Initial Explorations for Document Clustering Tasks in Latin Elegiac Poets[1]

Carlos Javier Nusch [1[0000-0003-1715-4228]] Gimena del Rio Riande [2[0000-0002-8997-5415]] Leticia Cecilia Cagnina [3[0000-0001-7825-2927]] Marcelo Luis Errecalde [3[0000-0001-5605-8963]] Leandro Antonelli [4,5[0000-0003-1388-0337]]

[1] PREBI, SEDICI, Universidad Nacional de La Plata, Argentina
carlosnusch@prebi.unlp.edu.ar
[2]IIBICRIT, Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
gdelrio@conicet.gov.ar
[3] LIDIC, Facultad de Ciencias Físico Matemáticas y Naturales, Universidad Nacional de San Luis, Argentina
lcagnina@unsl.edu.ar
merreca@email.unsl.edu.ar
[4] LIFIA, Facultad de Informática, Universidad Nacional de La Plata, Argentina
[5] CAETI, Facultad de Tecnología Informática, Universidad Abierta Interamericana, Argentina
lanto@lifia.info.unlp.edu.ar

Abstract. This article describes various Automatic Text Analysis tasks applying Natural Language Processing techniques on a corpus of Latin texts from the 1st century BC and 1st century AD. The motivation behind this work is to delve into and understand a historical literary trend revolving around the themes of love, spanning from antiquity through to the medieval period. The analyzed authors include Gaius Valerius Catullus, Albius Tibullus, and Sextus Propertius, who represent the literary movement of the neoterics, as a group of poets to be identified, and Publius Vergilius Maro and Marcus Annaeus Lucanus, epic poets with remarkably distinct styles, as control samples. The purpose of this preliminary and exploratory study is to investigate the potential and best features for document clustering. The clustering tasks were carried out using fixed ranges of character n-grams and word n-grams. For the clustering tasks, the K-Means method and the Silhouette Index were used for determining the optimal cluster sizes. Using optimal clusters as labels, decision trees were trained for each range of n-grams, aiming to identify features with the highest Information Gain and Information Gain Ratio. The trees were trained based on the criterion of Entropy, and calculations of Feature Importance were performed. Results show variations based on text preprocessing techniques: simple filtering of stopwords in the corpus yields better Silhouette scores, with one or two features showing potential classification value for the decision trees. The application of TF-IDF weighting results in Silhouette indices closer to zero, albeit with a more balanced distribution of Importance among different features.

Keywords: Latin Elegiac Poets, Document Clustering, K Means, Silhouette Coefficient, Decision Trees, Feature Importance, Information Gain Ratio.

---

[1] **Note**: This preprint has not undergone peer review or any post-submission improvements or corrections. The Version of Record of this contribution is published in *DECISIONING 2024: Collaboration in Knowledge Discovery and Decision Making*, CCIS, vol. 2019. Springer, Cham. https://doi.org/10.1007/978-3-031-91690-8_10. The NLP techniques and tasks applied here were further improved and also presented in: Nusch, C. J., Del Rio Riande, M. G., Cagnina, L., Errecalde, M. L., & Antonelli, R. L. (2024, November). *Clustering Tasks and Decision Trees with Augustan Love Poets: Cohesion and Separation in Feature Importance Extraction.* In *CEUR Workshop Proceedings* (Vol. 3834). https://sedici.unlp.edu.ar/handle/10915/175050.

1 Introduction

This study builds upon issues previously addressed in individual bachelor's and master's thesis works [19] considerations made by C. S. Lewis [12] on the influence of Courtly love and Occitan literature on the love imaginary of the 20th century. It was timely noted that there were some remarkable similarities between love themes, the treatment of the beloved, and certain terms derived from the political and military fields in erotic poems, both in Occitan and Latin literature of the 1st century BC. The overarching framework of this doctoral thesis is centered on attempting to identify textual patterns that could shed light on a potential literary tradition of love themes originating in antiquity and culminating in the development of the imagery of the "Religion of Love" by medieval Occitan poets. To narrow down the analysis corpus and circumvent the challenges inherent in working with vernacular languages, the research focused on texts from the Latin literary tradition. The comparative approach will be conducted by combining traditional close reading techniques with the capabilities of distant reading tools provided by computational methods [16, 23]. In the initial stages of the research, various document clustering techniques are being analyzed to uncover features that allow profiling and identifying poetry with love themes. For this specific article, various clustering techniques were evaluated to differentiate elegiac poems from other types of Latin poetry and to discover lexical features that could prove useful in document classification tasks. This study contributes to the field by testing the efficacy of clustering algorithms on a specific corpus of Latin poetry.

2 State of the Art

In the field of ancient text studies, several authors have applied clustering techniques to ancient texts, including Bracco et al. [2] worked on the automatic detection of literary genres in cuneiform texts using the K-means algorithm to group texts based on grammatical, graphemic, and other stylistic features. Regarding specifically Latin texts, Martins et al. [14] used the k-Nearest Neighbors (k-NN) approach in a multi-author classification task of the *Historia Augusta*. Cantaluppi and Passarotti [5] conducted an in-depth investigation of the corpus of Seneca's *opera omnia* using hierarchical cluster analysis and latent semantic analysis, comparing it with the corpora of Cicero's *orationes*, Jerome's Latin New Testament (*Vulgata*), and Thomas Aquinas's *opera maiora*. The authors demonstrated that a lexical-based approach is successful in classifying texts across different authors, genres, and eras. About the authors studied in this work, B. Nagy [18] used clustering and multivariate analysis to examine the conscious use of rhyme in twelve classical Latin poets (including Catullus, Propertius, and Tibullus), identifying stylistic differences between genres and authors. Forstall et al. [8, 9] have compared lexical and rhythmic characteristics at the level of character and word n-grams with other poets from the 1st century BC and later.

3 Problem Definition and Contributions

The primary aim of this work was to explore clustering techniques capable of distinguishing elegiac poems from other types of poetry and to determine which lexical characteristics might be useful in document classification tasks. The K-means algorithm [13] was selected for this purpose, and the optimal number of clusters was evaluated using the Silhouette Index [24], which evaluates the cohesion of each grouping and its distinctiveness from the other clusters. Given that the K-means method is based on Euclidean distance and, apart from identifying the points or documents closest to the centroid, does not allow for the extraction of detailed information about the features of the documents that comprise each cluster, it was decided to complement this approach with decision trees[22]. This combination allowed for the indirect extraction of the features. In this way, an effort was made to obtain the features with the greatest classification potential indirectly using the metrics of importance, information gain, and information gain ratio [26].

4 Research Methodology and Approach

4.1 Analysis Corpus and Used Editions

The working corpus includes the complete works of Gaius Valerius Catullus [15], Albius Tibullus [21], and Sextus Propertius [17] as the authors of love-themed poetry whose particular characteristics are sought to be identified. The various cantos from the epic poems *Aeneid* by Publius Vergilius Maro [10] and *Pharsalia* by Marcus Annaeus Lucanus [28] were used as control samples since their themes focus on political, historical, and martial matters. The analysis of the works of five Latin authors reveals differences in their use of words and verse structure. Catullus, with a total of 2,289 verses[2]and 12,887 words, has an average of 110.15 words per poem and 5,821 unique words. Tibullus wrote 1,930 verses and 12,705 words, with a rather high average of 343.38 words per poem and 5,328 unique words. Propertius, with 4,008 verses and 25,320 words, maintains an average of 241.14 words per poem and a repertoire of 9,021 unique words. Lucan stands out with 8,061 verses and 51,065 words, with an average of 5,106.50 words per canto and 14,766 unique words. Finally, Virgil leads in volume with 9,896 verses and 63,719 words, and an average of 5,309.92 words per canto, in addition to the widest variety of unique words, totaling 16,619.

To construct the analysis corpus, resources from the Perseus Project digital library [6, 29] at the Department of Classics at Tufts University were utilized. The

[2] In the Merrill edition of *Catullus*, the canonical total of 2296 verses is proposed by the editor, who notably includes 7 missing verses (vv. 79-82 and 112-114) in *Carmen* 61. I would also like to express my gratitude to Professor Benjamin Nagy of the Institute of Polish Language, Polish Academy of Sciences (IJP PAN), Kraków, Poland, for his invaluable advice and assistance in correcting the verse count in line with the authoritative editions of the studied authors. His expertise greatly contributed to ensuring accuracy in this text mining endeavor.

library also houses 2,412 works in 3,192 editions and translations (1,639 in Greek and 636 in Latin) and 69.7 million words: 32.1 million in Greek, 16.3 million in Latin, and the remainder in English translations and sources in other languages. The texts are carefully curated by specialists, and shared under a CC BY SA 3.0 (US) license, with the texts being available in XML format for download. Additionally, the project offers additional resources, including a model for grammatical tagging or Parts of Speech Tagging (PoS Tagging) and stopwords for Latin. The poems that constitute the corpus were harvested for previous Automatic Text Analysis tasks through web scraping procedures. The software used for performing the scraping was R. On the other hand, for the text analysis and mining, particular libraries of Python were used. R was utilized as the software for the scraping, while Python libraries were employed for text analysis and mining purposes.

In this study, we explored the characteristics of n-grams at different levels, including character n-grams (ranging from 2 to 7) and word n-grams (from 1 to 5). For our analysis, we employed the traditional Bag of Words (BOW)[25] method. We generated three types of matrices: the first matrix was created using Scikit-learn's module CountVectorizer, which tallies the frequency of simple terms and enables filtering common words or 'stopwords'. The second matrix used the TF-IDF (Term Frequency-Inverse Document Frequency) [27] weighting technique, which highlights important words by weighing their frequency relative to their rarity across the entire dataset. Finally, the third matrix combined TF-IDF weighting with 'stopwords' filtering, providing a more refined approach by excluding less significant terms while highlighting the importance of the remaining terms.

The difference between a matrix generated with CountVectorizer and one with TF-IDF is that the former simply counts the frequency of each word's occurrence in the documents, resulting in a direct numerical representation of the terms, whereas the latter considers not only the frequency of words but also their relative importance across the document set. TF-IDF weights common words that appear in many documents so that they have less impact, highlighting terms that are more unique to each document and therefore, potentially more significant for analysis.

4.2 Text Preprocessing Tasks

Before proceeding with the text analysis, cleaning and text preprocessing tasks were carried out. Consecutive empty lines and combinations of new lines and spaces in different sequences ("\n \n\n\n' and variants) that did not provide information were removed, as they were the remnants of scraping tasks. Furthermore, signs often used by editors to denote illegible gaps in the original codices (“†”) were also removed due to the problems these signs caused. Spanish double (“”) and single (‘’) quotation marks were replaced with English ones ("", ") to ensure compatibility with tools developed for English, thus preventing potential processing errors.

For working with character n-grams, punctuation marks were removed since they are an addition by the editors of the ancient texts and it is not possible to ascertain the original punctuation of the authors, given that modern punctuation did not exist in the 1st century BC.

Regarding stopwords, the package designed for Latin from Stopwords ISO [31], a project compiling a collection of stopwords in various languages, was used. The interest in using these particular stopwords over others like the Perseus Project [30] was motivated by the fact that they do not remove some words previously identified as important within the elegiac style, such as *ego*, thus allowing for the exploration of the prevalence of personal pronouns among the most significant features of the documents, a phenomenon already noted in previous works [19, 20].

5 Evaluation: Silhouette and Optimal Number of Clusters

5.1 Clustering with K Means and Silhouette Index

To conduct document clustering, the K-means method was used, enabling the partitioning of a dataset consisting of *n* observations into K different groups based on feature similarity. One of the limitations of the algorithm lies in its prerequisite knowledge of the number of K groups which the corpus will be distributed into. To reduce/mitigate bias in determining the K groups, the Silhouette calculation was used as it allows evaluating two important parameters: firstly, the cohesion within each cluster, i.e., the average distance of all the points that make it up among themselves, and secondly, the separation between the different clusters according to the distance of one point to all other points in the nearest cluster. The optimal k for clusters was determined by testing a range from 2 to 20 clusters and the k groups exhibiting the best coefficient were selected.

5.2 Character N-grams and Word N-grams

Tests were conducted utilizing different fixed ranges of character n-grams (from 2 to 7) and word n-grams (from 1 to 5). For each of these ranges, the Silhouette coefficient was calculated for k groups ranging from 2 to 20. This endeavor aimed to discern the optimal range of character and word n-grams for clustering tasks. The goal was to

find not only the best k for clusters but also the most effective n-gram ranges for optimizing the grouping process.

5.3 TF-IDF Weighting

Term Frequency-Inverse Document Frequency (TF-IDF) [27] a statistical technique, was used to evaluate the importance of a word in a document relative to a collection of documents or corpus. This technique weights a term's frequency of appearance in a specific document relative to the total number of terms in the document and the term's significance across the entire corpus. The calculation within the corpus helps to mitigate the impact of terms ubiquitous across multiple documents and are therefore not useful for classification tasks.

5.4 Decision Trees and Key Corpus Features

With data labeled by clusters, decision trees were trained using the entropy criterion to analyze the importance of different features across various n-gram levels. The importance of features was calculated, and additionally, Information Gain (IG) and Information Gain Ratio (IGR) were determined.

The importance of a feature reflects the extent to which that specific feature contributes to the improvement of the model's predictive capacity, based on its efficacy to help reduce impurity or disorder in the data set. Entropy measures the impurity or disorder in a dataset, calculated as the sum of the products of the probabilities of the labels and their logarithms. Information Gain is obtained by reducing the total entropy with the entropy weighted by the presence or absence of each feature. The Information Gain Ratio (IGR) is calculated on the entropy of the labels corresponding to each feature where the feature is present, obtaining the information gain by subtracting the feature's entropy from the total entropy.

6 Preliminary or Intermediate Results

Regarding the calculation of the optimal number of clusters, there were no major differences between using the two types of stopwords. It should be noted that a better performance in terms of the Silhouette index was observed when using the frequency matrix and simple stopwords filtering (CountVectorizer with Stopwords). The matrix resulting from the TF-IDF weighting shows Silhouette scores close to zero, indicating inadequate cluster separation and potential overlap.. In the case of 5-character n-grams, the negative value even suggests the likelihood of misassigned elements in the clusters (Table 1).

Table 1. Optimal clusters and corresponding Silhouette values for different ranges of n-grams.

Optimal Clusters (Stopwords ISO)				
N-gram Type	CountVec with Stopwords: Clusters	Score	TF-IDF: Clusters	Score

Char 2-grams	**2**	**0.94**	**2**	**0.16**
Char 3-grams	2	0.91	2	0.12
Char 4-grams	2	0.88	2	0.029
Char 5-grams	2	0.81	17	-0.00026
Char 6-grams	3	0.76	14	0.004
Char 7-grams	2	0.72	16	0.0025
Word 1-grams	**3**	**0.77**	**3**	**0.012**
Word 2-grams	2	0.704	17	0.0013
Word 3-grams	2	0.69	15	0.0075
Word 4-grams	2	0.69	17	0.0076
Word 5-grams	2	0.69	19	0.0074

Regarding the features deemed most crucial for classifying different types of clusters, the results obtained may suggest that it might be necessary to reevaluate the methodology and resources used. Although the Silhouette scores were high using the frequency matrix, when calculating the different metrics to evaluate the importance of the features, a very uneven distribution appears (Tables 2 and 3). In almost all cases, one or two attributes concentrate all the importance. Applying the TF-IDF technique, on the other hand, while the Silhouette scores had been close to zero, a greater number of features contributing information for the classification task can be observed (Tables 4 and 5).

Table 2. Most important features at the level of character n-grams according to Importance, Information Gain, Information Gain Ratio using the frequency matrix method and Stopwords filtering. Source: author's own work.

CountVectorizer with Stopwords. Char n-grams (2,2)					
Feature	Importance	Feature	IG	Feature	IGR
"a "	1	"gm"	0.206	"dh"	0.503
"us"	0	"dh"	0.1928	"gm"	0.4631
"ep"	0	"ze"	0.178	"dg"	0.4498
"ea"	0	"rh"	0.1738	" x"	0.4182
"eb"	0	"dv"	0.1686	"ae"	0.4182
"ec"	0	"yc"	0.166	"oï"	0.402
"ed"	0	"df"	0.1651	"sn"	0.392
"ee"	0	"lm"	0.1636	"ze"	0.39

"ef"	0	"ya"	0.1614	"mf"	0.3883
"eg"	0	"uq"	0.1614	"gg"	0.3873

Table 3. Most important features at the level of word n-grams according to Importance, Information Gain, Information Gain Ratio using the frequency matrix method and Stopwords filtering. Source: author's own work.

CountVectorizer with Stopwords. Word n-grams (1,1)					
Feature	Importance	Feature	IG	Feature	IGR
"iam"	0.8358	"fatis"	0.2599	"mundo"	0.6931
"omnipotens"	0.1642	"late"	0.2397	"bellorum"	0.6931
"flava"	0	"hos"	0.2285	"aeneas"	0.6931
"flammigeros"	0	"fatur"	0.2174	"teucrum"	0.6931
"flammis"	0	"metu"	0.2151	"divom"	0.6418
"flammisque"	0	"iamque"	0.2151	"teucros"	0.6418
"flamma"	0	"cursu"	0.2144	"civile"	0.6366
"flare"	0	"haud"	0.2127	"coelo"	0.6366
"flatibus"	0	"urbem"	0.2089	"caussa"	0.6366
"flatu"	0	"vires"	0.2082	"nocentes"	0.6366

Table 4. Most important features at the level of character n-grams ranked by Importance, Information Gain, Information Gain Ratio using the TF-IDF matrix method. Source: author's own work.

TF-IDF. Char n-grams (2,2)					
Feature	Importance	Feature	IG	Feature	IGR
"a "	0.4611	"rm"	0.2306	bh"	0.3652
"s "	0.1504	"fu"	0.2019	"rm"	0.2311
"ri"	0.0608	" t"	0.1904	"ta"	0.2306
" e"	0.0456	"fo"	0.1802	"ct"	0.229
"xi"	0.0441	"ct"	0.1801	"fu"	0.2231
"st"	0.0432	"aq"	0.179	"co"	0.2224
"lu"	0.0419	" e"	0.1786	"to"	0.2215
"or"	0.0374	"go"	0.1738	"ro"	0.2079
"m "	0.0338	"ph"	0.1729	" e"	0.203

"mu"	0.0243	"rb"	0.1723	"no"	0.2018

Table 5. Most important features at the level of character n-grams according to Importance, Information Gain, Information Gain Ratio using the TF-IDF matrix method. Source: author's own work.

TF-IDF. Word n-grams (1,1)					
Feature	Importance	Feature	IG	Feature	IGR
"et"	0.4448	"hic"	0.1912	"spes"	0.3198
"opus"	0.1748	"ab"	0.1818	"belli"	0.3042
"ille"	0.0894	"signa"	0.1744	"acies"	0.299
"per"	0.0797	"manus"	0.1736	"labor"	0.2938
"liquor"	0.0476	"per"	0.1698	"fatis"	0.2938
"turpis"	0.0433	"ad"	0.1637	"marte"	0.2886
"iam"	0.0407	"arma"	0.1586	"late"	0.2886
"altera"	0.0322	"tellus"	0.1548	"tellus"	0.2855
"fugaci"	0.0269	"ubi"	0.1526	"gentis"	0.2834
"classe"	0.0207	"spes"	0.1496	"iuventus"	0.2834

Regarding a qualitative reading of the results, assessing the relevance of specific character n-grams for text classification proves challenging. This analysis would require a more detailed stylistic study of the preferences of each author.

Concluding with the findings, although at the level of features the techniques used did not yield terms predominantly typical of the elegiac love repertoire, the division of documents into clusters seems to have shown efficacy in certain instances. With the Bag of Words technique and frequency counting at the level of 2 character n-grams, the 2 resulting clusters separate the elegiac poets from the epic poets (Figs. 1 and 2).

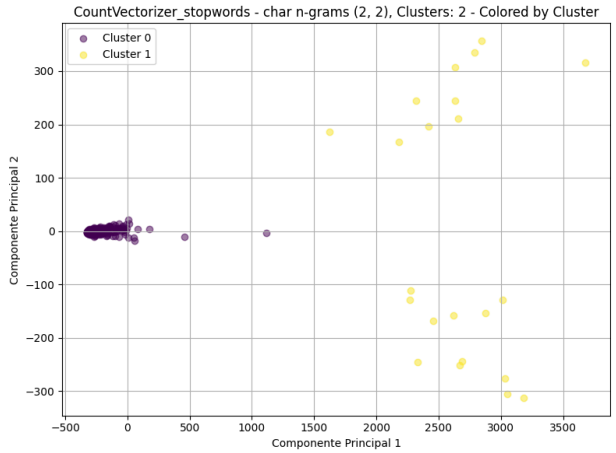

Fig. 1. Scatter plot of clustering by K Means using a frequency matrix of 2 character n-grams indicating the clusters with different colors. Source: author's own work.

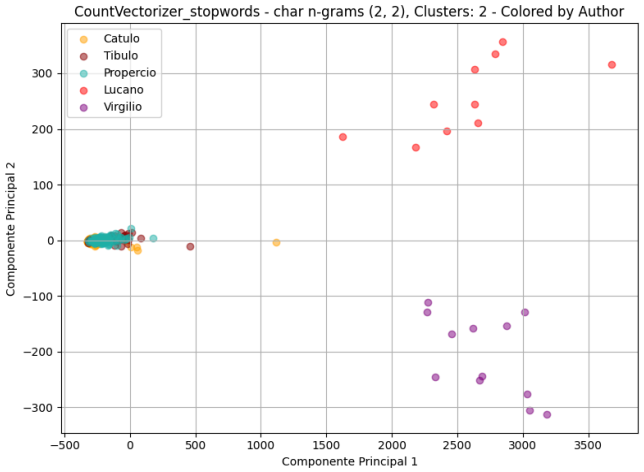

Fig. 2. Scatter plot of clustering by K Means using a frequency matrix of 2 character n-grams indicating the authors with different colors. Source: author's own work.

Applying the same technique but at the level of word n-grams, produced 3 clusters separating the elegiac poets on one side and Virgil and Lucan into two other clearly differentiated clusters.

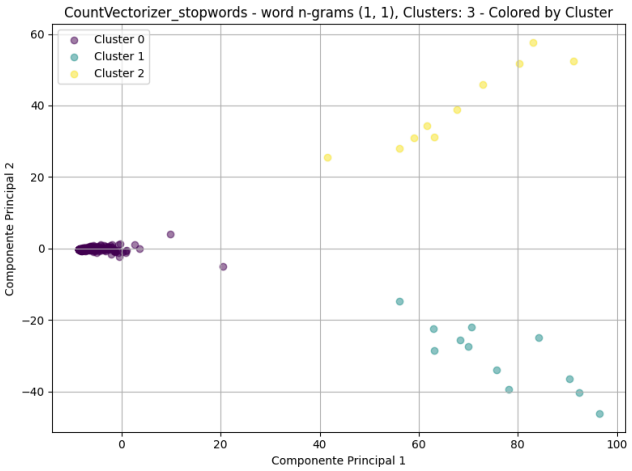

Fig. 3. Scatter plot of clustering by K Means using a frequency matrix of 1 word n-gram indicating the clusters with different colors. Source: author's own work.

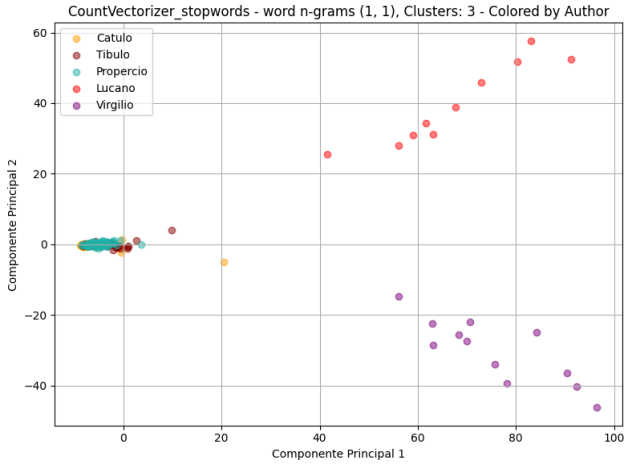

Fig. 4. Scatter plot of clustering by K Means using a frequency matrix of 1 word n-gram indicating the authors with different colors. Source: author's own work.

The use of a TF-IDF matrix both at the character and word levels produced scatter plots where the points (documents) were more dispersed across the plane. Specifically, at the level of 2 character n-grams, the poems of the epic authors are near those of Tibullus and Propertius, but occupy a distinct sector within the cluster. Conversely, the majority of Catullus's poetry is separated into a much less cohesive cluster (Figs. 5 and 6). At the level of word n-grams, the poetry of the epic authors coalesced into one cluster, and that of Tibullus and Propertius formed another, albeit in close proximity to a third cluster encompassing nearly all of Catullus's poetry along

with some texts from Tibullus. An interesting avenue for future investigation would be to explore why, using this technique, these documents from Tibullus appear to bear such a degree of similarity to those of Catullus.

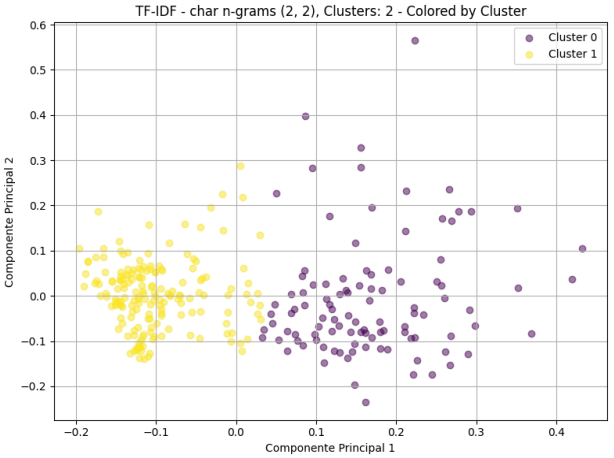

Fig. 5. Scatter plot of clustering by K Means using a TF-IDF matrix of 2 character n-grams indicating the clusters with different colors. Source: author's own work.

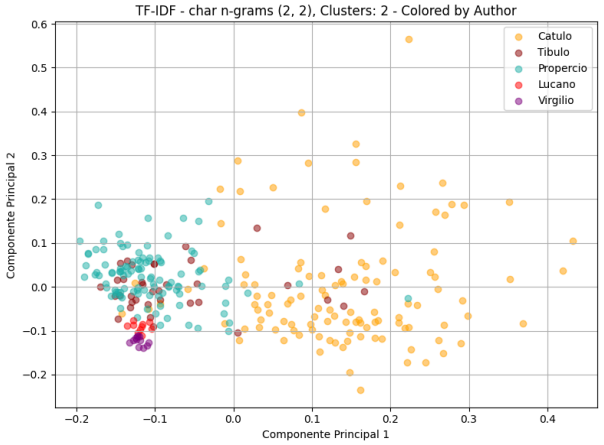

Fig. 6. Scatter plot of clustering by K Means using a TF-IDF matrix of 2 character n-grams indicating the authors with different colors. Source: author's own work.

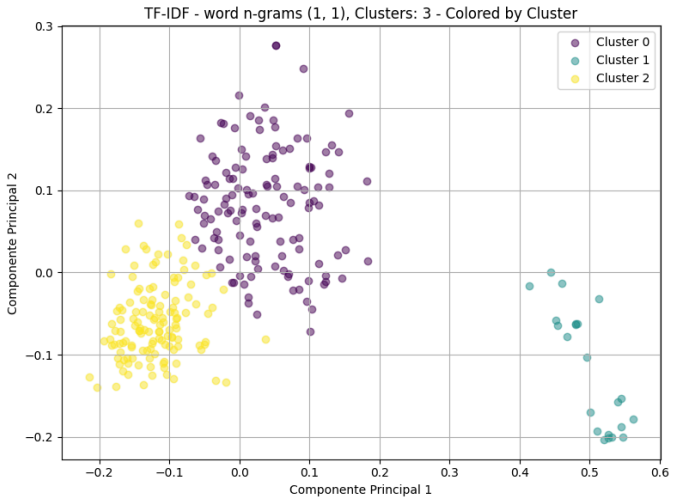

Fig. 7. Scatter plot of clustering by K Means using a TF-IDF matrix of 1 word n-gram indicating the clusters with different colors. Source: author's own work.

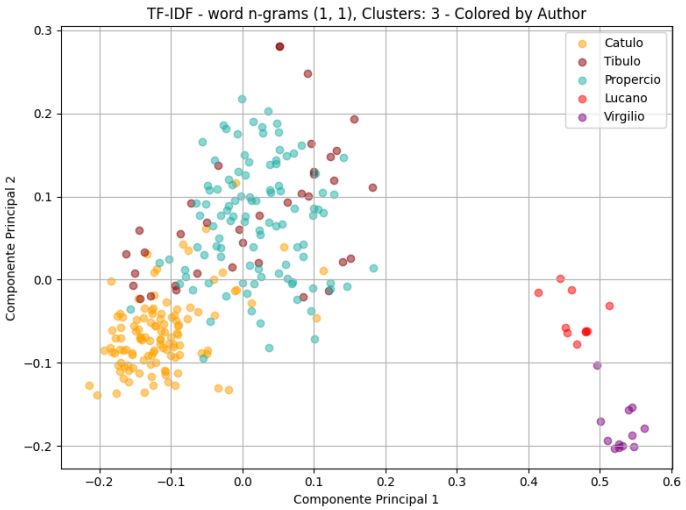

Fig. 8. Scatter plot of clustering by K Means using a TF-IDF matrix of 1 word n-gram indicating the authors with different colors. Source: author's own work.

7 Conclusions and Learned Lessons

Tests were conducted using two different types of stopwords, with negligible disparities observed for the K Means method using the Silhouette index to define the optimal number of clusters.

Different representations for the corpus were tested using various ranges of characters and word n-grams. The clusters with the highest Silhouette values were those using 2-character n-grams and 1-word n-gram, both applying the traditional Bag of Words method with stopwords filtering and the TF-IDF weighting. Contrary to the Silhouette results, which were close to 1 for the traditional frequency counting and stopwords filtering method and close to zero for the use of TF-IDF matrices, the outcomes from decision tree training presented conflicting data. The calculation of feature importance using Entropy, Information Gain, and Information Gain Ratio showed a better distribution of importance among the different features for TF-IDF matrices.

However, irrespective of the technique employed, the optimal number of clusters recommended by Silhouette remained consistent at the level of 2-character n-grams (two clusters) and 1-word n-gram (three clusters). The scatter plots obtained from the different representations show a match with the stylistic distribution reported by Forstall et. al. [8] who used a Support Vector Machine (SVM) approach to test the influence of Catullus on the poetry of Paul the Deacon[3].

Since this is a preliminary exploratory analysis, there is still a need to apply some other techniques such as variable ranges of character and word n-grams (only fixed ranges were used in this study), other similarity measures such as Jaccard, Cosine, or Soft Cosine, or other clustering methods like Gaussian Mixture Models, Density-based spatial clustering of applications with noise (DBSCAN), or even hierarchical clustering methods. As for the representation of the documents, there is also a need to explore representation techniques with Embeddings like those recently developed by Burns et al., Bamman et al., and Johnson et al. [1, 3, 4, 11].

References

1. Bamman, D., Burns, P.J.: Latin BERT: A Contextual Language Model for Classical Philology, http://arxiv.org/abs/2009.10053, (2020). https://doi.org/10.48550/arXiv.2009.10053.
2. Bracco, G. et al.: Data mining tools and GRID infrastructure for Assyriology text analysis (an Old-Babylonian situation studied through text analysis and data mining tools). In: RAI-Rencontre Assyriologique Internationale- Private and State in the Ancient Near East. , Belgium (2013).
3. Burns, P.J.: Building a Text Analysis Pipeline for Classical Languages. In: Berti, M. (ed.) Digital Classical Philology. Ancient Greek and Latin in the Digital Revolution. pp. 159–176 Walter de Gruyter GmbH, Berlin - Boston (2019).
4. Burns, P.J.: LatinCy: Synthetic Trained Pipelines for Latin NLP, http://arxiv.org/abs/2305.04365, (2023).

[3] For a colored version of the scatter plot, see Coffe et al.[7]

https://doi.org/10.48550/arXiv.2305.04365.
5. Cantaluppi, G., Passarotti, M.: Clustering the Corpus of Seneca: A Lexical-Based Approach. In: Carpita, M. et al. (eds.) Advances in Latent Variables: Methods, Models and Applications. pp. 13–25 Springer International Publishing, Cham (2015). https://doi.org/10.1007/10104_2014_6.
6. Cerrato, L.M., Chavez, R.F.: Perseus Classics Collection: An Overview, https://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:1999.04.0053, last accessed 2023/03/24.
7. Coffee, N. et al.: Modelling the Interpretation of Literary Allusion with Machine Learning Techniques Journal of Digital Humanities. J. Digit. Humanit. 478–479 (2013).
8. Forstall, C.W. et al.: Evidence of intertextuality: investigating Paul the Deacon's Angustae Vitae. Lit. Linguist. Comput. 26, 3, 285–296 (2011). https://doi.org/10.1093/llc/fqr029.
9. Forstall, C.W., Scheirer, W.: A Statistical Stylistic Study of Latin Elegiac Couplets. Presented at the (2010).
10. Greenough, J.B.: The Bucolics, AEneid, and Georgics of Virgil. Ginn, Boston (1900).
11. Johnson, K.P. et al.: The Classical Language Toolkit: An NLP Framework for Pre-Modern Languages. In: Ji, H. et al. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. pp. 20–29 Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.acl-demo.3.
12. Lewis, C.S.: La alegoría del amor: un estudio sobre tradición medieval. Encuentro, Madrid (1936).
13. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory. 28, 2, 129–137 (1982). https://doi.org/10.1109/TIT.1982.1056489.
14. Martins, A. et al.: Historia Augusta authorship: an approach based on Measurements of Complex Networks. Appl. Netw. Sci. 6, 1, 1–23 (2021). https://doi.org/10.1007/s41109-021-00390-7.
15. Merrill, E.T.: Catullus; edited by Elmer Truesdell Merrill. Boston Ginn (1893).
16. Moretti, F.: Distant Reading. Verso (2013).
17. Müller, L.: Sex. Propertii Elegiae. Teubner, Leipzig (1898).
18. Nagy, B.: Rhyme in classical Latin poetry: Stylistic or stochastic? Digit. Scholarsh. Humanit. 37, 4, 1097–1118 (2022). https://doi.org/10.1093/llc/fqab105.
19. Nusch, C.J.: Las Edades del Amor: una propuesta para el proyecto Aetates Amoris destinado a la poesía amorosa. Universidad Nacional de Educación a Distancia (UNED) (2021). https://doi.org/10.35537/10915/125629.
20. Nusch, C.J.: Una breve exploración de la terminología amorosa en los corpora catullianum, tibullianum y propertianum con métodos y herramientas computacionales: etiquetado gramatical, lemas, bigramas y co-apariciones. Rev. Humanidades Digit. 9, (2024). https://doi.org/10.5944/rhd.vol.9.2024.38680.
21. Postgate, J.P.: Tibulli aliorumque carminum libri tres. Scriptorum classicorum bibliotheca Oxoniensis, Oxford (1915).
22. Quinlan, J.R.: Induction of decision trees. Mach. Learn. 1, 1, 81–106 (1986).

https://doi.org/10.1007/BF00116251.
23. Ramsay, S.: Reading Machines: Toward an Algorithmic Criticism. University of Illinois Press (2011).
24. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65 (1987). https://doi.org/10.1016/0377-0427(87)90125-7.
25. Salton, G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill, Inc., New York (1986).
26. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. 27, 3, 379–423 (1948). https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.
27. Spärck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. J. Doc. 28, 1, 11–21 (1972). https://doi.org/10.1108/eb026526.
28. Weise, C.H.: Pharsaliae Libri X. M. Annaeus Lucanus. G. Bassus, Leipzig (1935).
29. Perseus Digital Library Homepage, https://www.perseus.tufts.edu/hopper/.
30. Perseus Digital Library Stopwords, https://www.perseus.tufts.edu/hopper/stopwords.
31. Stopwords ISO, https://github.com/stopwords-iso/stopwords-iso/blob/master/README.md.