Haris Muneer (15-4149)

Hassaan Elahi (15-4028)

Identifying Different Writing Styles in a Document Intrinsically Using Stylometric Analysis

Abstract:

In this project, we have developed an intelligent system that takes a single document and classifies different writing styles within the document using stylometric analysis. The classification is done using K-Means Clustering (an unsupervised machine learning method). First, the document is divided into chunks of text using a standard chunk size. Then for each chunk of text, a vector of stylometric features is computed. After that the chunks are clustered using their vectors of stylometric features. Thats where unsupervised machine learning comes into play. The chunks with same style are clustered together. Hence, the number of clusters made correspond to the number of different writing styles that the document has. In this approach, the value of K is determined using Elbow Method. We also ran an experiment for a document with two writing styles and our system was successfully able to identify that the document had two different writing styles. Our approach separates out text with same style from that of different style and can be used to detect plagiarism also.

Background:

The study of measurable features of literary style, such as sentence length, readability scores, vocabulary richness and various frequencies (of words, word lengths, word forms, etc.), has been around at least since the middle of the 19th century, and has found numerous practical applications of interesting problems in the modern era of Artificial Intelligence and Machine Learning. Study of a literary style of a document is called **Stylometry**.

Stylometry grew out of earlier techniques of analyzing texts for evidence of authenticity, author identity, and other questions. The development of computers and their capacities for analyzing large quantities of data enhanced this type of effort by orders of magnitude. In the current era of research, Stylometry is widely used in the problems of intrinsic plagiarism detection, genres separation, authorship verifications and authorship attribution, author gender detections and many more. However, the main task is to classify different writing styles from the text, which can further be used to solve above-mentioned problems.

Literature Review:

We did an in depth study on stylometry, authorship attribution and authorship obfuscation along with little theory on linguistics (e.g zipf's law). These fields used stylometric features to solve the problem. As our work is related to literary writing style so studying linguistics was necessary to actually understand the nuances of these features. Studying them really matured our understanding regarding the project. We started our hunt from Wikipedia**[1]**, studying the basics of stylometry. We learned its basics and advent in the field of computer science, its applications in different fields of computer science and its history. We gathered that currently many researches are doing text analytics using stylometric features. We took some links of research papers with good citations and studied them thoroughly. One research from University of Illinois**[2]** at Chicago exploited stylometry to distinguish actual human readable text vs paraphrased machine written text using stylometric analysis. We have learned some features from this research paper such as n-grams and some lexical features such as frequency count, punctuations and special characters. Intrinsic plagiarism detection is also an application of stylometric analysis and used in **[3]**. Given a document, they have identified suspicious sections of the document for plagiarism. We came across another survey**[4]** which described the use of machine learning and statistics with stylometry.

Introduction:

Our system determines variations in writings in a text document. These variations can be due to different authors or different genres of writing for example stories, research papers, dramas play etc.

Other approaches on Intrinsic Plagiarism Detection (essentially different writing styles detection) needed a large corpus of texts of different authors to train their models to know which text belonged to which author. They learnt the writing styles of each author and then given a document with an author they predicted whether that author plagiarised work of some other author or not. While our approach doesn't need training on a large corpus of texts. It just extracts the essence of text style of each chunk of text using stylometric features and then groups together the chunks that have same writing styles. This process is repeated for every new document.

In this report, first we elaborate our whole methodology including the features selection and data preprocessing. It also includes the machine learning method used. After that in order to demonstrate our approach an experiment is run on a document with two different writing styles. The results are explained and some limitations of our work are also presented. At the end we conclude our report.

Methodology:

First, the document is divided into chunks of text using a standard chunk size window. Then for each chunk of text a vector of stylometric features is computed. After that the chunks are clustered using their vectors of stylometric features. Thats where unsupervised machine learning comes into play. The chunks with same style are clustered together.

1) Data Set Selection:

We selected our data set from the internet. http://textfiles.com/stories/ is an online repository, which encompasses prodigious set of stories ranging from different authors to different difficulty levels. While we want to cluster different literary styles, we have used this data set for now to perform clustering. This dataset's main purpose is just demonstration of our approach. Our system can be run on any document.

2) Features Selection:

The heart of our system lies in feature extraction. We have to use such features which inherit the style of that text, hence we carefully crafted features for our project from the ones we studied during literature review. In order to distinguish a chunk of text on the basis of its literary style we first needed to define its writing style. A literary style spans a lot of things but we rather focused on three major ones: Lexical Features, Vocabulary Richness Features and Readability Scores. These include features like Shannon Entropy and Simpson's Index. Simpson’s index stems from the concept of biodiversity. We used that in our project as we wanted to measure the diversity of a text. We used python to code these features. Following is the list of features we have extracted:

Lexical Features:

1. Average Word Length
2. Average Sentence Length By Word
3. Average Sentence Length By Character
4. Special Character Count
5. Average Syllable per Word
6. Functional Words Count
7. Punctuation Count

These are the most basic features one can extract from the text. These features tell us about the structure of the text. For example averages of different counts like word lengths, special characters, punctuations and functional words etc. **Functional words** are used to express grammatical relationships among other words within a sentence. Secondly, if a word has more syllables then it is most likely to be a difficult word (although not necessary). **Avg Syllable per word** being the measure of complexity, is used in calculations of many other features related to readability scores described in the sections ahead. **Punctuation Count** and **Special Character Count** are straight forward ways to differentiate different genres. For example narrative story and research paper.

Vocabulary Richness Features:

Many quantitative studies rely on the concept of vocabulary richness. A text has low vocabulary richness if the same limited vocabulary is repeated over and over again, while it has high vocabulary richness if new words continually appear. In essence, these features tell us about the diversity and richness of the vocabulary used in the text.

1. Hapax Legomenon
2. Hapax DisLegemena
3. Honores R Measure
4. Sichel's Measure
5. Brunets Measure W
6. Yules Characteristic K
7. Shannon Entropy
8. Simpson's Index

Hapax Legomena and Hapax DisLegemena:

Hapax Legomena (sometimes abbreviated to *hapax*) is a word that occurs only once within a context, either in the written record of an entire language, in the works of an author, or in a single text. The term is sometimes incorrectly used to describe a word that occurs in just one of an author's works, but more than once in that particular work. *Hapax legomenon* is a Greek word meaning "(something) being said (only) once." Similarly, **Hapax DisLegemena** is the word that is used twice.

Now to further explain remaining features. we make use of the following notation:

- Tokens **N** length of text in words
- Types **V** number of different words in the text
- Hapax legomena **V1** number of words occurring just once in the text
- Dislegomena **V2** number of words occurring exactly twice in the text
- **Vi** number of words occurring exactly **i** times

The type / token ratio depends on the length of the text (being generally less for longer texts), but is a useful measure of vocabulary richness when the comparison texts are of equal length.

Honore's measure R [5] depends on the hapax legomena:

$$R = 100 * \log N / (1 - (V1 / V))$$

Sichel's measure S [6] depends on the dislegemena, and is relatively constant with respect to N:

$$S = V2 / V$$

Brunet's measure W is:

$$W = N^{v-a} ,$$

where a is a constant (usually 0.17). W was found to be relatively unaffected by text length and to be author specific [7].

Yule's characteristic K [8] depends on words of all frequencies:

$$K = 10{,}000 * (M - N) / (N*N) , \text{ where } M = \sum_{i}^{n} i^2 \; . \; V_i$$

Shannon Entropy

In general Entropy tells us the about the disorder in a system. We have used this concept in our project on text. Claude Shannon, the inventor of information theory gave this shannon entropy formula to measure the amount of information a word is giving

$$E = \sum_{i=0}^{N-1} P_i log\ P_i$$

P is the probability of word occuring in the passage.

Simpson's index

Simpson's Diversity Index is a measure of diversity. In ecology, it is often used to quantify the biodiversity of a habitat. It takes into account the number of species present, as well as the abundance of each species. Simpson's Index (D) measures the probability that two individuals randomly selected from a sample will belong to the same species (or some category other than species). This concept can be applied in NLP to find the diversity of a chunk of text. We have used this biodiversity concept as a feature in our project to find the diversity of different passages of texts..

$$\textbf{S Index (D)} = \sum (n / N^{2})$$

N = total number of words in a text.

n = total number of unique tokens

Readability Scores:

Readability is the ease with which a reader can understand a written text. Readability is more than simply legibility—which is a measure of how easily a reader can distinguish individual letters or characters from each other. Features for readability stems from the field of linguistics and researchers have frequently used linguistics' laws (e.g zipfs law) and lemmas to pull out the currently used features to calculate readability scores of text in the modern computer science. Following is the list of features we are using.

1. Flesch Reading Ease
2. Flesch-Kincaid Grade Level
3. Gunning Fog Index
4. Dale Chall Readability Formula
5. Shannon Entropy
6. Simpson's Index

Flesch Reading Ease was created in 1948 as a readability test [6]. The score on the test will tell us roughly what level of education someone will need to be able to easily read a piece of text. The Reading Ease formula generates a score between 1 and 100. (although it is possible to generate scores below and above this banding) A conversion table is then used to interpret this score. For example, a score of 70-80 is equivalent to school grade level 7. It should be fairly easy for the average adult to read. The Flesch Reading Ease test originated from research in the education sector. Teachers needed to choose texts appropriate to the reading level of their student. Yet, its use has always been far more wide-ranging.

$$\textbf{FR Score} = 206.835 - 1.015 \left(\frac{Total\ Words}{Total\ sentences} \right) - 84.6 \left(\frac{Total\ Syllables}{Total\ words} \right)$$

In the mid 70s, the US Navy were looking for a way of measuring the difficulty of technical manuals used in training. The Flesch Reading Ease test was revisited and, along with other readability tests, the formula was amended to be more suitable for use in the navy. The new calculation was named **Flesch-Kincaid Grade Level** [7]. Grade levels are based on the scores of participants in a trial group.

$$\textbf{FKG Level} = 0.39 * \left(\frac{Total\ Words}{Total\ sentences} \right) + 11.8 \left(\frac{Total\ Syllables}{Total\ words} \right) - 15.59$$

Gunning Fog index

In linguistics, the Gunning fog index is a readability test for English writing. The index estimates the years of formal education a person needs to understand the text on the first reading. For instance, a fog index of 12 requires the reading level of a United States high school senior (around 18 years old). The test was developed in 1952 by Robert Gunning, an American businessman who had been involved in newspaper and textbook publishing. The formula to calculate gunning fog index is given below.

$$\mathbf{G} = 0.4 * [(\frac{Words}{Sentences}) + 100 ((\frac{Complex\ Words}{Words}))]$$

"complex" words are those consisting of three or more syllables.

3) Data Pre-processing:

After downloading the data set from textfiles.com which consists of different text files of different authors and from different genres, we took a children story and a research paper from it for proof of concept. A document is then divided into small chunks. Here determining the size of chunk was a challenge for us. If it was too large then we won't be able to extract the crux of different passages. Had it been too small it would have lost the significance. Hence we went for the average i.e 10 sentences (it can be changed according to needs too) Now first of all the lexical features are computed for each chunk. Then for all other features we removed punctuations and special characters and performed tokenization (because lexical features use punctuations and special characters).

4) Machine Learning Algorithm:

As we are using unsupervised learning approach to cluster our data, we have used the most famous algorithm in this domain i.e. K-Means algorithm for our purpose.

5) PCA and Data Visualization:

As mentioned earlier, we have calculated almost 20 features. After that K-Means algorithm is run on the vectors of all chunks and centroids are identified. Now at this stage the number of centroids correspond to the number of different writing styles identified and this was what our system was meant to do but in order to visually see those clusters we had to convert our 20 dimensional vector into a 2D vector using Principal Component Analysis which extracted the essence from that 20D vector and converted it into a 2D vector. We then plot these vectors and color those chunks same which were grouped together under a centroid by K-Means. This way the chunks with different styles are visualized further strengthening our results.

Experimental Settings:

For the proof of concept, we chose two documents. One is a story named Jim(Story) while the other document is a research paper named AuthAttr(Paper). Now we merge these two documents into one and use that document for experiments. Since, this new document contains two different writing styles (one of a story and one of a research paper) hence, our system should essentially identify that this new document has two writing styles.

K - Means

We used K-Means algorithm to identify **K** different centroids in a text having different writing styles. Each centroid spans those chunks which have the same writing style. Hence the number of centroids correspond to the different number of writing styles that a document has.

Value of K:

We can choose the number of clusters by visually inspecting user data points first using their vector of stylometric features, but we soon realized that there is a lot of ambiguity in this process for all except the simplest data sets. This is not always bad, because we are doing unsupervised learning and there's some inherent subjectivity in the labeling process. Still tt is necessary to know the value of K before hand to run K-means effectively.

We used the ***Elbow method*** to find the optimal value of K.

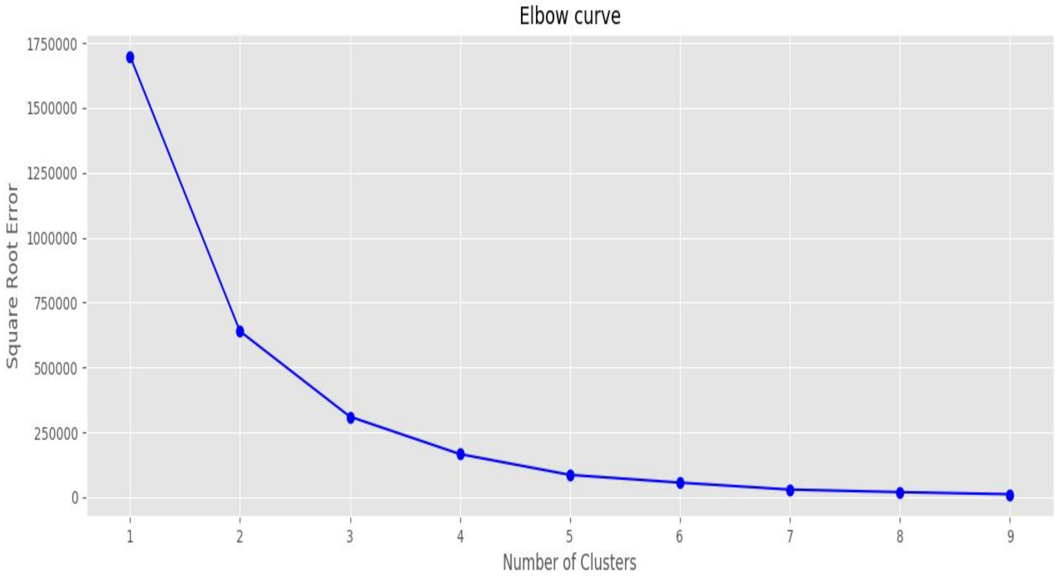
Elbow method:

The elbow method is described below:

First of all, compute the sum of squared error (SSE) for some values of k (for example 2, 4, 6, 8, etc.). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid. Mathematically:

$$\text{SSE} = \sum_{i=1}^{k} \sum_{x \in c_i} \text{dist}\,(x, c_i)^{\;2}$$

If we plot k against the SSE, we will see that *the error decreases as k gets larger*; this is because when the number of clusters increases, distortion gets smaller. The idea of the elbow method is to choose the k at which the SSE decreases abruptly. This produces an "elbow effect" in the graph, as can be seen in the following picture:



In this case, the most suitable value for K is k = 2**.**

Take into account that the Elbow method is an heuristic and, as such, it may or may not work well in usr particular case. Sometimes, there are more than one elbow, or no elbow at all. In those situations us usually end up calculating the best k by evaluating how well k-means performs in the context of the particular clustering problem us are trying to solve.

Parameter Tuning of K-Means:

We have used K-Means algorithm from sklearn library of python. At first we selected the value of K from Elbow method but there are other parameters whose values are very important to be well taken care of. After running multiple experiments we found out following parameter values to be best in our scenario

n_init:

As K means is heuristic based it depends upon the start seeds value of centroids we place at the start of the algorithm. It may stuck at local optima so We used **n_init value = 10.** It's basically randomly re initializes the centroids. So K-Means will be run **n_init**

Number of time with different centroid seeds. The final results will be the best output of **n_init** consecutive runs in terms of inertia.

Max_iter:

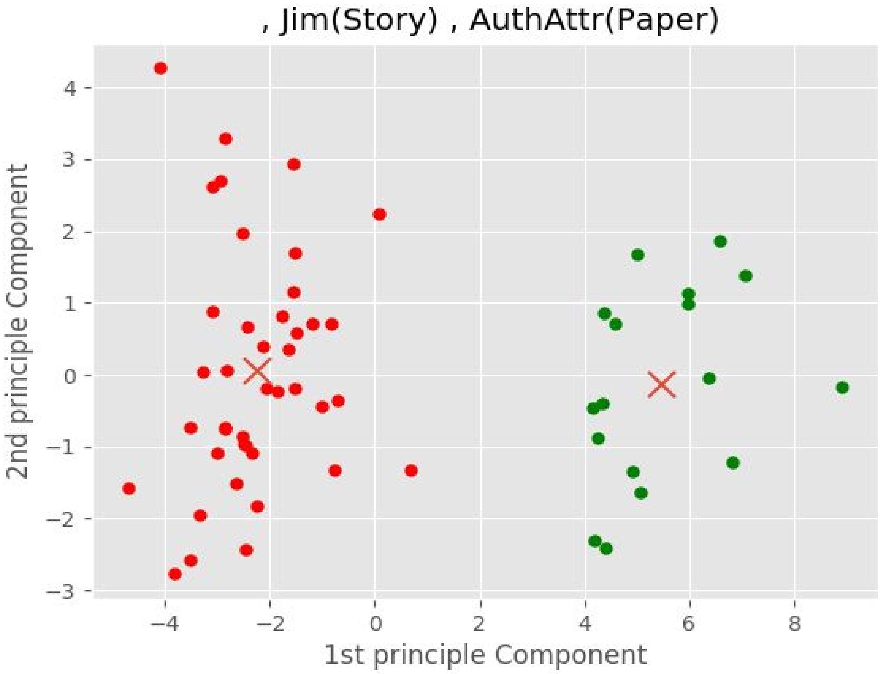
It is the maximum number of iterations K-Means algorithm for a single run. We used **max_iter = 500** With minimum tolerance for convergence.

n_Jobs:

The number of jobs to use for the computation. This works by computing each of the n_init runs in parallel. We have used **n_jobs = -1** Which will utilize all the cpu's available on the host machine.

Results:

We ran this experiment with a document file containing two different styles (one of a story and one of a research paper). And our system was correctly able to detect those two different writing styles and also color those chunks with the same writing style. The elbow method successfully returned K = 2 which was our goal.



A document containing texts with 2 different writing styles (a story and a research paper) are clearly distinguished indicating the correctness of our approach

Limitations:

As PCA smudges the higher dimensional vector into a 2D vector, so there is a high possibility of some loss of significance during the conversion process. Due to this there is a possibility that after plotting the clusters don't seem distinguishable as expected due to the loss of information while doing PCA. This is not as such a limitation to our system, since our system correctly identifies the number of writing styles through the value of K which is enough to proof the correctness of our approach but if and only if there is a need to visually see those clusters then it is only possible by converting the high dimensional vectors using PCA to a 2D one.

In a scenario where the document is written by a single author, one may assume that in this case there should be only one cluster implying one writing style. However, this is not the case every time. As we know each paragraph written even by the same author has a bit different writing style than its other counterparts like e.g there are some minor differences in lexical features (number of punctuation marks and others). Also there might be a poetry in paragraphs, mathematical equations etc. So our method will create clusters according to the writing style within that document. But the major difference in such a setting is the **distance between those clusters.** The more different the writing styles are the more far away the clusters are from each other. In the case of a document with the same writing style, even if some clusters are made, the distance between those clusters should be small as compared to the case if the document had really different writing styles. Hence this is the crucial essence which makes our algorithm excel because it not only identifies one part of the story i.e number of writing styles but also shows **how different those writings styles are from each other** which is implied by the distance between the clusters.

Conclusion:

To sum up:

1. Our system takes a document.
2. Divides it into chunks of 10 sentences.
3. Computes stylometric features for each chunk.
4. Then uses the elbow method on these vectors to identify the value of centroids K.
5. The value of K corresponds to the number of different writing styles the document had.

6. In order to visualize the clusters, PCA is used to convert the high dimensional features vector to a 2D one and then the chunks are plotted.
7. The chunks with same style are grouped under one centroid with same color, hence implying the number of writing styles implied in that document.

The heart of our approach is correctly extracting the style from the chunk which is successfully achieved using a mix of different categories of linguistic features like lexical, vocabulary richness and readability scores. Our method is repeated every time for a new document. Since tt identifies the different writing styles in that document, hence our approach can also be used to detect plagiarism.

References:

1. https://en.wikipedia.org/wiki/Stylometry
2. http://homepage.divms.uiowa.edu/~mshafiq/files/shehroze-text-spinner-icdm2017.pdf
3. https://www.uni-weimar.de/medien/webis/publications/papers/stein_2011a.pdf
4. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.440.1634&rep=rep1&type=pdf
5. Honore´, A. (1979), Some Simple Measures of Richness of Vocabulary. In: Association for Literary and Linguistic Computing Bulletin 7(2), 172177.
6. R. Flesch. A new readability yardstick. Journal of Applied Psychology, 32:221–233, 1948.
7. J. Kincaid, R.P. Fishburne, R.L. Rogers, and B.S. Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.
8. https://sites.google.com/site/parthochoudhury/aMToC_CShannon.pdf?attredirects=0