Poem author identification

First Author
Oproiu Matei

Second Author
Barbarasa Maria

Third Author
Ignat Mihaela

Abstract

In this paper, we will discuss two ways for determining the author of a poem. The purpose is to comprehend and investigate these strategies in order to advance knowledge in the field of natural language processing.

1 Introduction

In the field of natural language processing (NLP), authorship identification has been a fascinating challenge. For this project, we chose to work with Romanian poetry because it has a rich and expressive language, making it a great test case for this kind of analysis. We were curious to see how well different NLP techniques could distinguish between poets based on their writing patterns.

2 Related Work

The subject of authorship attribution has received extensive attention in the domains of natural language processing (NLP) and computational linguistics. This section presents a comparative study of two pertinent research in the field: Our course lecturer wrote a paper titled "Authorship Identification of Romanian Texts with Controversial Paternity".

Prof. Dinu's paper examines a specific literary debate in Romanian literature, namely whether Radu Albala replicated Caragiale's style in the sequel to his work. Their study uses Support Vector Machines (SVM) and string kernels to classify texts based on stylistic similarities. Furthermore, they present a novel classification-based approach that uses the frequency of functional words to determine authorship. This method ensures resilience by minimizing noise caused by text fluctuation. The dataset used consists of extended literature, including novels and novellas, that were hand-picked to ensure a balanced and high-quality corpus. Their method is tested using leave-one-out cross-validation, which yields accuracy scores of 62.56% and 50.56%, demonstrating that Albala closely resembled Caragiale's style, but with minor stylistic differences.

On the other hand, our work approaches authorship attribution from a broader NLP perspective, focusing on classifying Romanian poems by their authors. Our study employs two established techniques: TF-IDF with Cosine Similarity and Bag-of-Words with Multinomial Naive Bayes. Unlike our professor's work, their research does not introduce new methodological advancements but applies standard NLP methods to poetry classification. The dataset consists of Romanian poems automatically extracted from the Romanian Voice website using the ParseHub tool. While this method enables large-scale data collection, it may introduce inconsistencies and noise.

Our models achieve high accuracy: TF-IDF with Cosine Similarity reaches 90% accuracy for texts longer than 300 characters, whereas Bag-of-Words with Naive Bayes attains 80% accuracy. The data is split into 80% training and 20% testing sets, making it a more conventional machine learning setup compared to the leave-one-out validation used in our professor's work.

The two studies differ in several key aspects. The first paper analyzes prose while our work focuses on poetry. The former investigates a literary forensic case study, whereas the latter explores NLP-based classification for Romanian poets. Our professor's work uses a manually curated dataset, whereas our work rely on automatically extracted data. Methodologically, our professor's work proposes an original ranking method based on function words, whereas our applies widely used NLP techniques without introducing new methodologies. The validation approach also differs: our professor's work uses leave-one-out cross-validation, while ours split their data into training and testing sets.

Both investigations contribute to authorship identification, although for different reasons. Our professor's study addresses a specific literary case with a unique function-word ranking method, making it especially relevant to forensic linguistics and literary stylometry. In contrast our methods present a practical application of NLP classification algorithms for poetry, proving the efficacy of classical text categorization approaches in a literary context.

3 Method

In this project, we explore two distinct techniques for determining the authors of various Romanian poems. The first approach leverages a Bag-of-Words representation coupled with a Multinomial Naive Bayes classifier, using lexical frequencies to distinguish stylistic patterns among poets. The second method employs TF-IDF vectors and cosine similarity, thereby emphasizing the relative importance of words and measuring semantic closeness between documents.
We chose these methods primarily due to the limited availability of data. Romanian poets often have a relatively small number of published poems, which makes it challenging to train a large language model effectively. We chose a probabilistic approach, which performs well on small datasets by making strong assumptions about word distributions and a similarity-based method, which allows for comparisons between texts without extensive training data.

3.1 Dataset

The dataset was automatically extracted using the ParseHub application from the website Romanian Voice. The data was stored in json files with the authors name and the annotations were done automatically by the application.

3.2 Preprocessing

Initially, the json files contained the data in a very compact unreadable manner. The data was reformatted into line long blocks ensuring the file becomes human-readable:

```
Text:\n\nAuthor's   Name\n\n\n\n
Poem Title\n\n\n\n Poem
```

Then we applied a cleaning algorithm to remove unnecessary white spaces and new line elements. We were especially careful to preserve special Romanian characters.

3.3 Method

3.3.1 Method 1: TF-IDF + COSINE SIMILARITY

We start by extracting the poem text, title and author name from each json file and store them in a pandas DataFrame for efficient processing.
Before comparing texts, we convert poems into numerical vectors using the TF-IDF (Term Frequency-Inverse Document Frequency) method.
TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents, it weights words based on how unique they appear in different documents.

The TF-IDF score of a term t in a document d within a corpus D is given by:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t, D) \quad (1)$$

where:

- $\text{TF}(t, d)$ represents the Term Frequency, measuring how often term t appears in document d.
- $\text{IDF}(t, D)$ represents the Inverse Document Frequency, measuring how unique term t is across the entire document collection D.

Term Frequency (TF) measures how often a term t appears in a document d, normalized by the total number of words in the document. However, when using raw term counts, words that appear very frequently tend to dominate the representation,which may reduce the effectiveness of the model. To mitigate this, we use **logarithmic scaling**, which smooths the term frequency values and prevents overly frequent words from having an excessive impact.
The standart term frequency (TF) is calculated as:

$$\text{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in } d}{\text{Total number of words in } d} \quad (2)$$

The logarithmic scaling formula is calculated as:

$$\text{TF}_{\log}(t, d) = 1 + \log\left(\text{TF}(t, d)\right) \quad (3)$$

Inverse Document Frequency (IDF) measures the importance of a word in the entire corpus. Words that appear frequently across all documents receive lower weights, while words that are unique to certain documents receive higher weights.

$$\text{IDF}(t, D) = \log\left(\frac{N}{\text{DF}(t) + 1}\right) \tag{4}$$

where:

- N is the total number of documents in the corpus.
- $\text{DF}(t)$ is the number of documents containing term t.
- The "+1" in the denominator prevents division by zero.

In our implementation we dicided to use unigrams (single words) and bigrams (two-word phrases), we limited the vocabulary to 100,000 most important words, used logaritmic scaling for term frequency, and considered all words

Once all poems are converted into TF-IDF vectors, we compare a new poem against the dataset using cosine similarity.

Cosine Similarity calculates the angle between two vectors in an n-dimensional space. The closer the angle is to zero, the more similat the vectors (documents) are. The formula for cosine similarity between two vectors A and B is:

$$\text{cosine\_sim}(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} \tag{5}$$

where:

- $A \cdot B$ is the **dot product** of the vectors.
- $\|A\|$ and $\|B\|$ are the **Euclidean norms** (magnitudes) of the vectors.

Expanding the formula:

$$\text{cosine\_sim}(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{6}$$

where:

- A_i and B_i represent the TF-IDF values of the i^{th} word in documents A and B.
- n is the total number of unique words in the vocabulary.

The combination of TF-IDF and Cosine Similarity provides an efficient and interpretable method for authorship identification. Despite the limitations of a small dataset, this method proves to be highly effective, consistently achieving an accuracy of over 90% when tested on randomly selected text samples of at least 300 characters from various poems. However, for significantly shorter section, the method struggles to establish meaningful connections, as the reduced number of words limits the ability of Cosine Similarity to capture distinct stylistic patterns and author-specific language.

3.3.2 Method 2: Bag-of-Words + Multinomial Naive Bayes

For our second method, we employ a Bag-of-Words (BoW) representation in combination with a Multinomial Naive Bayes (MNB) classifier. This probabilistic model classifies Romanian poems based on word frequency distributions.

We begin by extracting the poem text, title, and author name from each JSON file and storing them in a Pandas DataFrame for structured processing. The dataset consists of multiple authors, each associated with a collection of poems.

Step 0: Preprocessing in Romanian Before building the vocabulary, we preprocess all poem texts to standardize and clean the data. This includes:

- Lowercasing all characters;
- Removing non-alphanumeric characters using regular expressions;
- Eliminating Romanian stopwords using the `nltk.corpus.stopwords` list;
- Applying stemming with a Romanian SnowballStemmer to reduce words to their root forms.

This step ensures that word forms such as *"cântăreț", "cântarea"* are stemmed to a common root, reducing sparsity in the feature vectors.

Step 0.5: Filtering and Vocabulary Pruning To improve model reliability, we discard authors who have fewer than two poems in the dataset. This avoids overfitting on underrepresented classes.

Additionally, we compute the global frequency of all tokens and retain only those that appear more than once. This vocabulary pruning removes rare or unique words that do not generalize well and would otherwise introduce noise.

$$\text{Filtered Vocabulary} = \{w_i \in \text{Vocabulary} \mid \text{count}(w_i) > 1\} \tag{7}$$

This results in a more compact and robust feature space.

Step 1: Building the Vocabulary To create a numerical representation of poems, we first construct a vocabulary set containing all unique words across the corpus. The vocabulary is used to encode each poem as a feature vector, where each dimension corresponds to a word in the dataset.

$$\text{Vocabulary} = \{w_1, w_2, \ldots, w_N\} \tag{8}$$

where N represents the total number of unique words across all poems.

Step 2: Computing the BoW Vectors Each poem is tokenized and represented as a Bag-of-Words frequency vector, where the value at each position represents the number of occurrences of the corresponding word in the poem:

$$\text{BoW}(d) = [f(w_1, d), f(w_2, d), \ldots, f(w_N, d)] \tag{9}$$

where:

- $f(w_i, d)$ represents the frequency of word w_i in document d.

For example, given a small dataset with three poets (Mihai Eminescu, George Bacovia, Alexandru Macedonski), a simplified vocabulary and the corresponding BoW representations for three sample poems are:

Poem	lumina	cer	soare	mare
Luceafărul	2	1	0	0
Plumb	0	0	1	2
Noapte de Mai	1	0	1	1

Table 1: Example of Bag-of-Words representation for three poems.

Step 3: Transforming Data for Classification The BoW vectors are then converted into a sparse matrix format using **DictVectorizer**, ensuring efficient memory usage for large vocabularies. The transformed dataset is split into 80% training and 20% testing sets.

Step 4: Applying Multinomial Naive Bayes We apply the Multinomial Naive Bayes classifier, a probabilistic model designed for discrete features such as word counts. This model calculates the probability of a poem belonging to an author based on the occurrence frequencies of words.

Using Bayes' theorem, the probability of a poem d belonging to an author C_k is given by:

$$P(C_k|d) = \frac{P(d|C_k)P(C_k)}{P(d)} \tag{10}$$

where:

- $P(C_k|d)$ is the probability that document d belongs to author C_k.
- $P(d|C_k)$ is the likelihood of observing document d given class C_k.
- $P(C_k)$ is the prior probability of class C_k.
- $P(d)$ is the probability of document d occurring in the dataset.

Step 5: Prediction and Evaluation After training, the model predicts the author of unseen poems and assigns probability scores to each author based on word distributions. The accuracy of the model is evaluated using:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \tag{11}$$

Our BoW-based classifier achieves an accuracy of over 80% on the test data, demonstrating its effectiveness in capturing distinct linguistic patterns of Romanian poets.

Step 6: Handling New Poems For a new poem fragment, the model follows these steps:

1. Tokenizes and converts the poem into a BoW vector.
2. Transforms the BoW vector into a numerical representation using the trained vectorizer.
3. Computes probability scores for each author.
4. Assigns the poem to the author with the highest probability.

Step 7: Saving Prediction Results Once the model predicts the author of a new poem, it outputs not only the predicted class but also a vector of class probabilities. These results are stored in a CSV file using the Pandas library.

Observations and Limitations: While BoW is an efficient and interpretable method, it has some limitations:

- It ignores **word order**, meaning that "the night is dark" and "dark is the night" have the same representation.
- It treats **all words equally**, without considering their importance (which TF-IDF addresses).
- It is susceptible to **out-of-vocabulary (OOV) words**, meaning it cannot handle new words outside the training set.

Despite these limitations, the BoW + Multinomial Naive Bayes model performs well in classifying Romanian poetry authors, particularly for datasets with limited text samples.

4 Future Work

We plan on continuing working, the next steps could involve experimenting with deep learning models such as Word2Vec, FastText, or Transformer-based models like BERT to capture more complex linguistic patterns and increase the dataset to capture more romanian poets and their unique style. This could be used in plagiarism detection to spot if someone is copying one of the great Romanian poets.

5 Conclusion

What have you learned from this project? What did you like and what did you hate?

With the addition of both BoW + Multinomial Naive Bayes model and TF-IDF + Cosine Similarity, this project has become more versatile and accurate. It provides a strong foundation for text classification.

Both BoW + Multinomial Naive Bayes model and TF-IDF + Cosine Similarity provide high classification accuracies for this short dataset, and because they are rapid approaches, they can quickly determine whose author the poem belongs to. They are an excellent starting point for learning about Natural Language Processing.

Limitations

Our approaches work quickly and efficiently for any language, with few adjustments required because each language has its own set of special circumstances. The approaches are scalable, with the requirement that the Poets/Authors have a comparable amount of poems, as introducing a Poet with a large number of poems brings bias into the process.

Ethical Statement

What unethical uses could there be for your research? Considering the nature of your project, do you think it's possible to have included any biases? How?

Did you take any measures to prevent / combat bias? How do you recommend others use your research to avoid them? What is your personal opinion on the matter?

We took all necessary precautions to ensure that our process is free of biases, and we chose only poets with more than 20 poems to ensure that we can extract each one's unique style.

References

Anca Dinu Liviu Petrisor Dinu, Marius Popescu. 2008. Authorship identification of romanian texts with controversial paternity.

- (Liviu Petrisor Dinu, 2008)
- Poem Website: https://poezii.romanianvoice.com/index.php
- Webscraping Tool: https://www.parsehub.com/
- TF-IDF 1:https://en.wikipedia.org/wiki/Tf
- TF-IDF 2:https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/
- Cosine Similairty 1: https://en.wikipedia.org/wiki/Cosine_similaritydocument-frequency/
- Cosine Similarity 2: https://www.geeksforgeeks.org/cosine-similarity/
- Bag of Words: https://www.datacamp.com/tutorial/python-bag-of-words-model
- Multinomial Naive Bayes: https://www.geeksforgeeks.org/applying-multinomial-naive-bayes-to-nlp-problems/