Initial Explorations for Document Clustering Tasks in Latin Elegiac Poets

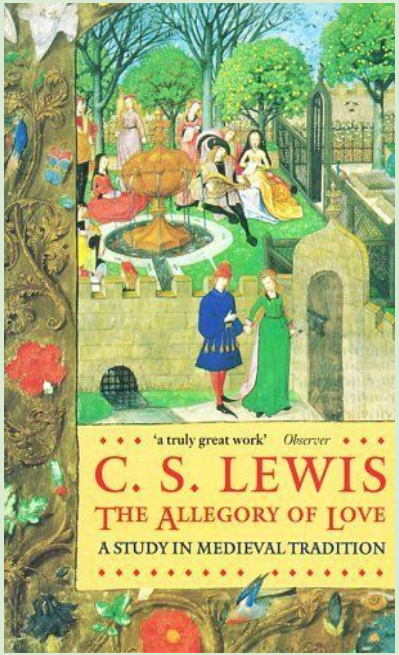
Carlos Javier Nusch, Gimena del Rio Riande, Leticia Cecilia Cagnina, Marcelo Luis Errecalde, Leandro Antonelli.

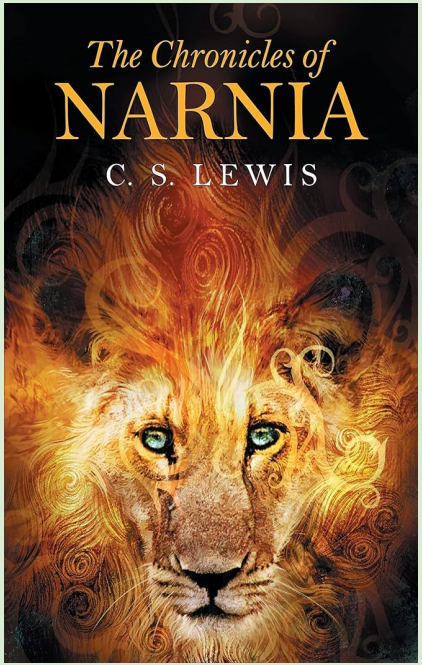
Motivation, Corpus and Methods

- The motivation behind this work is to explore a historical literary trend revolving around the themes of love.
- Automatic Text Analysis tasks applying Natural Language Processing techniques on a corpus of Latin texts (1st century BC and AD).
- The methods: Frequency Matrix, TF-IDF Matrix, Different Ranges of N grams, K-Means, Decision Trees.
- The metrics: Silhouette Score, Importance, Information Gain, Information Gain Ratio.

Motivation: A Reading of C. S. Lewis

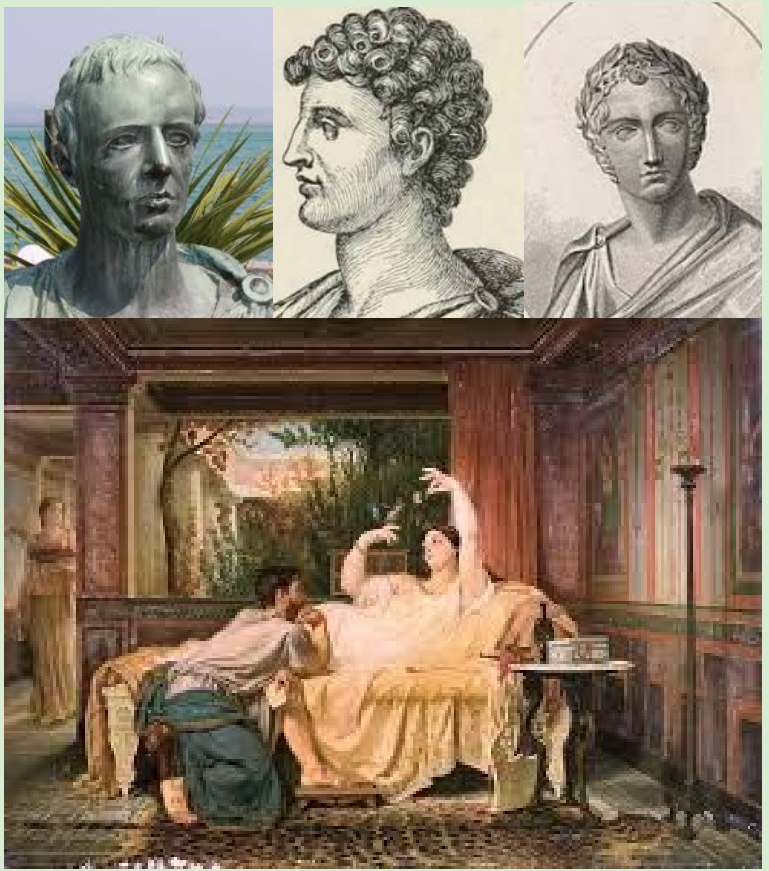
- **“French poets, in the eleventh century, discovered or invented, or were the first to express, that romantic species of passion which English poets were still writing about in the nineteenth.”** (Lewis, 1936)
- Lewis suggested that the shape of the love motifs and imaginary we had, at least at the beginnings of the XX Century, belong to the occitan poets in the XI century.





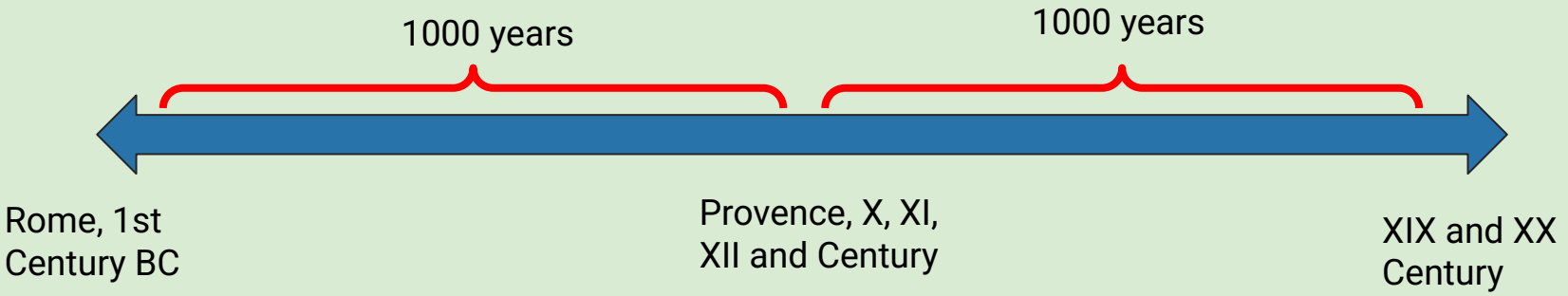
But... What happened before?

- Roman poets called *neoteroi* or elegiac poets also had a similar style and wrote about the same themes than the occitan poets.
- So we have 1000 years between roman poets and occitan poets and we could ask...
    - Is there any tradition we could track or is it just a coincidence?



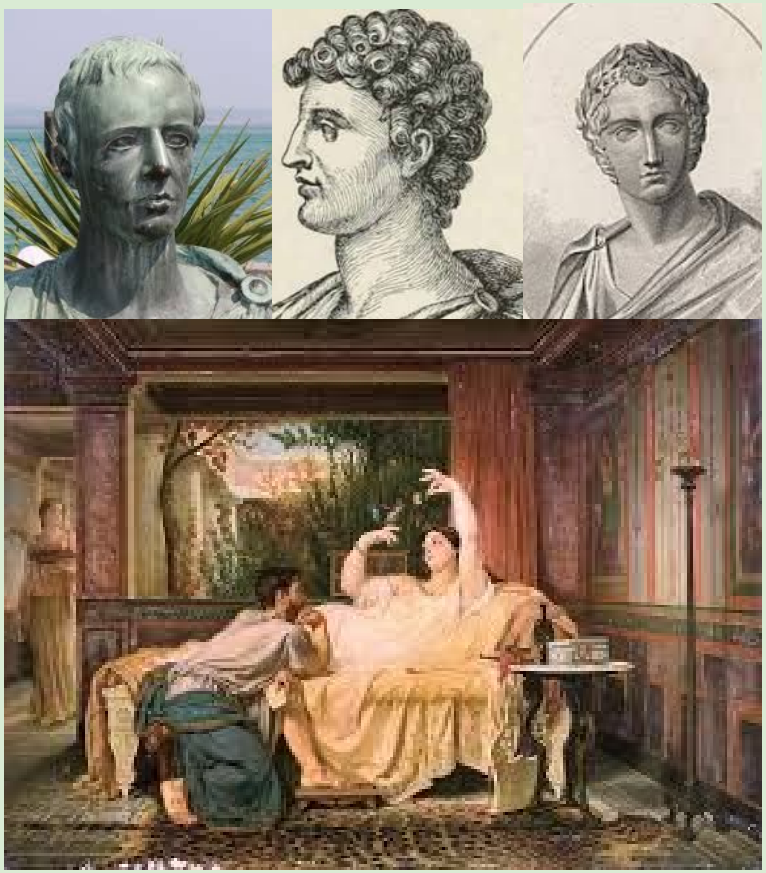
Motivation: What happened before?

- Years ago the comparing task of thousands of texts would be impossible to do in a lifetime...
- But now we have the advantages of the computational techniques:
  - We can perform clustering task to explore the corpora
  - And try to group documents with a similar style
  - and find features and patterns on those clusters.



The positive class

- The analyzed and targeted authors:
  - Gaius Valerius Catullus
  - Albius Tibullus
  - Sextus Propertius
- The control sample authors:
  - Publius Vergilius Maro
  - Marcus Annaeus Lucanus



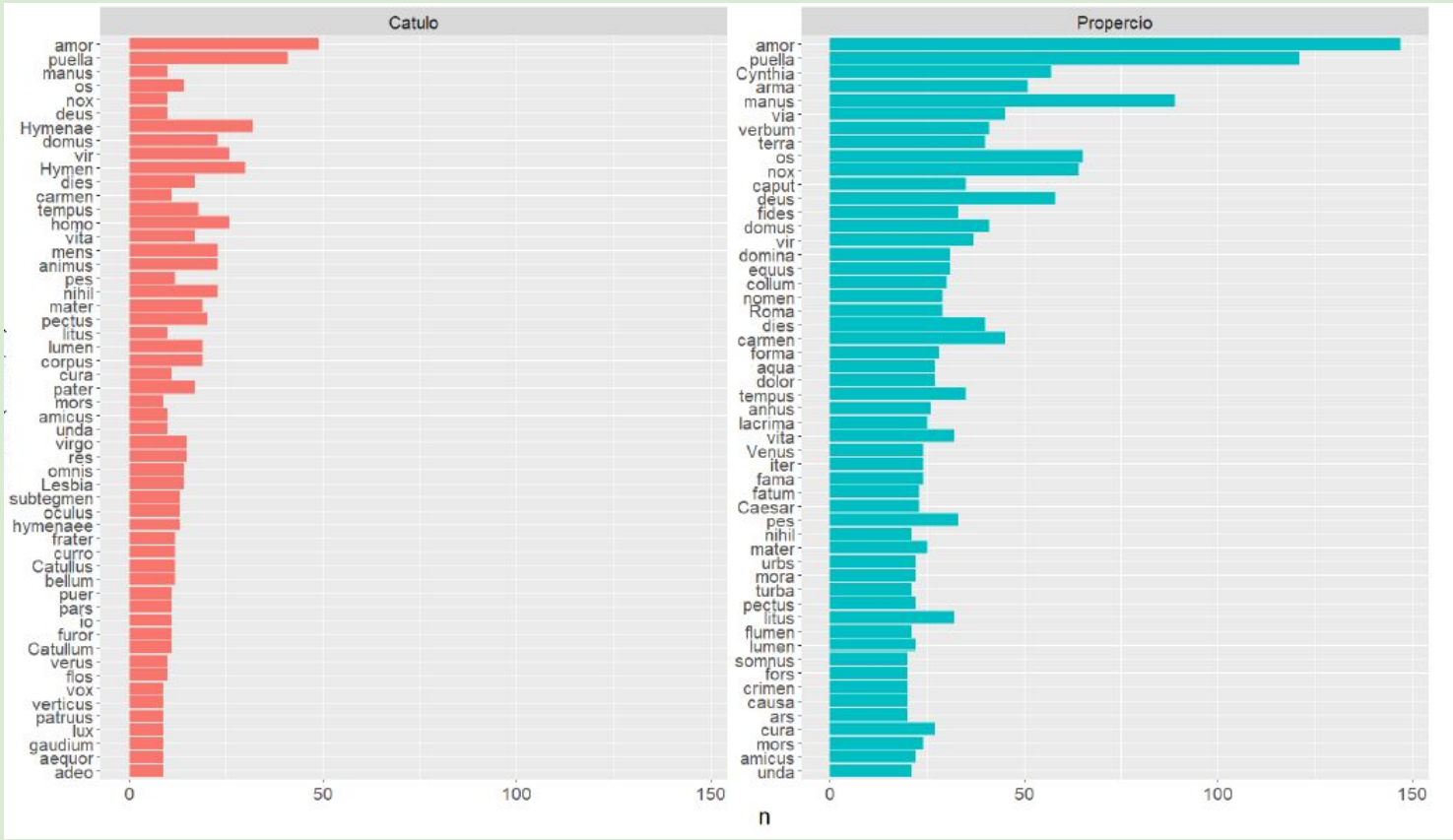
- Their style:
  - Lyric poetry
  - Mostly short poems
  - Love theme
  - Love motifs

The control sample authors

- The analyzed and targeted authors:
  - Gaius Valerius Catullus
  - Albius Tibullus
  - Sextus Propertius
- The control sample authors:
  - Publius Vergilius Maro
  - Marcus Annaeus Lucanus
- Their style:
  - Epic poetry
  - Large poems
  - Mythic and war themes

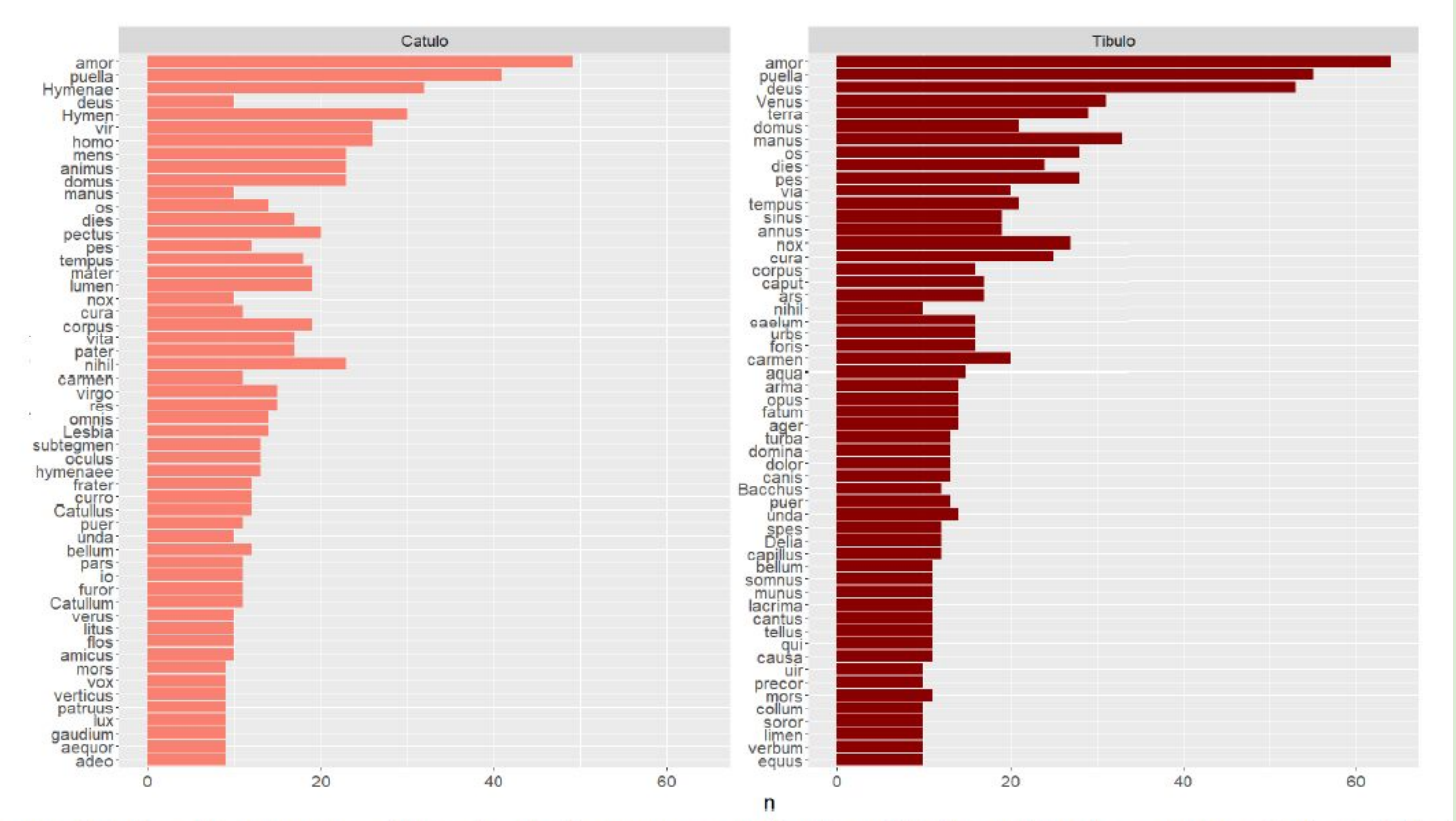
Previous Findings

The 50 most frequent nouns in Catullus and Propertius ordered by lemma



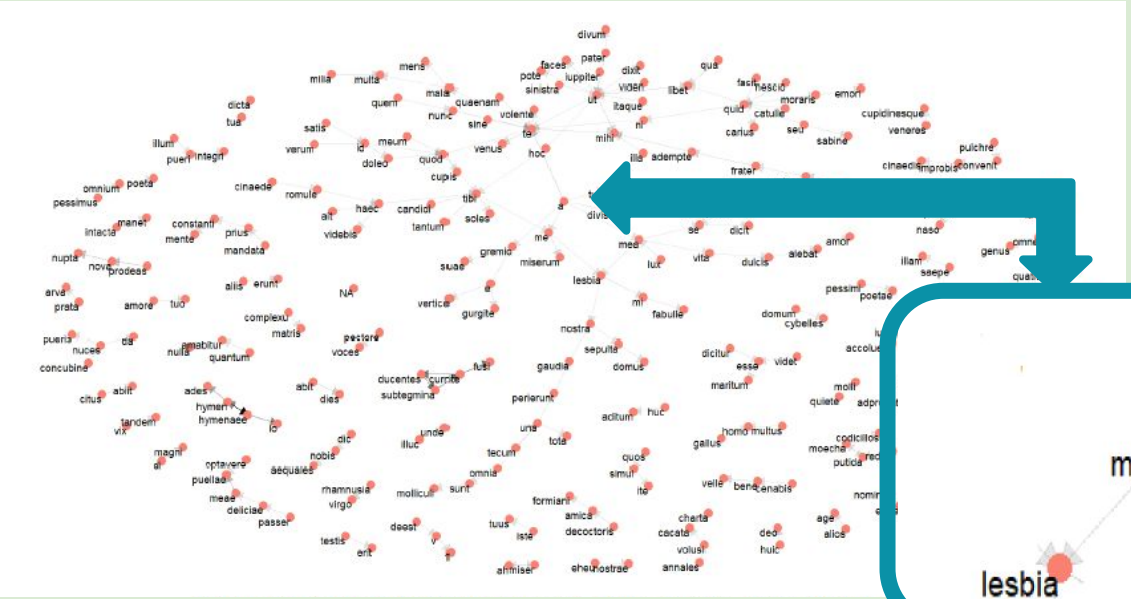
Previous Findings

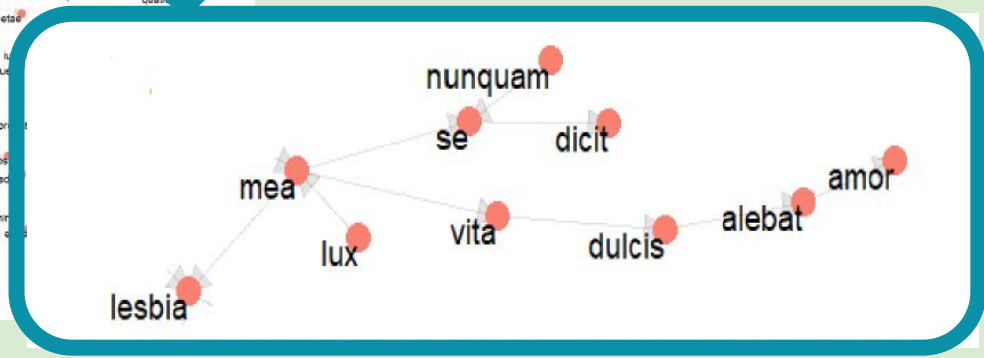
The 50 most frequent nouns in Catullus and Tibullus ordered by lemma



Previous Findings

Bigram graph of Catullus and the sector of the graph that refers to Lesbia



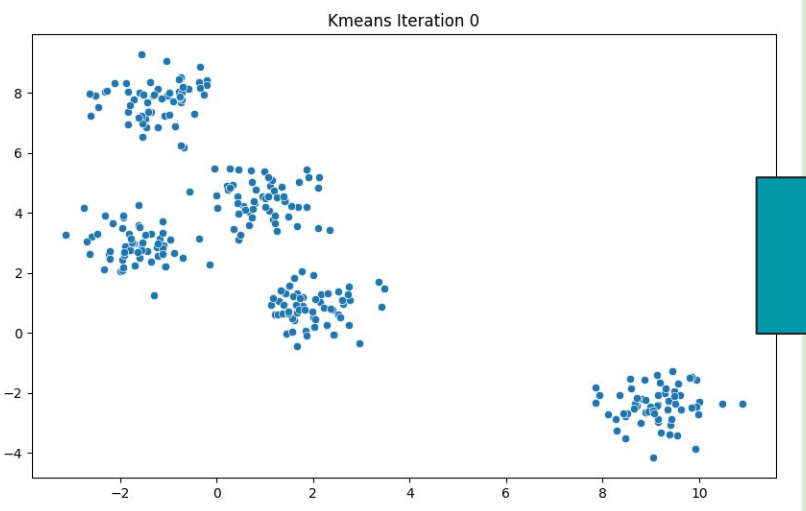


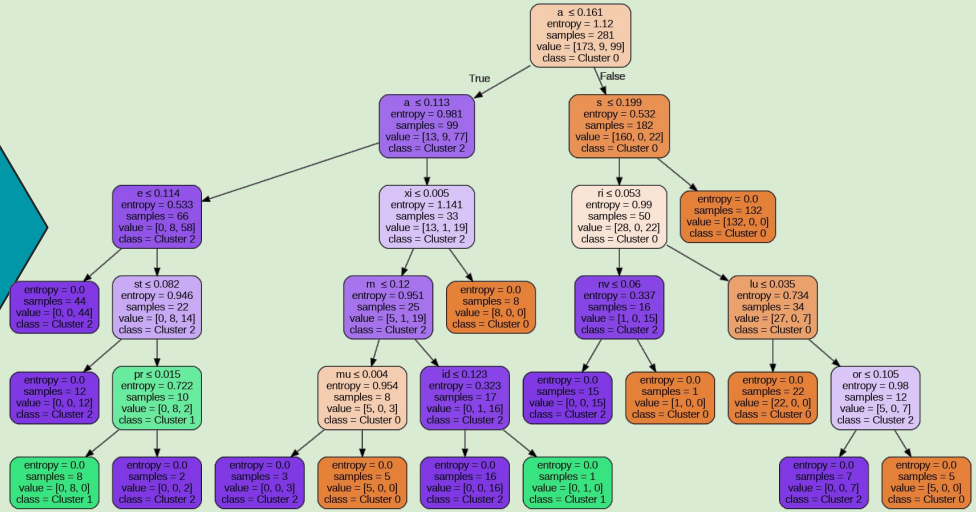
Previous Findings

We found a set of distinctive words related to the theme of love in the Elegiac Poets:

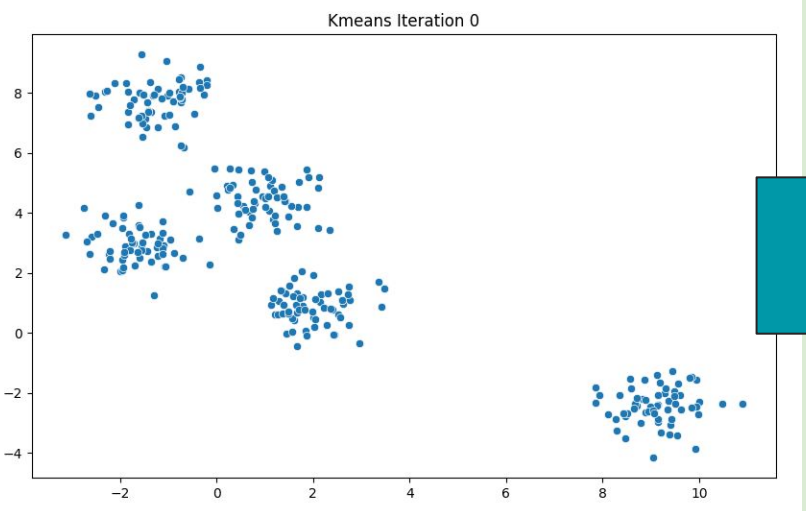
- **amor:** love
- **puella:** girl, but the beloved girl
- **domina:** lady, the beloved has the status of a masculine medieval master (dominus)
- **manus:** hands
- **oculi:** eyes
- **vita:** life
- **dulcis:** sweet
- **lux:** light
- **Lesbia:** the pseudonym of Catullus beloved
- **Cynthia:** the pseudonym of Tibullus beloved
- **Delia:** the pseudonym of Propertius beloved
- **personal pronouns:** ego (I), tu (you), me (me), te (you), mihi (for/to me), tibi (for/to you)...

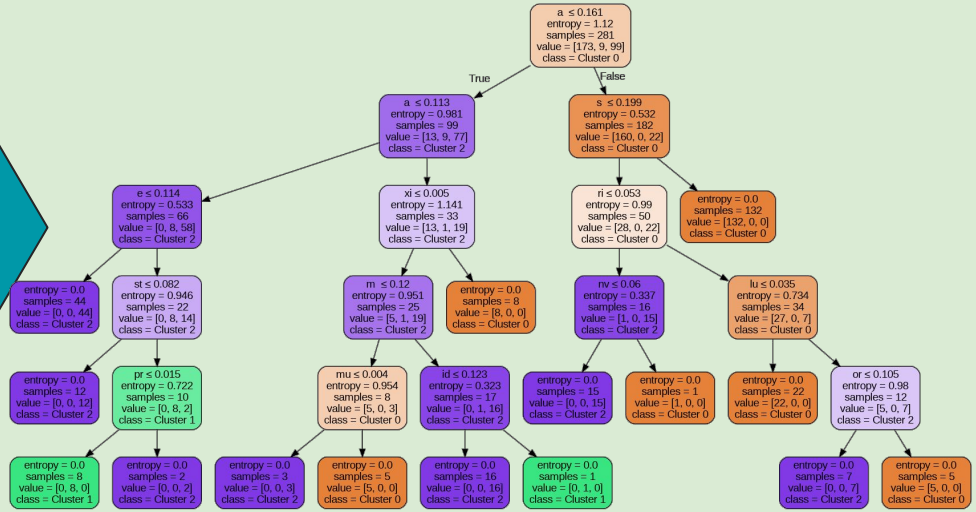
The NLP tasks





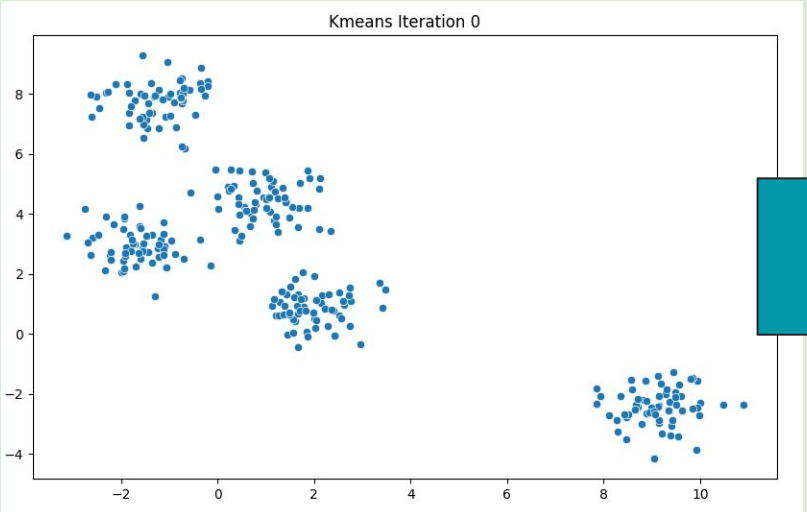
The NLP tasks

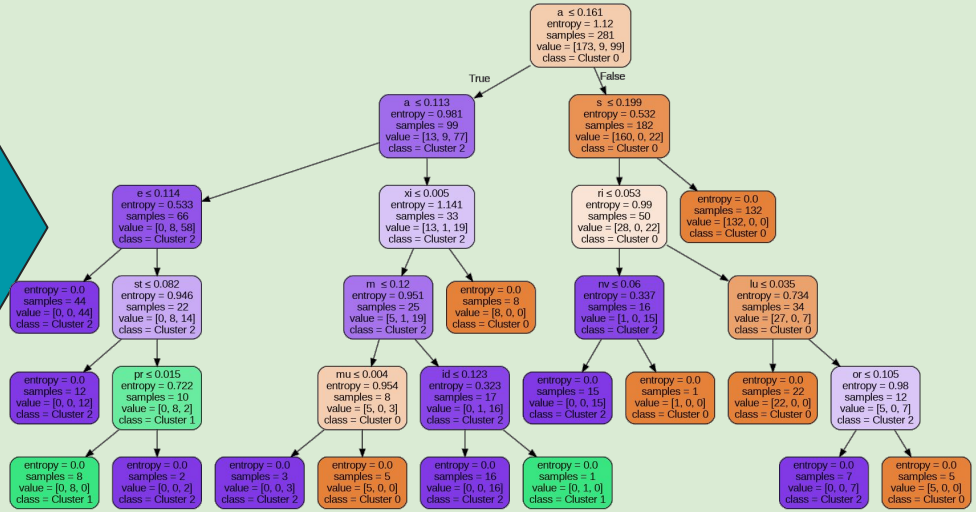




- We used the K-Means algorithm to cluster the different documents.
- The Silhouette Score was employed to determine the optimal number of clusters in the different ranges of character and word n-grams.

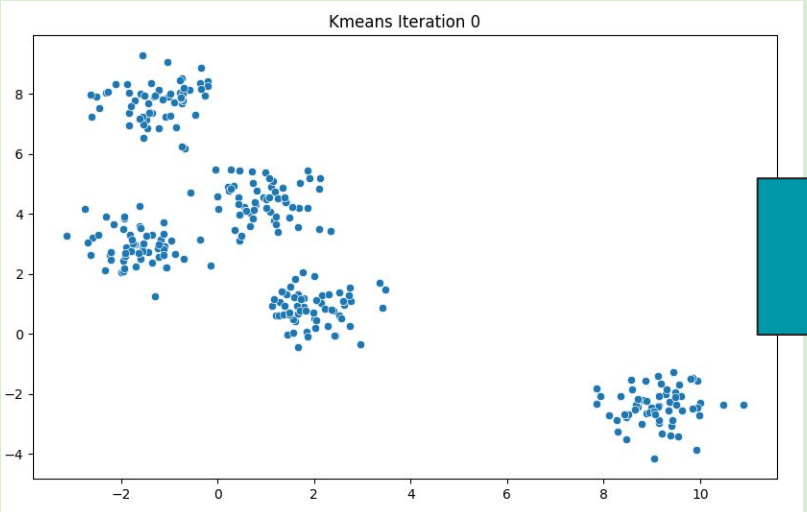
The NLP tasks

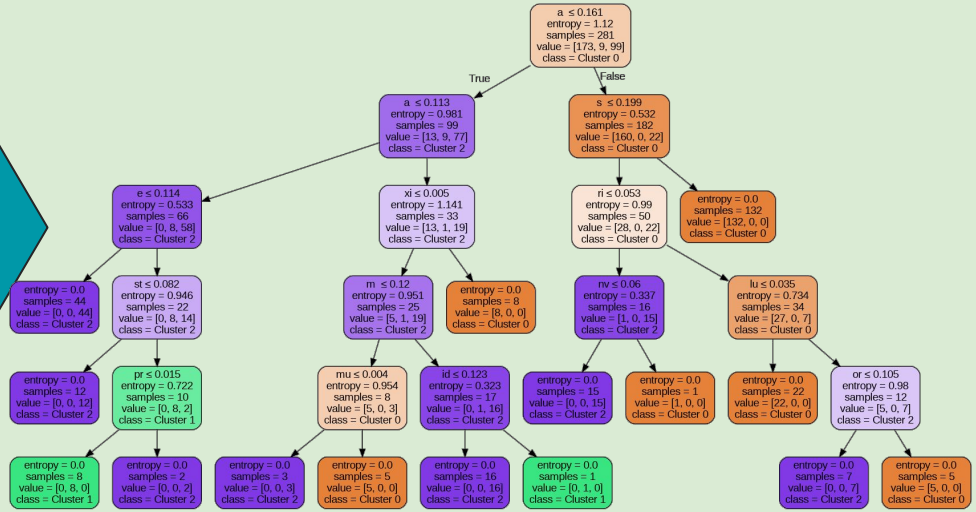




- K-Means has certain limitations. I does not allow us to select the most important features. The algorithm works with a distance measure, so in this case, we could only ask for the closest documents to the centroid.
- Therefore, we used decision trees to select the best features by an indirect method.

The NLP tasks





- In the first step, we separated the documents into clusters.
- In the second step, using the assigned clusters as labels, we trained a decision tree to extract the main features.

N Gram Features and Range

- The clustering tasks were carried out using fixed ranges of character n-grams and word n-grams:
    - Character range n-gram: 2 to 7.
    - Word range n-gram: 1 to 5.
    - Cluster range: 2 to 20.
- **Why do we used character n-grams?**
    - We often underestimate the potential of character n-gram parsing for classification tasks,
    - so it was an excellent chance to evaluate it too.

Document Representation Techniques: Frequency Matrix

- **Frequency Matrix**
  - The Bag-Of-Words (BoW) model is a model of text which uses a representation of text that is based on an unordered collection (or "bag") of words.

Word	Catullus	amat	Lesbiam	Tibullus	quoque	sed	Deliam	Lesbia	placet	ludi
Document 1	1	2	1	1	1	1	1	0	0	0
Document 2	1	1	1	0	0	0	0	1	1	1

Document Representation Techniques: TF-IDF Matrix

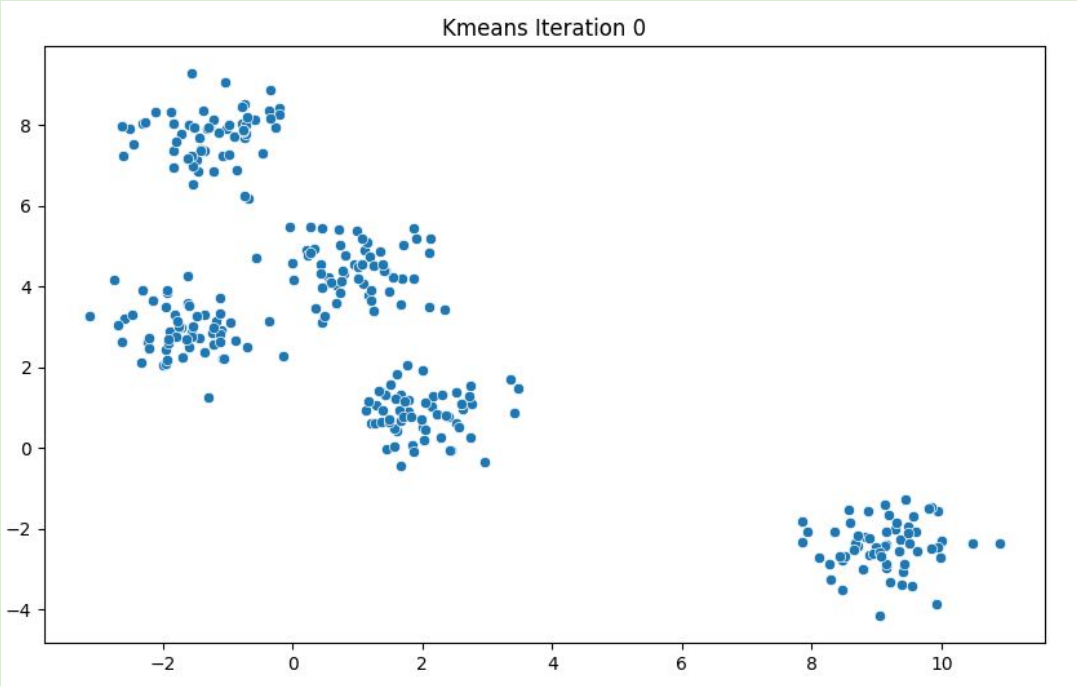
- **TF-IDF Weighting**
  - Term Frequency-Inverse Document Frequency is a statistical technique used to evaluate the importance of a word in a document relative to a collection of documents or corpus.

  - **This method penalizes**
    - words that appear in the majority of documents and
    - words that appear in very few documents.
      - **They are less useful for classification tasks**
      - **The matrix size is smaller and the process should be more efficient**

Word	Catullus	amat	Lesbiam	Tibullus	quoque	sed	Deliam	Lesbia	placet	ludi
Document 1	0.268208	0.536416	0.268208	0.376957	0.376957	0.376957	0.376957	0	0	0
Document 2	0.334712	0.334712	0.334712	0	0	0	0	0.470426	0.470426	0.470426

Clustering Task: K Means

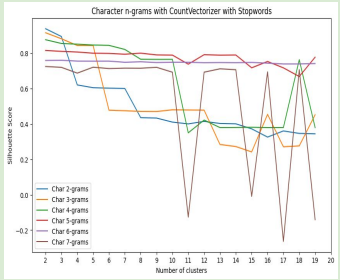
- The K-Means Method:
  - We need to choose a number of centroids to separate the clusters.
  - A centroid is an ideal point from which we will calculate the position.
  - The positions of the centroids are calculated over several iterations until they occupy the central positions of the clusters.

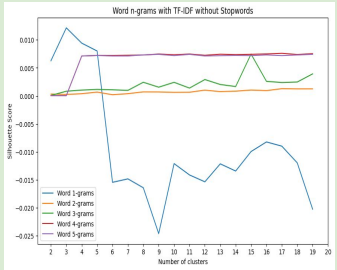


Weakness: By manually choosing the number of clusters (centroids), we could easily introduce bias.

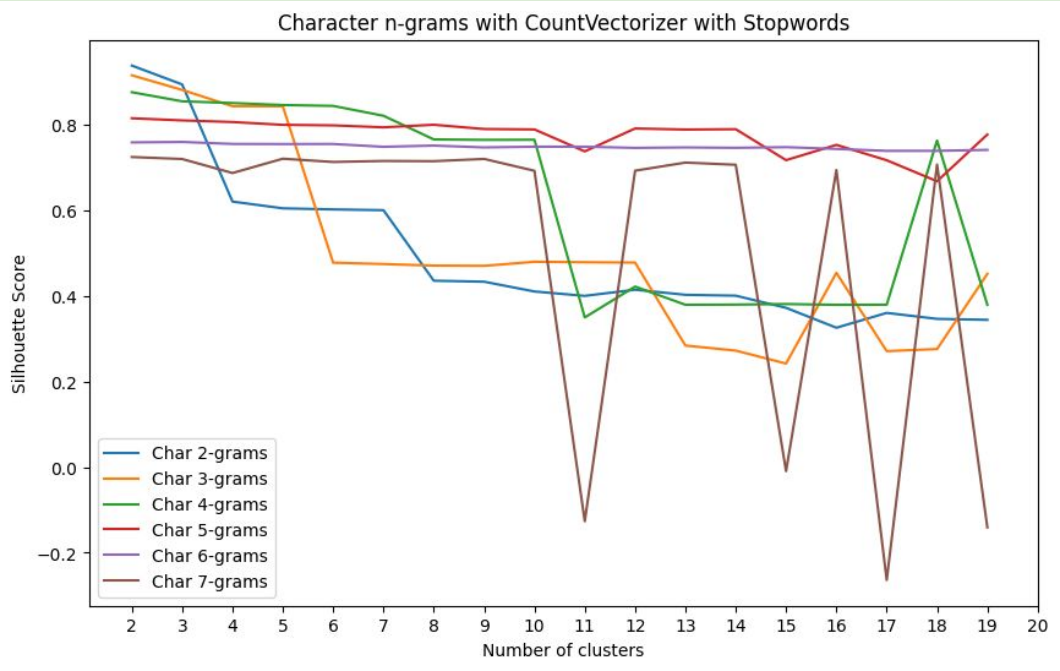
Silhouette Score

- Silhouette score measures:
  - **cohesion:** how similar a data point is to its own cluster
  - **separation:** how similar a data point compared to other clusters
- The Silhouette Score ranges from -1 to 1.
  - **A score close to 1 is considered a good measure.**
  - A score close to 0 indicates that the clusters are not very cohesive and not well separated.
  - A negative Silhouette Score tells us that the clusters are wrongly grouped and the documents are wrongly assigned.



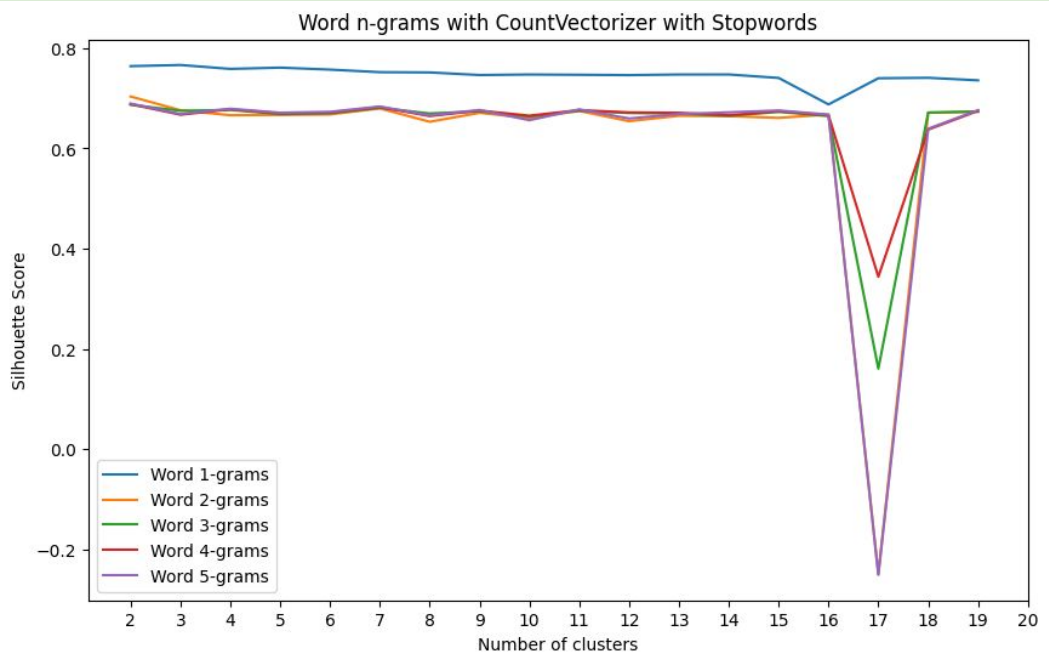


Silhouette Score: Frequency Matrix



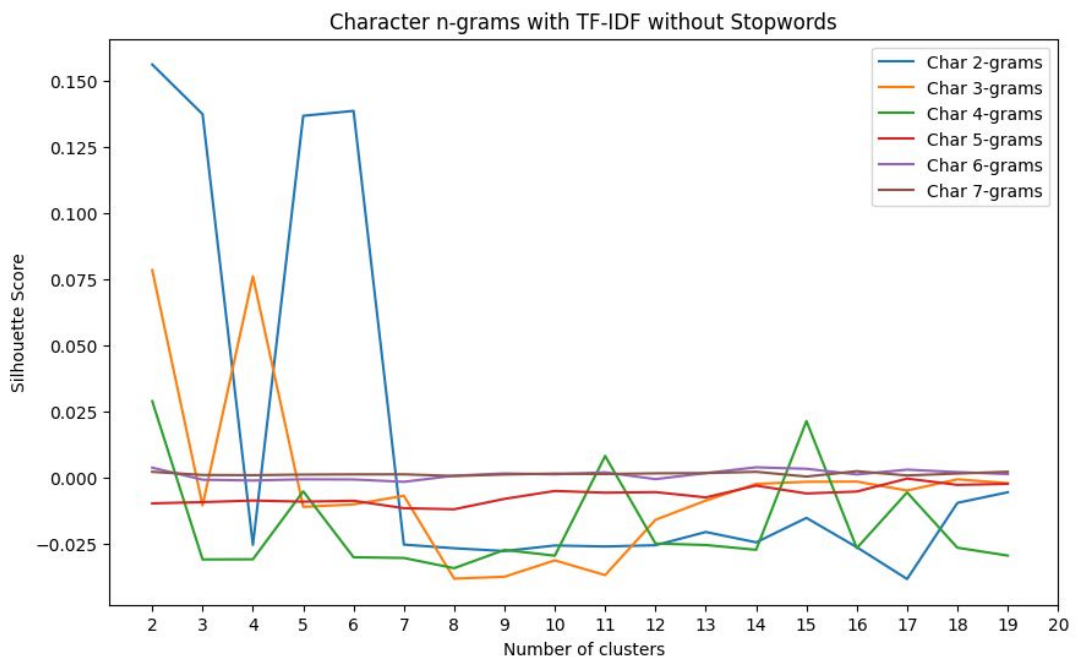
By calculating the Silhouette Score on a range of clusters (from 2 to 20 this time) we can approach the optimal number of clusters on a dataset.

Silhouette Score: Frequency Matrix



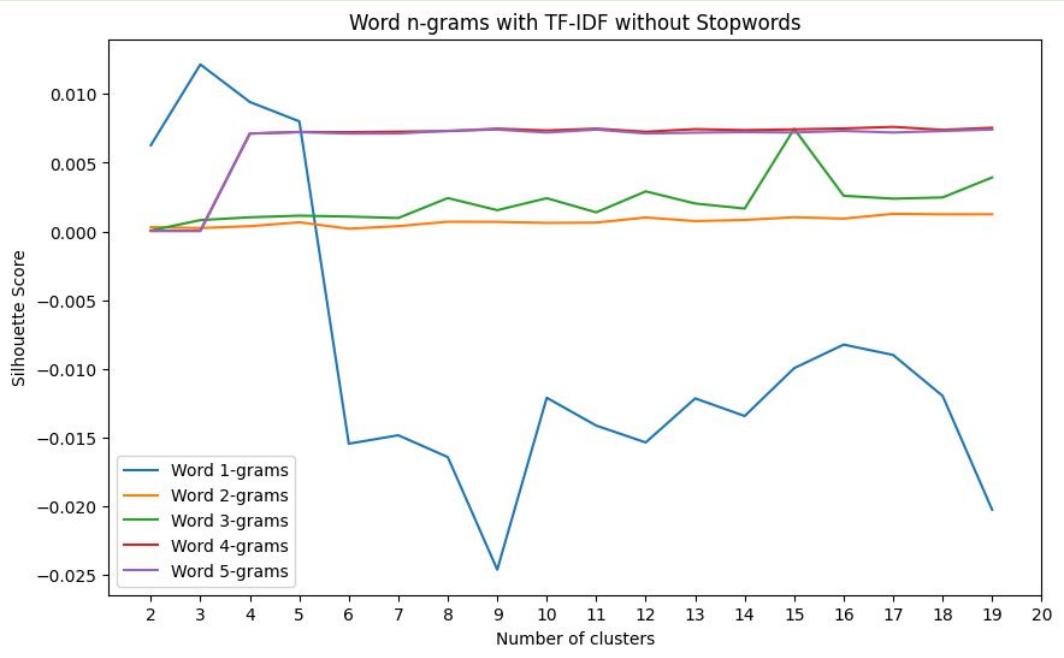
By calculating the Silhouette Score on a range of clusters (from 2 to 20 this time) we can approach the optimal number of clusters on a dataset.

Silhouette Score: TF-IDF Matrix



By calculating the Silhouette Score on a range of clusters (from 2 to 20 this time) we can approach the optimal number of clusters on a dataset.

Silhouette Score: TF-IDF Matrix



By calculating the Silhouette Score on a range of clusters (from 2 to 20 this time) we can approach the optimal number of clusters on a dataset.

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

Better Score: 2 clusters for 2-character grams.

Better Score: 3 clusters for 1-word 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

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

TF-IDF values are close to zero.

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

TF-IDF negative value.

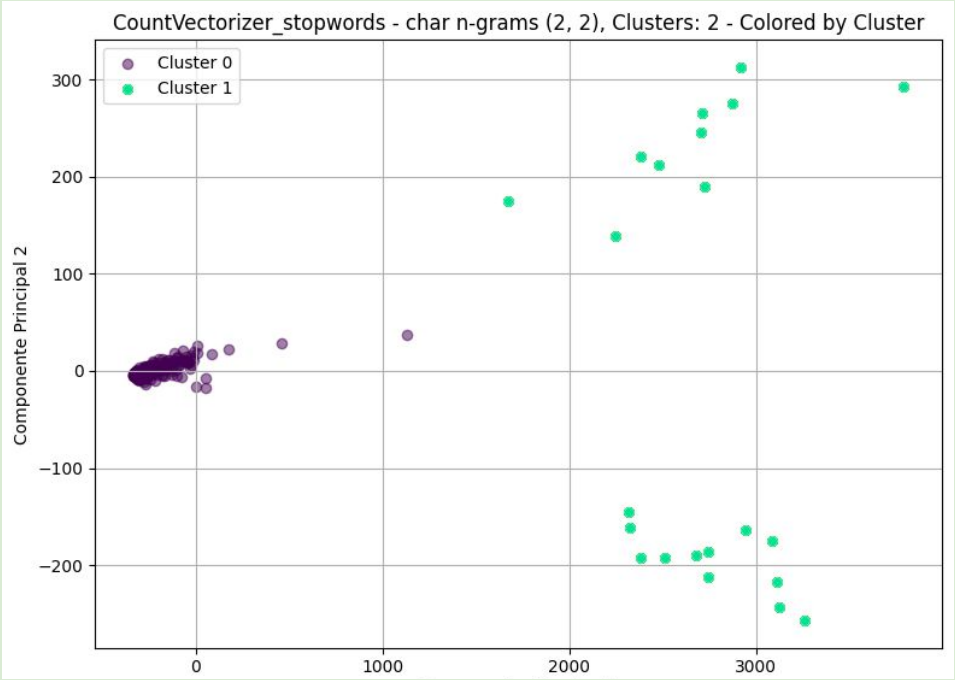
The highest Silhouette values matched the optimal number of clusters for both word and character n-gram ranges in both types of matrices

Same number of optimal clusters for both matrix techniques

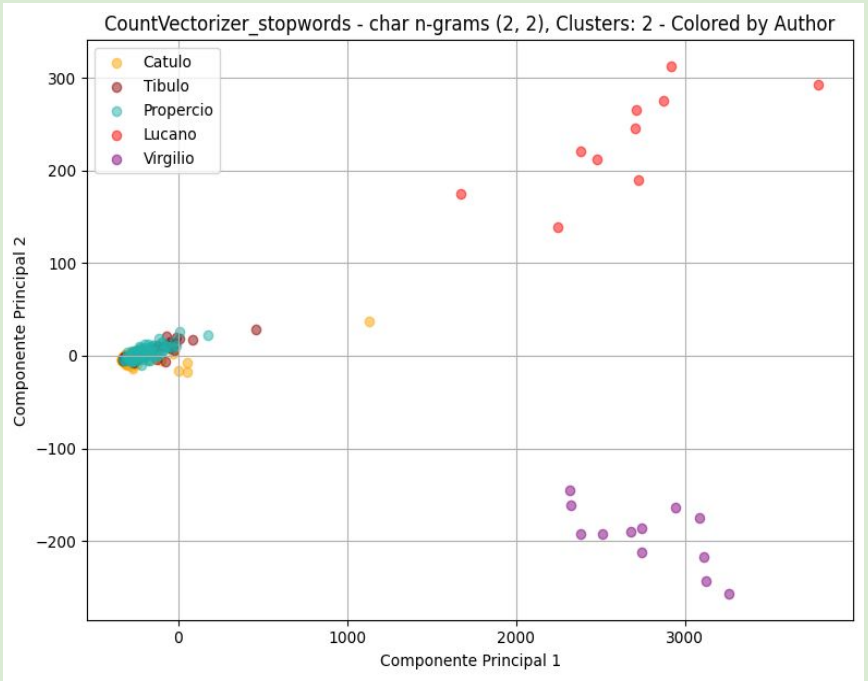
Optimal Clusters (Stopwords ISO)

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

Scatter plot of document clustering by K Means using a frequency matrix of 2 character n-grams

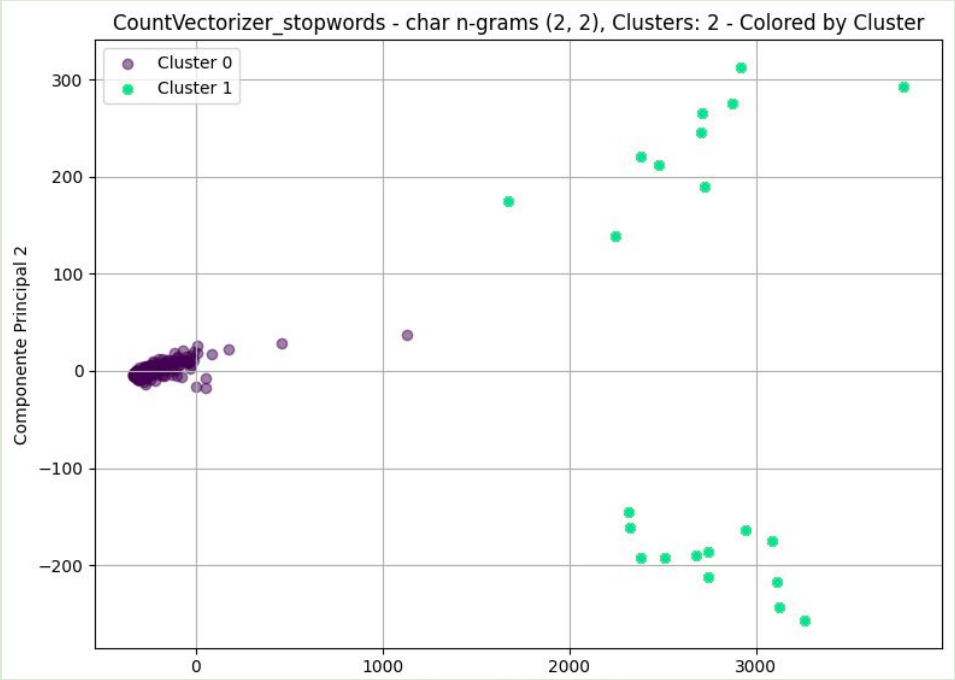


Colored by cluster

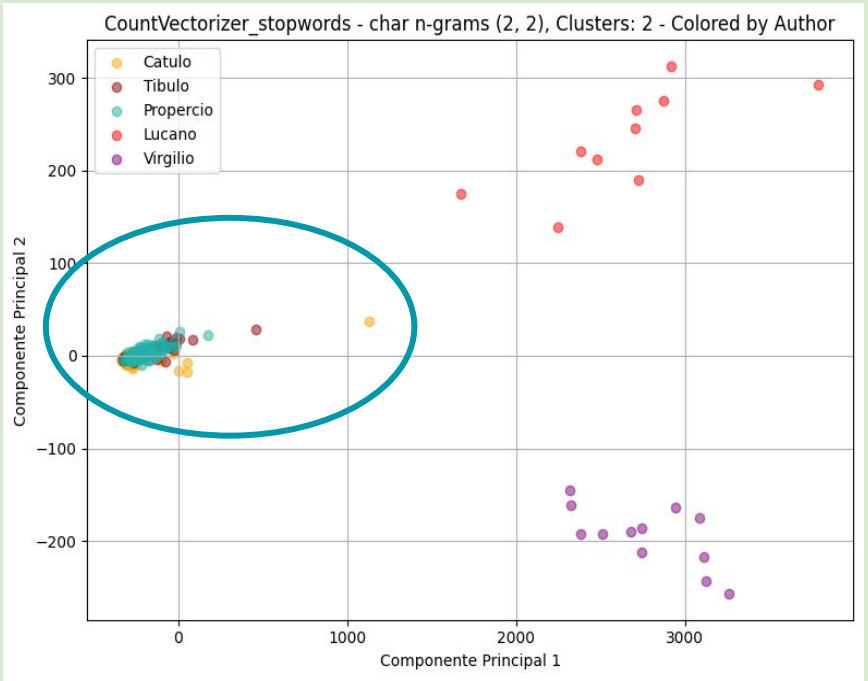


Colored by author

Scatter plot of document clustering by K Means using a frequency matrix of 2 character n-grams

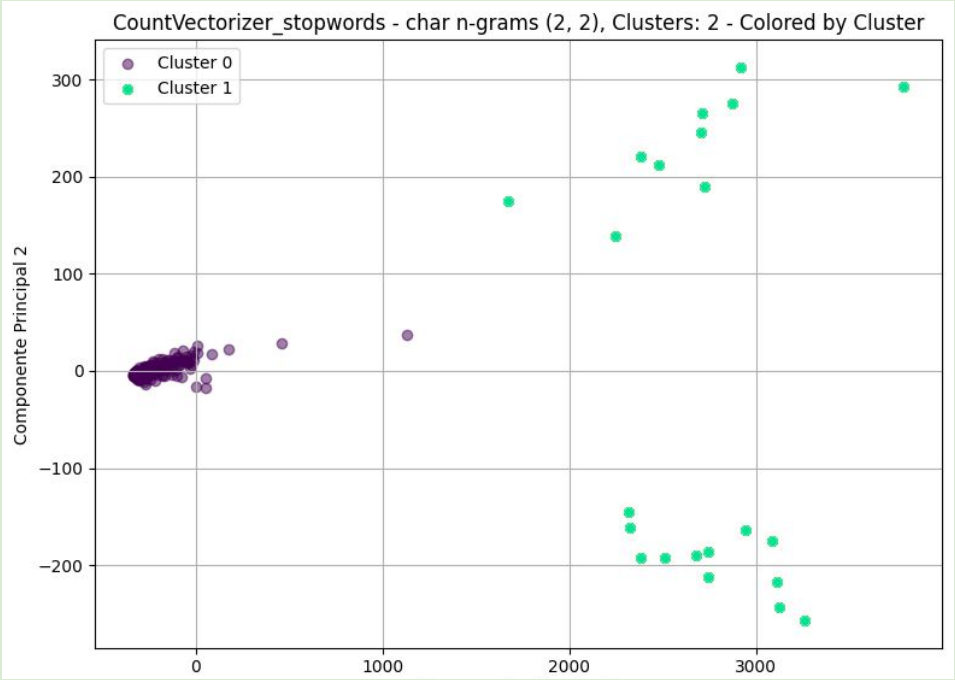


Colored by cluster

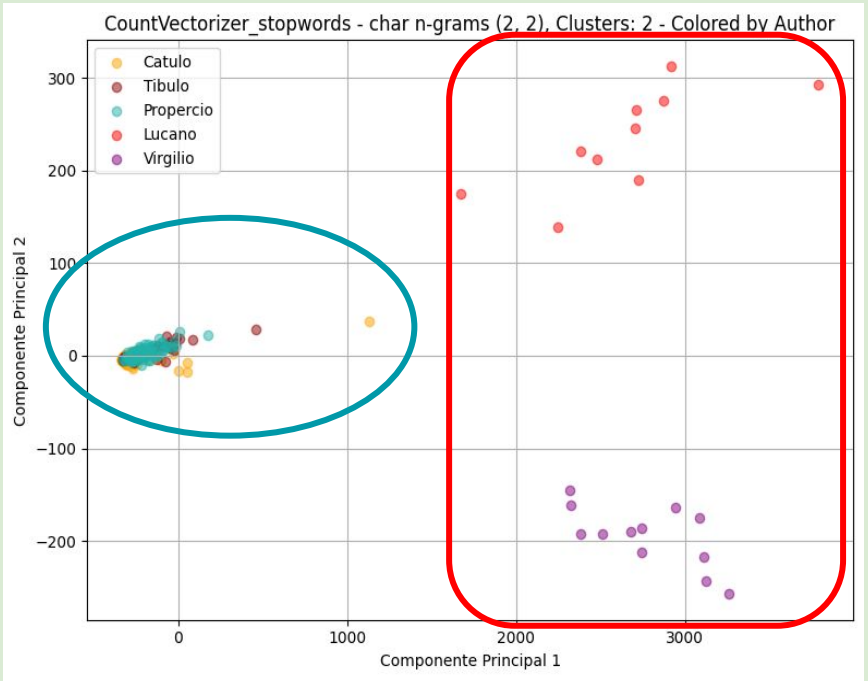


Colored by author

Scatter plot of document clustering by K Means using a frequency matrix of 2 character n-grams

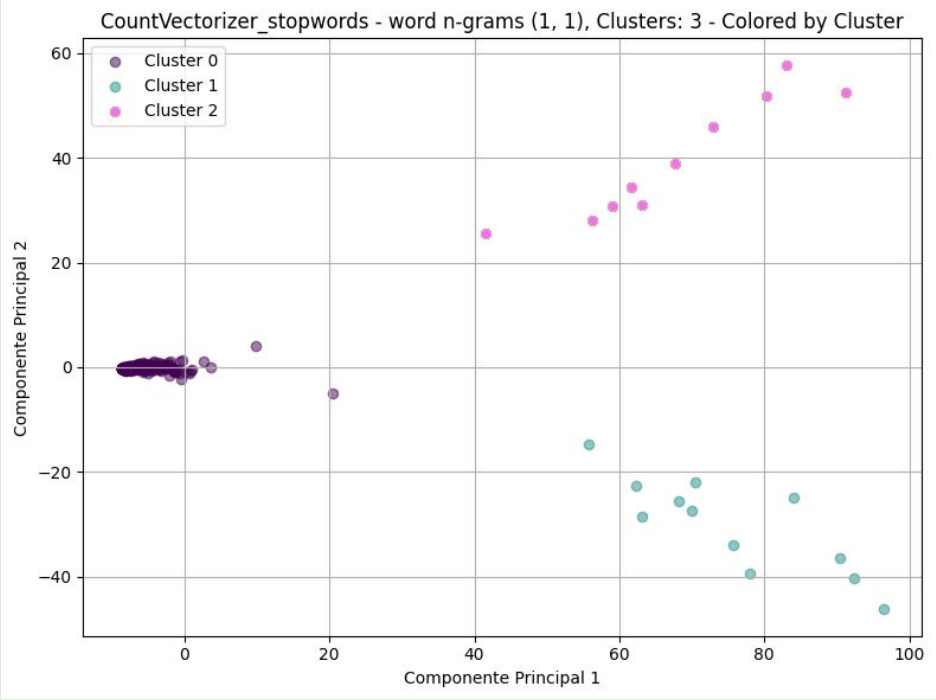


Colored by cluster

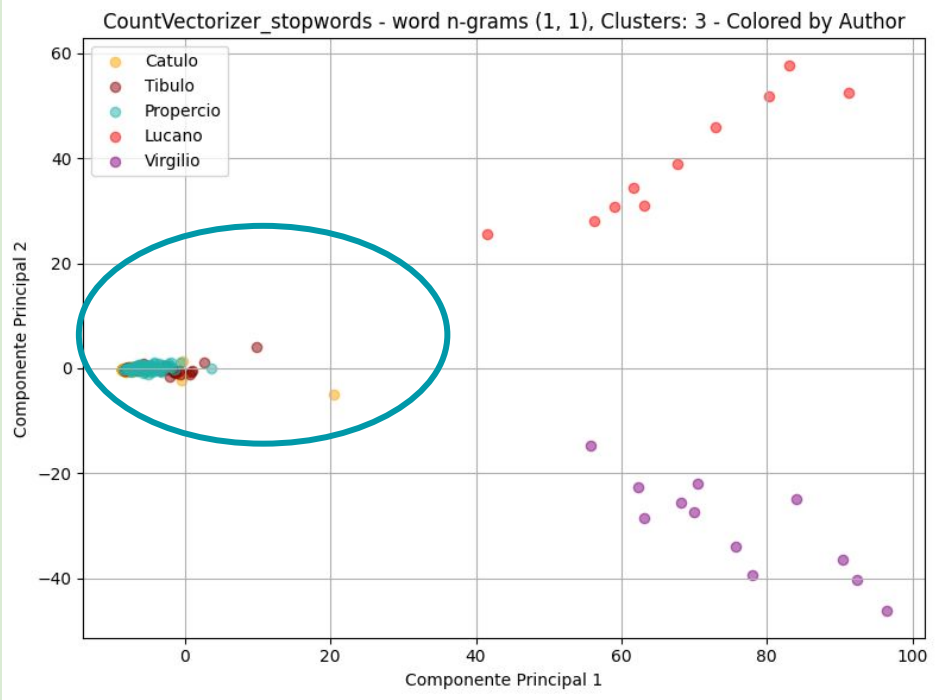


Colored by author

Scatter plot of document clustering by K Means using a frequency matrix of 1 word n-grams

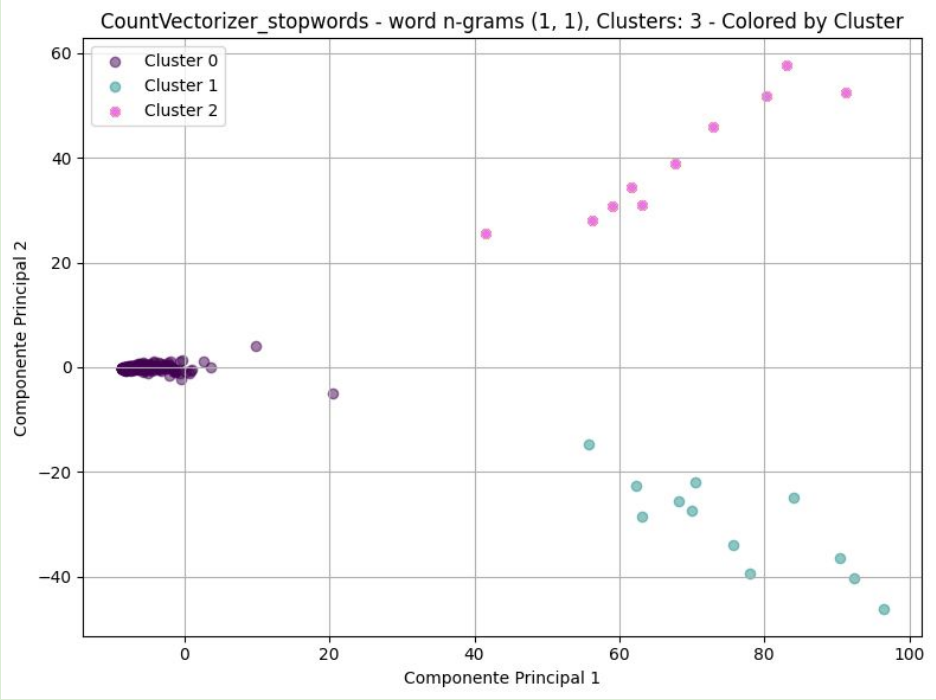


Colored by cluster

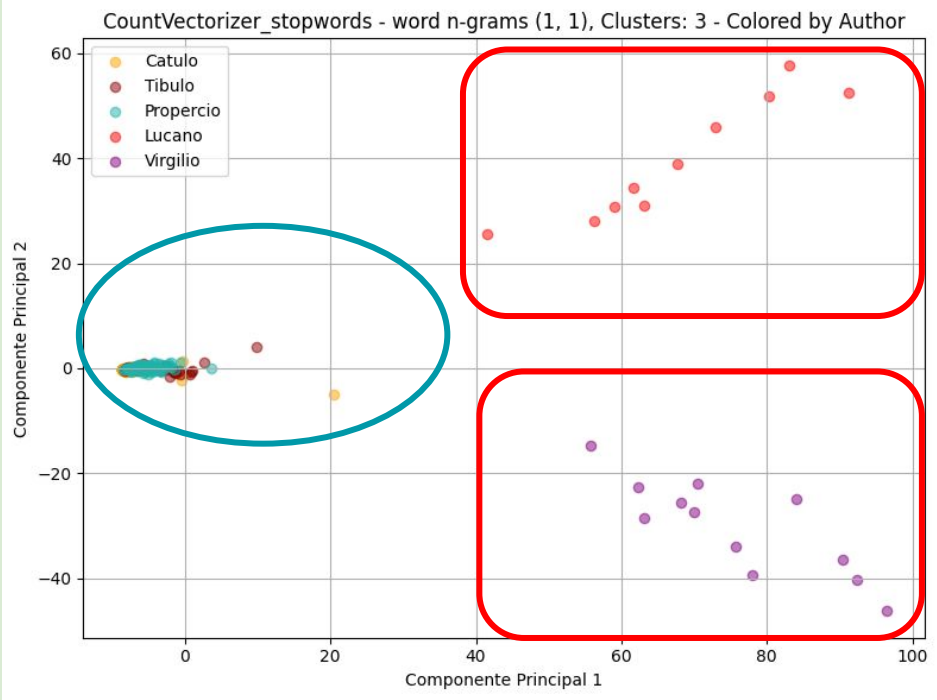


Colored by author

Scatter plot of document clustering by K Means using a frequency matrix of 1 word n-grams

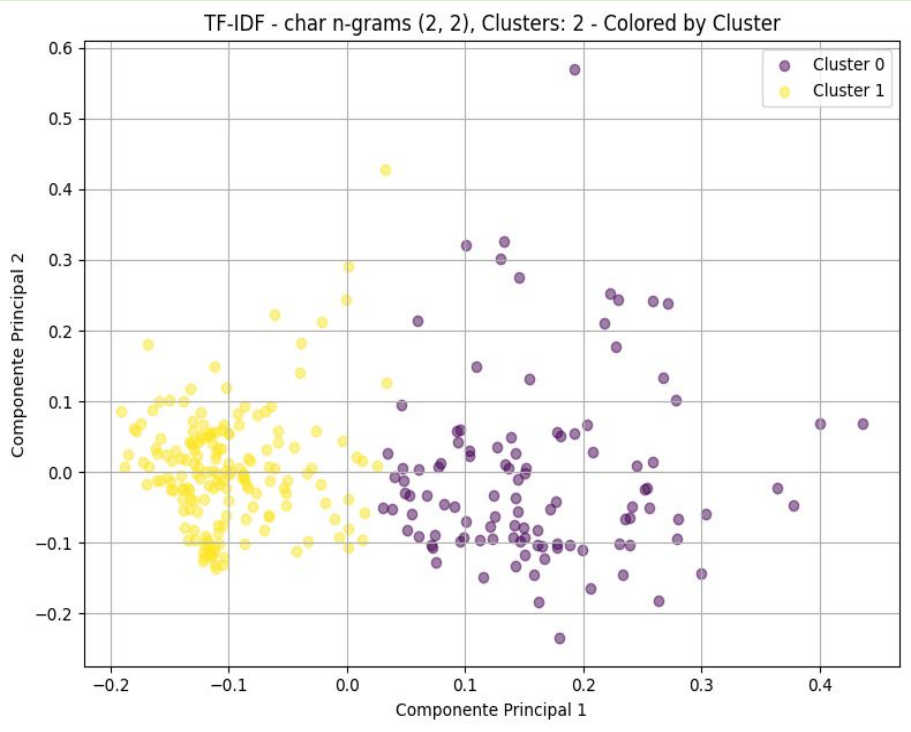


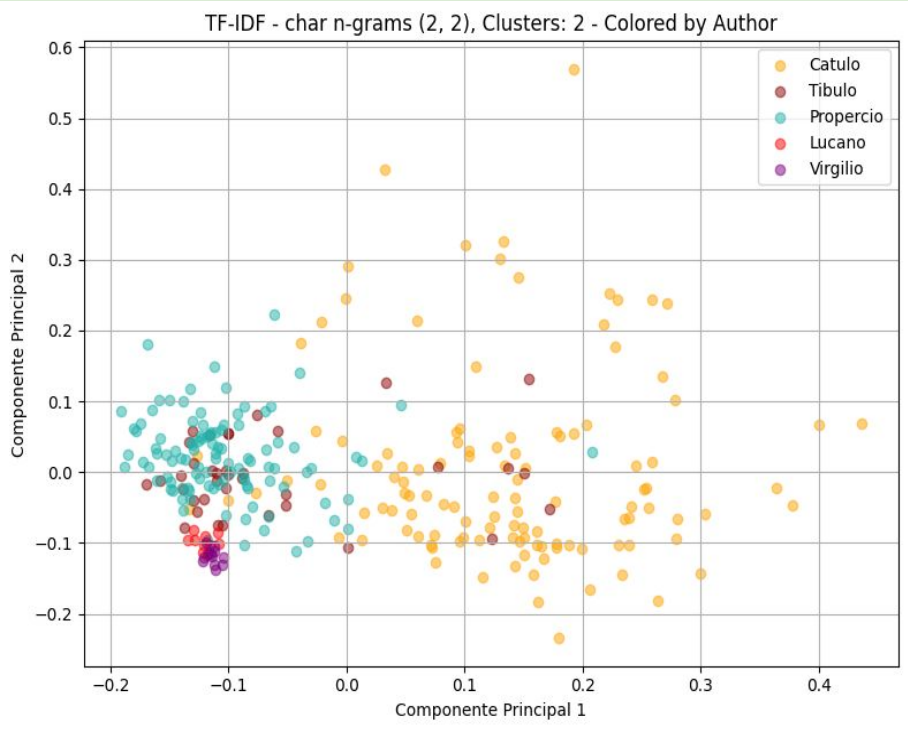
Colored by cluster



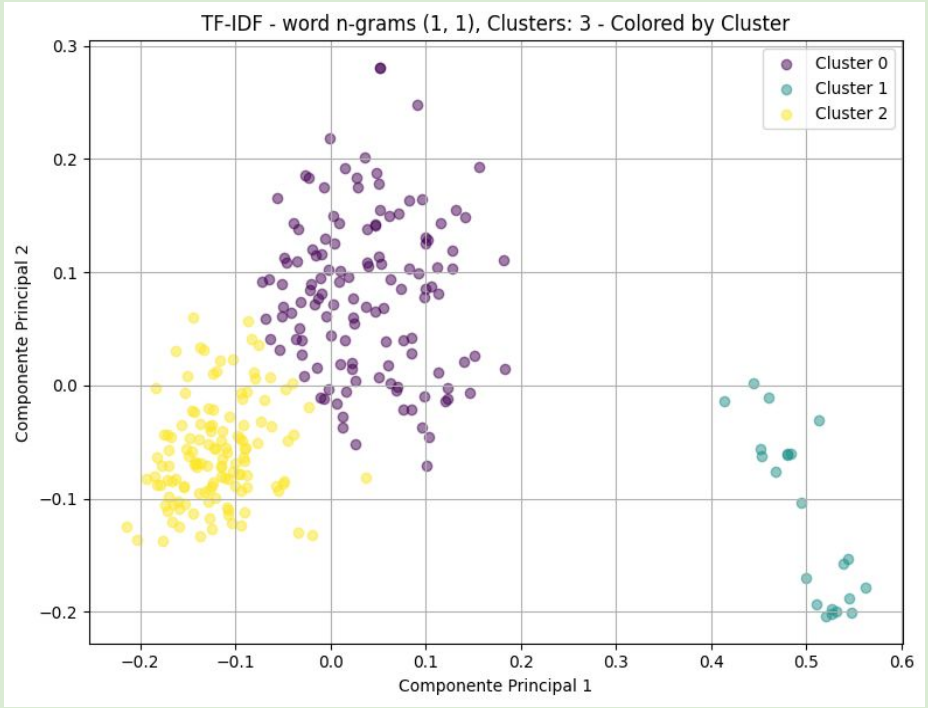
Colored by author

Scatter plot of document clustering by K Means using a TF -IDF matrix of 2 character n-grams

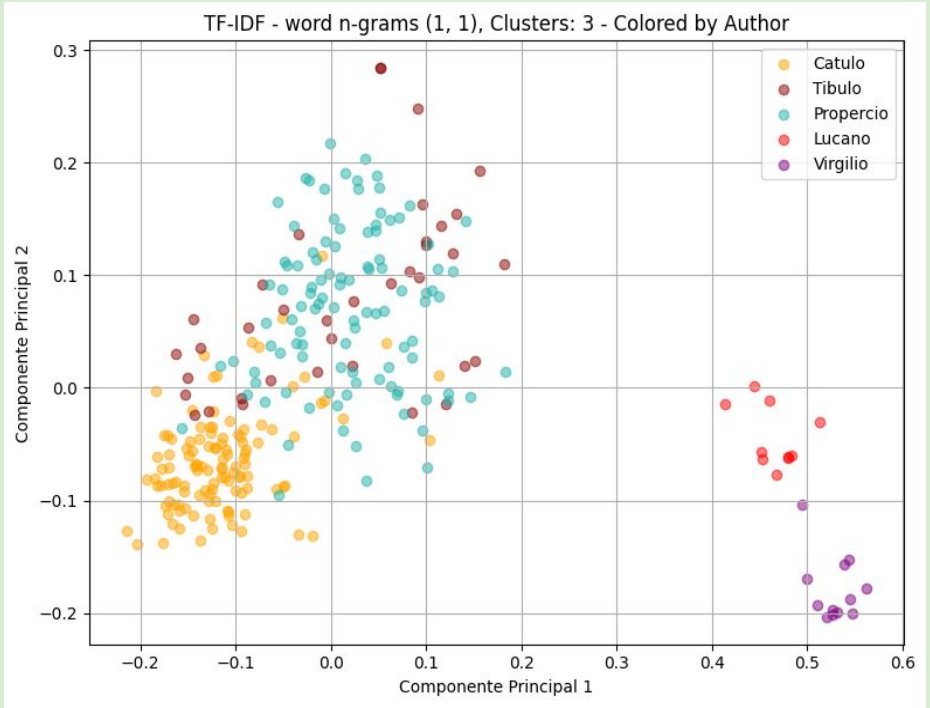




Scatter plot of document clustering by K Means using a TF IDF matrix of 1 word n-grams



Colored by cluster

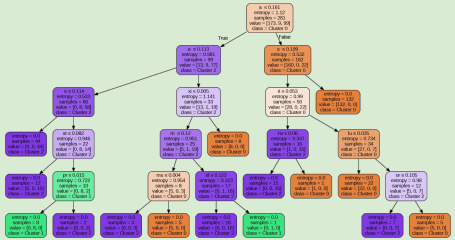


Colored by author

Extracting features: metrics on the Decision Trees

- **Importance:** It tells us how useful a feature is for making good predictions. It helps to see how much the model improves thanks to that feature.
- **Information Gain (IG):** It measures how much the disorder in the data is reduced by knowing a feature. In other words, how much better we understand the data thanks to that feature.
- **Information Gain Ratio (IGR):** Information Gain tends to favor attributes with many values. This measure improves it by correcting the bias.

- **The ideal outcome is to achieve an even distribution in the measurement of importance across several features.**



Most important features at the level of character n-grams (I, IG, IGR) using the frequency matrix method and Stopwords filtering

The feature importance is unbalanced.

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

Most important features at the level of word n-grams (I, IG, IGR) using the frequency matrix method and Stopwords filtering

The feature importance is unbalanced.

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

Most important features at the level of character n-grams (I, IG, IGR) using the TF-IDF matrix method

The feature importance is balanced.

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

Most important features at the level of word n-grams (I, IG, IGR) using the using the TF-IDF matrix method

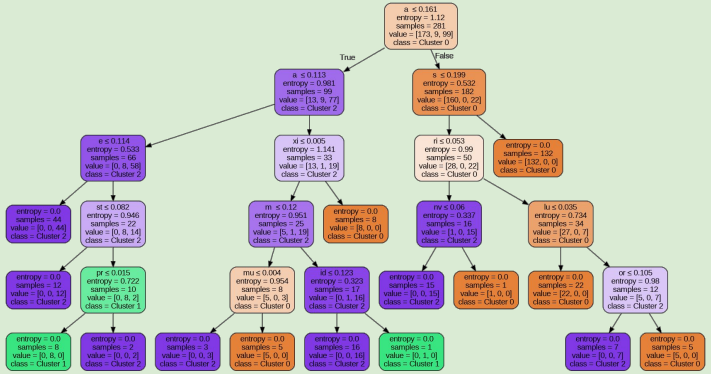
The feature importance is balanced.

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

Conclusions

- Results showed variations based on text preprocessing techniques:
  - A simple frequency matrix produced better Silhouette scores.
  - TF-IDF weighting produced Silhouette scores closer to zero, albeit with a more balanced distribution of Importance among different features.
- 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).

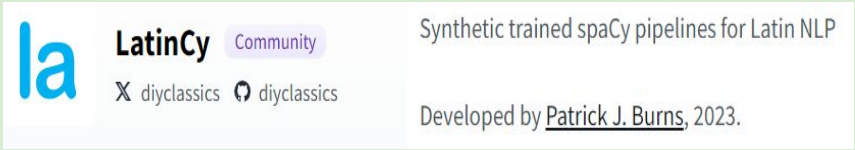


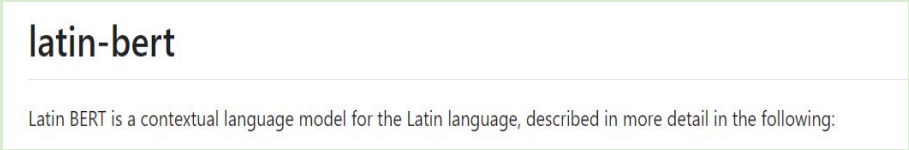
Conclusions

- The scatter plots obtained showed a match with the stylistic distribution reported by Forstall et al. (2011) who used a Support Vector Machine (SVM) approach to test the influence of Catullus on the poetry of Paul the Deacon.

Conclusions

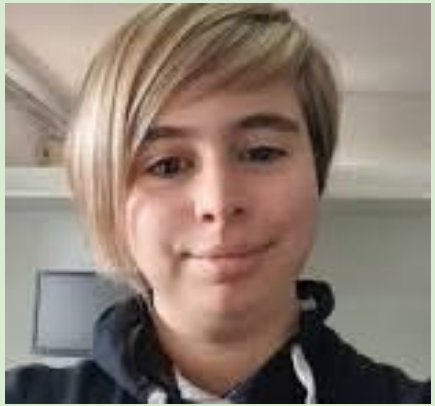
- The clustering tasks produced positive results by grouping authors of different styles into distinct and well-defined clusters.
- Features obtained from Decision Trees were not very promising, so we might need to investigate other techniques:
  - such as variable ranges of character and word n-grams,
  - 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, we could explore other representation techniques using Embeddings.





A Special Thanks to My Colleagues

Luciana Tanevitch



Juliana Delle Ville

Joaquín Bogado

Juana Borrelli Zara

Carlos J. Nusch

carlosnusch@prebi.unlp.edu.ar

UNIVERSIDAD NACIONAL DE LA PLATA

PREBI prebi.unlp.edu.ar SEDICI sedici.unlp.edu.ar

Lifia

IIBICRIT

LIDIC