UNIVERSIDAD NACIONAL DE LA PLATA

UNIVERSIDAD NACIONAL DEL NORDESTE

Breve Introducción a la Minería de Textos

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

Lifia

Mg. Carlos J. Nusch

1 de octubre 2024

Objetivos del Curso

Objetivos del Curso*

- Comprender los fundamentos del PLN.
- Aprender técnicas básicas de minería de textos y análisis automático.
- Familiarizarse con herramientas en Python para PLN.
- Ejecutar algunos ejemplos breves de PLN
  - Nubes de Palabras
  - Modelado de Tópicos

* Quiero agradecer muy especialmente a mi directora, la Dra. Leticia Cagnina de la UNSL por haberme ayudado en la revisión y corrección de este curso.

¡Manos a la obra!


poderosa
pasión
universales
es
una
de
sobre
alegría
escribieron
el
antigua
Amor
y
tiempo
culturas
trascendiendo
Roma
En
Los
Propercio
poemas
la
emoción
trae
Catulo
dolor
son

	Tópico 1	Tópico 2	Tópico 3	Tópico 4
Palabra 1	amor	muerte	tractor	política
Palabra 2	pasión	entierro	agricultura	gobierno
Palabra 3	corazón	luto	motor	elección
Palabra 4	romántico	fallecimiento	mecánica	democracia
Palabra 5	sentimiento	pérdida	ruedas	ley
Palabra 6	pareja	cementerio	campo	voto
Palabra 7	beso	tristeza	fuerza	ciudadanos
Palabra 8			máquina	

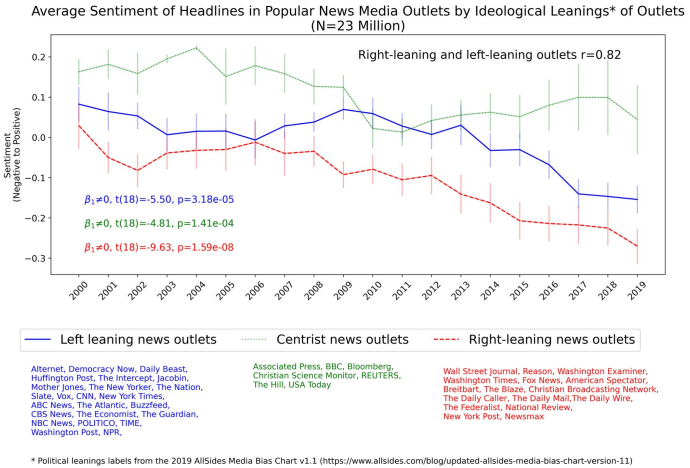
Algunos deslindes terminológicos

Antes de comenzar es necesario realizar una serie de precisiones terminológicas para comprender mejor qué campos del conocimientos abarca el contenido del curso:

- Análisis Automático de Textos
- Minería de Textos
- Procesamiento del Lenguaje Natural
- Otros términos y áreas cercanas:
  - Lingüística de Corpus,
  - Recuperación de la Información,
  - Lingüística Computacional,
  - Aprendizaje Automático,
  - Aprendizaje Profundo, etc.

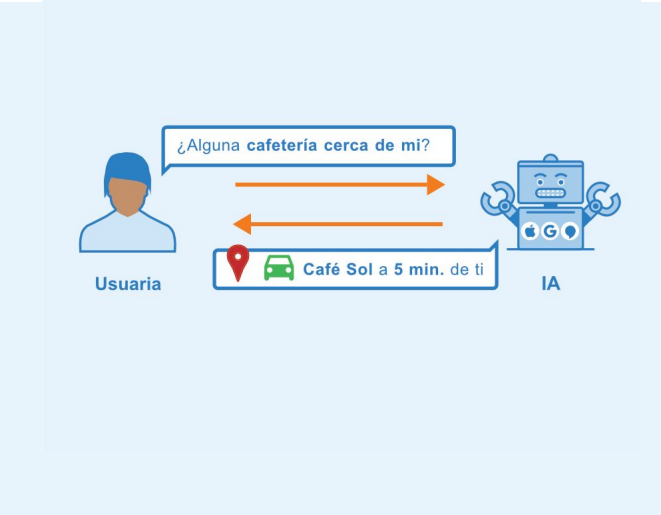
Procesamiento del Lenguaje Natural

- Tareas de comprensión del lenguaje: traducción automática, resumen de textos, extracción de información y análisis de sentimientos, donde el sistema intenta entender el texto proporcionado.



Procesamiento del Lenguaje Natural

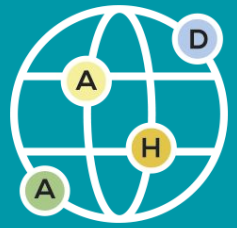
- Tareas de generación de lenguaje: creación de texto coherente y relevante a partir de datos no lingüísticos o en respuesta a alguna entrada de lenguaje natural.



Otras áreas relacionadas

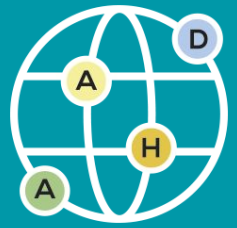
- **Lingüística de Corpus:** es un área de estudio dentro de la lingüística que utiliza **colecciones grandes y estructuradas de textos reales (corpus)** para el análisis del lenguaje. Se enfoca en el **examen sistemático de datos lingüísticos naturales**, utilizando métodos estadísticos y computacionales para identificar patrones, tendencias y características del uso del lenguaje en diferentes contextos y géneros.
- La Lingüística de Corpus apoya investigaciones en **semántica, sintaxis, lexicografía**, y más, permitiendo un acercamiento empírico y descriptivo a la investigación lingüística.

Otras áreas relacionadas

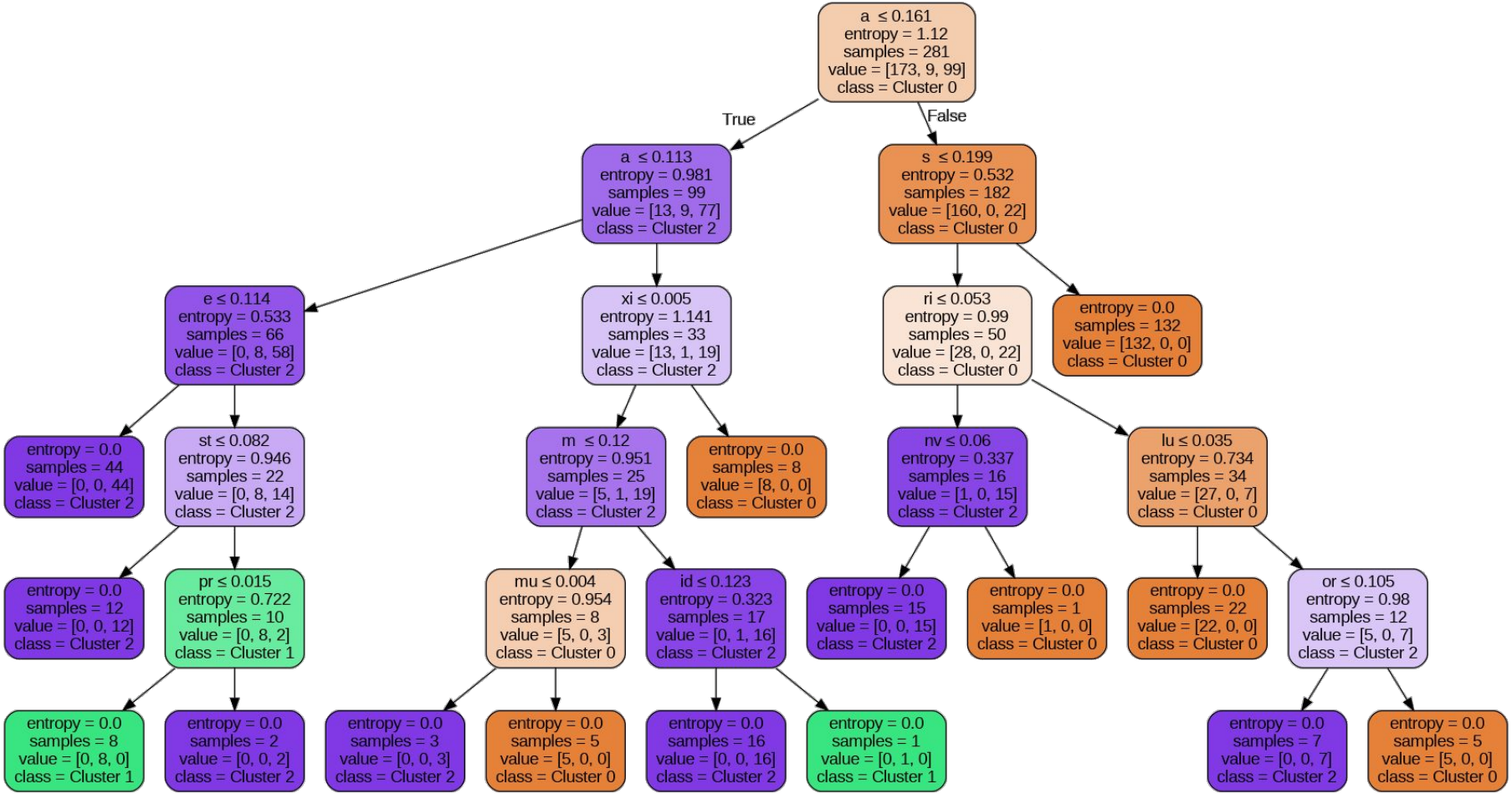


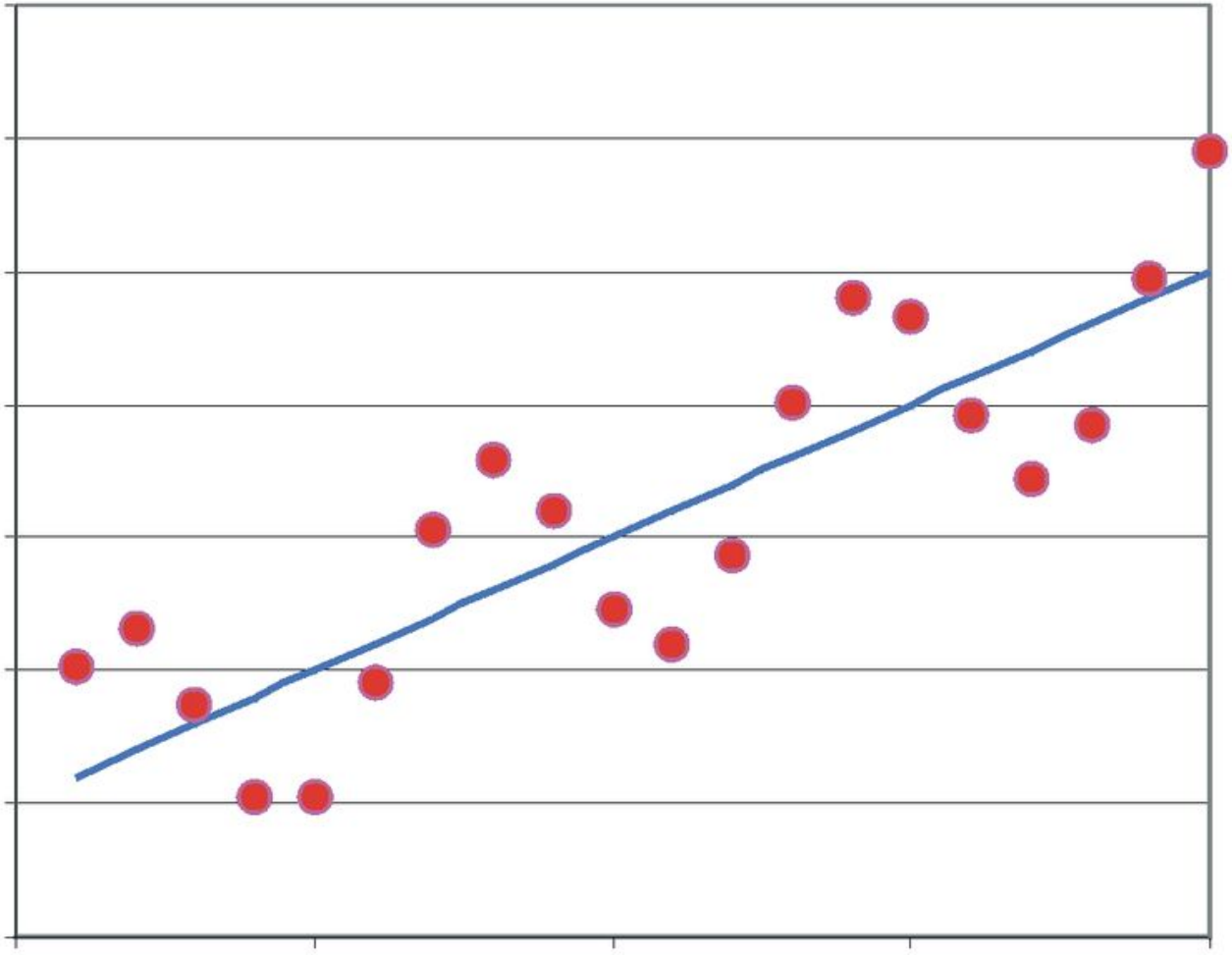
- **Recuperación de la Información:** es un campo de la informática que se ocupa de la **organización, almacenamiento, búsqueda y recuperación** de información. El objetivo principal es encontrar material (usualmente documentos) de una naturaleza no estructurada (texto libre) que satisfaga una necesidad de información dentro de grandes colecciones (a menudo almacenadas en bases de datos o en la web).
- La RI utiliza técnicas y algoritmos para mejorar la precisión y la relevancia de los resultados de búsqueda, abarcando desde la búsqueda en texto completo hasta sistemas más complejos de búsqueda semántica y filtrado de información.

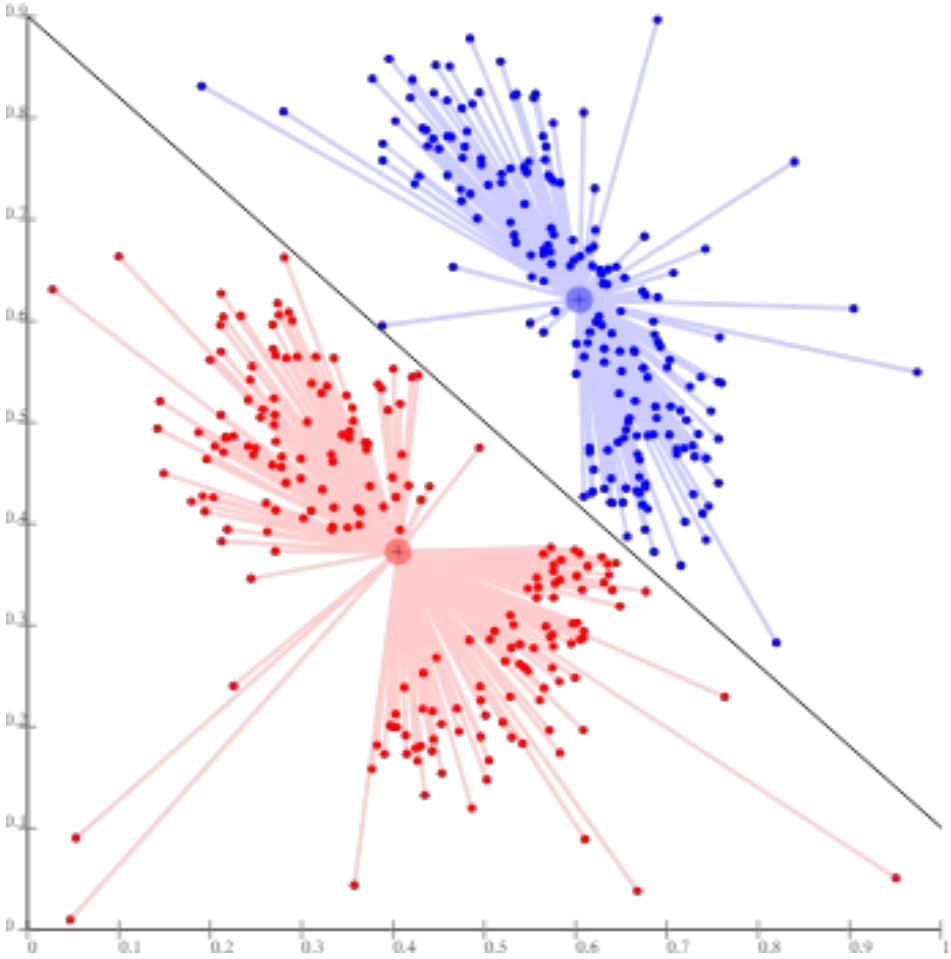
Otras áreas relacionadas

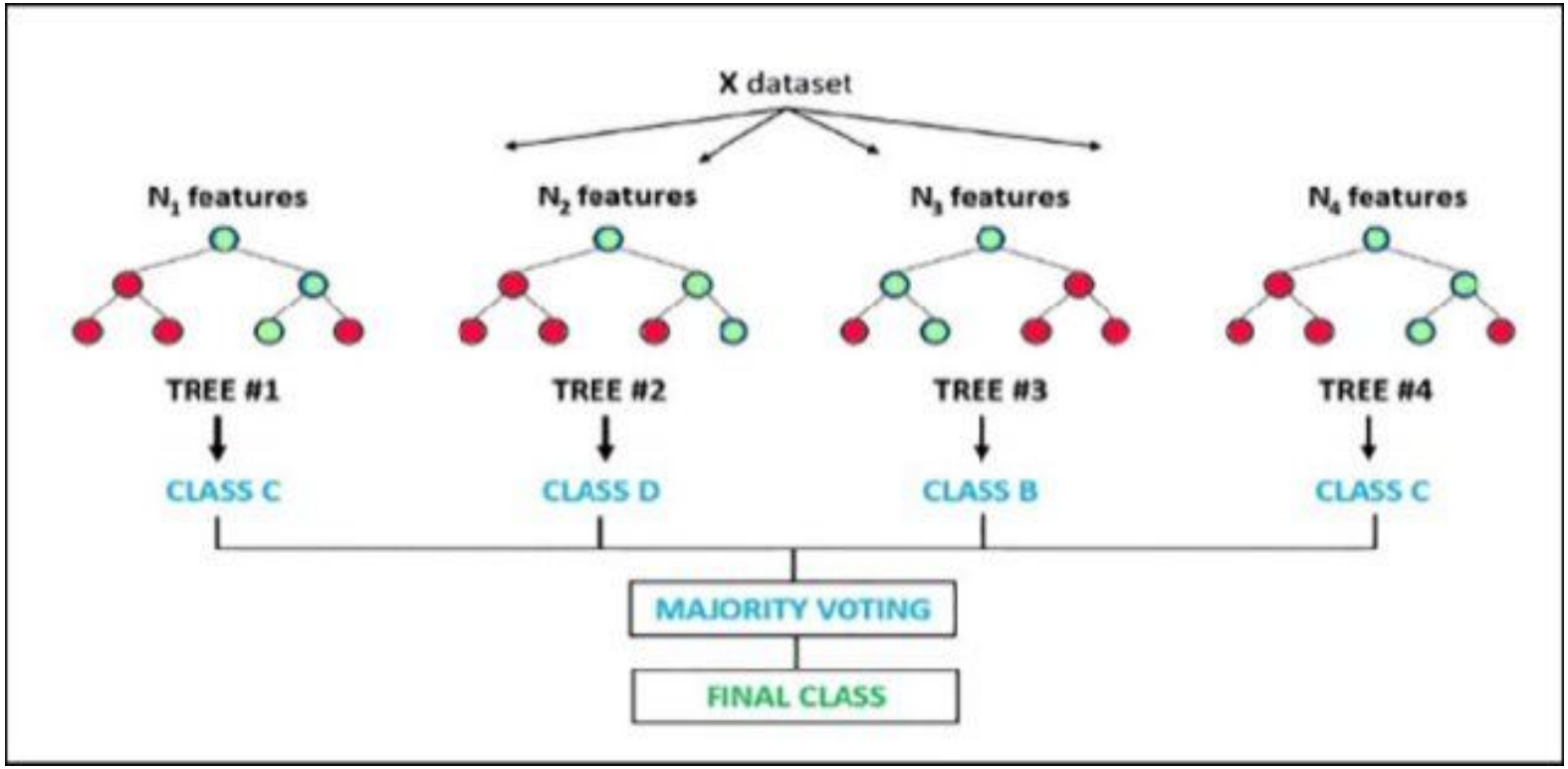


- **Aprendizaje Automático:** es una rama de la inteligencia artificial que se enfoca en el **desarrollo de algoritmos que permiten a las máquinas mejorar su rendimiento** y aprender características. En el caso de aprendizaje supervisado se mejora el desempeño en una tarea dada con un previo entrenamiento con datos (experiencia). Se basa en la idea de que los sistemas pueden *aprender* de los datos, identificar patrones y tomar decisiones con mínima intervención humana. El aprendizaje automático se aplica en una variedad de dominios, desde el reconocimiento de voz y la visión por computadora hasta el filtrado de correos electrónicos no deseados y la recomendación personalizada de productos.
- Las técnicas específicas del aprendizaje automático incluyen: Árboles de Decisión, Máquinas de Vectores de Soporte (SVM), Regresión Lineal, Regresión Logística, K-Vecinos más Cercanos (K-NN), Bosques Aleatorios, Algoritmos de Agrupamiento.

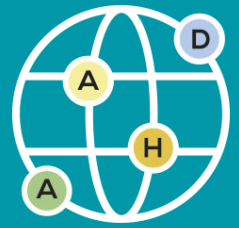

a ≤ 0.161
entropy = 1.12
samples = 281
value = [173, 9, 99]
class = Cluster 0
True
False
a ≤ 0.113
entropy = 0.981
samples = 99
value = [13, 9, 77]
class = Cluster 2
s ≤ 0.199
entropy = 0.532
samples = 182
value = [160, 0, 22]
class = Cluster 0
e ≤ 0.114
entropy = 0.533
samples = 66
value = [0, 8, 58]
class = Cluster 2
xi ≤ 0.005
entropy = 1.141
samples = 33
value = [13, 1, 19]
class = Cluster 2
ri ≤ 0.053
entropy = 0.99
samples = 50
value = [28, 0, 22]
class = Cluster 0
entropy = 0.0
samples = 132
value = [132, 0, 0]
class = Cluster 0
entropy = 0.0
samples = 44
value = [0, 0, 44]
class = Cluster 2
st ≤ 0.082
entropy = 0.946
samples = 22
value = [0, 8, 14]
class = Cluster 2
m ≤ 0.12
entropy = 0.951
samples = 25
value = [5, 1, 19]
class = Cluster 2
entropy = 0.0
samples = 8
value = [8, 0, 0]
class = Cluster 0
nv ≤ 0.06
entropy = 0.337
samples = 16
value = [1, 0, 15]
class = Cluster 2
lu ≤ 0.035
entropy = 0.734
samples = 34
value = [27, 0, 7]
class = Cluster 0
entropy = 0.0
samples = 12
value = [0, 0, 12]
class = Cluster 2
pr ≤ 0.015
entropy = 0.722
samples = 10
value = [0, 8, 2]
class = Cluster 1
mu ≤ 0.004
entropy = 0.954
samples = 8
value = [5, 0, 3]
class = Cluster 0
id ≤ 0.123
entropy = 0.323
samples = 17
value = [0, 1, 16]
class = Cluster 2
entropy = 0.0
samples = 15
value = [0, 0, 15]
class = Cluster 2
entropy = 0.0
samples = 1
value = [1, 0, 0]
class = Cluster 0
entropy = 0.0
samples = 22
value = [22, 0, 0]
class = Cluster 0
or ≤ 0.105
entropy = 0.98
samples = 12
value = [5, 0, 7]
class = Cluster 2
entropy = 0.0
samples = 8
value = [0, 8, 0]
class = Cluster 1
entropy = 0.0
samples = 2
value = [0, 0, 2]
class = Cluster 2
entropy = 0.0
samples = 3
value = [0, 0, 3]
class = Cluster 2
entropy = 0.0
samples = 5
value = [5, 0, 0]
class = Cluster 0
entropy = 0.0
samples = 16
value = [0, 0, 16]
class = Cluster 2
entropy = 0.0
samples = 1
value = [0, 1, 0]
class = Cluster 1
entropy = 0.0
samples = 7
value = [0, 0, 7]
class = Cluster 2
entropy = 0.0
samples = 5
value = [5, 0, 0]
class = Cluster 0



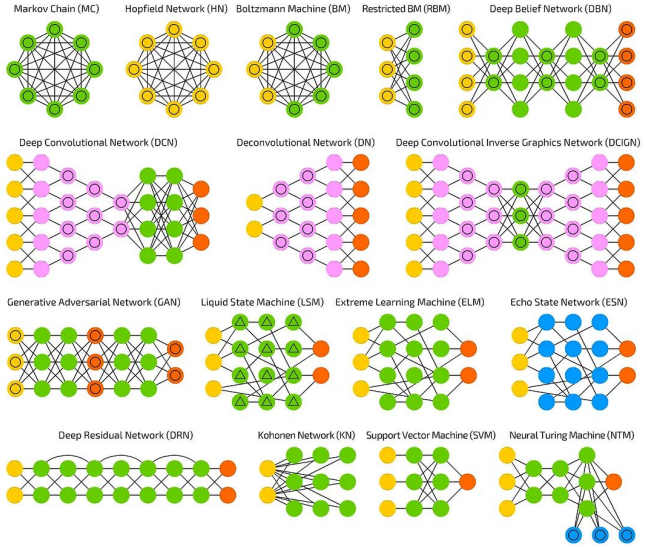

0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


X dataset
N1 features
N2 features
N3 features
N4 features
TREE #1
TREE #2
TREE #3
TREE #4
CLASS C
CLASS D
CLASS B
CLASS C
MAJORITY VOTING
FINAL CLASS

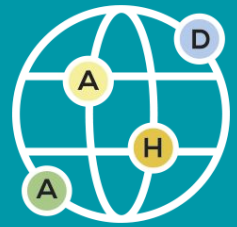
Otras áreas relacionadas



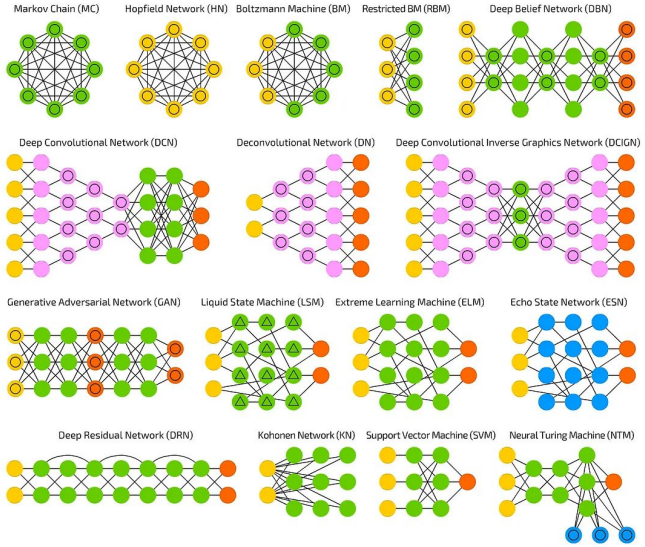
- **Aprendizaje Profundo:** es un subcampo del aprendizaje automático que utiliza **redes neuronales artificiales con múltiples capas (profundas)** para modelar abstracciones complejas en los datos. Inspirado por la estructura y función del cerebro humano, el aprendizaje profundo ha sido fundamental para realizar avances significativos en áreas desafiantes como el reconocimiento de imágenes, procesamiento del lenguaje natural y juegos estratégicos complejos.



Otras áreas relacionadas



- El aprendizaje profundo se caracteriza por su capacidad para aprender representaciones de datos a diferentes niveles de abstracción, permitiendo que el modelo mejore su precisión en tareas de clasificación, regresión y generación de datos.
- Las técnicas específicas del aprendizaje profundo incluyen: Redes Neuronales Convolucionales (CNN), Redes Neuronales Recurrentes (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Autoencoders, Redes Generativas Adversarias (GANs), Transformers.



¿Qué es un Modelo de Lenguaje?

¿Qué es un Modelo de Lenguaje?

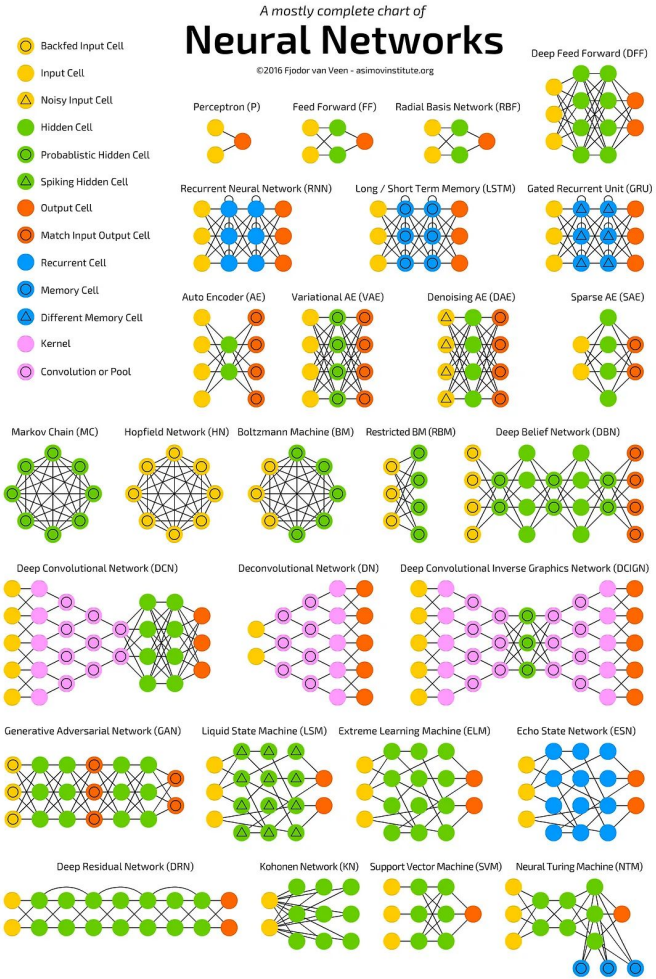
- ¿Qué es un modelo?
  - Un modelo en el contexto de la ciencia, la ingeniería y las matemáticas es una representación simplificada o una abstracción de un fenómeno, sistema o proceso del mundo real. Los modelos se crean para ayudarnos a comprender, predecir o controlar sistemas complejos al enfocar las características más importantes o relevantes de dichos sistemas.

¿Qué es un Modelo de Lenguaje?

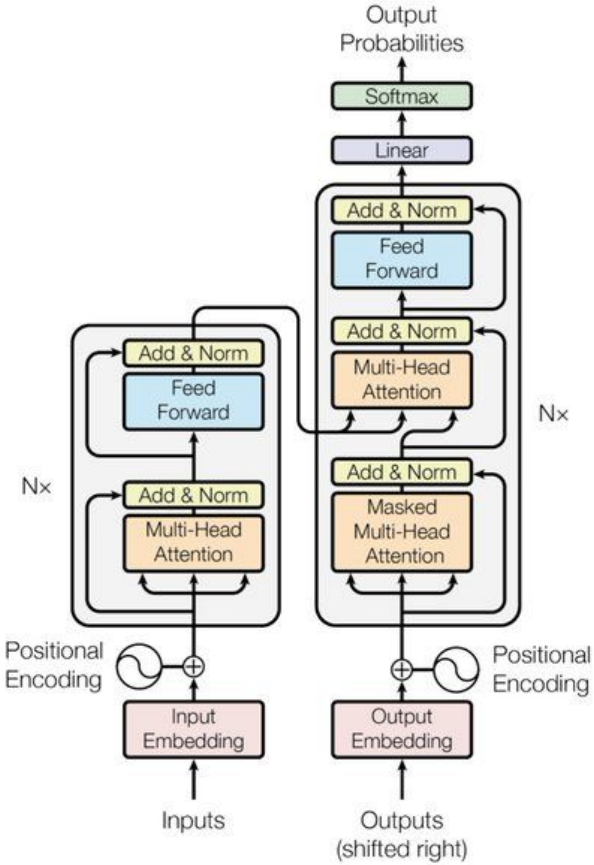
- **Modelo descriptivo**
  - *Un modelo descriptivo se utiliza para explicar o representar un sistema o fenómeno tal como es, proporcionando una descripción clara de su estructura o comportamiento actual."*
- **Modelo predictivo**
  - *Un modelo predictivo se diseña para anticipar resultados futuros utilizando datos históricos o patrones observados, permitiendo hacer proyecciones sobre el comportamiento de un sistema.*

¿Qué es un Modelo de Lenguaje?

- Un modelo de lenguaje es un sistema basado en inteligencia artificial que se entrena para comprender y generar datos (texto, imagen, sonido). Su objetivo principal es predecir la probabilidad de una secuencia de palabras o generar texto coherente.

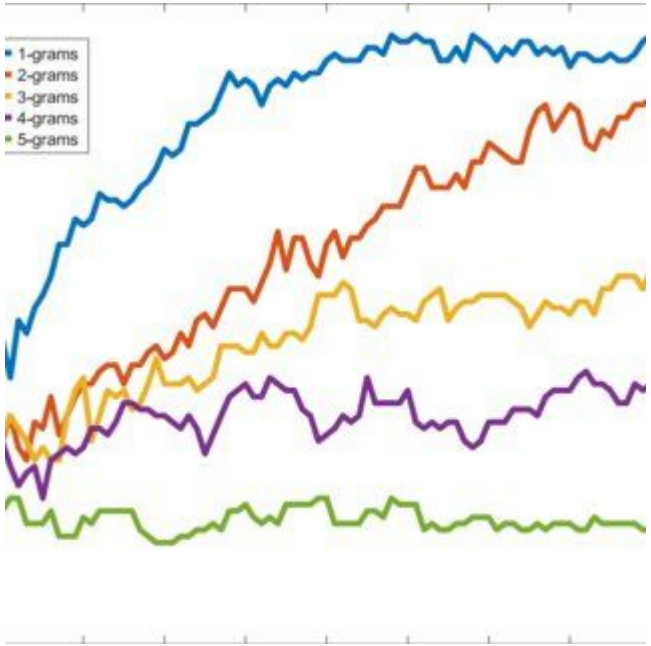


How do Transformers Work?



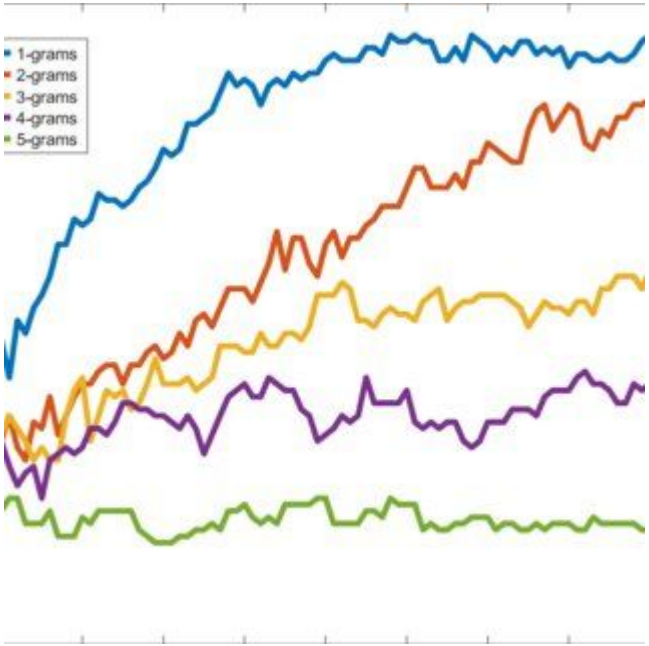
Otras áreas relacionadas

- **Estilometría:** es un campo de estudio que se enfoca en el **análisis cuantitativo del estilo literario**, principalmente a través del uso de métodos estadísticos y computacionales. Originada en los estudios literarios y la lingüística, la estilometría examina las características únicas de los textos escritos para identificar, atribuir o diferenciar a sus autores, así como para analizar tendencias literarias y evoluciones en el uso del lenguaje a lo largo del tiempo.



Otras áreas relacionadas

- Al analizar elementos como la frecuencia de palabras, el uso de estructuras gramaticales, patrones de puntuación y otros marcadores lingüísticos, los estilómetras pueden descubrir la *huella digital* de un autor en sus obras. Esto ha sido particularmente útil en casos de autoría disputada o desconocida, en el estudio de la evolución literaria y en la comprensión de cómo el contexto histórico y personal influye en el estilo de escritura.



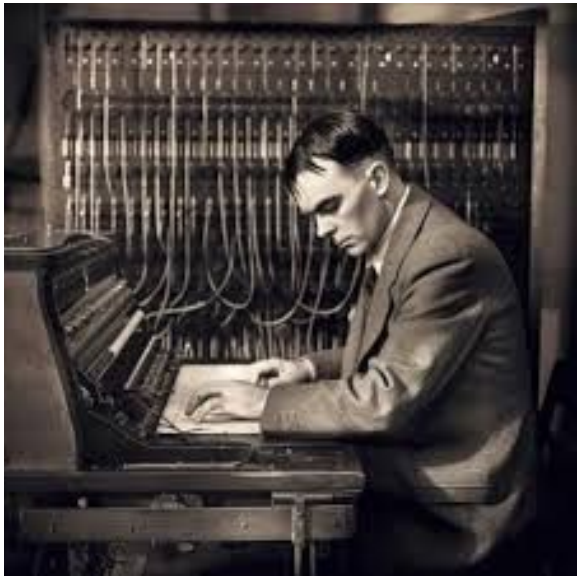
Otras áreas relacionadas

- Las Técnicas más usadas en estilometría incluyen:
  - Análisis de frecuencia de palabras
  - Análisis de n-gramas
  - Índice de diversidad léxica
  - Análisis de longitud de oraciones y párrafos
  - Análisis de función de palabras
  - Perfilado estilístico
  - Métricas de legibilidad
  - Análisis de patrones de puntuación

Hitos en la Historia del PLN

- El campo del Procesamiento PLN ha evolucionado desde reglas manuales hasta el uso de algoritmos de aprendizaje automático y aprendizaje profundo. Algunos hitos:

- **1950s: la Prueba de Turing**
  Alan Turing publica *Computing Machinery and Intelligence*, proponiendo la famosa Prueba de Turing como criterio de inteligencia en máquinas, lo que indirectamente impulsó la investigación en PLN.



Hitos en la Historia del PLN

- **1949-1958 Roberto Busa**
  Roberto Busa, intentó convertir la lexicología en un método de investigación científica con la intención de elaborar un índice de palabras o lematización de toda la obra del santo Tomás de Aquino. Tamaña empresa lo llevó a pensar en la informática como herramienta necesaria.

Hitos en la Historia del PLN

En 1949 se reunió con Thomas Watson, fundador de la compañía IBM para gestionar los ordenadores que usaría en la tarea de compilar toda la obra tomista.
Utilizó el sistema de tarjetas perforadas y clasificó 10.500.000 de palabras a lo largo de 30 años.
En 1958 presentó resultados iniciales en el pabellón de IBM en la Expo de Bruselas.

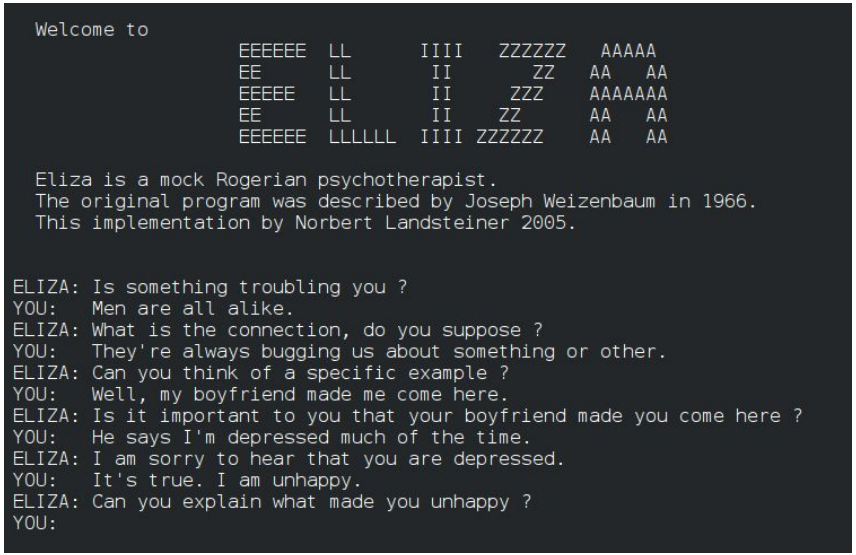
Hitos en la Historia del PLN

- **1949-1958 Roberto Busa**
  Examinó terminológicamente los 118 libros del santo, además de la obra de otros 61 autores relacionados con el filósofo. Este corpus fue enormemente útil para el trabajo y el estudio en varias áreas de la informática moderna:

  - cuantificación lingüística
  - traducción automatizada
  - hipertexto
  - indización automatizada
  - recuperación de información

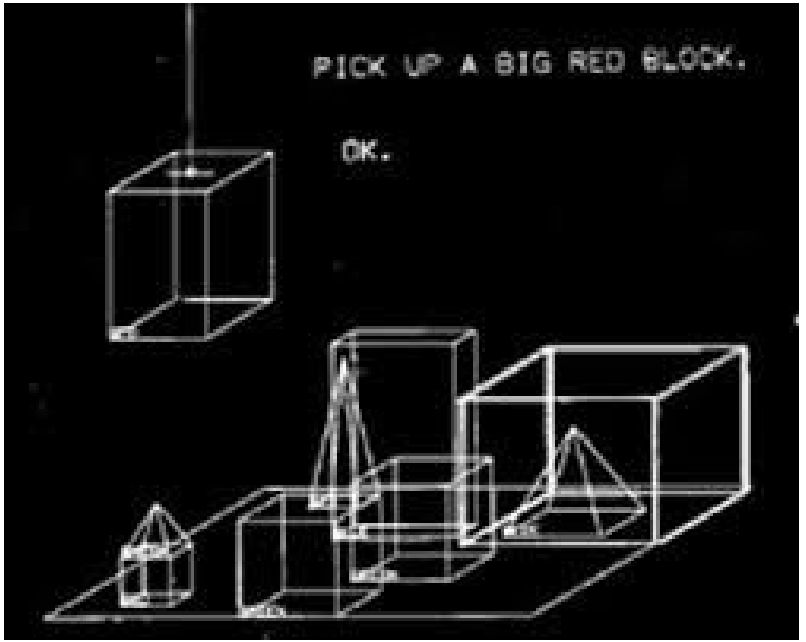
Hitos en la Historia del PLN

- **1960s: ELIZA**
  Joseph Weizenbaum crea ELIZA, un programa que simulaba una conversación al adoptar el rol de un psicoterapeuta Rogeriano. Se demostró cómo las máquinas podrían simular la comprensión del lenguaje humano.



Hitos en la Historia del PLN

- **1970s: SHRDLU**
  SHRDLU, desarrollado por Terry Winograd, era capaz de manipular objetos en un mundo de bloques a través de instrucciones en lenguaje natural, demostrando una comprensión más profunda del lenguaje y su relación con el entorno.



Hitos en la Historia del PLN

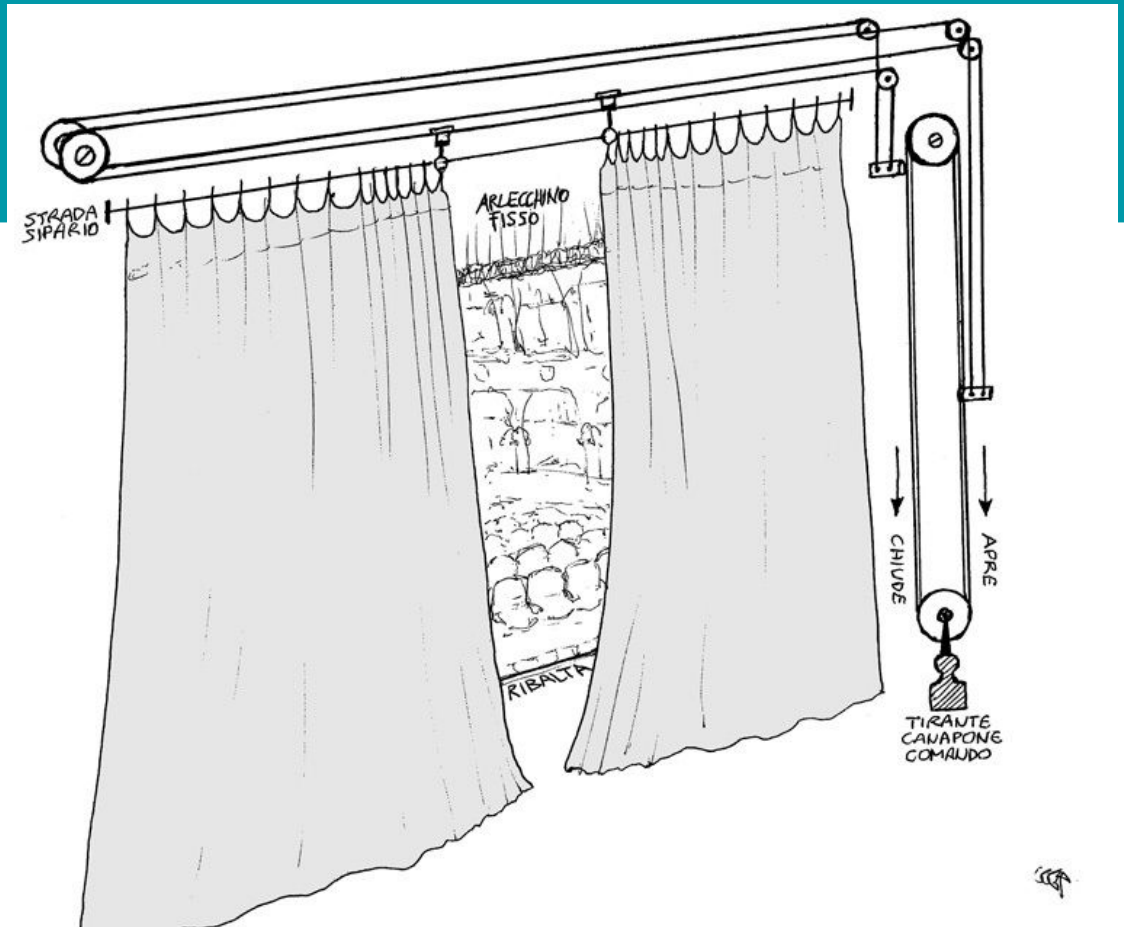
- **1980s: Sistemas Basados en Reglas**
  El foco estaba puesto en el desarrollo de gramáticas complejas y lexicones para analizar y entender el texto.
  Se utilizan actualmente y consisten en conjuntos de reglas codificadas a mano para interpretar texto. Son excelentes para tareas estructuradas que no cambian mucho, como un chat de soporte al cliente.

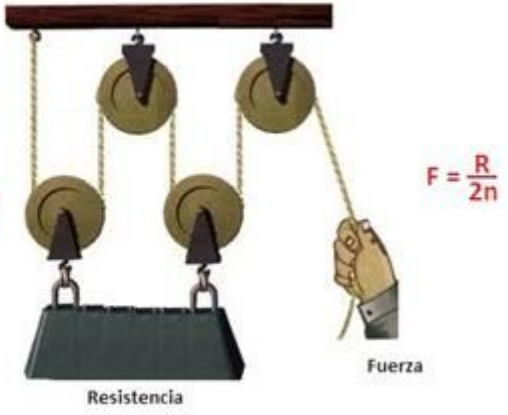
Hitos en la Historia del PLN

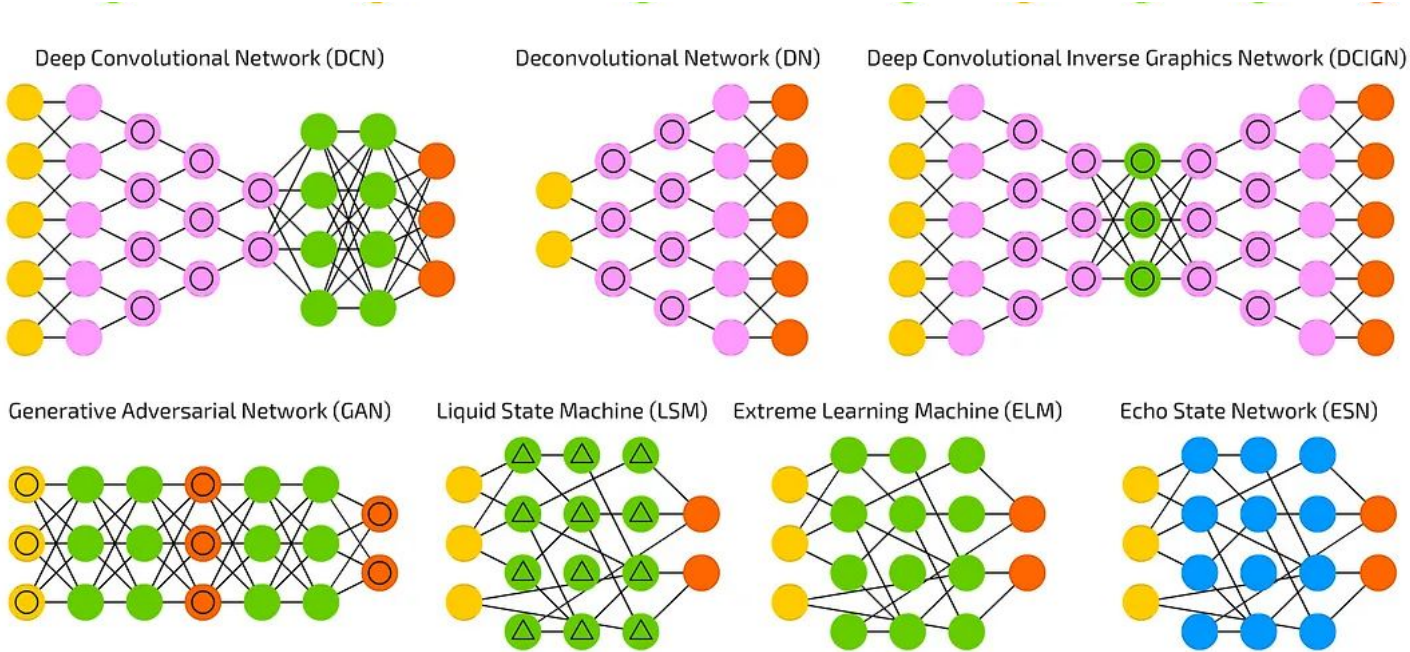
- **1990s: Aprendizaje Estadístico y Machine Learning**
  Llegan los métodos estadísticos y de aprendizaje automático lo cual transforma el PLN, permitiendo a los sistemas aprender de grandes cantidades de datos de texto en lugar de depender exclusivamente de reglas codificadas manualmente.

Hitos en la Historia del PLN

Los Métodos Estadísticos *miran* los datos e infieren estadísticamente lo que es más probable que sea cierto.
Los sistemas de Aprendizaje Automático o *Machine Learning* aprenden de sus experiencias, ajustando sus métodos a medida que reciben más datos. Inician con un entendimiento básico de un idioma y se vuelven más inteligentes, versátiles y precisos con el tiempo.


STRADA SIPARIO
ARLECCHINO FISSO
RIBALTA
CHIUDE
APRE
TIRANTE CANAPONE COMANDO


F = R/2n
Resistencia
Fuerza

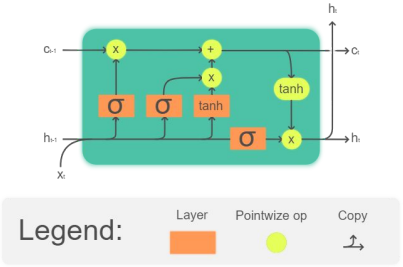

Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)

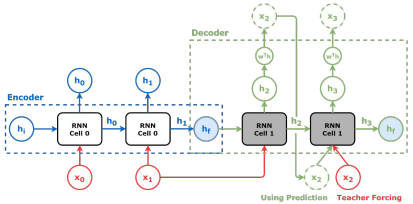
Hitos en la Historia del PLN

- **2000s: Modelos de Espacio Vectorial y Aprendizaje Profundo**
  La implementación de modelos de espacio vectorial para el procesamiento de texto con técnicas como Frecuencia de Término-Frecuencia Inversa de Documentos (TF-IDF) y Análisis Semántico Latente (LSA), mejoró significativamente la búsqueda y recuperación de información.

Hitos en la Historia del PLN

- **Finales de la década:** el resurgimiento de las redes neuronales en forma de aprendizaje profundo comenzó a tener un impacto significativo en el PLN, liderando las mejoras en la traducción automática, reconocimiento de voz y generación de texto.



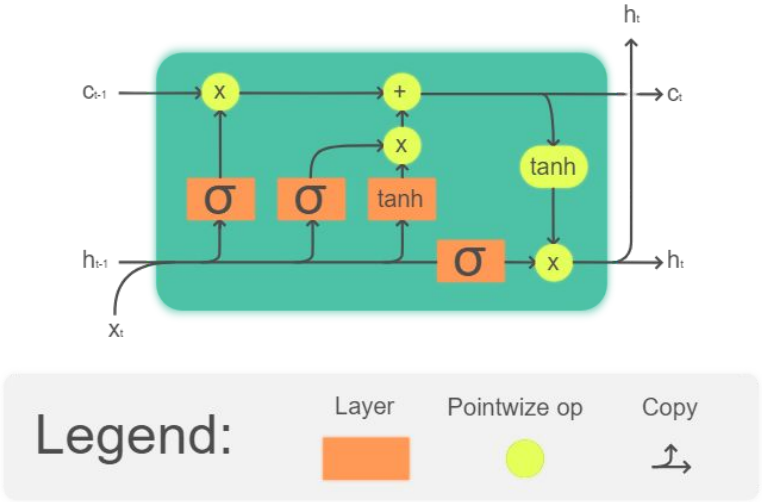


MARIANNMT

Fast Neural Machine Translation in C++

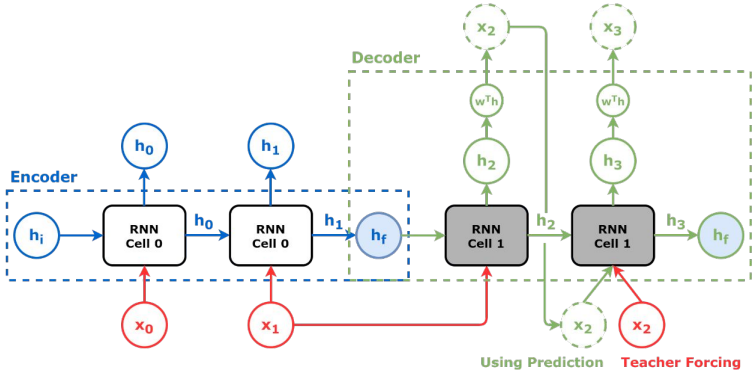
Hitos en la Historia del PLN

- **Encoders-Stacked Long Short-Term Memory (LSTM):** una variante de las redes neuronales recurrentes (RNN), comenzaron a mostrar potencial para secuencias de texto largas, como las que se encuentran en traducción automática. Tienen una estructura de *celdas de memoria* que son capaces de retener información por períodos largos de tiempo.



Hitos en la Historia del PLN

- **Modelos de secuencia a secuencia (Seq2Seq):** de 2014 pero comenzó a gestarse al final de la década del 2000. Este enfoque permitió entrenar modelos de traducción de extremo a extremo utilizando redes neuronales recurrentes para mapear oraciones entre lenguajes. Pueden atender a diferentes puntos de una oración. Sentó las bases para los avances de Google Translate , Transformers y otros.



Hitos en la Historia del PLN

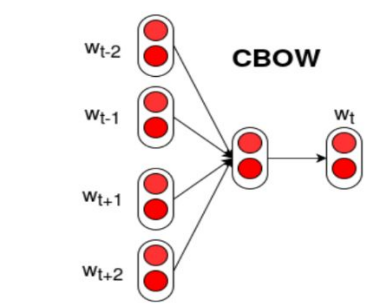
- **Google Translate (2006 - 2016):** inicialmente utilizaba traducción automática estadística (SMT), pero empezó a explorar modelos basados en redes neuronales hacia el final de la década del 2000. En 2016 adoptó completamente un sistema de traducción automática neuronal (NMT), basado en deep learning.

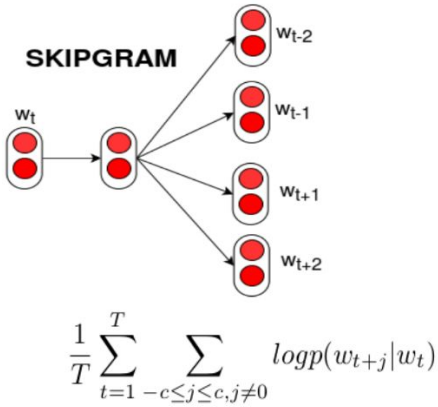
Hitos en la Historia del PLN

- **Marian NMT:** (con prototipos a finales de los 2000) se popularizó más adelante, Marian es un motor de traducción automática neuronal basado en deep learning, optimizado para el uso en GPU.

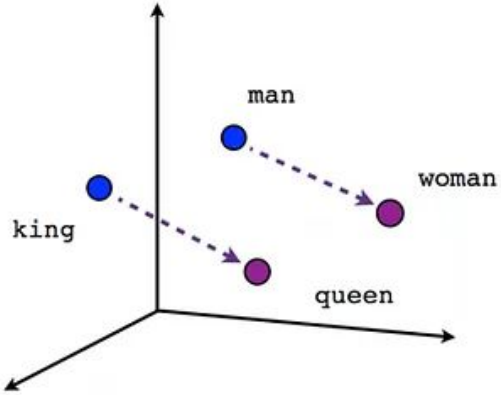
Hitos en la Historia del PLN

- **2010s: Transformadores y Modelos de Lenguaje Pre Entrenados**
- **2013** Se introduce **word2vec**, una técnica revolucionaria para la generación de vectores de palabras (de entre 100 y 300 dimensiones), lo que permite capturar similitudes semánticas y sintácticas.

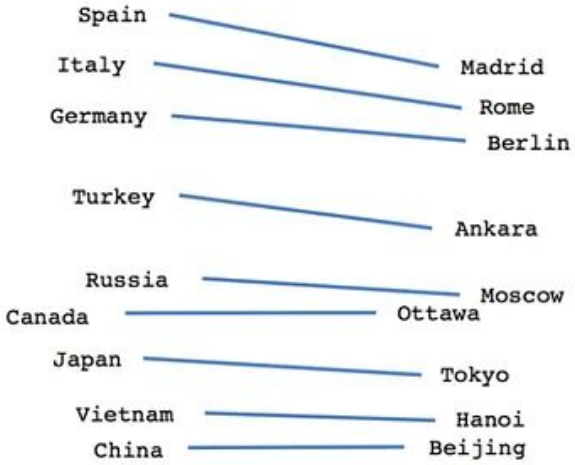




Word2vec



Male-Female



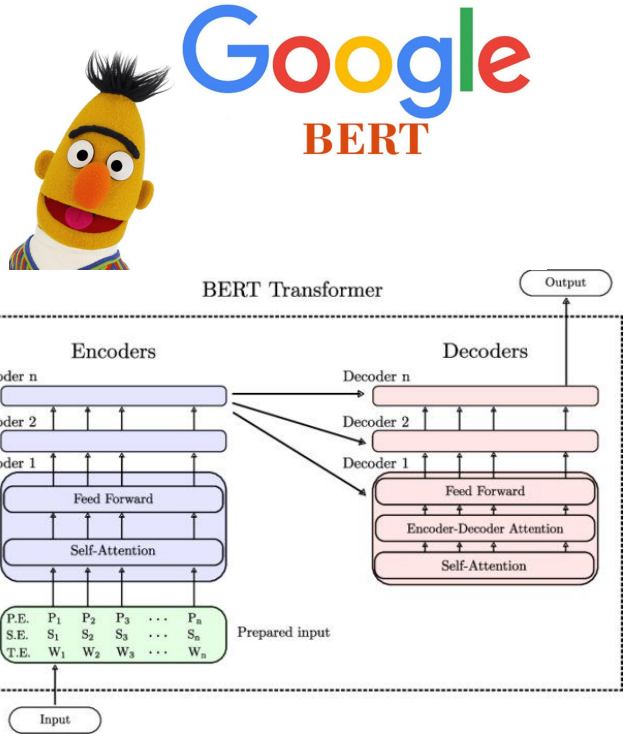
Country-Capital

Hitos en la Historia del PLN

- **2016 FastText**, una biblioteca para el aprendizaje de representaciones de palabras y clasificación de texto, fue introducida por investigadores de Facebook AI Research (FAIR). FastText es más eficiente en cuanto al tiempo de entrenamiento como de precisión. Misma cantidad de dimensiones.
- Ofrece la capacidad de generar representaciones vectoriales para palabras fuera del vocabulario al considerar las subunidades de las palabras (n-gramas de caracteres).

Hitos en la Historia del PLN

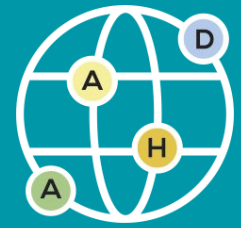
- **2018: Google introduce BERT** (Bidirectional Encoder Representations from Transformers), un modelo basado en la arquitectura de transformadores que establece nuevos estándares en la comprensión del lenguaje, siendo capaz de entender el contexto de las palabras en el texto de manera bidireccional.



Hitos en la Historia del PLN

- **GPT (Generative Pretrained Transformer)** OpenAI lanza GPT y sus versiones subsiguientes, modelos que pueden generar texto coherente y convincente, impulsando la generación de lenguaje y abriendo nuevas posibilidades en la interacción hombre-máquina.

Hitos en la Historia del PLN



Algunas diferencias:

- **BERT:** es bidireccional (más precisamente, deeply bidirectional), ya que tiene en cuenta tanto el contexto a la izquierda como a la derecha de cada palabra durante el entrenamiento.
  - Esto le permite tener una visión completa del contexto en el que se encuentra cada palabra.

OpenAI

- **GPT:** es unidireccional o autoregresivo. Solo tiene en cuenta las palabras anteriores en la secuencia y predice la siguiente palabra de manera progresiva.

Hitos en la Historia del PLN

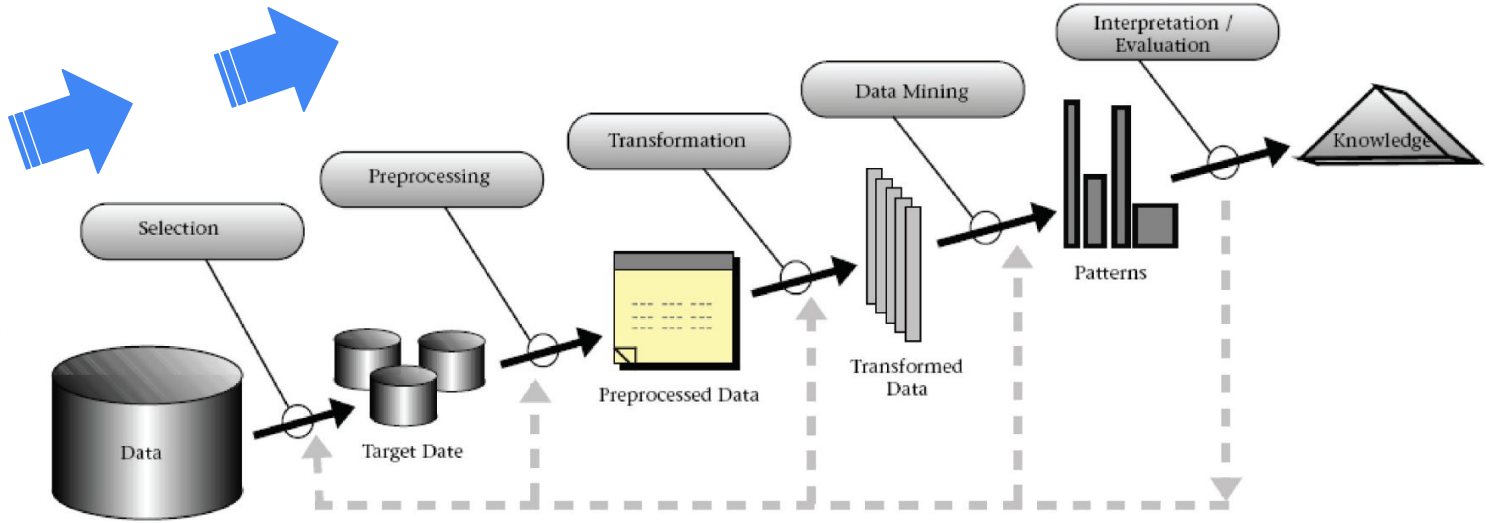
Algunas diferencias:

- **BERT:** está diseñado principalmente para tareas de comprensión del lenguaje. Se utiliza en tareas como clasificación de texto, reconocimiento de entidades, respuesta a preguntas y análisis de sentimientos. BERT es excelente para tareas que requieren una buena comprensión del contexto completo de una oración.

OpenAI

- **GPT:** se utiliza más comúnmente en tareas de generación de texto debido a su naturaleza autoregresiva. Es adecuado para tareas como generación de diálogos, escritura creativa o finalización de texto, ya que está diseñado para predecir la próxima palabra en una secuencia.

Minería de Textos



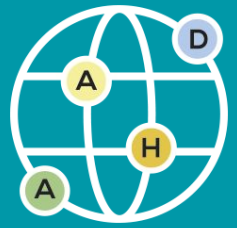
Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996, August). Knowledge Discovery and Data Mining: Towards a Unifying Framework. In *KDD* (Vol. 96, pp. 82-88).

Alnoukari, M., & El Sheikh, A. (2012). Knowledge discovery process models: from traditional to agile modeling. In *Business Intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications* (pp. 72-100). IGI Global.

Minería de Textos: qué veremos

- **Preprocesamiento**
  - Importancia del preprocesamiento.
- **Tokenización y eliminación de palabras vacías.**
  - Lematización y stemización.
  - Ejemplos prácticos de preprocesamiento.
- **Exploración de Datos**
  - Enriquecimiento
    - Etiquetado de Partes del Discurso (POS Tagging)
    - Detección de Entidades Nombradas (NER)
- **Extracción de Características y Vectorización de Texto**
  - Bag of Words.
  - TF-IDF.
- **Aplicaciones prácticas**
- **Nube de Palabras**
- **Herramientas y Librerías en Python para PLN**
  - NLTK.
  - spaCy.
- **Modelado de Tópicos**
  - Latent Dirichlet Allocation (LDA)
- **Algunos ejemplos de uso propios**

Minería de Textos: etapas

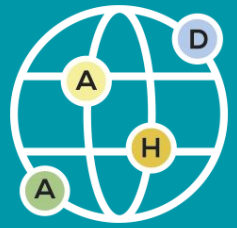


Implica varias etapas y técnicas:

- **Preprocesamiento de Texto:** el primer paso es la limpieza y preparación de los datos de texto, que puede incluir la eliminación de formato, corrección ortográfica, eliminación de palabras vacías, tokenización, lematización y stemización.

```
# Tokenización, lematización y stemming
tokens = [token.text for token in doc]
lemmas = [token.lemma_ for token in doc]
stems = [stemmer.stem(token.text) for token in doc]

# Obtener n-gramas de caracteres (por ejemplo, bigramas o trigramas)
n = 3  # Puedes cambiar 'n' al número de caracteres que desees para el n-grama
ngrams_caracteres = [obtener_ngrams_caracteres(token, n) for token in tokens]
```

Preprocesamiento

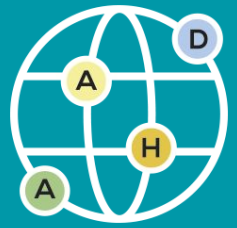


Es una etapa fundamental para limpiar y preparar los datos de texto para el análisis, con el objeto de mejorar significativamente la eficacia y la precisión de los resultados obtenidos. ¿En qué consiste?

Limpieza de Datos: el texto recopilado de fuentes como Internet, correos electrónicos, redes sociales o documentos, a menudo viene con una gran cantidad de ruido e irregularidades: errores ortográficos, abreviaturas, jerga, etiquetas HTML, información no relevante, entre otros. La limpieza es esencial para eliminar el ruido y asegurar que el texto esté en un formato estándar.

Normalización: el texto puede contener variaciones que no afectan su significado, como mayúsculas/minúsculas ("Casa" vs. "casa") o diferentes formas de una palabra (por ejemplo, "correr", "corriendo"). La normalización busca reducir estas variaciones y convertir el texto a una forma base o estándar (conversión a minúsculas, la lematización y la stemización (eliminar prefijos y sufijos para reducir las palabras a una forma raíz).

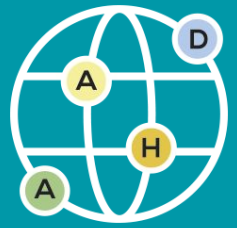
Preprocesamiento



Eliminación de Palabras Vacías: las palabras vacías son términos comunes como "el", "la", "y", "o", que aparecen con frecuencia en el texto pero generalmente no aportan significado relevante para el análisis. Eliminar estas palabras reduce el volumen de datos a procesar y puede mejorar la relevancia de los resultados en tareas como la extracción de temas o la clasificación de textos.

Tokenización: implica dividir el texto en unidades más pequeñas, como palabras o frases. La tokenización es fundamental para transformar el texto continuo en datos estructurados que puedan ser analizados. Facilita la identificación de elementos clave en el texto y es un paso previo necesario para muchas técnicas de PLN, como el análisis de sentimientos el modelado de tópicos.

Tokenización y Eliminación de Palabras Vacías



- La tokenización y la eliminación de palabras vacías son dos procesos fundamentales que ayudan a preparar los datos para un análisis más eficiente y efectivo.
- Tokenización: es el proceso de dividir el texto en unidades más pequeñas, llamadas tokens, que pueden ser palabras, frases o incluso caracteres. Este paso es esencial porque el texto, en su forma original, es solo una larga secuencia de caracteres sin una estructura clara que las computadoras puedan procesar directamente para tareas de PLN.
- Al tokenizar el texto, se convierte en una lista de elementos discretos sobre los cuales se pueden aplicar técnicas analíticas.

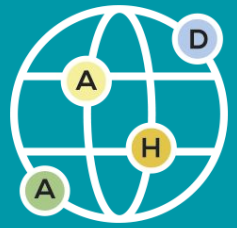
Eliminación de Palabras Vacías

- Las palabras vacías son términos comunes en un idioma que aparecen con mucha frecuencia pero generalmente no aportan significado relevante al contexto del texto (mayormente en tareas en las que se analizan ciertos contenidos temáticos). Estas incluyen preposiciones, conjunciones, artículos y pronombres, como "y", "en", "el", "pero", entre otros.
- Aunque son esenciales para la estructura gramatical en la comunicación humana, su presencia puede ser innecesaria o incluso ruidosa para muchas tareas de PLN, como el modelado de tópicos o la clasificación de texto.

Eliminación de Palabras Vacías

- La eliminación de palabras vacías reduce el tamaño del dataset de texto y puede mejorar la eficiencia de los procesos de análisis al concentrarse en palabras que tienen más probabilidades de ser significativas para el análisis.
- No siempre es necesario eliminarlas y dependiendo de la tarea de PLN que se esté realizando se puede decidir si eliminar o no las palabras vacías.
- En algunas aplicaciones, como el análisis de estilo o la autoría, estas palabras pueden proporcionar información valiosa.

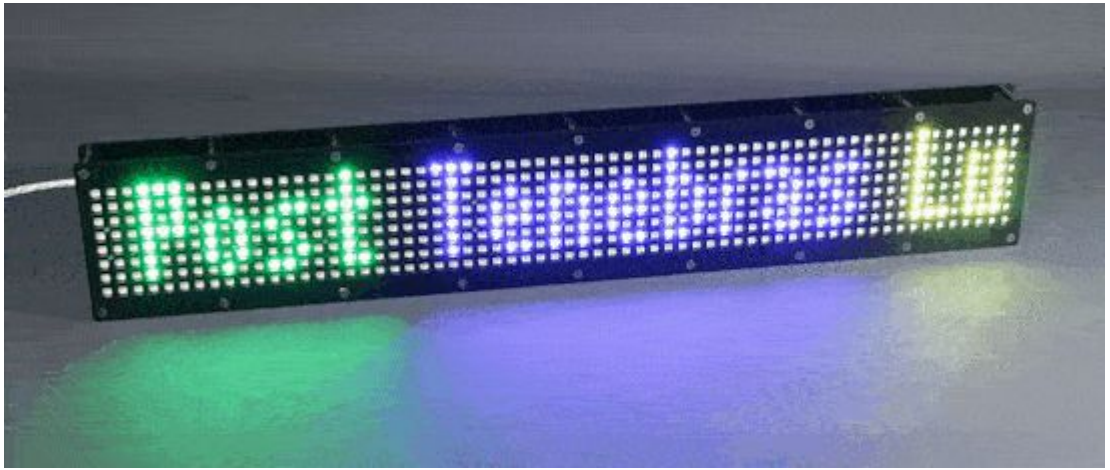
Tokenización



- Hay varias maneras de realizar la tokenización, dependiendo de la granularidad deseada:
  - Tokenización por palabras: divide el texto en palabras individuales.
  - Tokenización por oraciones: separa el texto en oraciones completas.
  - Tokenización por caracteres: descompone el texto en caracteres individuales.

- La elección del método de tokenización depende de la tarea específica de PLN que se esté realizando. Por ejemplo, la tokenización por palabras es común en el análisis de sentimientos, mientras que la tokenización por oraciones puede ser preferible para la traducción automática.

Tokenización



Stemming

Es un proceso de reducción de palabras a su raíz o forma básica. Por ejemplo, las palabras "corriendo", "corre", y "corrí" se reducen a "corr". Es una técnica utilizada en procesamiento de lenguaje natural (NLP) para mejorar la eficiencia de tareas como búsqueda y análisis de textos.

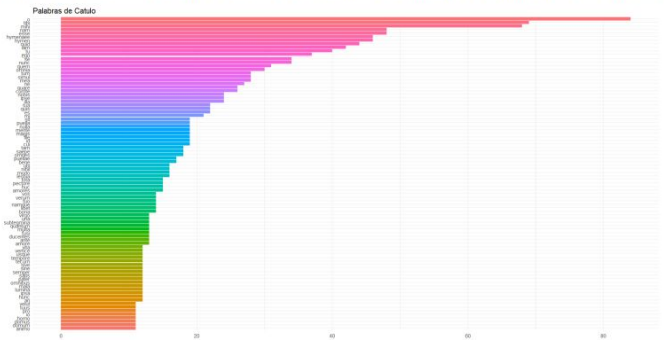
- El objetivo del stemming es eliminar afijos (prefijos y sufijos) y obtener el "tallo" o base léxica de la palabra. Aunque a veces este proceso puede generar raíces no reales, es útil en muchas aplicaciones prácticas como motores de búsqueda y análisis de sentimientos.
- El concepto de stemming tiene sus raíces en la década de 1960. Uno de los algoritmos más famosos es el de Porter, desarrollado por Martin Porter en 1980, que sigue siendo uno de los algoritmos de stemming más utilizados en la actualidad.

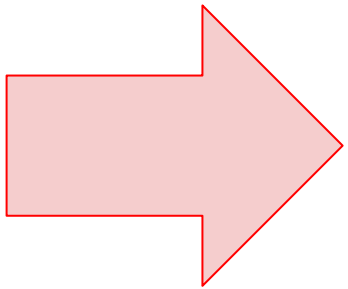
Lematización

- Es un proceso que busca reducir la palabra a su lema, es decir, a su raíz o forma base o de diccionario.
- Tiene en cuenta el análisis morfológico de las palabras, lo que significa que identifica y considera el rol gramatical de una palabra (como su verbo base, el singular de un sustantivo, etc.), su contexto y su etimología para convertirla a su forma canónica.
    - Por ejemplo, la lematización transformaría las palabras "corriendo", "corrió" y "corre" al lema "correr".

Minería de Textos: etapas

- **Exploración de Datos:** la siguiente etapa implica la exploración de los datos para entender mejor su estructura, contenido y las posibles relaciones entre ellos. Esto puede incluir el análisis de la frecuencia de palabras, la identificación de términos clave y frases, y la visualización de datos para obtener una perspectiva inicial sobre el corpus de trabajo.





Minería de Textos: etapas

Muchas veces se llevan a cabo otros procesos que pueden implicar el Enriquecimiento, la Transformación y/o la Extracción de características para facilitar la conversión de texto no estructurado en datos estructurados y analizables, permitiendo la aplicación de algoritmos estadísticos y de inteligencia artificial para extraer conocimientos y patrones.

- **Enriquecimiento:** implica añadir metadatos al texto para proporcionar más contexto sobre su contenido. Esto se logra mediante la asignación de etiquetas a los términos encontrados en el texto.

Minería de Textos: enriquecimiento

- ***Partes del Discurso (POS):*** se clasifican las palabras según su papel gramatical en la oración, como sustantivos, verbos, adjetivos, adverbios, etc. Al etiquetar las palabras con su categoría gramatical, se facilita la comprensión estructural y semántica del texto, permitiendo análisis más sofisticados como la identificación de patrones gramaticales o la extracción de frases significativas.

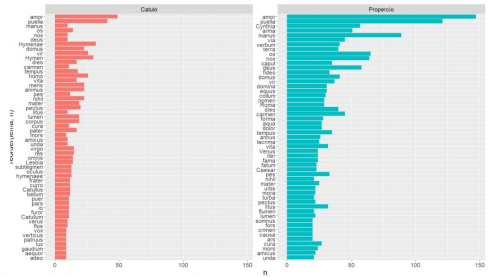


Figura 17. Los 50 sustantivos (lemas) más frecuentes en Catulo y Propercio ordenados por la raiz (lema) de la palabra. Fuente: Elaboración propia.

Minería de Textos: enriquecimiento

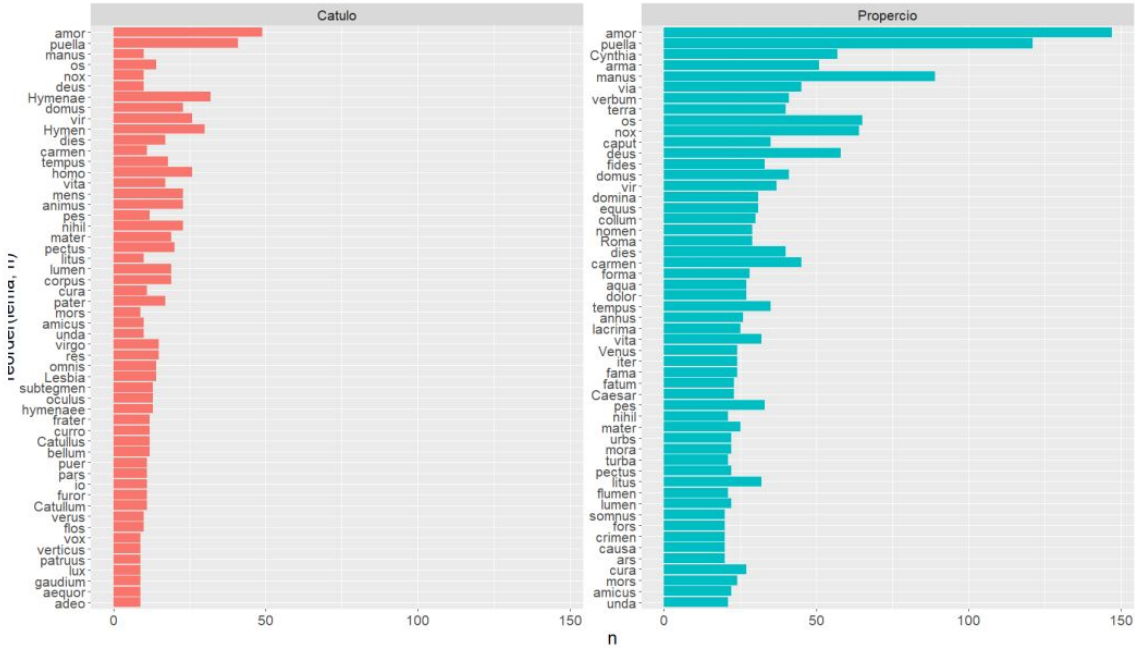


Figura 17. Los 50 sustantivos (lemas) más frecuentes en Catulo y Propercio ordenados por la raíz (lema) de la palabra. Fuente: Elaboración propia.

Minería de Textos: enriquecimiento

- ***Entidades Nombradas (NER):*** las palabras o frases son etiquetadas como nombres de personas, organizaciones, lugares, fechas, etc. Identificar estas entidades es crucial para muchos análisis que requieren comprender el contexto o los sujetos específicos mencionados en el texto, como en la extracción de relaciones, la construcción de bases de conocimiento o en sistemas de recomendación personalizada.

```
Procesando archivos en Catulo:
Entidades en Catulo_Carmen_029.txt:
Mamurram (PERSON)
Britannia (LOC)
Cinaede (PERSON)
Romule (PERSON)
Adoneus (PERSON)
Romule (PERSON)
Tagus (LOC)
Galliae (LOC)
Britanniae (LOC)
```

Minería de Textos: enriquecimiento


```
Procesando archivos en Catulo:
Entidades en Catulo_Carmen_029.txt:
Mamurram (PERSON)
Britannia (LOC)
Cinaede (PERSON)
Romule (PERSON)
Adoneus (PERSON)
Romule (PERSON)
Tagus (LOC)
Galliae (LOC)
Britanniae (LOC)
```

```
Entidades en Propercio_Liber_3_Poema_20.txt:
Africa (LOC)
Luna (LOC)
Venus (PERSON)
Amor (PERSON)
```

Vectorización

Para que los algoritmos de machine learning puedan procesar texto, este debe convertirse en un formato numérico. La vectorización es el proceso de transformar el texto en vectores de números.
Técnicas comunes incluyen el modelo de bolsa de palabras (Bag of Words) y TF-IDF (Frecuencia de Término - Frecuencia Inversa de Documento). Estos métodos permiten representar textos basándose en la presencia y la importancia de las palabras, preparándolos para su análisis con algoritmos de aprendizaje automático.

Minería de Textos: etapas

- **Transformación:** para que el texto sea compatible con algoritmos estadísticos o de inteligencia artificial, que típicamente operan sobre datos numéricos, es necesario convertir el texto en una representación numérica. Esto se consigue mediante la vectorización del texto, que puede adoptar varias formas:
  - Bolsa de Palabras (Bag of Words): representa el texto como un vector donde cada elemento cuenta la presencia o frecuencia de una palabra específica dentro del documento. Aunque es un método simple y efectivo para muchos casos de uso, no mantiene el orden de las palabras.
  - TF-IDF (Frecuencia de Término - Frecuencia Inversa de Documento): similar a la Bolsa de Palabras, pero pondera las frecuencias de las palabras por su importancia en el conjunto del corpus, reduciendo el peso de las palabras comunes y resaltando términos más distintivos.

Minería de Textos: etapas

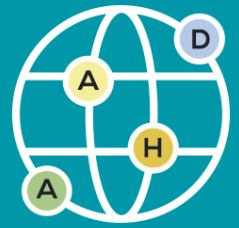


Tabla Bag of Words (BoW):

	abismo	abrasador	acabar	alboroz	amado	amar	amistad	amor	andar	aprieto	burlar	cansado	ca
0	0	0	1	0	0	0	0	0	0	1	2	0	
1	1	1	0	0	1	1	1	1	1	0	0	1	
2	0	0	0	1	0	0	0	1	0	0	0	0	

==========

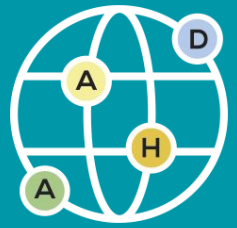
Tabla TF-IDF:

	abismo	abrasador	acabar	alboroz	amado	amar	amistad	amor	andar	aprieto	burlar	cansado
0	0	0	0.142786	0	0	0	0	0	0	0.142786	0.285572	0
1	0.171104	0.171104	0	0	0.171104	0.171104	0.171104	0.130129	0.171104	0	0	0.171104
2	0	0	0	0.170324	0	0	0	0.129536	0	0	0	0

Minería de Textos: etapas

- **Extracción de Características:** con el texto ya convertido a formato numérico, se pueden aplicar diversos algoritmos para identificar y extraer las características más relevantes o informativas del texto. Esto puede implicar:
  - Análisis Estadístico: utilizar técnicas estadísticas para identificar patrones, tendencias, y correlaciones en los datos textuales. Por ejemplo, encontrar las palabras o frases que más contribuyen a la diferencia entre categorías de documentos.
  - Modelos de Aprendizaje Automático e Inteligencia Artificial: aplicar modelos como Naive Bayes, SVM, redes neuronales, entre otros, para clasificación, predicción, agrupamiento, y otras tareas analíticas basadas en las características extraídas del texto.

Minería de Textos: etapas



- **Análisis y Modelado**: aquí se pueden aplicar técnicas estadísticas, de aprendizaje automático y de procesamiento del lenguaje natural que varían según objetivo de la tarea específica. Por ejemplo:
  a. *Clasificación de Textos:* asignar categorías predefinidas a los textos basándose en su contenido.
  b. *Agrupamiento (Clustering):* agrupar textos similares sin categorías predefinidas, basándose en patrones y similitudes encontradas en los datos.
  c. *Extracción de Entidades Nombradas:* identificar y clasificar entidades clave dentro del texto, como nombres de personas, organizaciones o lugares.
  d. *Análisis de Sentimientos:* determinar la actitud o el sentimiento expresado en el texto, como positivo, negativo o neutral.
  e. *Modelado de Tópicos:* descubrir los temas o asuntos subyacentes en una colección de textos.

Minería de Textos: etapas

- **Evaluación y Validación:** tras aplicar los modelos y técnicas de análisis se debe evaluar su rendimiento y validar los resultados. Esto puede incluir el uso de métricas específicas para clasificación (precisión, el recall y el F1-score) o medidas de coherencia para el modelado de tópicos.
- **Interpretación de Resultados**: en esta etapa corresponde analizar los patrones, tendencias y relaciones descubiertas durante el análisis y modelado, y comprender su significado en el contexto del problema o pregunta de investigación específica.

Minería de Textos: etapas

- **Visualización de Datos:** es una herramienta poderosa para presentar los hallazgos de manera comprensible y accesible. Mapas de calor, nubes de palabras, gráficos de barras y otras visualizaciones pueden ayudar a destacar los resultados más importantes y facilitar la interpretación de grandes volúmenes de datos textuales.
- **Comunicación de Resultados:** finalmente, los conocimientos extraídos deben ser comunicados efectivamente a las partes interesadas, lo cual puede incluir la elaboración de informes, presentaciones y recomendaciones basadas en los hallazgos de la minería de textos.

Aplicaciones Prácticas en la Vida Cotidiana

- **Asistentes Virtuales:** Siri, Alexa, Google Assistant y otros asistentes virtuales utilizan el PLN para entender las consultas en lenguaje natural de los usuarios, procesarlas y ofrecer respuestas útiles o realizar acciones.
- **Traducción Automática:** servicios como Google Translate y DeepL aplican técnicas de PLN para traducir textos o voz de un idioma a otro.
- **Análisis de Sentimientos:** empresas utilizan la minería de textos y el análisis de sentimientos para monitorear y analizar opiniones y emociones en redes sociales, reseñas de productos, y foros en línea.
- **Sistemas de Recomendación:** plataformas de streaming como Netflix y Spotify analizan las preferencias y comportamientos de sus usuarios para realizar recomendaciones personalizadas.

Aplicaciones Prácticas en la Vida Cotidiana

- **Detección de Spam y Filtros de Contenido:** el correo electrónico y las plataformas de redes sociales aplican técnicas de PLN para filtrar y clasificar mensajes, identificando y eliminando automáticamente contenido no deseado o spam.
- **Búsqueda y Extracción de Información:** motores de búsqueda y herramientas de investigación utilizan PLN para entender las consultas de los usuarios, buscar información relevante y resumir contenido, haciendo que la navegación y la adquisición de conocimientos sean más eficientes.
- **Herramientas de Escritura y Corrección:** programas de procesamiento de texto y aplicaciones como Grammarly emplean PLN para detectar y corregir errores gramaticales, ortográficos y de estilo en la escritura, ayudando a mejorar la calidad de los textos producidos.


SPAM®
Classic
SIGNATURE RECIPE COLLECTION 3 OF 3
Crazy Tasty
U.S. INSPECTED AND PASSED BY DEPARTMENT OF AGRICULTURE
Serving Suggestion
NET WT 12 OZ (340g)
Hormel Foods


MENU
EGG + SPAM
EGG BACON + SPAM
EGG BACON SAUSAGE + SPAM
SPAM BACON SAUSAGE + SPAM
SPAM EGG SPAM SPAM BACON + SPAM
SPAM SPAM SPAM EGG + SPAM
SPAM SPAM SPAM SPAM SPAM SPAM BAKED BEANS SPAM SPAM SPAM + SPAM
LOBSTER THERMIDOR AUX CREVETTES WITH MORNAY SAUCE GARNISHED WITH TRUFFLE PATE


THIS IS A SPECIAL ECONOMY LABEL FOR PERIOD OF EMERGENCY
SPAM


la
de
en
cielo
el
como
y
un
que
una
si
casi
porque
por
ciudades
vacías
todo
bajo
esas
éramos
centro
cambio
Más
vez
eran
acribillada
azul
negrura
grandes
abundancia
amarilla
quedó
intemperie
ahora
viejo
ellas
playa
Allá
mano
estrellas
soy
horizontal
mis
días
desierto
sobre
es
hormigas
Estaban
costas
ese
dormíamos
diminuto
dilatado
disimulan
paso
alcance
vida
entrever
pared
al
chisporroteantes
innumerables
noche
sentí
incandescencia
actividad
orificios
hubiese
sin
sus
otra
entre
mucha
aplastados
volcán
dejase
sido
interna

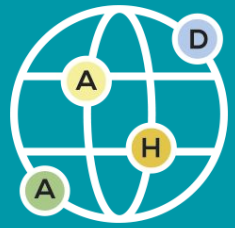

la
de
en
cielo
el
y
como
un
que
una
si
ciudades
vacías
todo
bajo
esas
centro
éramos
cambio
Más
vez
eran
acribillada
azul
negrura
grandes
dejase
abundancia
volcán
amarilla
aplastados
entre
mucha
otra
sus
sin
hubiese
orificios
incandescencia
actividad
quedó
casi
intemperie
sido
porque
horizontal
ahora
viejo
ellas
playa
Allá
mano
estrellas
soy
mis
por
sobre
es
días
desierto
hormigas
Estaban
costas
ese
diminuto
dilatado
disimulan
interna
alcance
vida
dormíamos
entrever
paso
pared
al
chisporroteantes
innumerables
noche
sentí


estrellas
quedó
noche
sentí
mano
abundancia
cambio
disimulan
actividad
días
paso
desierto
si
ciudades
vida
grandes
alcance
intemperie
acribillada
sido
diminuto
vacías
innumerables
orificios
dejase
casi
costas
entrever
azul
aplastados
viejo
dormíamos
playa
incandescencia
interna
cielo
vez
Allá
bajo
hormigas
pared
mucha
chisporroteantes
ahora
negrura
horizontal
amarilla
volcán
dilatado
centro


la
de
en
cielo
el
como
un
que
una
y
si
ciudades
todo
bajo
esas
éramos
vacías
centro
amarilla
quedó
casi
sobre
es
días
porque
por
ahora
viejo
ellas
playa
Allá
costas
ese
noche
sentí
vida
alcance
dilatado
diminuto
horizontal
disimulan
paso
dormíamos
desierto
hormigas
estrellas
intemperie
abundancia
azul
negrura
grandes
Más
cambio
vez
eran
acribillada
mucha
entre
aplastados
otra
sin
hubiese
orificios
incandescencia
actividad
volcán
mis
soy
mano
Estaban
interna
innumerables
al
pared
entrever
chisporroteantes
dejase
sido
sus


si
ciudades
cielo
casi
costas
vacías
diminuto
abundancia
sentí
quedó
estrellas
noche
mano
cambio
disimulan
actividad
días
paso
desierto
vida
grandes
alcance
intemperie
acribillada
sido
dejase
innumerables
orificios
entrever
azul
aplastados
viejo
dormíamos
playa
incandescencia
interna
Allá
vez
hormigas
bajo
ahora
pared
mucha
chisporroteantes
negrura
horizontal
amarilla
volcán
dilatado
centro

Herramientas en Python: NLTK (Natural Language Toolkit)



- Fue lanzada en 2001 como parte de un proyecto de investigación en la Universidad de Pensilvania y desde entonces se ha convertido en una de las herramientas más utilizadas en la comunidad de PLN para la enseñanza y la investigación.
- Proporciona una fácil acceso a más de 50 recursos léxicos y una suite de bibliotecas para la clasificación, tokenización, lematización, etiquetado de partes del discurso, y análisis sintáctico.

Herramientas en Python: NLTK (Natural Language Toolkit)

- Incluye una amplia gama de corpora y funcionalidades para trabajar con datos de texto, lo que lo hace particularmente útil para prototipos rápidos y para la enseñanza de conceptos fundamentales de PLN. Sin embargo, debido a su naturaleza altamente modular y su enfoque en la enseñanza, puede ser menos eficiente para aplicaciones de PLN en producción en comparación con otras bibliotecas más recientes.

Herramientas en Python: spaCy

- Fue lanzada en 2015 y se ha destacado por su rendimiento y eficiencia.
- Ofrece implementaciones optimizadas de tareas comunes de PLN como tokenización, etiquetado de partes del discurso, lematización, y reconocimiento de entidades nombradas.

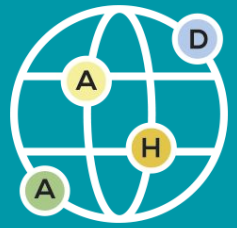
spaCy

Herramientas en Python: spaCy

- spaCy se centra en ofrecer modelos preentrenados altamente eficientes para una variedad de idiomas, lo que permite a los desarrolladores implementar soluciones de PLN robustas y escalables con relativa facilidad.
- spaCy también integra capacidades para el análisis de dependencias sintácticas y ofrece una API limpia y coherente que facilita la creación de aplicaciones de PLN complejas.

spaCy

Ejemplo de tokenización y lematización




```python
import spacy

# Cargar el modelo de lenguaje en español
nlp = spacy.load('es_core_news_sm')

# Texto de ejemplo en español
text = "Los gatos quieren cazar a los ratones"

# Procesar el texto
doc = nlp(text)

# Tokenización y lematización
tokens = [token.text for token in doc]
lemmas = [token.lemma_ for token in doc]

# Resultados
print("Tokens:", tokens)
print("Lemas:", lemmas)

```

```
Tokens: ['Los', 'gatos', 'quieren', 'cazar', 'a', 'los', 'ratones']
Lemas: ['el', 'gato', 'querer', 'cazar', 'a', 'el', 'ratón']
```

Análisis estilométrico básico: frecuencias y n-gramas

```python
import spacy
from collections import Counter
from spacy.lang.es.stop_words import STOP_WORDS
from spacy.lang.es import Spanish

# Crear un objeto de procesamiento de lenguaje natural en español
nlp = Spanish()


# Texto de ejemplo en español
text = "Y que yo me la llevé al río creyendo que era mozuela, pero tenía marido."

# Procesar el texto con spacy
doc = nlp(text)

# Tokenización y limpieza de stopwords y puntuación
tokens = [token.text.lower() for token in doc if not token.is_stop and not token.is_punct]

# Frecuencia de palabras (unigramas)
word_freq = Counter(tokens)

# Generar bigramas
bigrams = list(ngrams(tokens, 2))
bigram_freq = Counter(bigrams)

# Mostrar la frecuencia de palabras
print("Frecuencia de palabras (unigramas):", word_freq)

# Mostrar la frecuencia de bigramas
print("Frecuencia de bigramas:", bigram_freq)

```

```
Frecuencia de palabras (unigramas): Counter({'llevé': 1, 'río': 1, 'creyendo': 1, 'mozuela': 1, 'marido': 1})
Frecuencia de bigramas: Counter({('llevé', 'río'): 1, ('río', 'creyendo'): 1, ('creyendo', 'mozuela'): 1, ('mozuela', 'marido'): 1})
```

Pequeño ejemplo: Nube de Palabras

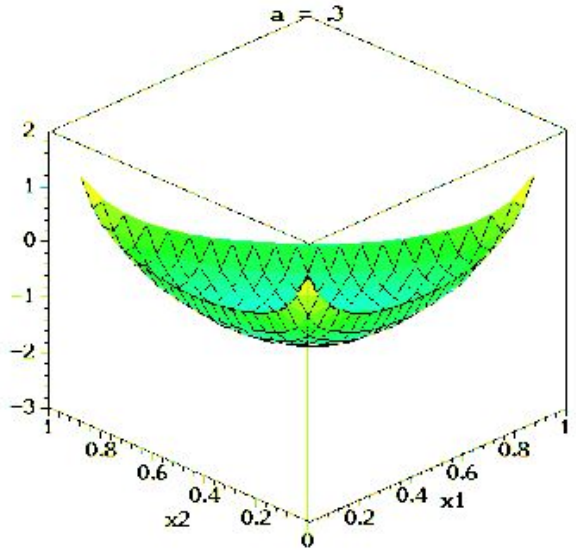
```python
# Importar las librerías necesarias
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

# Descargar la lista de stopwords en español
nltk.download('stopwords')
nltk.download('punkt')

# Definir las stopwords en español
stop_words = set(stopwords.words('spanish'))

# Texto de ejemplo
texto = """De esas costas vacías me quedó sobre todo la abundancia de cielo. Más de una vez me sentí
diminuto bajo ese azul dilatado: en la playa amarilla, éramos como hormigas en el centro de
un desierto. Y si ahora que soy un viejo paso mis días en las ciudades, es porque en ellas la
vida es horizontal, porque las ciudades disimulan el cielo. Allá, de noche, en cambio, dormíamos,
a la intemperie, casi aplastados por las estrellas. Estaban como al alcance de la mano y
eran grandes, innumerables, sin mucha negrura entre una y otra, casi chisporroteantes, como
si el cielo hubiese sido la pared acribillada de un volcán en actividad que dejase entrever por
sus orificios la incandescencia interna."""

# Tokenizar el texto
tokens = word_tokenize(texto, language='spanish')

# Filtrar las stopwords
filtered_text = ' '.join([word for word in tokens if word.lower() not in stop_words])

# Generar la nube de palabras sin stopwords
wordcloud = WordCloud(width = 800, height = 400,
                      background_color ='white').generate(filtered_text)

# Mostrar la nube de palabras
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()

```

Modelado de Tópicos

- Es una técnica de minería de textos que tiene como objetivo descubrir automáticamente temas ocultos o *tópicos* en grandes colecciones de documentos. Estos tópicos representan conjuntos de palabras que tienden a aparecer juntas y ayudan a entender la estructura semántica del texto.
- Permite explorar grandes corpus de documentos sin necesidad de etiquetado previo.
- Es útil para resumir contenido, clasificar documentos, mejorar la búsqueda y entender tendencias en datos textuales.
- Un tópico está representado por un conjunto de palabras relevantes, mientras que cada documento puede ser visto como una mezcla de estos tópicos en diferentes proporciones.

Latent Dirichlet Allocation (LDA)

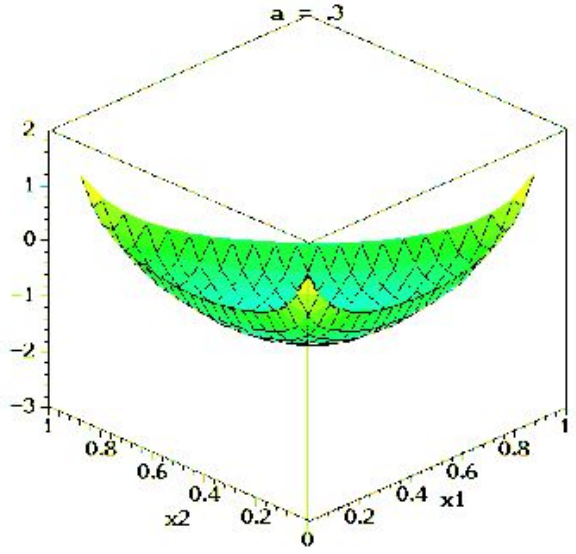
- Es uno de los algoritmos más populares para el modelado de tópicos.
- LDA es un modelo generativo probabilístico que asume que cada documento es una combinación de varios tópicos y que cada tópico es una distribución sobre palabras.
- El algoritmo busca asignar palabras de los documentos a tópicos de forma que maximice la coherencia dentro de los tópicos.



Latent Dirichlet Allocation (LDA)

Funcionamiento básico:

- Cada documento es representado como una mezcla de tópicos en proporciones variables.
- Cada tópico está representado como una mezcla de palabras.
- LDA utiliza la distribución de Dirichlet para controlar cómo se distribuyen los tópicos en los documentos y las palabras en los tópicos.

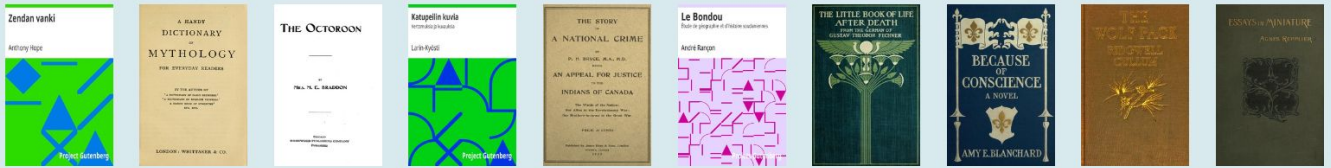


Proyecto Gutenberg

Welcome to Project Gutenberg

Project Gutenberg is a library of over 70,000 free eBooks

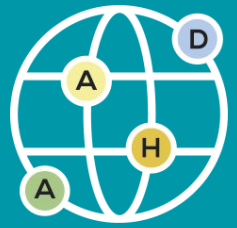
Choose among free epub and Kindle eBooks, download them or read them online. You will find the world's great literature here, with focus on older works for which U.S. copyright has expired. Thousands of volunteers digitized and diligently proofread the eBooks, for you to enjoy.



Zendan vanki by Anthony Hope | A handy dictionary of mythology by | The Octoroon by M. E. Braddon | Katupeilin kuvia by Larin-Kyösti | The story of a national crime by P. H. Bryce | Le Bondou: étude de géographie et | The little book of life after death by | Because of conscience by Amy Ella | The wolf pack by Ridgwell Cullum | Essays in miniature by Agnes Repplier

Some of our latest eBooks Click Here for more latest books!

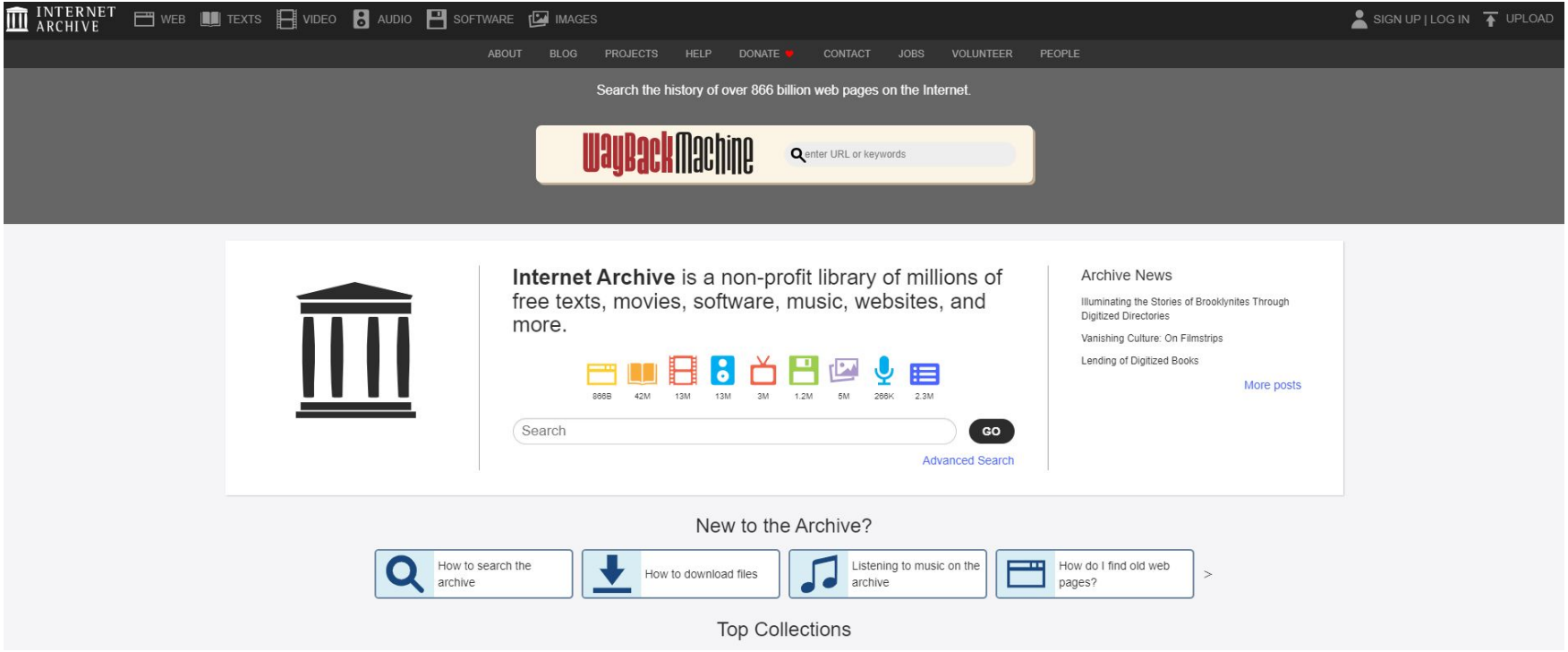
Proyecto Gutenberg



Es una iniciativa que tiene como objetivo digitalizar, archivar y distribuir obras literarias de dominio público en formato electrónico. Fue fundado en 1971 por Michael S. Hart, quien creó el primer libro electrónico (e-book) digitalizando la "Declaración de Independencia de los Estados Unidos".

- Acceso libre y gratuito: todos los libros están disponibles de manera gratuita porque son de dominio público.
- Variedad de formatos: los libros están disponibles en múltiples formatos (texto plano, HTML, ePub, Kindle, etc.) para facilitar su acceso en ordenadores, e-readers y smartphones.
- Colaboración voluntaria: la digitalización y corrección de los libros es realizada por una comunidad de voluntarios.

Internet Archive

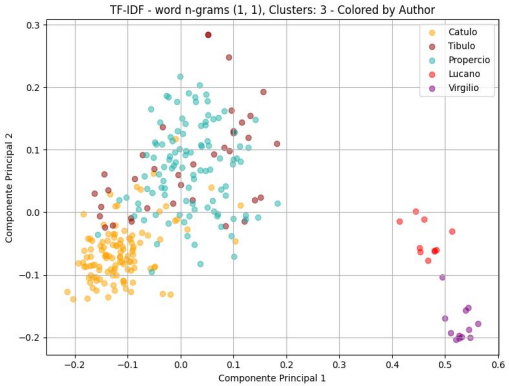


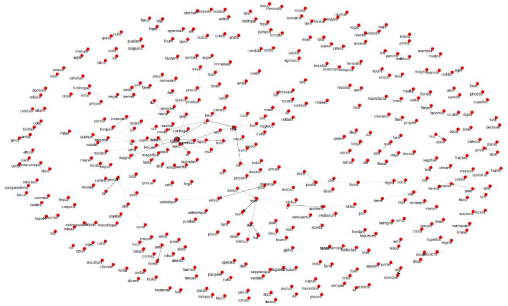
Internet Archive

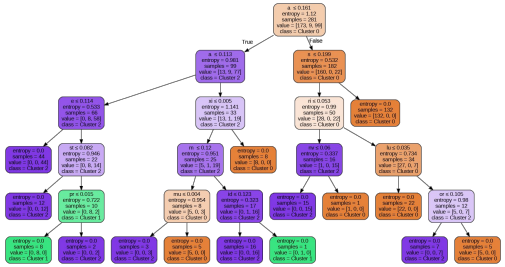
Es una biblioteca digital sin fines de lucro fundada en 1996 con el objetivo principal es preservar y proporcionar acceso gratuito a una vasta cantidad de contenido digital, como libros, música, vídeos, software, y páginas web.

- Los recursos que ofrece Internet Archive incluyen:
  - Libros y textos: digitalización y acceso a millones de libros y documentos, muchos de ellos en dominio público.
  - Páginas web: a través de la *Wayback Machine*, permite consultar versiones archivadas de sitios web.
  - Películas y vídeos: acceso a películas, vídeos caseros, y grabaciones de interés histórico.
  - Audio y música: grabaciones de música, programas de radio y podcasts.
  - Software: preserva y proporciona acceso a software antiguo, videojuegos y aplicaciones históricas.

Algunos ejemplos de uso personales







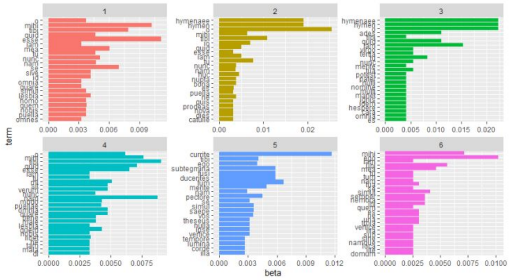
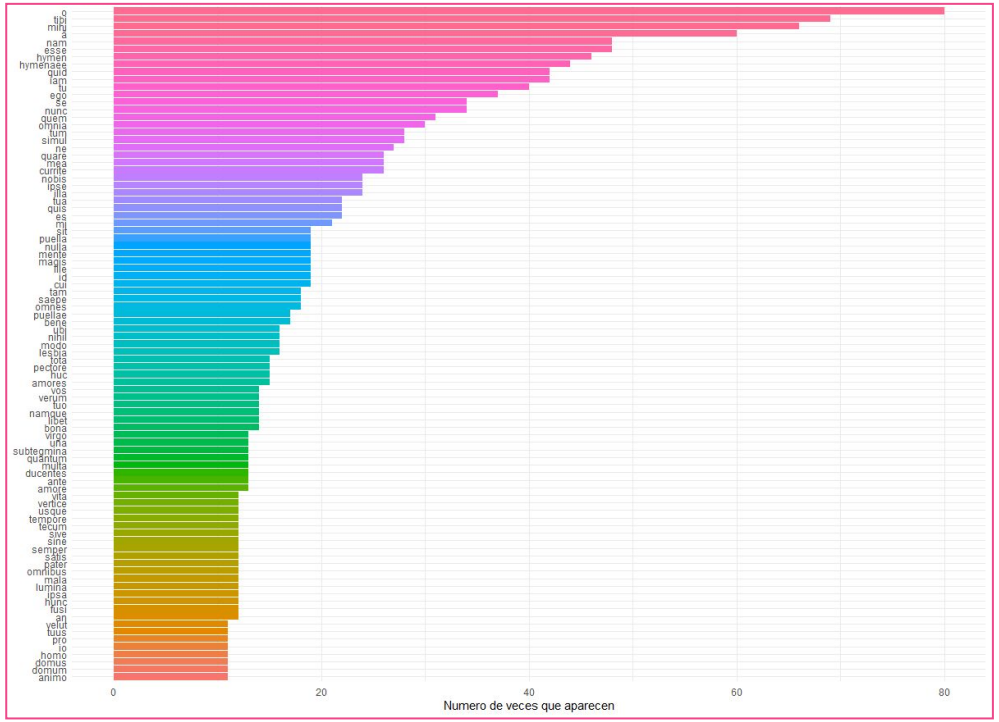


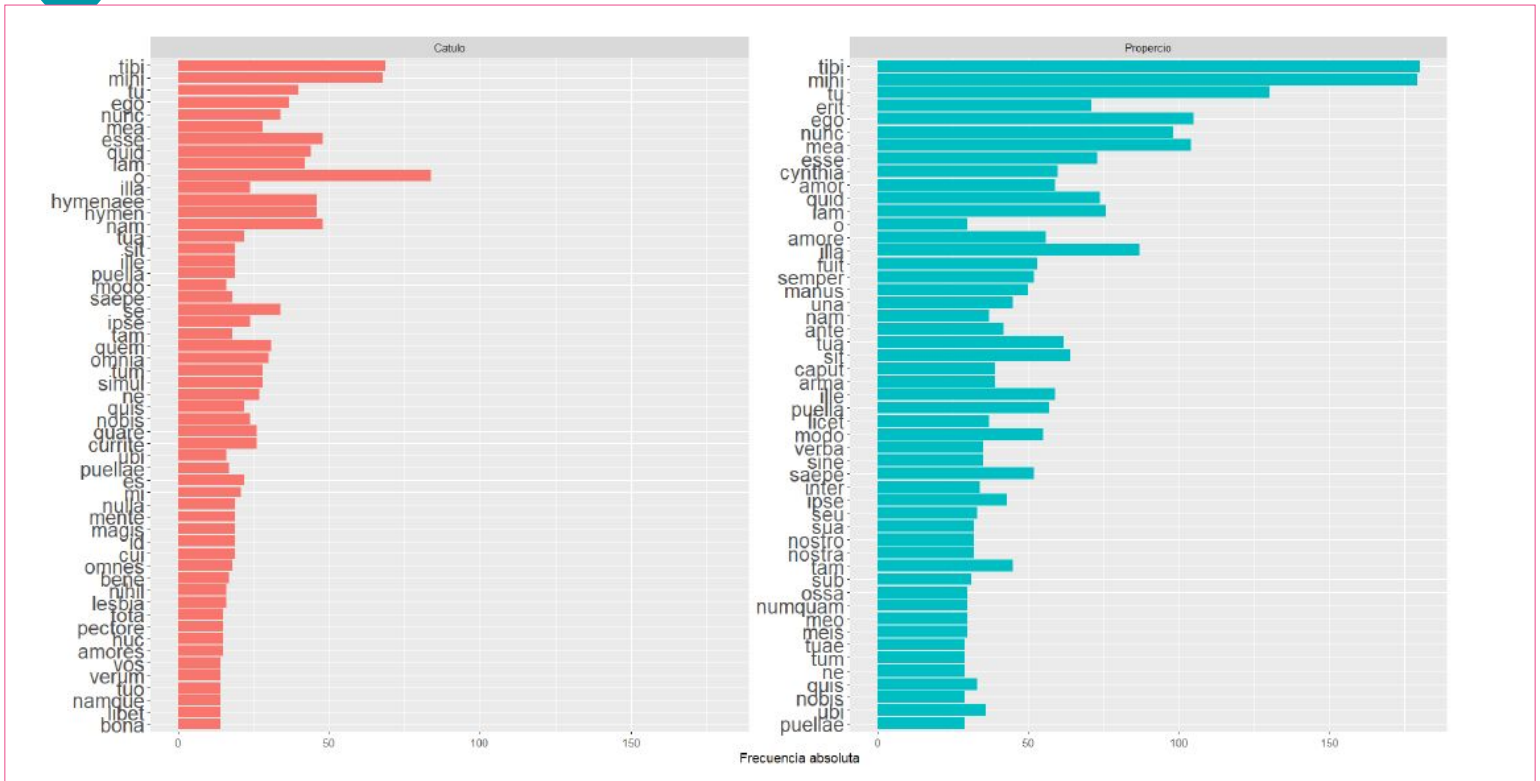
FIGURA 32

Cuadro de tópicos para el *Corpus Catullianum*

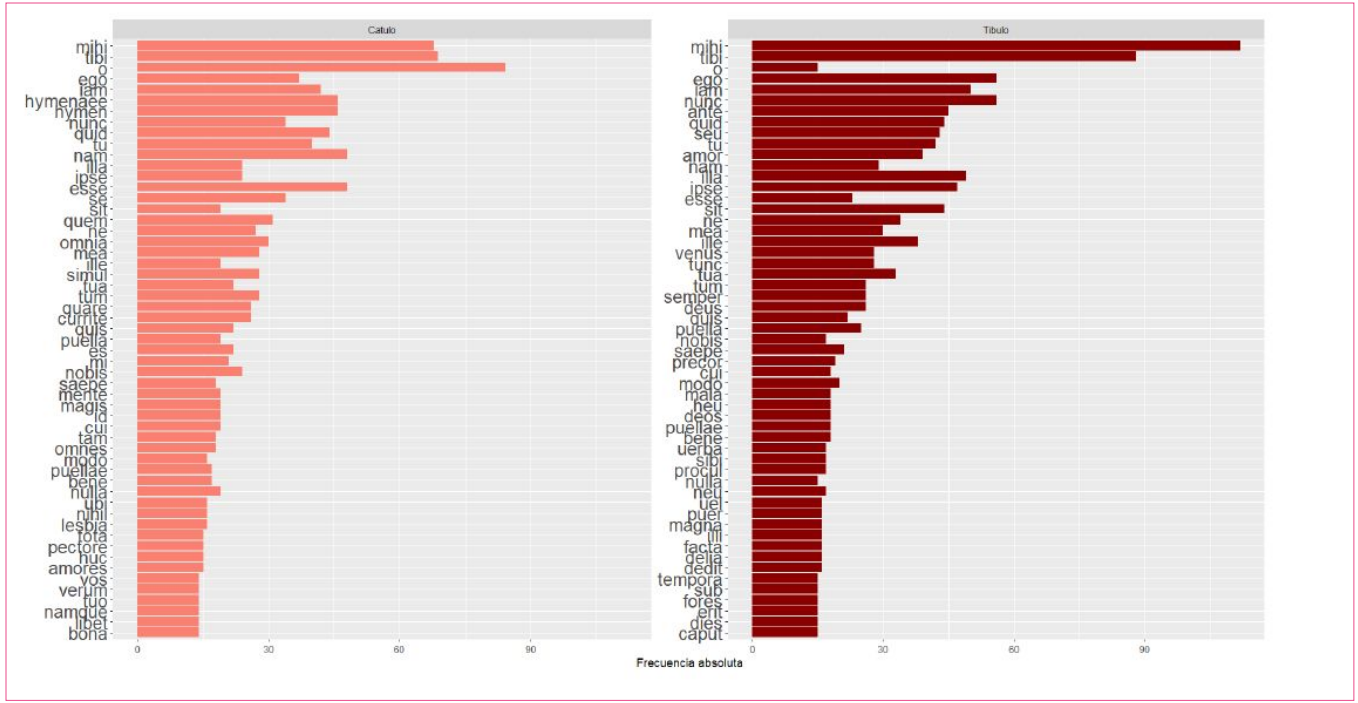
Palabras más frecuentes en Catulo



Palabras más frecuentes en Catulo y Propercio

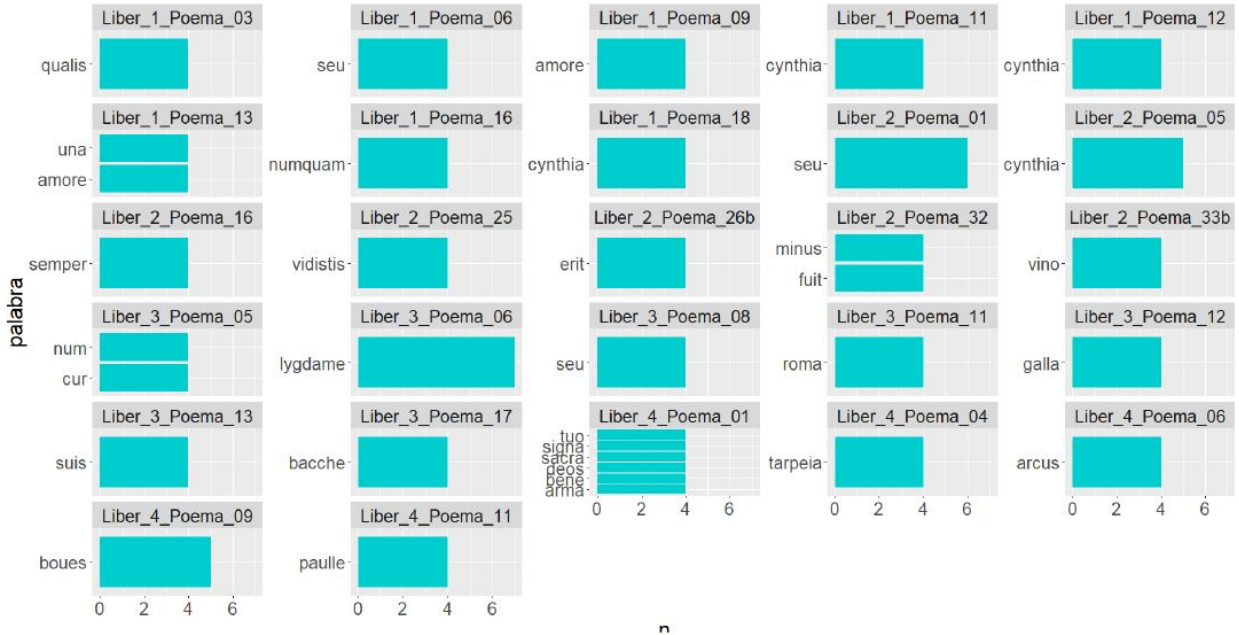


Palabras más frecuentes en Catulo y Tibulo

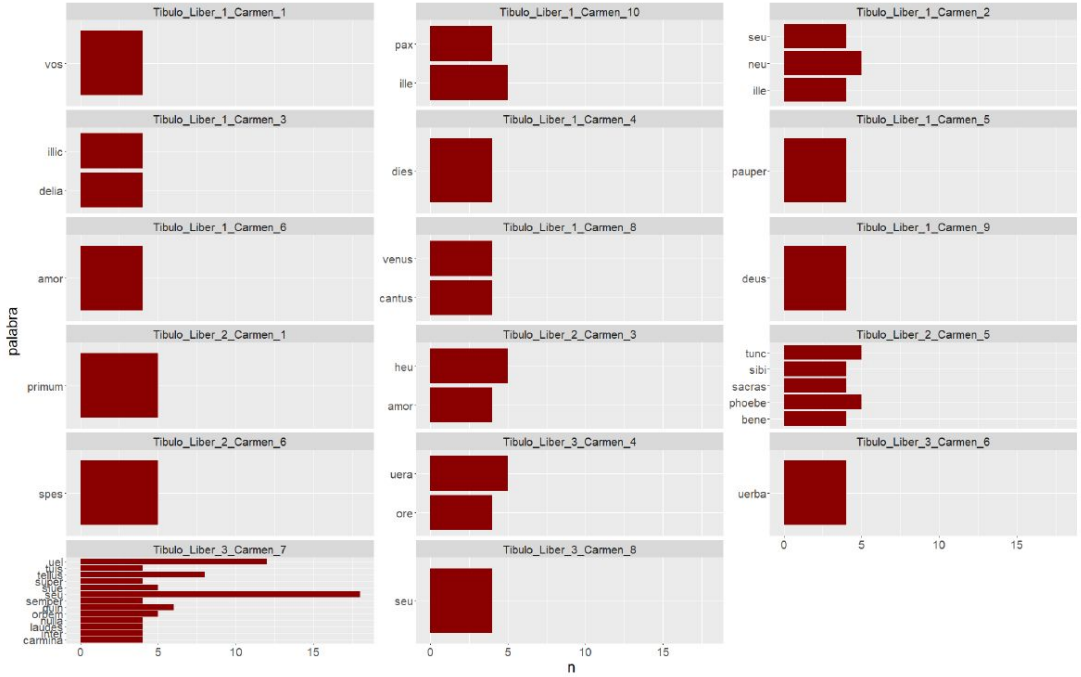


Palabras repetidas más de tres veces en Catulo

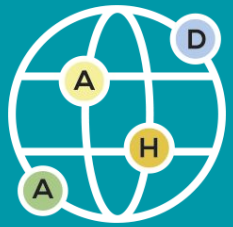
Palabras repetidas más de tres veces en Propercio



Palabras repetidas más de tres veces en Tibulo



Bigramas



sed totum hoc studium luctu fraterna mihi mors

abstulit. o misero ***frater adempte mihi,*** (*Carm. 68*, 19-20)

Pero la muerte fraterna con su luto de toda esa dedicación

me apartó. ¡Oh pobre mi hermano arrebatado!

heu miser indigne ***frater adempte mihi.*** (*Carm. 101*, 6)

¡Oh mi pobre hermano arrebatado indignamente!

Troia (nefas) commune sepulcrum Asiae Europaeque,

Troia virum et virtutum omnium acerba cinis:

quaene etiam nostro letum miserabile fratri

attulit. Hei misero ***frater adempte mihi,*** (*Carm. 68*, 89-92)

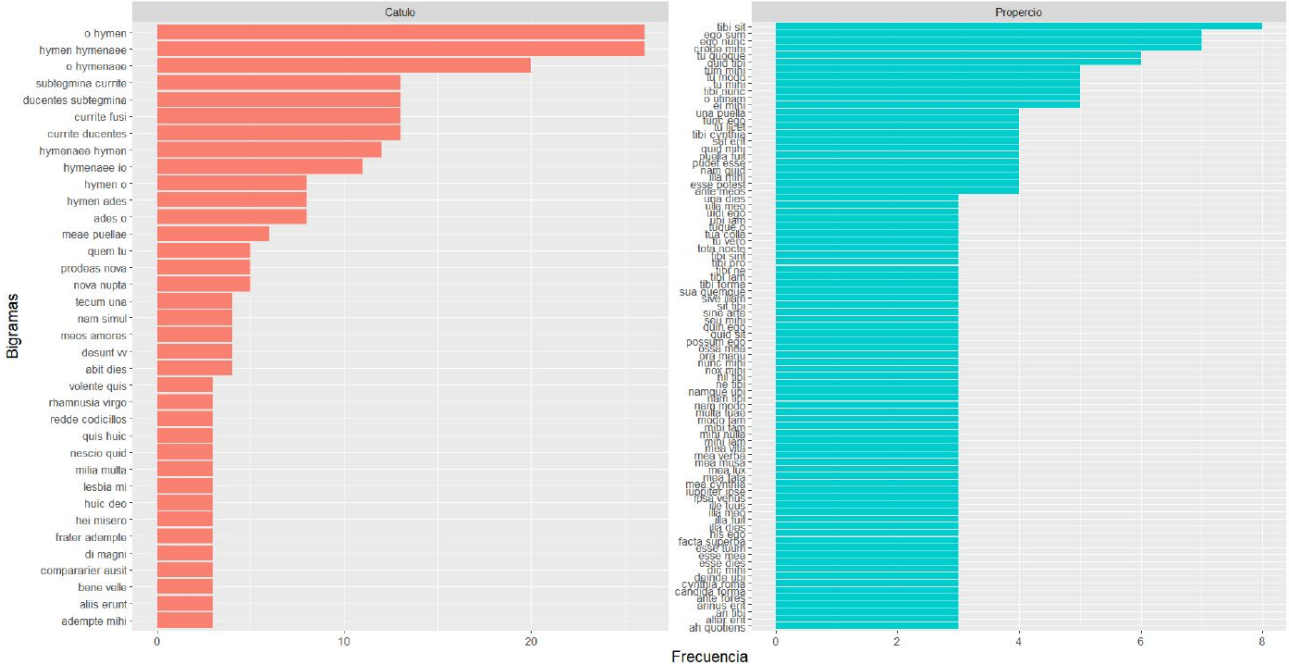
Troya nefasta, sepulcro común de Asia y de Europa

Troya amarga ceniza de todos los varones y las virtudes:

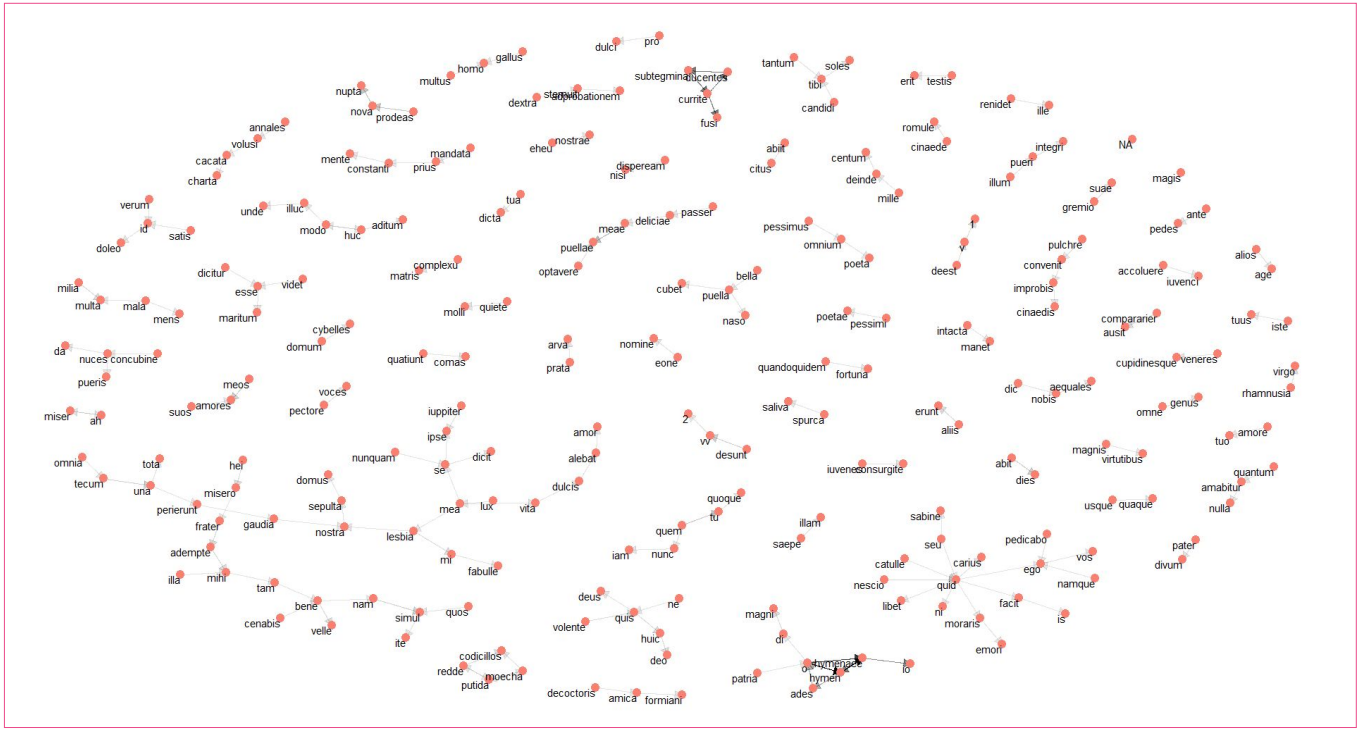
ella inclusive trajo la muerte a nuestro pobre hermano.

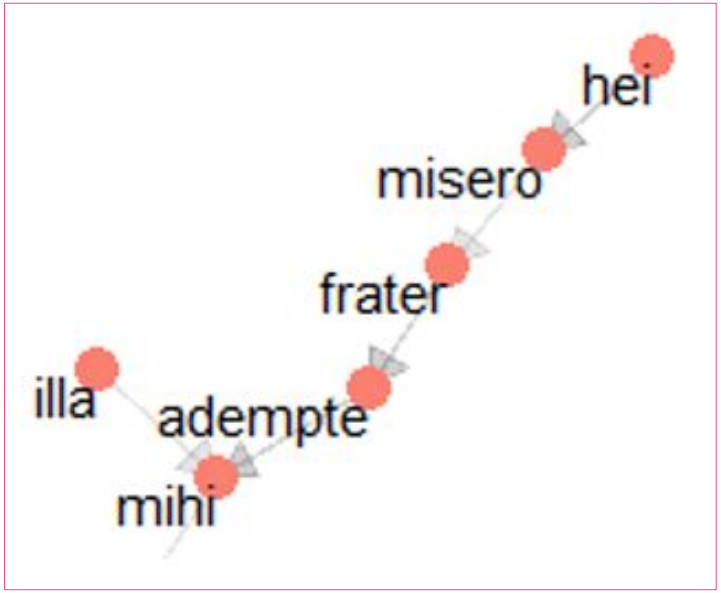
¡Oh mi pobre hermano arrebatado!

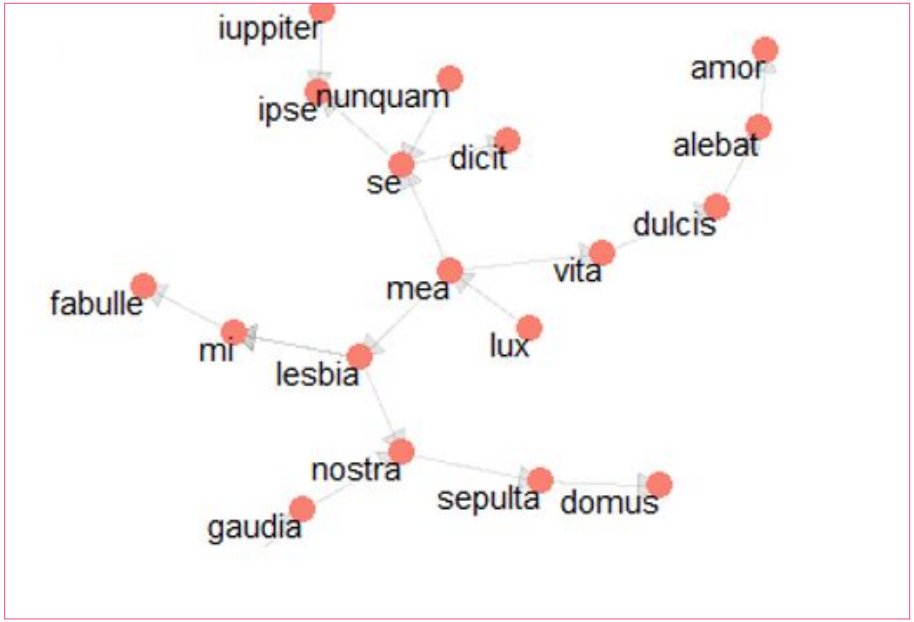
Bigramas más frecuentes en cada poeta



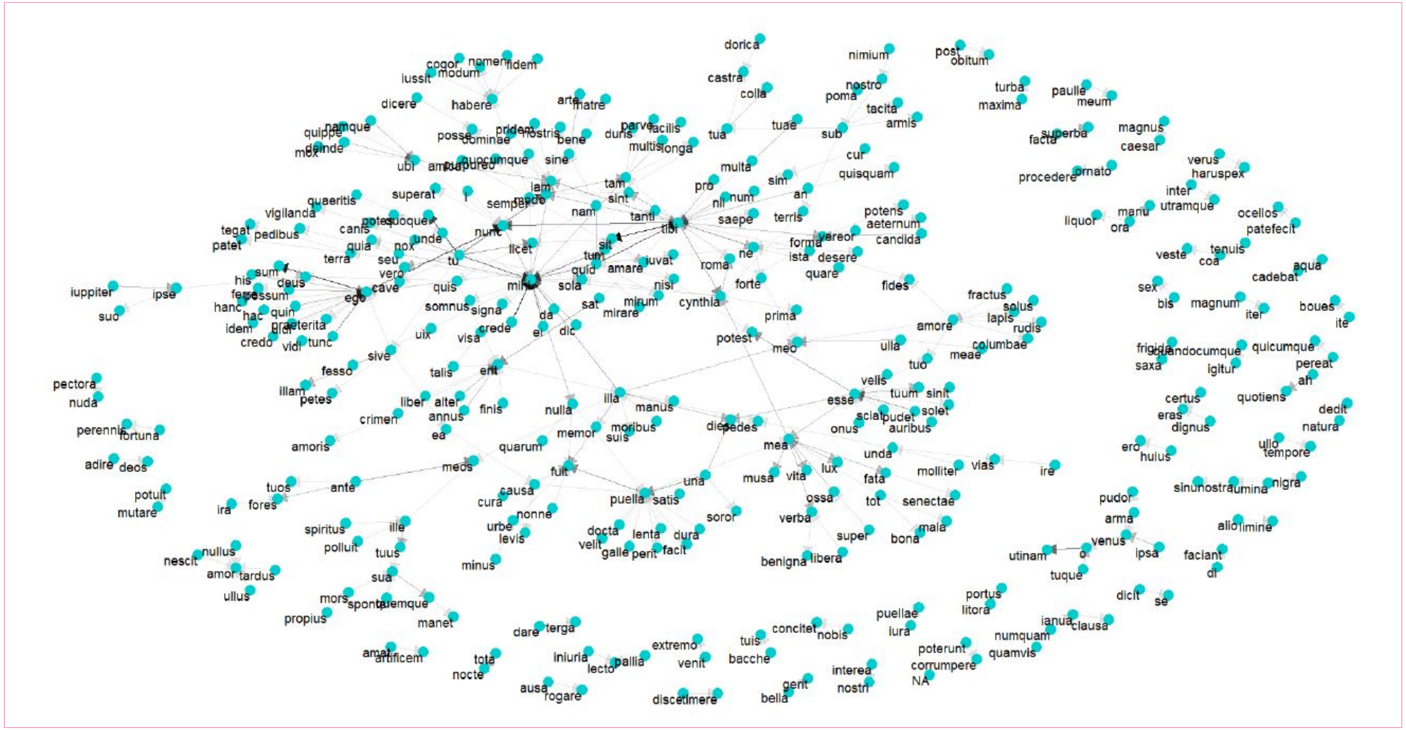
Grafo de bigramas en Catulo con términos de estribillos



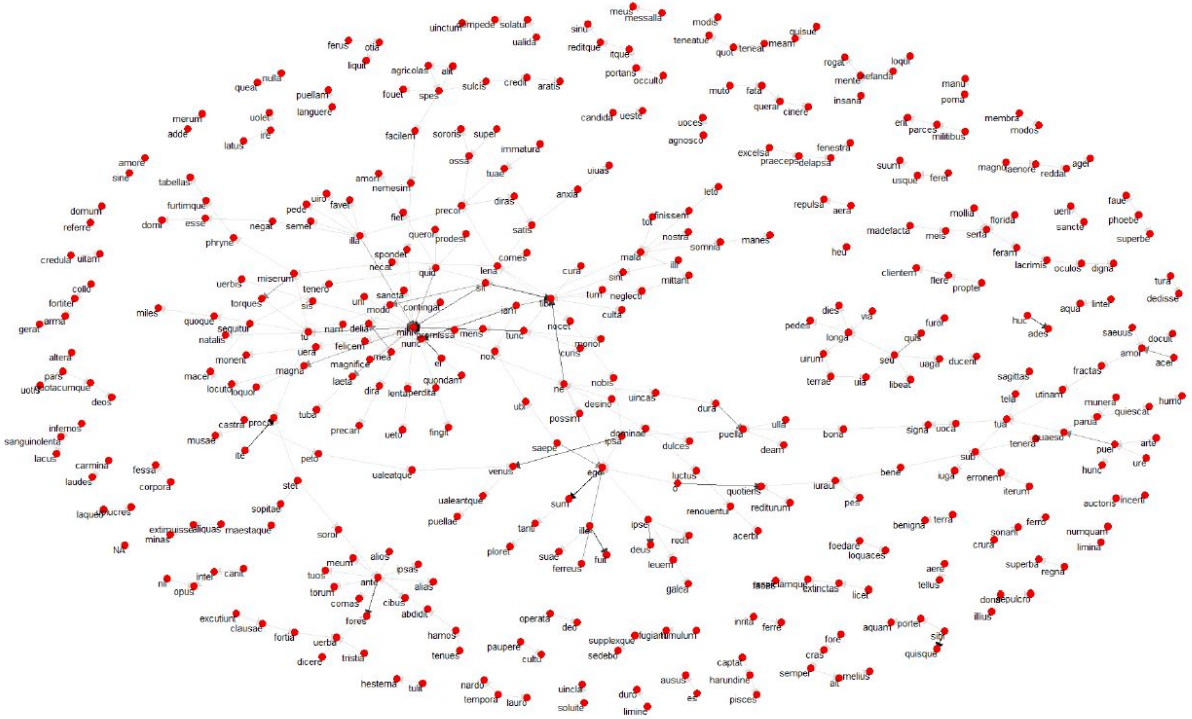

hei
misero
frater
illa
adempte
mihi


iuppiter
amor
nunquam
ipse
alebat
dicit
se
dulcis
vita
mea
fabulle
lux
mi
lesbia
nostra
sepulta
domus
gaudia

Grafo de bigramas en Propercio



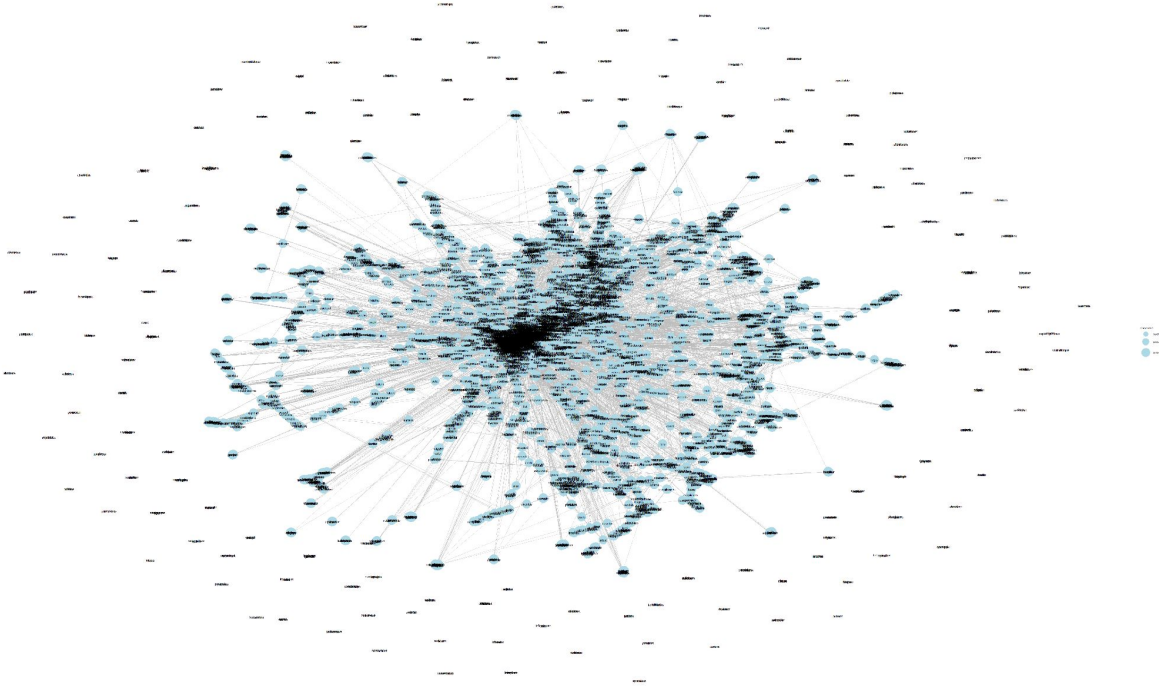
Grafo de bigramas en Tibulo



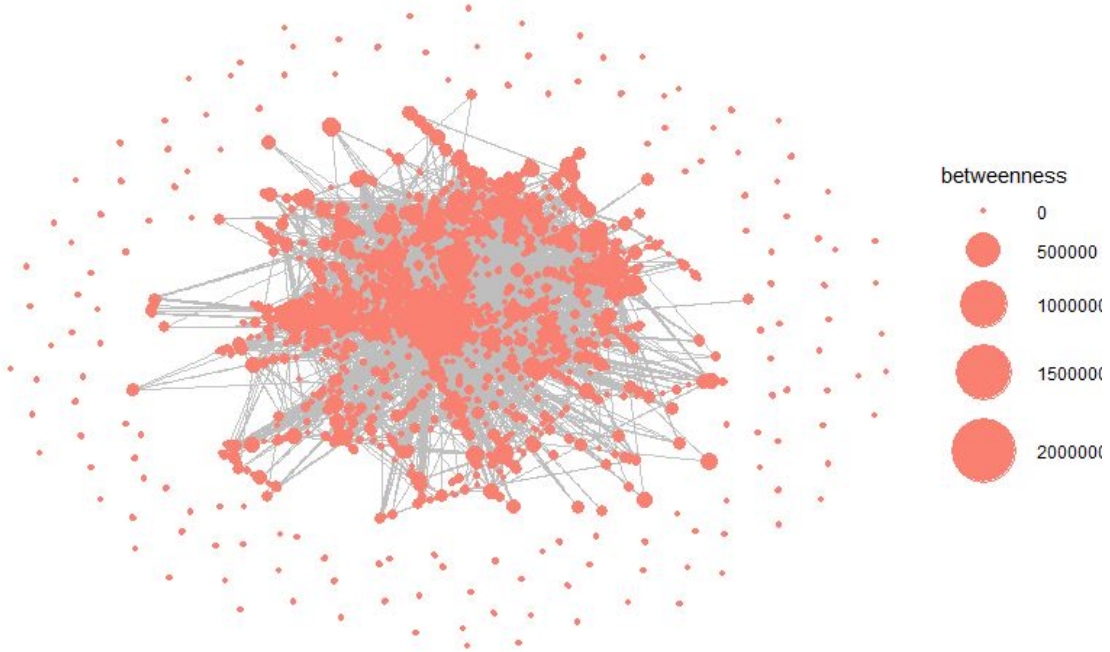
Lestrigones y Cíclopes

- Desarrollar tolerancia al error!!!

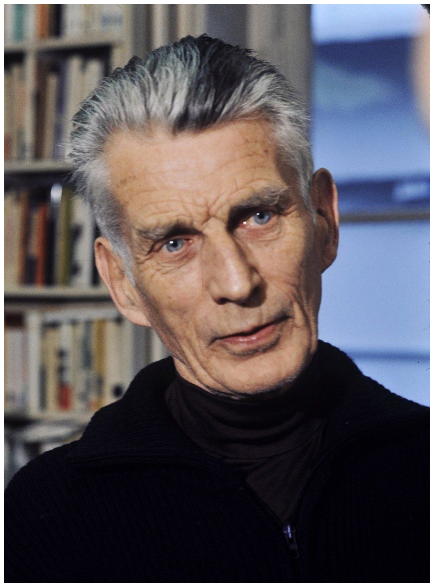
Grafo de bigramas en Catulo (Puede fallar)

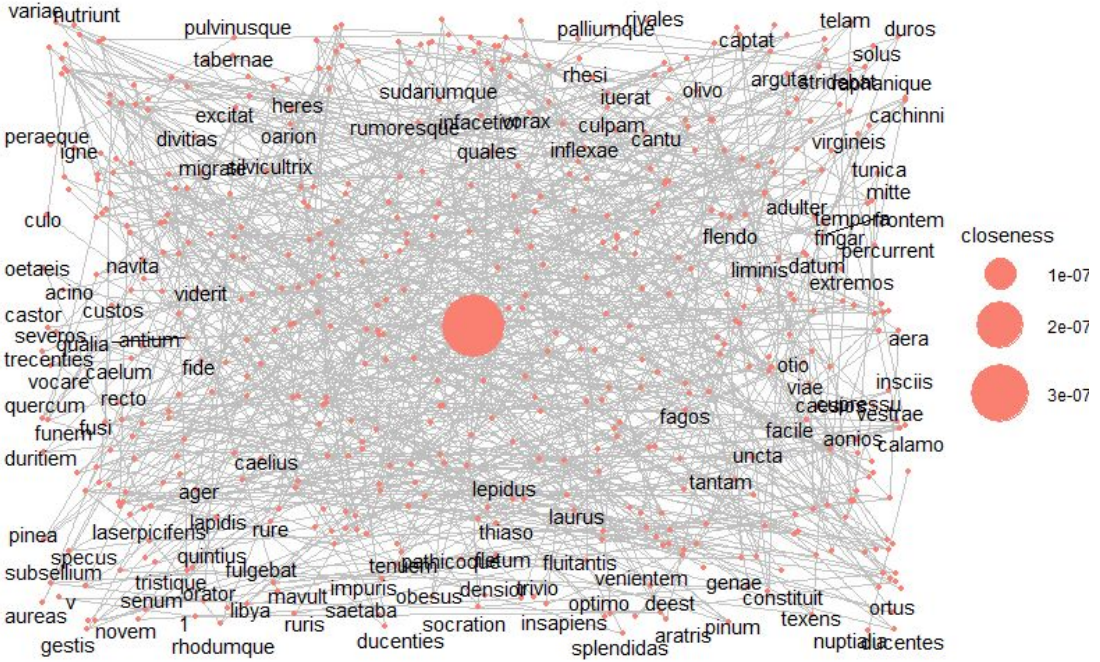


Grafo de bigramas en Catulo (intentando nuevamente)



Grafo de bigramas en Catulo (FRACASANDO MEJOR!!!)





Erro: errare, erravi, erratum.

1. apartarse del camino, errar.
2. fallar (un objetivo).
3. viajar sin rumbo.

Cuadro de tópicos para el Corpus Catullianum

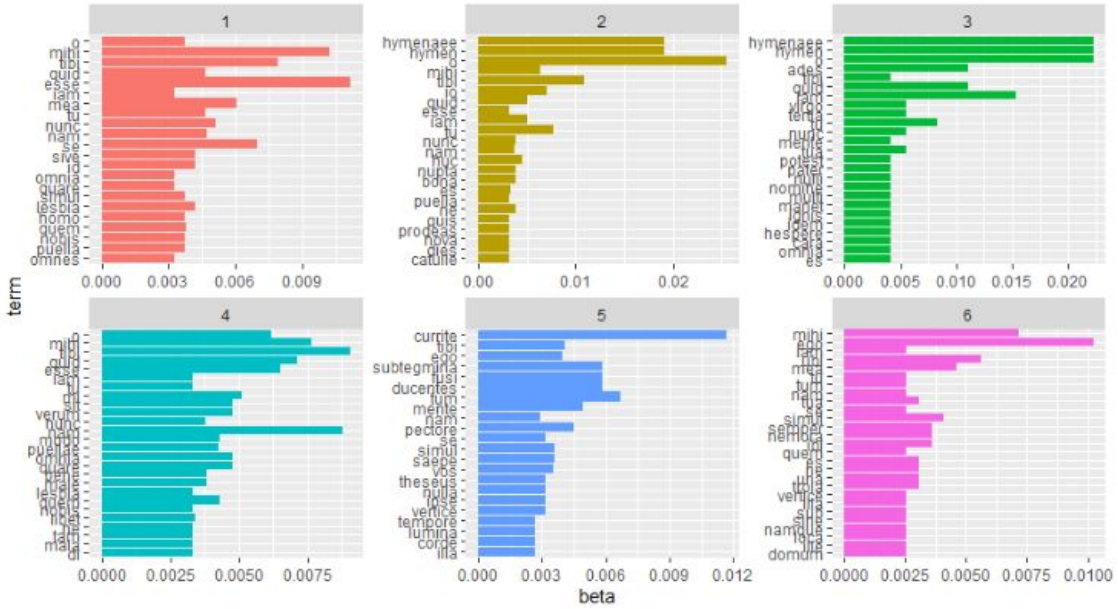
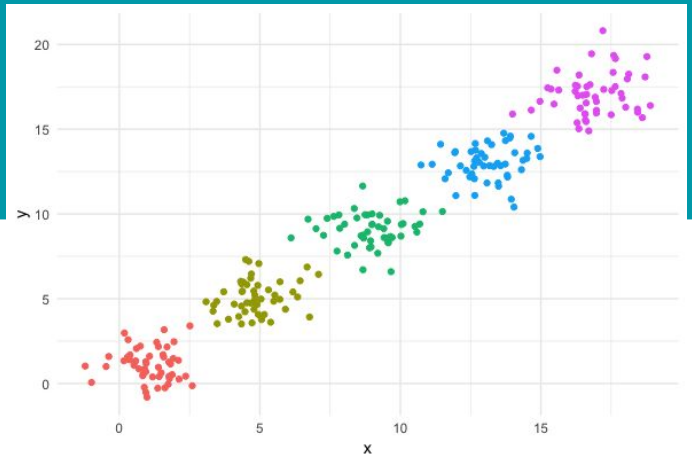
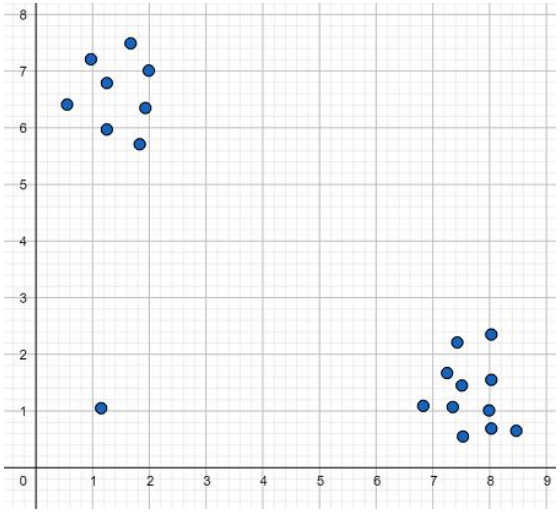


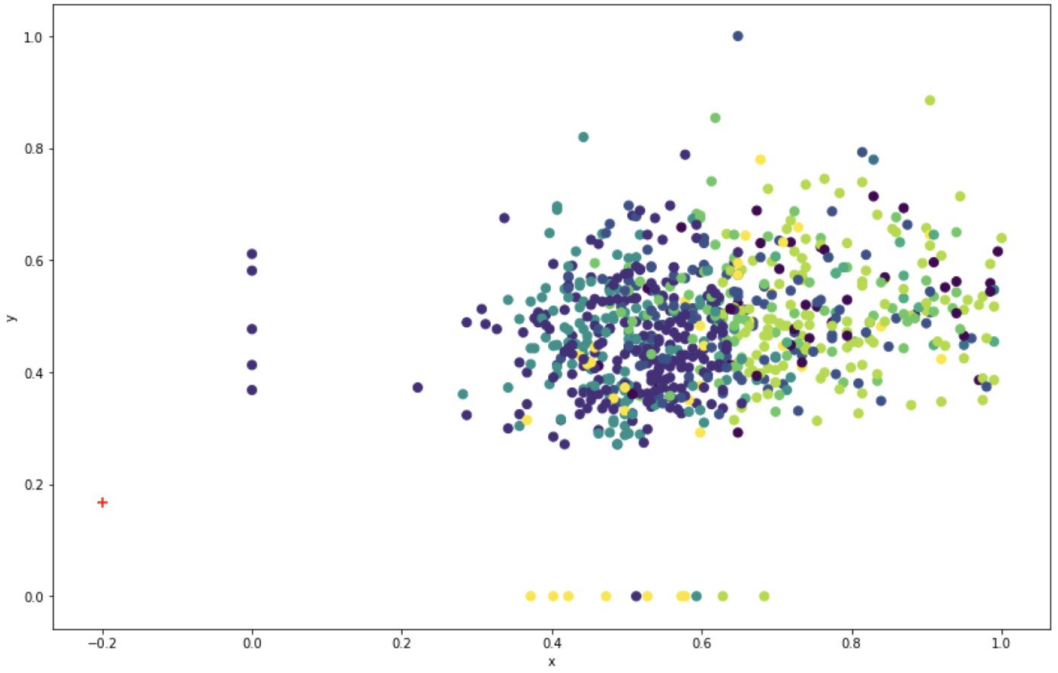
FIGURA 32

Cuadro de tópicos para el *Corpus Catullianum*

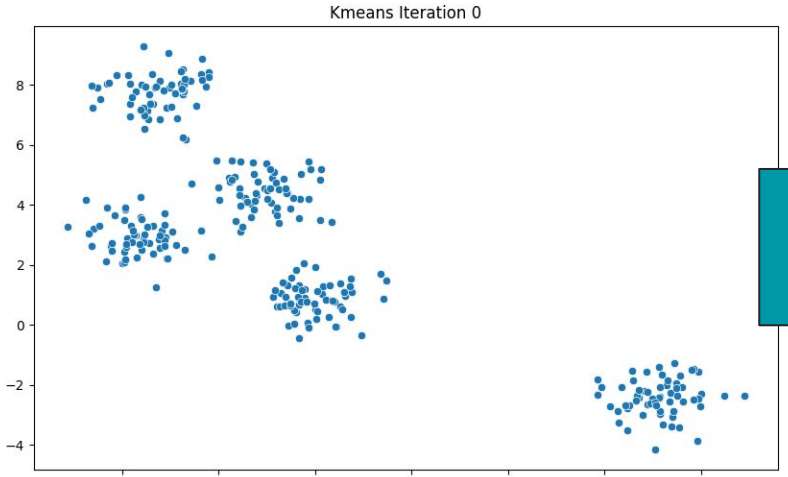
Clusters

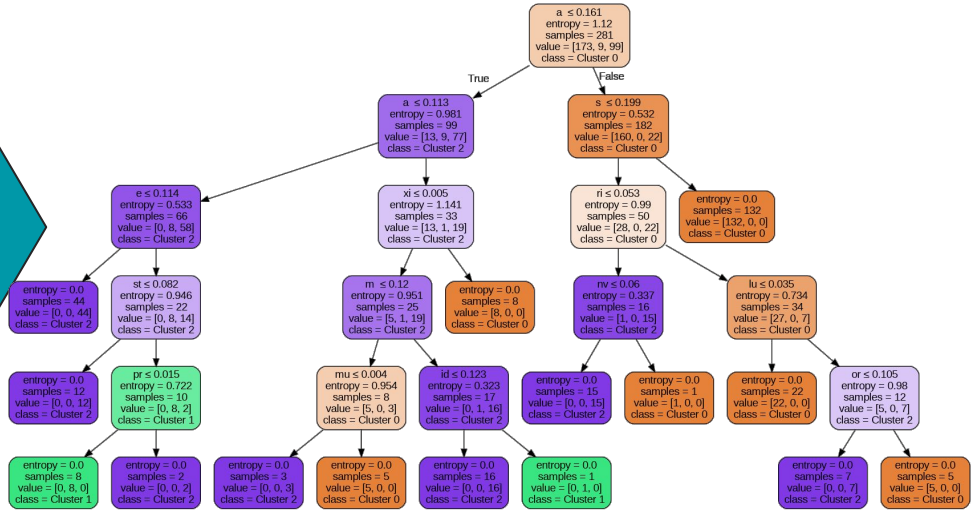

20
15
10
5
0
y
0
5
10
15
x


8
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8
9

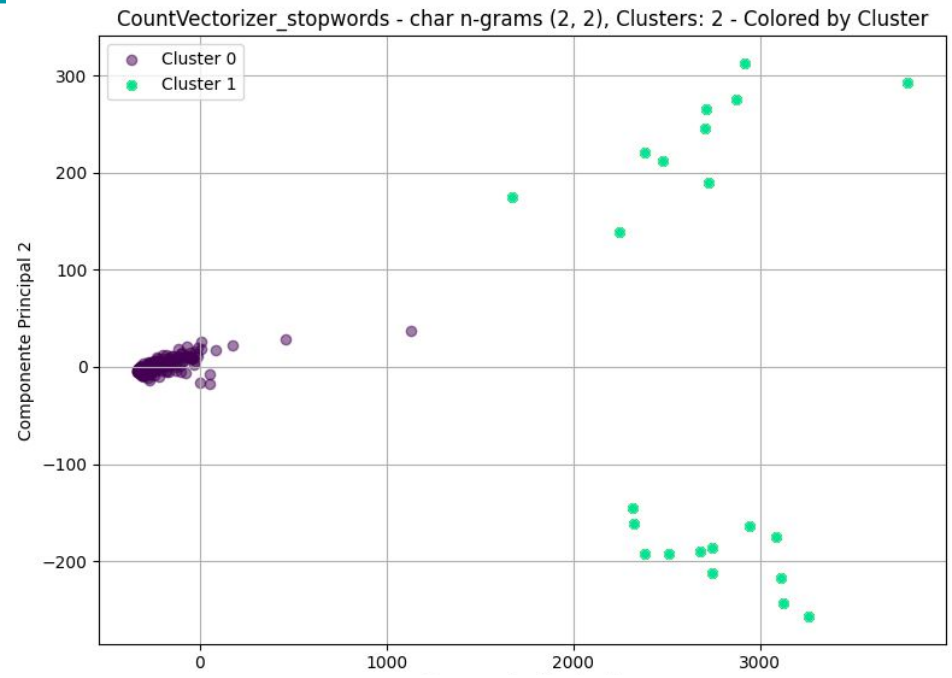

1.0
0.8
0.6
0.4
0.2
0.0
y
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
x

Clusters

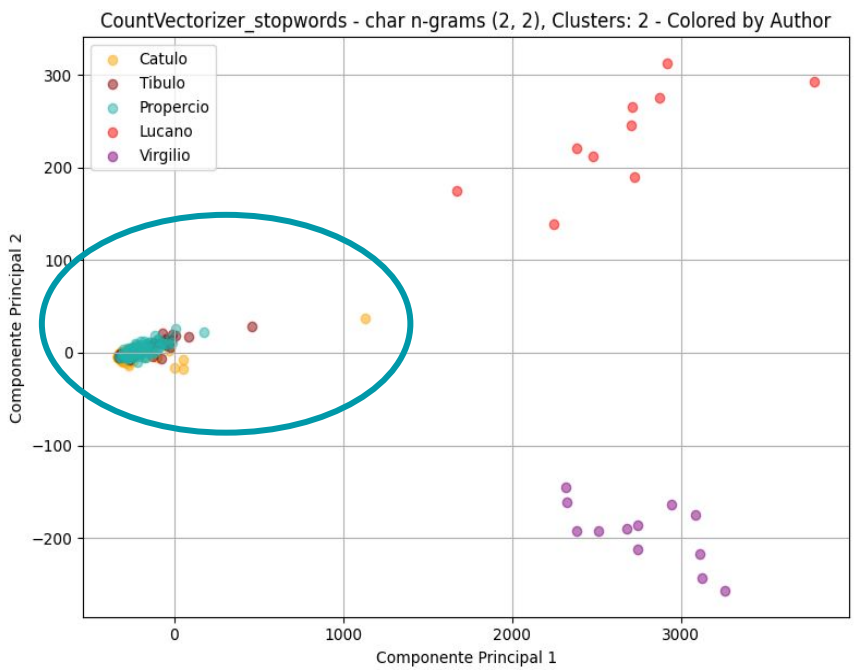

Kmeans Iteration 0
8
6
4
2
0
−2
−4
−2
0
2
4
6
8
10


a ≤ 0.161
entropy = 1.12
samples = 281
value = [173, 9, 99]
class = Cluster 0
True
False
a ≤ 0.113
entropy = 0.981
samples = 99
value = [13, 9, 77]
class = Cluster 2
s ≤ 0.199
entropy = 0.532
samples = 182
value = [160, 0, 22]
class = Cluster 0
e ≤ 0.114
entropy = 0.533
samples = 66
value = [0, 8, 58]
class = Cluster 2
xi ≤ 0.005
entropy = 1.141
samples = 33
value = [13, 1, 19]
class = Cluster 2
ri ≤ 0.053
entropy = 0.99
samples = 50
value = [28, 0, 22]
class = Cluster 0
entropy = 0.0
samples = 132
value = [132, 0, 0]
class = Cluster 0
entropy = 0.0
samples = 44
value = [0, 0, 44]
class = Cluster 2
st ≤ 0.082
entropy = 0.946
samples = 22
value = [0, 8, 14]
class = Cluster 2
m ≤ 0.12
entropy = 0.951
samples = 25
value = [5, 1, 19]
class = Cluster 2
entropy = 0.0
samples = 8
value = [8, 0, 0]
class = Cluster 0
nv ≤ 0.06
entropy = 0.337
samples = 16
value = [1, 0, 15]
class = Cluster 2
lu ≤ 0.035
entropy = 0.734
samples = 34
value = [27, 0, 7]
class = Cluster 0
entropy = 0.0
samples = 12
value = [0, 0, 12]
class = Cluster 2
pr ≤ 0.015
entropy = 0.722
samples = 10
value = [0, 8, 2]
class = Cluster 1
mu ≤ 0.004
entropy = 0.954
samples = 8
value = [5, 0, 3]
class = Cluster 0
id ≤ 0.123
entropy = 0.323
samples = 17
value = [0, 1, 16]
class = Cluster 2
entropy = 0.0
samples = 15
value = [0, 0, 15]
class = Cluster 2
entropy = 0.0
samples = 1
value = [1, 0, 0]
class = Cluster 0
entropy = 0.0
samples = 22
value = [22, 0, 0]
class = Cluster 0
or ≤ 0.105
entropy = 0.98
samples = 12
value = [5, 0, 7]
class = Cluster 2
entropy = 0.0
samples = 8
value = [0, 8, 0]
class = Cluster 1
entropy = 0.0
samples = 2
value = [0, 0, 2]
class = Cluster 2
entropy = 0.0
samples = 3
value = [0, 0, 3]
class = Cluster 2
entropy = 0.0
samples = 5
value = [5, 0, 0]
class = Cluster 0
entropy = 0.0
samples = 16
value = [0, 0, 16]
class = Cluster 2
entropy = 0.0
samples = 1
value = [0, 1, 0]
class = Cluster 1
entropy = 0.0
samples = 7
value = [0, 0, 7]
class = Cluster 2
entropy = 0.0
samples = 5
value = [5, 0, 0]
class = Cluster 0

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

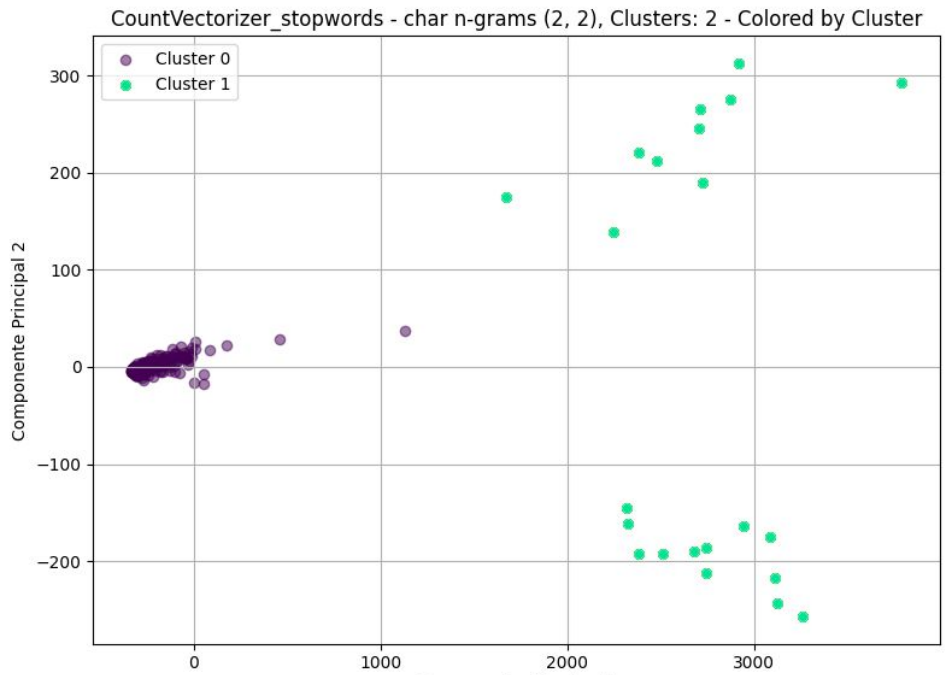


Colored by cluster

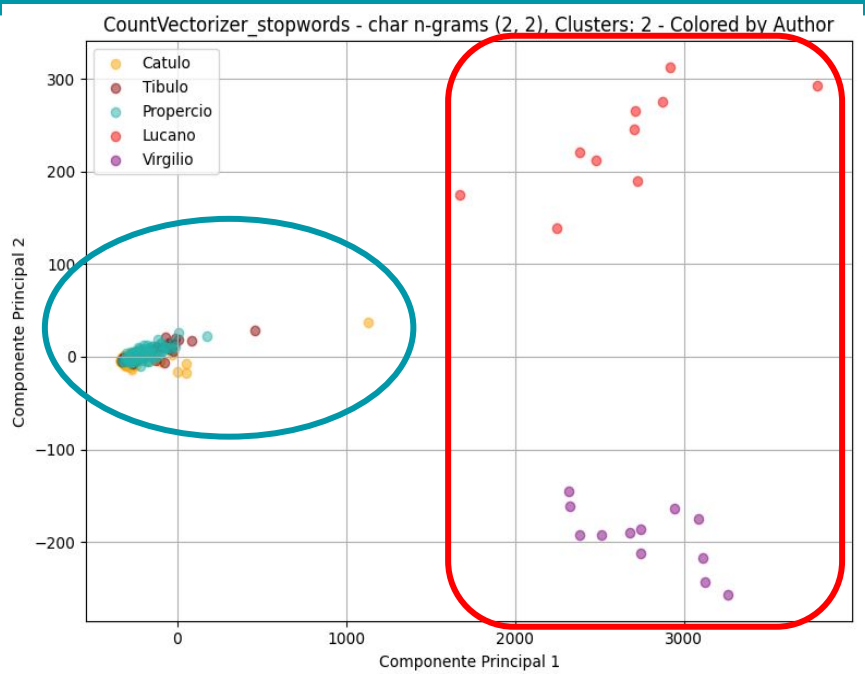


Colored by author

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

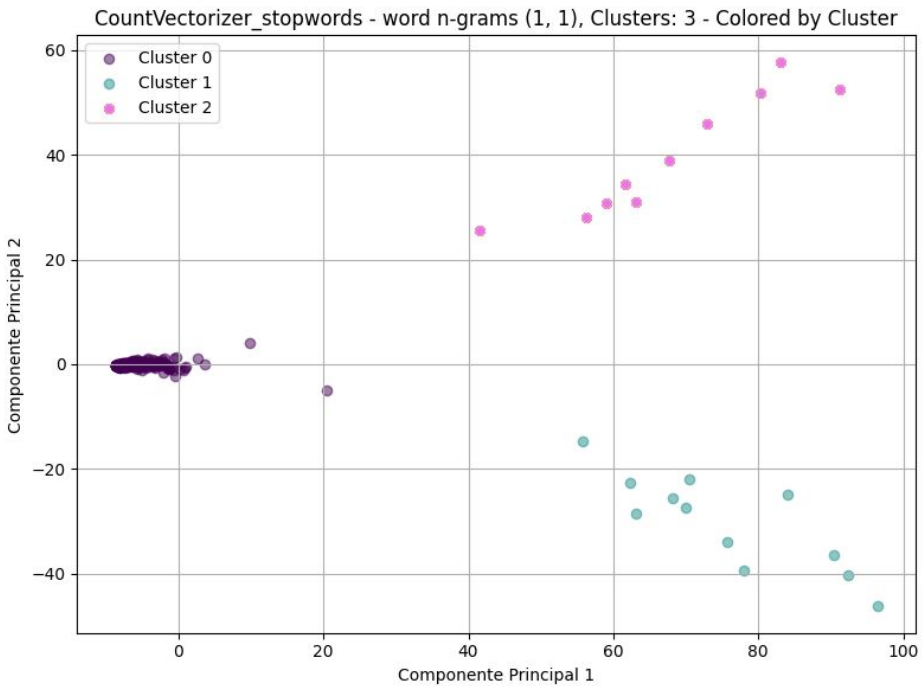


Colored by cluster

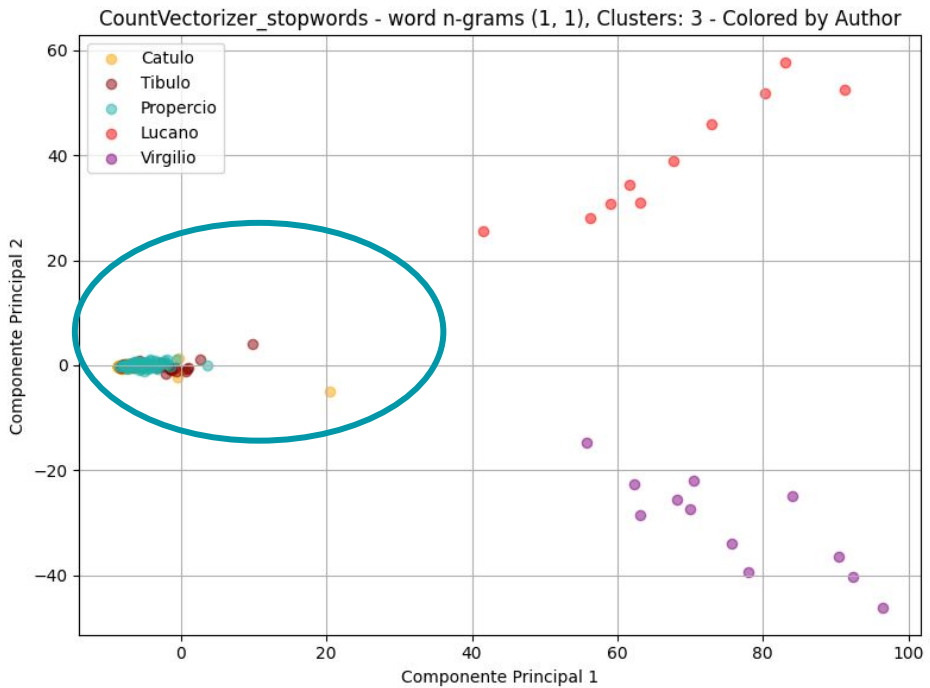


Colored by author

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

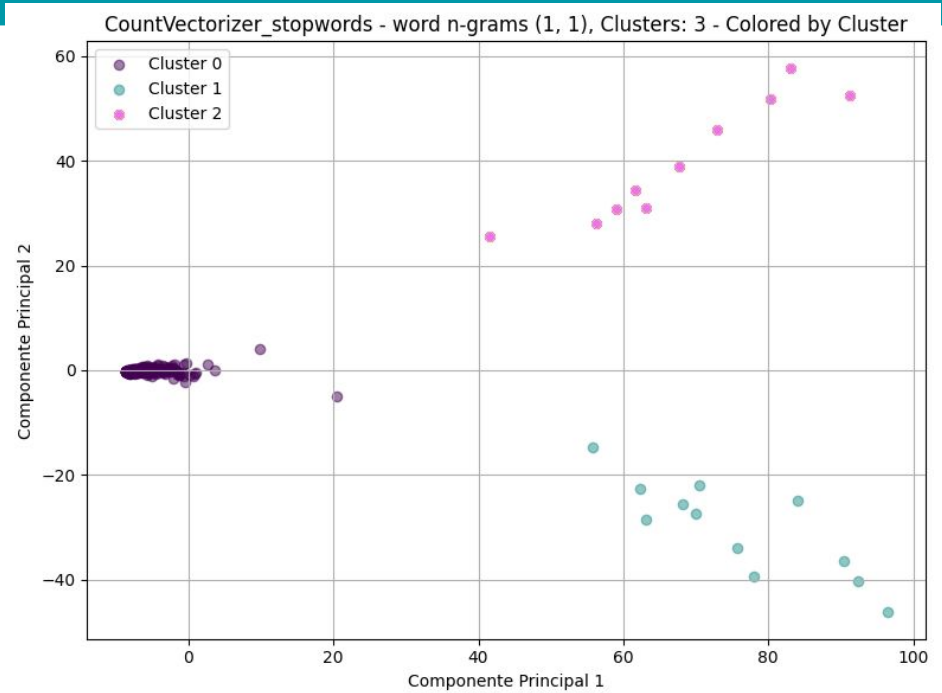


Colored by cluster

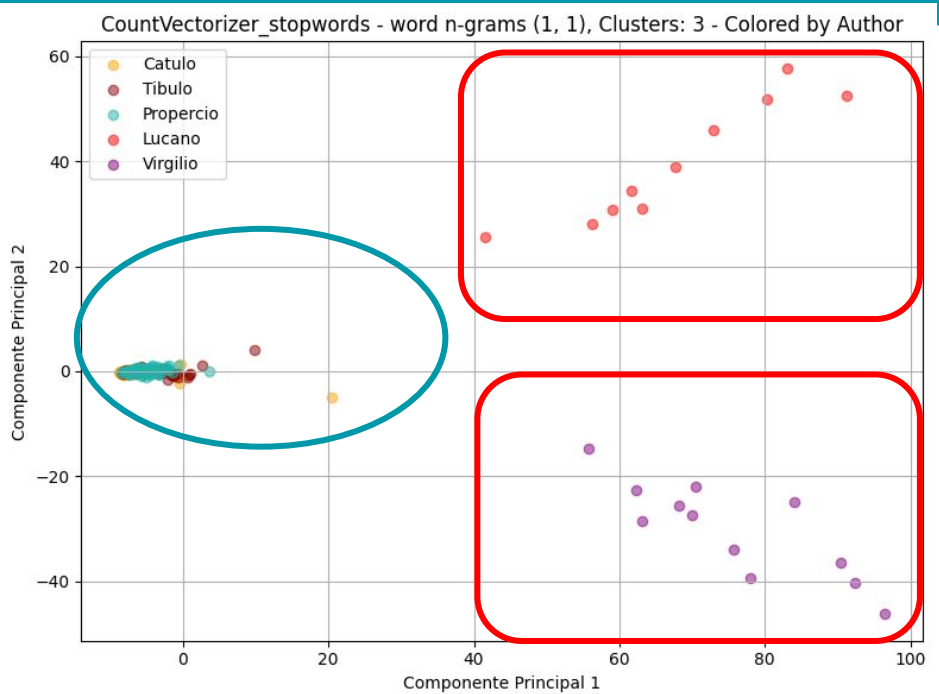


Colored by author

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

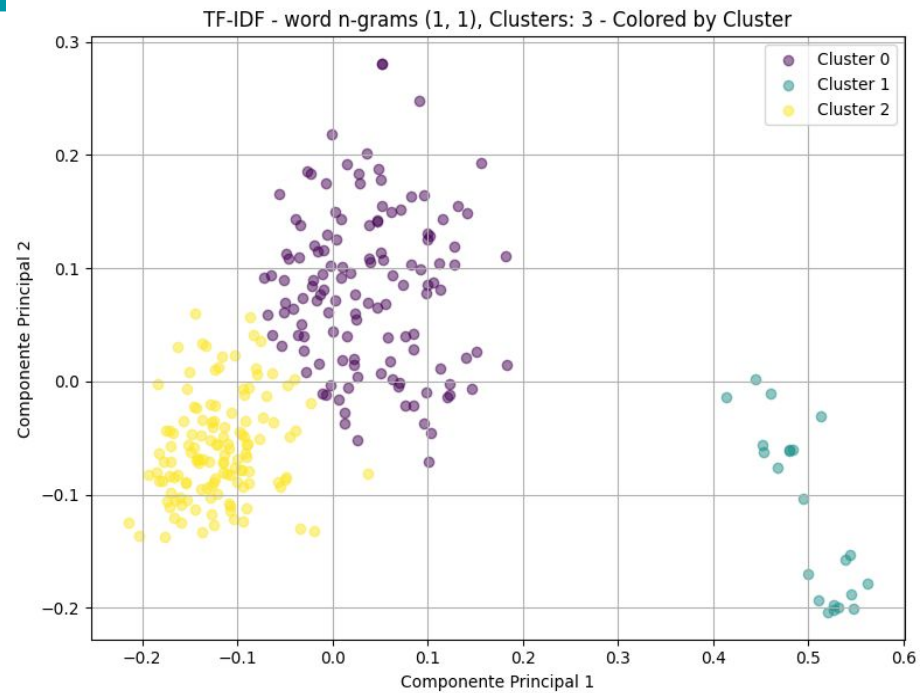


Colored by cluster

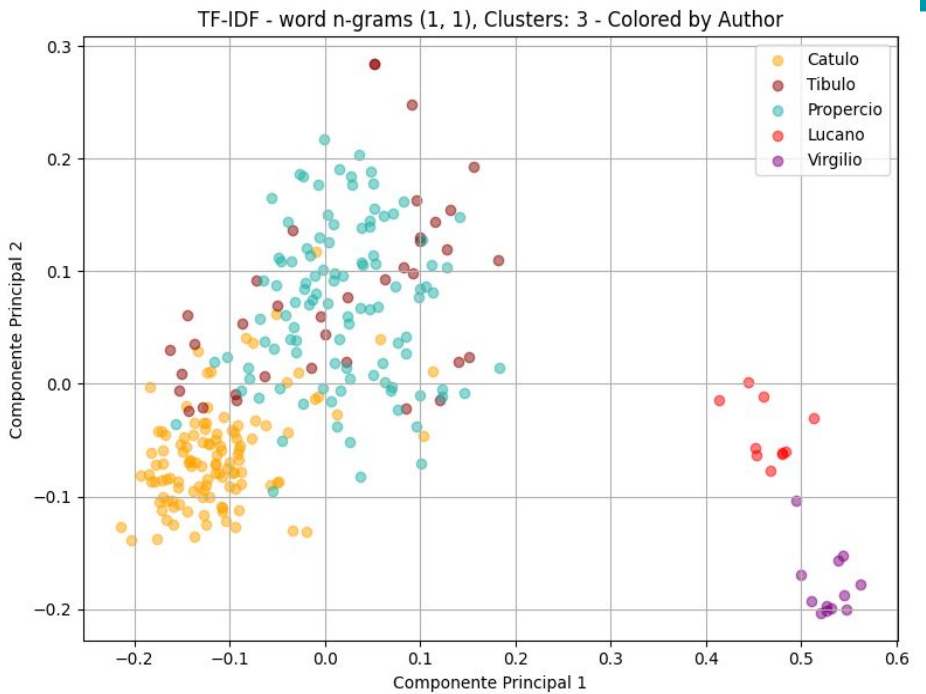


Colored by author

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

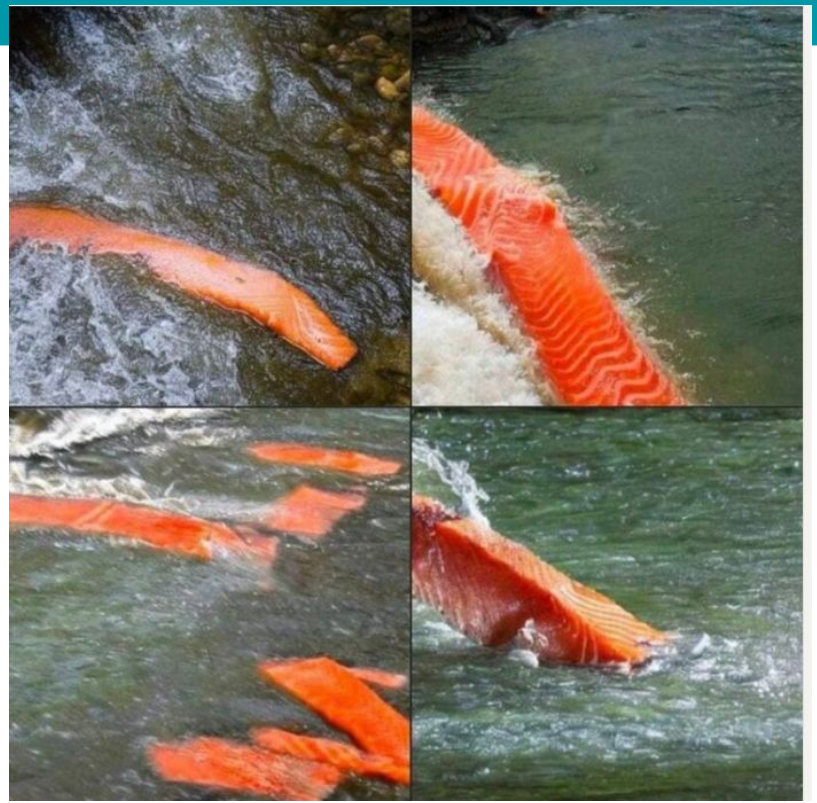


Colored by cluster



Colored by author

Los peligros del Sesgo de Entrenamiento



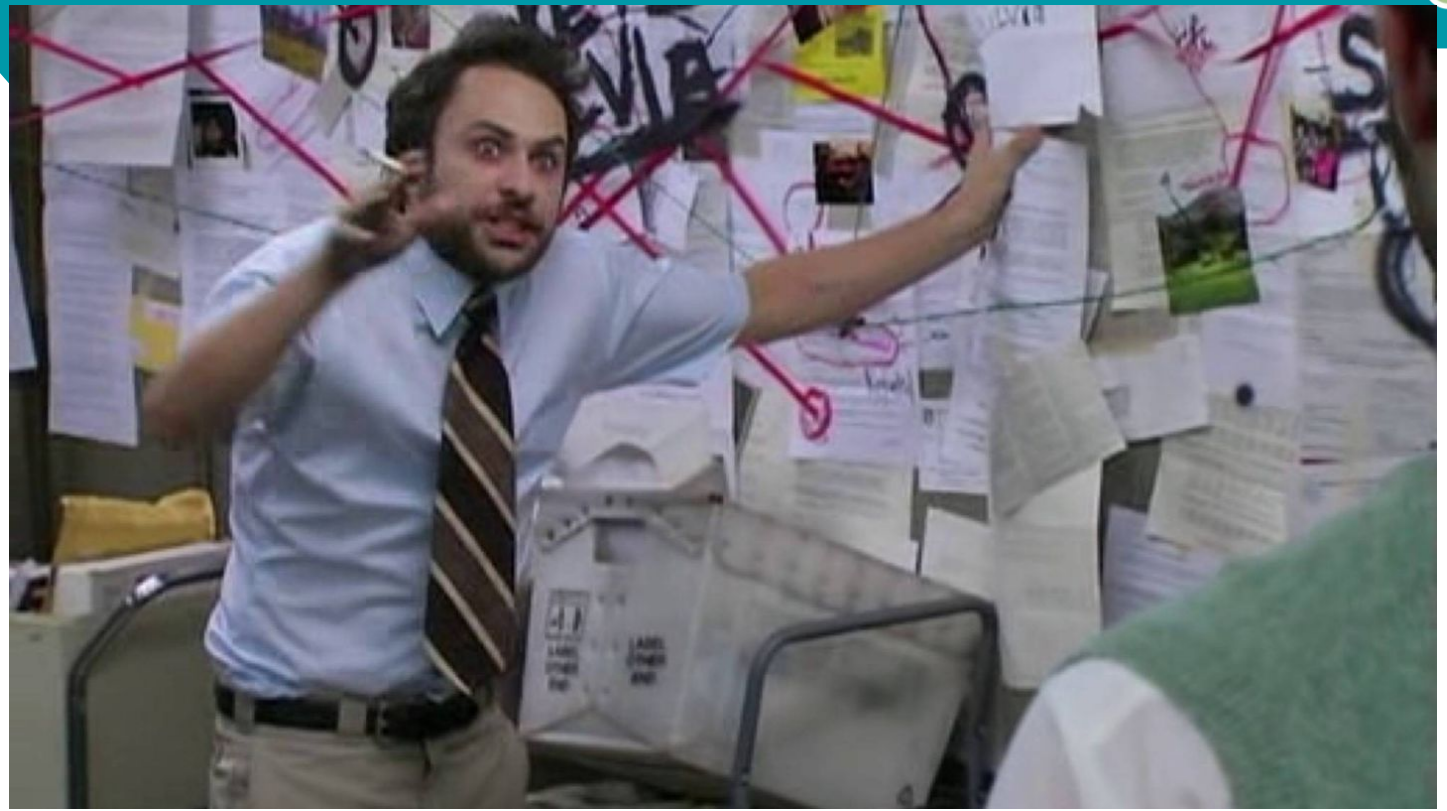
Insólito: en Iowa usan ChatGPT para decidir qué libros quitar de las bibliotecas escolares

Para cumplir con una normativa que exige no incluir en las bibliotecas escolares libros con descripciones de actos sexuales, la junta del distrito escolar de Mason City, en Iowa (EE.UU.) delegó el análisis en ChatGPT, pese a que ofrece respuestas diferentes según quién se lo pregunte

17 de agosto de 2023 • 12:28

Europa Press

¿Dudas, consultas?



Bibliografía y ¿Por dónde comenzar?

Esta gente sabe...

Daniel, J., H, M. J., Peter, N., & Stuart, R. (2014). *Speech and Language Processing*.

Fradejas Rueda, J. M. (2020). *Cuentapalabras: Estilometría y análisis de texto con R para filólogos*. http://www.aic.uva.es/cuentapalabras/

Cuello, C. Y., Jofre Caradonna, V., Garciarena Ucelay, M. J., & Cagnina, L. (2023). On the Importance of Data Representation for the Success of Text Classification. XXVIII Congreso Argentino de Ciencias de la Computación (CACIC) (La Rioja, 3 al 6 de octubre de 2022). http://sedici.unlp.edu.ar/handle/10915/149536

Este no tanto (pero para que vean que se puede)

Nusch, C. J. (2024). 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. *Revista de Humanidades Digitales*, *9*, 1-40. https://doi.org/10.5944/rhd.vol.9.2024.38680

Nusch, C. J., del Rio Riande, G., Cagnina, L. C. C., Errecalde, M. L., & Antonelli, L. (2024). Initial Explorations for Document Clustering Tasks in Latin Elegiac Poets. *Decisioning*. Decisioning 2024, Pereira, Colombia.

Nusch, C. J., Riande, G. del R., Cagnina, L. C., Errecalde, M. L., & Antonelli, L. (2024). Clustering Tasks and Decision Trees with Augustan Love Poets: Cluster Cohesion and Separation in Feature Importance Extraction. *Proceedings of the Computational Humanities Research Conference 2024 (CHR 2024)*.

UNIVERSIDAD NACIONAL DE LA PLATA

Carlos J. Nusch

carlosnusch@prebi.unlp.edu.ar

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