2.4180.

[6] Sayantan Polley and Suhita Ghosh, Comparing the qualitative impact of different features and similarities on fictional text using SIMFIC, github.com/sayantanpolley/fiction/blob/master/SIMFIC_LNCS.pdf

[7] Sayantan Polley, Suhita Ghosh, Maruc Theil, Michael Kotzyba and Andreas Nümberger, "SIMFIC: An Explainable Book Search Companion", github.com/sayantanpolley/fiction/blob/master/ICHMS2020_paper_42.pdf