Two assumptions:

* Large scale isotropy property of data
* Data independence over parallel processing

Concepts:

Style:

Throughout the history of literary analysis, writing style as a feature has always been considered something that can only be attributed to individual authors, that is every author has his own style of writing articles, novels etc. This trend came to an end by the endeavors of some linguist like Kenneth Dover. In an article, he showed1 that there is “a certain degree of dependence of style upon content” in the sense that one writer’s handling of a novel, might not only be his “mannerisms” in writing that particular novel, it might also be the case that other authors have chosen the similar mannerism while writing literary piece of the same genre. To elucidate his point, he took Thucydide and Isocrates’ literary works and found statistical dissimilarity in the same book that the writers wrote separately, but similarity in their writing style in handling similar situations, which certainly is counter-intuitive since we attribute writing style to individual author only. Perhaps, Susan MacDonald summarized it2 pretty well by saying, “Genres derive not from preexisting forms or previously dictated rules, but from the similarity of recurring rhetorical situations.”

One of the frontiers who initiated this trend of considering style features as part of genre rather than only an “authorship attribution tool” is most probably Douglas Biber. In his book3 he described at full length what the relation between style and genre. In his other book4, he provided a splendid list of 67 features that can be extracted from a text corpora to use in clustering the corpora along genres. To the best of our knowledge, Karlgren and Cutting, 19945 provided perhaps the first known implementation of this analysis of Biber. They selected, naively, 15 features from this list and used them in classifying Brown Corpus. On the other hand, Kessler6 et al. suggested a prudent way of selecting features from different generic cues. Further approaches and analyses can be found in Wolters7 et al. and in Cimino8 et al.

In this current work of ours, we tried to extract a subset of features given by D. Biber. For this, we haven’t applied any intelligence. We’re taking two types of features here: character level features and word level features. For character level features we have average character per word, average punctuation per sentence etc. For word level features, we have different POS ratios, article ratio, type token ratio etc. Furthermore, there are certain features like amplifiers, downtoners, demonstratives that are being used for the first time in the analysis of style feature. The list of all the features that were extracted is attached in the appendix A.

**Implementation Details**

Implementation details can be summarized by mentioning some challenges that were answered during the implementation.

* Data imbalancement: As we have already mentioned, the corpus has a severe data imbalancement problem. Some genres have more representatives than others. We’re defining it as ‘severe’ because the ratio of the highest and the lowest data representation is 2/794 = 0.0025 or 0.25%. In order to tackle this challenge, at first, we are not taking all the books in the corpus, rather we are selecting a decent number of books from each genre. Here we are assuming that our data has a **large scale isotropy** property i.e. if we take a couple of random samples without replacement (i.i.d.) we will get a good representative of the individual genre. That way we are selecting a few samples from the huge corpus to represent each genre. Highest number of books taken is 30 here. For categories where the total number of representative books are less than this 30 threshold, all the books were selected. It should be noted that this number 30 has no statistical significant whatsoever.

Furthermore, after we’ve acquired all our feature vectors, still, we have data imbalancement in our dataset as some genre has around 30 representative and some have around 4. If we keep our dataset as it is, our model will be biased to those classes with higher representative as it will have more data about them, it can tune itself to the variation of those classes. To tackle this issue, we are using Adaptive Synthetic Sampling Approach (ADASYN).9 It is a modification of Synthetic Minority Oversampling Technique (SMOTE10). Here, we are generating synthetic data points for classes that have very few representatives.

In the original paper of SMOTE10, the authors suggest that their method of oversampling works better if implemented with a combination of over and under sampling, in comparison to using over sampling alone. So after doing an over sampling of the minority classes, we are doing an under sampling of all the classes. This eventually gives us a dataset whereby all the classes has the same amount of data.

Also, we’re using ‘balanced’ class weight to penalize classes with more data, if there is any. Unfortunately, because of using over sampling technique, we had to drop the values of ‘allegory’ class as ADASYN cannot work if there is only one data point. If we split the two points of allegory class it eventually boils down to one data point in each of training and test set.

* Reducing run-time: We have divided each book into a number of chunks of size 1000. From each of these chunks, we are calculating 15 features. Keeping a count for all the 15 features, we are dividing them by the number of chunks at the end to generate a ratio of the feature in question. We’re doing this to avoid the issue of book length. As we’re taking a ratio here for each feature, book length did not become an issue while comparing one book with another.

But this chunking and calculating does not come without downside. If we have n books and m chunks per book, it means 15\*m\*n calculations are needed to get the feature vector out of the reduced dataset. If we do it sequentially, one book at a time then it would take 15\*m\*n\*t seconds considering each book takes ‘t’ seconds to complete. This number becomes O(n3) if we consider m=n=t. But in reality n << m, t. So, evidently the computation time becomes much higher than O(n3).

To reduce this, we are using parallel processing for each chunk and for each book. 10 books are being processed in parallel to each other and for each of those 10 books, 10 chunks are being processed in parallel. Here, we have a second assumption, that each of these chunks can be **processed independently of each other**. Although this can also be classified as a design choice, as this parallel processing assumption is not an assumption at all, rather it’s a fact that these features, as they’re style features, do not depend on chunks at all. It should be noted in this context that while we are parallel processing everything, still we are only keeping 10\*10 = 100 threads alive at a time, i.e. after creating 10 threads for file and 10 for chunks, we are waiting for them to finish to start another 10\*10 threads. That way we are keeping the register memory from flooding with tons of threads.

* Modularity: As this is a team project, we want all of our members to work as independently as possible from each other. To ensure that we have modularized the whole project into a number of sub processes and writing the output of each sub process into separate csv files. That way we can use plug and play philosophy to incorporate any members’ contribution with least effort. Besides, we don’t want to run the full pipeline, from data selection to feature extraction to model output all the time. So, writing results of individual sub processes into csv files helps us to resume from the place where we left off the last time.
* Data normalization and pruning: Upon calculating all the features that we set out to calculate, we have a 24 dimensional feature vector. Almost all of these features are some sort of ratio so we don’t have to normalize the data once again after it has been generated. But to regularize data abnormality in vertical axis, i.e. column-wise, we are performing standard scaling, which gives us a further normalized and well-behaved dataset.

As our feature vector has 24 dimensions, we want to select the best out of these before feeding it to any algorithm. After much experimentation, we arrived at the conclusion of using the 20 best features using chi-square feature selection test. This indeed gave us better result as experiment section will verify.

* Running machine learning models: We have decided to test multiple machine learning models for our dataset. Because of this reason, we have implemented Logistic Regression classifier, Multinomial Naïve Bayes classifier, Random Forest classifier, Support Vector Machine classifier (both polynomial and Gaussian RBF kernel), Multi-Layer Perceptron, and a majority voting Ensemble classifier that includes a polynomial kernel SVM, a Logistic Regressor, and a Random forest classifier.

For model parameters, we have done some experiment with different parameter combinations. But when we are running the entire project in one go, we are running a grid search and using the result of this search directly as the model parameters. No heuristic or opinionated value is being used in this case, the parameters are being selected solely based on their performance in the grid search.

As we have selected a subset of our original dataset, it is bound to be smaller than necessary. To tackle this issue, we are using 3 fold cross validation as suggested by Friedman, Tibshirani, and Hastie in their classic book The Elements of Statistical Learning. Before doing this cross validation, we are keeping a portion of our data (25%) separate for determining generalizing accuracy of our algorithms as they have repeatedly suggested.

* Selecting metric for evaluation: As this is a multi-label classification setting, we are using accuracy as our primary metric for evaluation. But it turns out, comparing them using only one metric is not as comprehensive as we would require. So, in addition to accuracy, we are using F1 measure, which is the harmonic mean of precision and recall.

Bibliography

1. Dover, Kenneth - Style Genre and Author. https://www.jstor.org/stable/23065423
2. MacDonald, Susan - Prose Styles, Genres, and Levels of Analysis https://www.jstor.org/stable/10.5325/style.36.4.618
3. Douglas Biber, Susan Conrad - Register, Genre, and Style
4. Douglas Biber – Variation across speech and writing
5. Karlgren, Jussi; Cutting, Douglass - Recognizing Text Genres with Simple Metrics Using Discriminant Analysis 1994
6. Kessler, Brett; Nunberg, Geoffrey; Schutze, Hinrich - Automatic Detection of Text Genre
7. Wolters, Maria; Kirsten, Mathias - Exploring the Use of Linguistic Features in Domain and Genre Classification
8. Cimino, Andrea et al. - Identifying Predictive Features for Textual Genre Classification the Key Role of Syntax
9. He, Haibo; Bai, Yang; Garcia, Edwardo; Li, Shutao - ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning
10. Chawla et al - SMOTE: Synthetic Minority Over-sampling Technique
11. Kelih, Emmerich et al. - Classification of Author and/or Genre? The Impact of Word Length

Appendix A

Word level

1. avg\_sentence\_length
2. long\_word\_ratio
3. noun\_ratio
4. verb\_ratio
5. adverb\_ratio
6. preposition\_ratio
7. adjective\_ratio
8. article\_ratio
9. gerund\_infinitive\_ratio
10. downtoner\_ratio
11. amplifier\_ratio
12. demonstrative\_ratio
13. type\_token\_ratio
14. flesch readability score
15. male\_oriented
16. female\_oriented
17. interjections

Character level

1. avg\_char\_per\_word11
2. avg\_punctuation\_per\_sentence
3. hyphens
4. colons
5. semi\_colon
6. comma
7. period