INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Atribución de autoría con aprendizaje automático

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

Ing. Carolina Martín del Campo Rodríguez

DIRECTORES DE TESIS:
Dr. Grigori Sidorov
Dr. Ildar Batyrshin

México, Cuidad de México Junio 2019

SIP-14 bis

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México siendo las 16:00 horas del día 03 del mes de junio de 2019 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Atribución de autoría con aprendizaje automático"

Presentada por el alumno:

MARTÍN DEL CAMPO	**RODRÍGUEZ**	**CAROLINA**
Apellido paterno	Apellido materno	Nombre(s)

Con registro:

A	1	7	0	6	2	3

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

Dr. Grigori Sidorov

Dr. Ildar Batyrshin

Dr. Alexander Gelbukh

Dr. Rolando Quintero Téllez

Dr. Francisco Viveros Jiménez

Dr. Luis Manuel Vilches Blázquez

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Marco Antonio Ramírez Salinas

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de **México** el día **14** del mes **junio** del año **2019**, el (la) que suscribe **Ing. Carolina Martín del Campo Rodríguez** alumno (a) del Programa de **Maestría en Ciencias de la Computación** con número de registro **A170623**, adscrito al **Centro de Investigación en Computación**, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección del **Dr. Grigori Sidorov y Dr. Ildar Batyrshin**, cede los derechos del trabajo intitulado **Atribución de autoría con aprendizaje automático**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección **cm.del.cr@gmail.com, sidorov@cic.ipn.mx y batyr1@gmail.com.** Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Carolina Martín del Campo Rodríguez

Resumen

En esta tesis se presenta un modelo computacional para la tarea de atribución de autoría no supervisada, también conocida como agrupación de documentos por autor.

El modelo se basa en un esquema tradicional de aprendizaje de máquina. Con este modelo, aplicando diferentes propuestas, se logró una mejora con respecto al resultado obtenido en el estado del arte (con el mismo corpora).

Para la extracción de características se propone:

- un método para separar los *tokens* por tipos, para asígnar solo una categoría a cada *token* al usar concatenación de características.
- la utilización de caracteres especiales como parte de los signos de puntuación, esto para mejorar el resultado obtenido al utilizar los n-gramas de caracteres tipados.

Además, se integran medidas de selección de características no supervisada propuestos en el estado del arte (agrupación) que no han sido utilizados para el problema de atribución de autoría no supervisada. Posterior a esto se utiliza el PCA como método de reducción de características para evaluar su comportamiento en base a la creación de grupos por autor.

Se plantean tres diferentes algoritmos de agrupación (jerárquico, k-means, espectral) para resolver la tarea, así como cinco índices (Calinski-Harabaz, Davies-Bouldin, SD, S_Dbw, Silhouette) para la validación de los grupos.

Se utiliza la medida *similitud del coseno con pesos* para reducir los valores entre los vectores (que representan los documentos) donde los atributos son exclusivos. Esta medida es la utilizada para determinar las distancias entre los documentos, las cuales son ocupadas posteriormente por el algoritmo de agrupación.

El modelo fue desarrollado y probado con el corpora proporcionado por el PAN, laboratorio de evaluación en detección de plagio, identificación de autoría y mal uso de software social (PAN 2017).

Como parte del desarrollo de esta tesis se participó en la tarea de atribución de autoría de dominio cruzado (*Cross-domain Authorship Attribution*) propuesta por el PAN en 2018 y 2019. Además se hizo la publicación de los artículos: *"CIC-GIL Approach to Cross-domain Authorship Attribution: Notebook for PAN at CLEF 2018"* y *"Enhancement of Performance of Document Clustering in the Authorship Identification Problem with a Weighted Cosine Similarity"* y se apoyo para realizar el artículo *"Hierarchical Clustering Analysis: The Best-Performing Approach at PAN 2017 Author Clustering Task"*.

Abstract

This thesis presents a computational model for the unsupervised authorship attribution task, also known as clustering of documents by author.

The model is based on a traditional scheme of machine learning. With this model, applying different proposals, an improvement was achieved with respect to the result obtained in the state of the art (with the same corpora).

For the extraction of characteristics it's proposed:

- a method to separate the tokens by types, to assign only one category to each token when using concatenation of characteristics.
- the use of special characters as part of the punctuation marks, this to improve the result obtained when using the typed character n-grams.

In addition, unsupervised features selection measures proposed in the state of the art (clustering) that have not been used for the problem of unsupervised authorship attribution are integrated. After this the PCA is used as a method of feature reduction to evaluate its behavior based on the creation of groups by author.

Three different cllustering algorithms are presented (hierarchical, k-means, spectral) to solve the task, as well as five indexes (Calinski-Harabaz, Davies-Bouldin, SD, S _Dbw, Silhouette) for the validation of the clusters.

The measure *weighted cosine similarity* is used to reduce the values between the vectors (representing the documents) where the attributes are exclusive. This measure is the one used to determine the distances between the documents, which are later occupied by the grouping algorithm.

The model was developed and tested with the corpora provided by the PAN, evaluation laboratory in plagiarism detection, identification of authorship and misuse of social software (PAN 2017).

As part of the development of this thesis, we participated in the Cross-domain Authorship Attribution task proposed by PAN in 2018 and 2019. In addition, the following articles were published: "CIC-GIL Approach to Cross-domain Authorship Attribution: Notebook for PAN at CLEF 2018" and "Enhancement of Performance of Document Clustering in the Authorship Identification Problem with a Weighted Cosine Similarity" and supported to carry out the article "Hierarchical Clustering Analysis: The Best-Performing Approach at PAN 2017 Author Clustering Task".

Agradecimientos

Agradezco a mis directores de tesis, el Dr. Grigori Sidorov y el Dr. Ildar Batyrshin por su orientación, apoyo, disposición y atenciones hacia mi persona.

Agradezco a mis padres y hermanos por su apoyo incondicional y consejos.

Agradezco al Centro de Investigación en Computación, al Instituto Politécnico Nacional y al CONACyT por las facilidades y recursos otorgados durante mi estancia en el programa de MCC del CIC.

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

La Atribución de Autoría (AA) consiste en identificar el autor de un documento anónimo. Muchos estudios sobre AA están enfocados en ver esta como un problema de clasificación de textos etiquetados multiclase; sin embargo, existen aplicaciones en donde no es fácil o incluso posible encontrar tales etiquetas en los datos y es necesario un modelo no supervisado para la AA.

En esta investigación se analiza como un conjunto de características determinado, la selección/reducción de características y algoritmos de agrupamiento pueden utilizarse para hacer la AA no supervisada en diferentes lenguajes (inglés, holandés y griego) y géneros (artículos y reseñas).

Este documento esta organizado de la siguiente manera: en el capítulo 2 se muestran algunos conceptos básicos para entender los pasos propuestos para la agrupación de documentos; en el capítulo 3 se muestran diversos trabajos que se han hecho relacionados con la AA no supervisada; en el capítulo 4 se explica paso a paso el método propuesto; en el capítulo 5 se muestran las diversas pruebas realizadas y los resultados obtenidos; en el capítulo 6 se explican las conclusiones y se presenta el trabajo a futuro.

1.1. Planteamiento del problema

La Atribución de Autoría (AA) consiste en identificar el autor de un documento anónimo. Existen muchas subtareas relacionadas con el campo de AA, como son: identificación de autor [2], ofuscación de autor [3], agrupación de documentos por autor [4], detección de plágio [5] y perfilado del autor [6]. Aplicaciones de este problema incluyen preprocesamiento automático de textos en repositorios (Web), recuperación de los documentos escritos por el mismo autor, entre otros.

La tarea de agrupación de autores esta definida como: dada una colección de documentos de autoría desconocida, se deben de agrupar los documentos escritos por el mismo autor, de tal forma que cada grupo corresponda a un autor direfente.

El **PAN**[1] es un foro internacional que se encarga de fomentar la investigación forense de texto digital mediante la organización de tareas de evaluación compartidas. Para esto, año con año lanza una serie de diferentes tareas enfocadas a ciertos temas en específico, tales como: identificación de autor (atribución de autoría, detección de cambio de estilo, verificación de autoría, agrupación de autores, entre otras), perfil de autor (predicción de género, perfilado de celebridades, predicción de lenguage, predicción de edad, entre otras) y ofuscación de autor (enmascarado de autor y evaluación de ofuscación).

[1] Véase https://pan.webis.de/

En el contexto de las tareas lanzadas por el PAN en 2017 [7] propusieron la tarea de agrupación de documentos por autor (*author clustering*). El corpora proporcionado para esta tarea será el utilizado en este trabajo de investigación.

1.2. Objetivo general

El objetivo general es proponer un sistema basado en técnicas de aprendizaje de máquina tradicionales capaz de realizar la tarea de atribución de autoría no supervisada, mejorando el resultado conocido en el estado del arte (con el mismo corpus).

1.3. Objetivos específicos

Los objetivos específicos de este trabajo de investigación son:

- Seleccionar el preprocesamiento específico que se debe utilizar para los textos.
- Mejorar el método de extracción de características.
- Mejorar la agrupación de documentos (B^3 *F-score*) por autor utilizando la selección/reducción de características.
- Seleccionar el algoritmo de agrupación que, de acuerdo a las características seleccionadas, mejore el resultado.

Capítulo 2

Marco teórico

Cuando hablamos de atribución de autoría no supervisada existen una serie de pasos que pueden seguirse (de manera tradicional) para resolver el problema. De manera general, estos pueden dividirse en:

- Pre-procesamiento y extracción de características (determinación del estilo de los autores).
- Representación de los textos.
- Pesado de las características.
- Selección y/o reducción de características.
- Análisis de la agrupación (clustering): determinar que algoritmo que debe utilizarse para la agrupación y el método para determinar el número de grupos, así como la medida de similitud o distancia a utilizar.
- Cálculo del error al hacer la agrupación mediante alguna métrica.

En las siguientes secciones se introduce cada uno de los pasos descritos anteriormente.

2.1. Pre-procesamiento y extracción de características (determinación del estilo de los autores)

El estilo de un autor puede ser identificado seleccionando características léxicas, sintácticas, semánticas, basadas en caracteres, entre otras. En [8] se hace un estudio de los diferentes tipos de características que pueden ser considerados para definir el estilo de un autor, como son: características léxicas (n-gramas de palabras), sintácticas (*Part-of-Speech PoS*), semánticas (errores), basadas en caracteres (n-gramas de caracteres), entre otras.

Estos tipos de características puede ser divididos en dos categorías: dependientes del idioma y no dependientes del idioma. Las características dependientes del idioma engloban a las características semánticas y sintácticas, puesto que es imposible hacer un etiquetado de las palabras en una oración o la detección de errores si no se conoce el idioma del texto. Las características léxicas pueden verse como características independientes de idioma, aunque el conocimiento del idioma puede ayudar a determinar correctamente cada palabra. Las basadas en caracteres son características no dependientes del idioma, puesto que solo identifican los consecutivos de caracteres; por otro lado, existen clasificaciones de los n-gramas de caracteres que pueden ayudar a definir mejor el estilo del autor, como son los n-gramas de caracteres tipados.

2.1.1. N-gramas de caracteres tipados

Los n-gramas de caracteres tipados, introducidos por [9], son subgrupos de n-gramas de caracteres que corresponden a tres diferentes aspectos lingüísticos: morfo-sintácticos (representado por los *affix n-gramas*), contenido temático (representado por *word n-gramas*) y estilo (representado por *punctuation n-gramas*). Estos subgrupos son llamados súper categorías; cada una de estas está dividida en diferentes categorías.

Affix n***-gramas*** [9]

Los *Affix n-gramas*, hasta cierto punto, capturan la morfología. Para esto se hace un análisis del inicio y el fin de las palabras. Estos n-gramas se dividen en cuatro categorías.

- ***prefix***: n-gramas de caracteres que cubre los primeros n caracteres de la palabra. La palabra tiene que tener una longitud de al menos $n - 1$.
- ***suffix***: n-gramas de caracteres que cubre los últimos n caracteres de la palabra. La palabra tiene que tener una longitud de al menos $n - 1$.
- ***space-prefix***: n-gramas de caracteres que comienza con un espacio.
- ***space-suffix***: n-gramas de caracteres que termina con un espacio.

Word n***-gramas*** [9]

Capturan palabras parciales y otras muestras relevantes de palabras. Se dividen en tres categorías.

- ***whole-word***: n-gramas de caracteres que cubre todos los caracteres de una palabra. La palabra tiene que tener una longitud de n.
- ***mid-word***: n-gramas de caracteres que cubre n caracteres de una palabra. La palabra tiene que tener una longitud de al menos $n + 2$. No cubre ni el primer ni el último caracter de la palabra.
- ***multi-word***: n-gramas que abarca múltiples palabras. Se identifica por la presencia de un espacio en medio del n-grama.

Punctuation n***-gramas*** [9]

Captura los patrones que existen en el uso de puntuaciones. Existen tres categorías de estos.

- ***beg-punct***: n-gramas de caracteres cuyo primer caracter es un signo de puntuación, pero los otros valores no lo son.
- ***mid-punct***: n-gramas de caracteres que tiene al menos un signo de puntuación que no esté al inicio ni al final.
- ***end-punct***: n-gramas de caracteres cuyo último caracter es un signo de puntuación, pero los otros valores no lo son.

Algunas de las categorías de los n-gramas de caracteres tipados han presentado mayores capacidades predictivas en la tarea de atribución de autoría que cuando se utilizan todos los tipos posibles de n-gramas (tipados y no tipados) [9]. La redefinición utilizada por [10] de estas categorías elimina la ambigüedad presentada en cada 3-grama a exactamente una categoría y no excluye ningún n-grama (como en el caso de puntuación consecutiva de la propuesta original). Además, los autores mostraron que algunas características tienen un mejor resultado que otras para la atribución de autoría.

2.2. Representación de los textos

Una vez extraídas las características de cada uno de los textos, es necesario representar estos de tal forma que sea posible hacer una comparación entre si, para posteriormente poder determinar la autoría de los mismos basados, hasta cierto punto, en su similitud.

En general, el modelo de espacio vectorial es ampliamente utilizado en las ciencias de la computación, puesto que permite comparar dos objetos de manera formal [11]. Para esto se represeta cada documento como un vector de n dimensiones, en donde n es el número de características extraídas. Este vector puede ser un vector binario (1 la características existe y 0 si no) o una valor entero (conteo del número de veces que aparece cada característica).

El siguiente ejemplo explíca de forma sencilla como pueden representarse dos oraciones en un modelo de espacio vectorial:

Oracion_1: *El perro está comiendo croquetas.*

Oracion_2: *El gato está durmiendo.*

Primero es necesario determinar cual será el tipo de características con los que las oraciones serán representadas, para este ejemplo una representación de bolsa de palabras será considerada. Las oraciones quedán representadas como:

Oracion_1: *'El', 'perro', 'está', 'comiendo', 'croquetas', '.'*

Oracion_2: *'El', 'gato', 'está', 'durmiendo', '.'*

Posteriormente es necesario obtener un diccionario que incluya todas las palabras de ambas oraciones. El diccionario queda representado como:

Diccionario: *'El', 'perro', 'gato', 'está', 'comiendo', 'croquetas', 'durmiendo', '.'*

Un vez obtenido el diccionario, se considera a cada una de las palabras en el diccionario como una característica para hacer la representación vectorial. Para esto consideraremos los vectores x (que representa la **Oracion_1**) y y (que representa la oración **Oración_2**). Para la formación de los vectores se considerarán cada unos de los valores en el diccionario; comenzando con 'El', puesto que existe en ambas oraciones la primera dimensión en ambos vectores queda representada con 1, posterioemente la palabra 'perro' existe solo en la primera oración, por lo que para la segunda dimensión de los vectores, para el vector x se tendrá un 1, para el vector y se tendrá un cero (puesto que no existe esa palabra en la segunda oración), y así posteriormente para cada una de las palabras en el diccionario, se tiene:

$\boldsymbol{x}$=*[1,1,0,1,1,1,0,1]*

$\boldsymbol{y}$=*[1,0,1,1,0,0,1,1]*

De esta forma, con los vectores x y y se puede determinar la similitud entre las oraciones. A este tipo de representación de los textos se le conoce como '*representacion en bolsa de palabras*', independientemente del tipo de características utilizadas.

Existen otros tipos de representaciones de los textos, como son las '*palabras embebidas*', las cuales no serán descritas puesto que van más allá del alcance de este trabajo de investigación.

2.3. Pesado de las características

Al extraerse las características y tener la representación de los textos (representacion en bolsa de palabras) es necesario implementar algún tipo de pesado en las características, puesto que los valores iniciales de las mismas pueden verse influenciados por la longitud de los textos, además de que si tenemos una palabra que existe para todos los textos la misma cantidad de veces, esta palabras en realidad no marca una diferencia entre los documentos. Es para esto que existen las *funciones de pesado global*.

Las *funciones de pesado global* miden la importancia de cada término en toda la colección de documentos [1]. Existen funciones que permiten asignar un peso específico a cada una de las características haciendo una relación entre la ocurrencia de las características por documento y por colección de documentos. Generalmente las principales son *tf-idf* y *Log-Entropy*.

2.3.1. *tf-idf*

La frecuencia de una palabra se denomina tf (term frecuency) y representa cuántas veces la palabra (término) ocurre en un documento [11]. Así se tiene que tf_{ij} representa cuántas veces la palabra i aparece en el documento j. Normalmente se utiliza el logaritmo de tf: $log(tf + 1)$ (se hace la suma de uno con el fin prever valores en cero).

La frecuencia inversa de documentos por sus siglas en inglés *idf* (inverse document frequency) se calcula con la ecuación (2.1):

$$idf_i = log\left(\frac{N}{DF_i}\right), \tag{2.1}$$

donde N es el número total de documentos en la colección, DF_i es el número de documentos en donde la palabra i se encuentra por lo menos una vez. Se utiliza el logaritmo para suavizar la influencia de las frecuencias altas. El *idf* se calcula para cada palabra en una colección dada, es decir, depende de la colección, pero no depende de un documento específico en la colección.

Así, se tiene el cálculo de *tf-idf* con la ecuación (2.2):

$$tf\text{-}idf_i = log(tf_i + 1) * log\left(\frac{N}{DF_i}\right), \tag{2.2}$$

2.3.2. *Log-Entropy*

Investigaciones anteriores en juicios de similitud de documentos [12, 13] han mostrado que el pesado global basado en entropía es generalmente mejor que el modelo *tf-idf*. El pesado *log-entropy* (*le*) es calculado con la ecuación (2.3):

$$le_{ij} = e_i \times \log(tf_{ij} + 1), \tag{2.3}$$

donde, tf_{ij} es la frecuencia del término i en el documento j, e_i es calculado mediante la ecuación (2.4):

$$e_i = 1 + \sum_j \frac{p_{ij} \times \log p_{ij}}{\log n}, \quad donde \quad p_{ij} = \frac{tf_{ij}}{gf_i}, \tag{2.4}$$

donde n es el número de documentos, y gf_i es la frecuencia del término i en la colección completa. Un término que aparece una vez en cada documento tendrá un peso de cero. Un término que aparece una vez en solo un documento tendrá un peso de uno. Cualquier otra combinación de frecuencias asigna a un término dado un preso entre cero y uno.

2.4. Selección y reducción de características

Los datos con dimensiones altas tienden a ser comunes en los problemas de aprendizaje, sobre todo en los relacionados con el procesamiento de lenguaje natural (PLN). Cuando se utiliza una representación de bolsa de palabras (descrita en 2.2) usualmente se obtiene una dimesionalidad muy alta, que en general se relaciona con la logitud de los textos y los diferentes tipos de características considerados.

Típicamente cuando se tiene muchas dimensiones existen características que son redundantes y/o irrelevantes. Es por esto que la reducción y la selección de características juegan un papel muy importante para las diversas tareas del aprendizaje de máquina.

2.4.1. Reducción de características

La reducción de características consiste en realizar un análisis de dependencia entre las características para hacer una combinación de dimensiones y así obtener un número de dimensiones reducido. Para esto existen varios métodos, como son en el Análisis de Componentes Principales (por sus siglas en inglés PCA-Principal Component Analysis), el Análisis Semántico Latente (por sus siglas en inglés LSA-Latent Semantic Analysis), entre otros.

Análisis de Componentes Principales

El Análisis de Componentes Principales (PCA) es un procedimiento estadístico que utiliza una transformación ortogonal para convertir un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables no correlacionadas linealmente llamadas componentes principales. El número de componentes principales es menor o igual que el número de variables originales [14].

Análisis Semántico Latente

El Análisis Semántico Latente (LSA) es la aplicación de una técnica de procesamiento de matrices tomada de álgebra lineal. Esta técnica se llama descomposición en valores singulares (SVD, Singular Value Decomposition) y lo que permite es encontrar en las matrices aquellas filas con mayor información (valores grandes) y así eliminar las filas de menor información (valores pequeños). En este sentido, esencialmente el análisis semántico latente no es nada más que una técnica de reducción de dimensiones de un espacio vectorial [11].

2.4.2. Selección de características

La selección de características (SC), a diferencia de los métodos de reducción de características que hace una combinación lineal de dimensiones, se encarga de descartar aquellas dimensiones que son redundantes y/o irrelevantes. Normalmente la selección de características se divide en dos enfoques: de envoltura (*wrappers*) y filtros (*filters*).

De envoltura (*wrappers*)

Este tipo de selección de características trabaja en conjunto con un clasificador que va evaluando el comportamiento del subconjunto de características seleccionado para así hacer un aprendizaje de los datos de entrenamiento. Ya que para cada subconjunto de características es necesario hacer la evaluación con el algoritmo de aprendizaje este tipo de selección de características no es recomendable para datos con un valor alto de dimensiones.

Filtros

Determina que tan buenos son los subconjuntos de características utilizando las mismas características, es decir, no necesita de un clasificador para determinar el mejor subconjunto. Generalmente se hace un

ordenamiento de las características basado en su peso, posteriormente se establece un umbral que determina el número de características (o porcentaje de las características) que se conservará, todas las características que estén por debajo de este umbral son descartadas [15]. Algunos tipos de filtros son los siguientes:

- **Varianza de término** (TV, por sus siglas en inglés, Term Variance), fue introducido por [16] para evaluar la calidad de cada término. Esto mediante el cálculo de la varianza de cada término en todos los conjuntos de datos. La fórmula para el cálculo de TV está descrita en la ecuación (2.5):

$$\begin{aligned} TV_i &= \frac{1}{n}\sum_{j=1}^{n}(X_{ij} - \overline{X}_i)^2 \\ \overline{X}_i &= \frac{1}{n}\sum_{j=1}^{n} X_{ij}, \end{aligned} \tag{2.5}$$

  donde n es el número de muestras y X_{ij} representa la característica i del documento j.

- **Diferencia promedio absoluta** (MAD, por sus siglas en inglés, Mean Absolute Variance), obtiene la diferencia absoluta del valor promedio [15], está definida por la ecuación (2.6). Como se puede observar, la única diferencia entre MAD y TV es el cambio del cuadrado por el valor absoluto:

$$\begin{aligned} MAD_i &= \frac{1}{n}\sum_{j=1}^{n}|X_{ij} - \overline{X}_i| \\ \overline{X}_i &= \frac{1}{n}\sum_{j=1}^{n} X_{ij}, \end{aligned} \tag{2.6}$$

  donde n es el número de muestras y X_{ij} representa la característica i del documento j.

2.5. Análisis de la agrupación (clustering)

El objetivo de la agrupación es que los objetos en el mismo grupo deben ser similares (o estar relacionados) y ser diferentes (o no relacionados) con los objetos en otros grupos [17]. La agrupación es considerada como buena cuando la distancia entre los miembros de cada grupo es pequeña y la distancia entre los grupos es grande.
De esta forma, teniendo que $X = \{X_1, X_2, ..., X_n\}$ es un conjunto de n elementos de datos, cada uno con f características de valor real, es decir, $X_i \in \mathbb{R}^f$, $i = 1, 2, ..., n$ y $X_i = \{X_{i1}, X_{i2}, ..., X_{if}\}$ son las f características de X_i. Dado X, la agrupación es una partición $C = \{C_1, C_2, ..., C_k\}$ de X para un k, donde $\{k|k > 1, k < n\}$ tal que:

- $\bigcup_{i=1}^{k} C_i = X$
- $C_i \bigcap C_j = \emptyset \quad \forall i, j \in \{1, 2, ..., k\}, i \neq j$
- $C_i \neq \emptyset \quad \forall i \in \{1, 2, ..., k\}$

Generalmente, la agrupación se divide en dos tipos, agrupación jerárquica y agrupación particional. Pese a que este tipo de algoritmos son los más utilizados existen otros tipos de agrupación importantes, como es la agrupación espectral. A continuación, se describen cada uno de estos tipos de agrupación.

2.5.1. Agrupación jerárquica (*Hierarchical Clustering*)

Este enfoque de agrupación produce una secuencia anidada de particiones con un solo (todos incluidos) grupo en la parte alta y grupos de un solo individuo en el fondo. Generalmente la agrupación jerárquica se divide en dos tipos:

- Aglomerativa: Empieza con grupos conteniendo un solo miembro y va uniendo estos grupos.
- Divisiva: Empieza un grupo que contiene a todos los miembros y va dividiéndolos.

La mayoría de los algoritmos utiliza de dos en dos grupos ya sea para la forma aglomerativa o la divisiva. Además, para estos tipos de agrupación existen diferentes técnicas, las principales son [17]:

- Unión simple o MIN: la proximidad entre dos grupos está definida como el mínimo de las distancias entre dos puntos cualquiera en diferentes grupos.
- Unión completa o MAX: la proximidad entre dos grupos está definida como el máximo de las distancias entre dos puntos cualquiera en diferentes grupos.
- Promedio de grupo: la proximidad entre dos grupos está definida como el promedio de las proximidades por pares de puntos en los diferentes grupos. Este es un enfoque intermedio entre MIN y MAX.

2.5.2. Agrupación particional

Este enfoque crea un solo nivel de partición de puntos de datos. Si k es el número de grupos deseados el enfoque particional típicamente encuentra todos los k grupos al mismo tiempo. Esto contrasta con los típicos enfoques tradicionales, que parten un grupo para obtener dos nuevos o grupos o unen dos grupos para obtener un nuevo grupo.

K-Means

K-means es un algoritmo iterativo que trata de particionar el conjunto de datos en k distintos subgrupos no superpuestos, donde cada punto de datos pertenece a un solo grupo. Este algoritmo trata que los puntos de datos que pertenecen a un mismo grupo sean lo más similares posible mientras que trata de mantener cada grupo lo más diferente posible. Esto lo hace asignando puntos de datos a cada grupo de tal forma que la suma de la distancia al cuadrado entre los puntos de datos de un grupo y el centroide (media aritmética de todos los puntos de datos que pertenecen a ese grupo) del grupo es la mínima. Si existe una menor variación dentro de los grupos, significa que los puntos de datos de cada grupo son más homogéneos (similares) [18].

El algoritmo k-means sigue los siguientes pasos para la agrupación:

1. Se especifica el número k de grupos.
2. Inicializa los centroides seleccionando aleatoriamente (sin reemplazo) k puntos de datos del conjunto de datos, para ser los centroides.
3. Se mantiene iterando hasta que no haya cambios en los centroides, es decir hasta que no haya cambios en la asignación de los puntos de datos a los grupos.
    - Calcula la suma de la distancia al cuadrado entre los puntos de datos de todos los centroides.
    - Asigna cada punto de datos al grupo más cercano (centroide).
    - Calcula los centroides para cada grupo tomando el promedio de todos los puntos de datos que pertenecen al grupo.

2.5.3. Agrupacion espectral

En el agrupamiento espectral hacen uso de los valores propios de la matriz de similitud de los datos para realizar una reducción de dimensiones antes de la agrupación. Posteriormente un algoritmo clásico de agrupación es aplicado para determinar los grupos [19]. La reducción de dimensiones la hace a través de los conocidos valores propios (*eigenvalues*) y vectores propios (*eigenvectors*): hace el cálculo de los *eigenvectors* y mapea cada punto de una dimensionalidad menor basándose en uno o más *eigenvectors*.

2.5.4. Validación de la agrupación

La validación de la agrupación es usada para determinar qué tan bueno es el resultado del algoritmo de agrupación. Este paso es importante, puesto que sirve para evitar patrones en datos aleatorios, así como para la comparación de dos algoritmos de agrupación. Los métodos para hacer la validación de la agrupación pueden ser divididos en 3 clases [20]: validación interna de la agrupación, validación externa de la agrupación y validación relativa de la agrupación.

Validación interna de la agrupación

Usa la información interna de los grupos para determinar qué tan buena es la estructura de los mismos, sin utilizar como referencia información externa. Para realizar la validación interna de la agrupación existen algo llamado índice de validación. Existen varios tipos de índices que, considerando diferentes aspectos de los grupos, indican que tan bien están formados los mismos. Algunos de ellos son [21, 22, 23, 1, 24]:

- **Índice Calinski-Harabaz:** Para un k número de grupos, el índice Calinski-Harabaz está dado como la razón del promedio de la dispersión entre grupos y la dispersión dentro de cada grupo. La ecuación (2.7) muestra el cálculo del índice:

$$hc(k) = \frac{SS_B}{SS_W} \times \frac{N-k}{k-1}, \tag{2.7}$$

  donde k es el número de grupos y N es el número de muestras, SS_W es la variación dentro del grupo (equivalente al total de la suma de los cuadrados de grupo), SS_B es la variación entre los grupos. El valor de SS_W se calcula con la ecuación (2.8):

$$SS_W = \sum_{i}^{k} \sum_{x \epsilon C_i} ||x - m_i||^2, \tag{2.8}$$

  donde k es el número de grupos, x el punto de datos (documento), C_i es el grupo i, m_i es el centroide del grupo i y $||x - m_i||$ es la norma $L2$ (distancia euclidiana) entre dos vectores.

  La variación total entre los grupos es calculada usando el total la suma del total de cuadrados (STT) menos SS_W. La SST es la distancia al cuadrado de todos los puntos de datos al centroide del grupo; esta medida es independiente del número de grupos. SS_B mide la variación de todos los centroides de los grupos al centroide del conjunto de datos (cuando los centroides de cada grupo se extienden y no están muy cerca uno del otro, el valor de SS_B es mayor). SS_W seguirá disminuyendo a medida que aumenta el tamaño del grupo. Por lo tanto, para el índice de Calínski-Harabasz, la mayor proporción de $\frac{SS_B}{SS_W}$ indica el tamaño de agrupamiento óptimo. Este valor es más alto cuando los grupos son densos y están bien separados.

- **Índice Davies-Bouldin:** Esta definido como la similitud promedio entre cada grupo C_i para $i = 1, ..., k$ y el más similar C_j. En el contexto de este índice, la similitud está definida como la medida R_ij que se compuesta por:

- s_i es la distancia promedio entre cada punto del grupo i y el centroide de ese grupo (también conocida como diámetro del grupo).
- d_{ij} la distancia entre los centroides de los grupos i y j.

Una elección simple para R_{ij} de tal forma que no sea negativa y sea simétrica se muestra en la ecuación (2.9):

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}, \tag{2.9}$$

así el índice Davies-Bouldin queda definido por la ecuación (2.10):

$$DB = \frac{1}{k}\sum_{i=1}^{k} max(R_{ij}), \qquad \text{donde} \quad i \neq j, \tag{2.10}$$

donde un valor bajo de DB indica que los grupos no son muy similares entre sí, lo que significa que son compactos y están bien separados.

- **Índice SD:** La idea del índice SD está basada en los conceptos de dispersión promedio de los grupos (*Scatt*) y la separación total de los grupos. La ecuación (2.11) indica cómo calcular la dispersión promedio de los grupos (*Dis*):

$$\begin{aligned} Scatt(k) &= \frac{1}{k}\sum_{i=1}^{k}\frac{||\sigma(v_i)||}{||\sigma(X)||}, \\ \sigma(v_i) &= \frac{1}{k}\sum_{k=1}^{n}(x_k^p - \overline{v_i^p})^2 \qquad \text{varianza de los centroides,} \\ \sigma(X) &= \frac{1}{n}\sum_{j=1}^{n}(x_j^p - \overline{X^p})^2 \qquad \text{varianza del conjunto de datos,} \end{aligned} \tag{2.11}$$

donde k es el número de grupos, $\sigma(v_i)$ es la varianza de los centroides y $\sigma(X)$ es la varianza del conjunto de datos. La separación entre los grupos se calcula con la ecuación (2.12):

$$Dis(k) = \frac{D_{max}}{D_{min}}\sum_{j=1}^{k}\Big(\sum_{z=1}^{k}||v_j - v_z||\Big)^{-1}, \tag{2.12}$$

donde, $D_{max} = max(||v_i - v_j||)\forall i, j \in \{1, ..., k\}$ es el máximo de las distancias entre los centros de los grupos, $D_{min} = min(||v_i - v_j||)\forall i, j \in \{1, ..., k\}$ es el mínimo de las distancias entre los centros de los grupos. Así, es índice SD queda definido por la ecuación (2.13):

$$SD(k) = \alpha * Scatt(k) + Dis(k), \tag{2.13}$$

donde, α es un factor de pesado igual a $Dis(c_{max})$ donde c_{max} es el número máximo de grupos de entrada. Un valor pequeño de SD indica una configuración mejor de grupos (grupos compactos y separados).

- **Índice S_Dbw:** De forma similar que la definición del índice SD, este índice está basado en que tan compactos son los grupos y la separación entre ellos, pero también toma en consideración la densidad de los grupos. Formalmente el S_Dbw mide la varianza entre grupos y la varianza entre los miembros

de los grupos. La varianza entre los grupos mide la dispersión promedio de los grupos, esta se describe en la ecuación (2.11). La densidad de los grupos está definida por la ecuación (2.14):

$$Dens\_bw = \frac{1}{k(k-1)} \sum_{i=1}^{k} \Big(\sum_{j=1, i \neq j}^{k} \frac{density(u_{ij})}{max\{density(v_i), density(v_j)\}} \Big), \tag{2.14}$$

donde, k es el número de grupos, u_{ij} es el punto medio del segmento de línea que está definido para los centroides v_i y v_j.La función de densidad alrededor de un punto está definida como: cuenta el número de puntos en una hiper-esfera cuyo radio es igual a la desviación estándar promedio de los grupos. La desviación estandar promedio de los grupos está definida en la ecuación (2.15):

$$stdev = \frac{1}{k} \sqrt{\sum_{i=1}^{k} ||\sigma(v_i)||}, \tag{2.15}$$

así el índice S_Dbw queda definido por la ecuación (2.16):

$$S\_Dbw = Scatt + Dens\_bw, \tag{2.16}$$

donde un valor pequeño de S_Dbw indica un mejor esquema de agrupación.

- **Índice Silhouette:** Mide que tan similar es un punto de datos a su mismo grupo (cohesión) comparado con otros grupos (separación). Asumiendo que se tienen k grupos; para cada punto de datos $i \in C_i$ (punto de datos i que pertenece al grupo C_i), se calcula la distancia promedio entre i y todos los otros puntos de datos en el mismo grupo como se muestra en la ecuación (2.17):

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j), \tag{2.17}$$

donde, $a(i)$ es la distancia promedio entre i y todos los otros puntos de datos en el mismo grupo, $d(i, j)$ es la distancia de los punto i y j en el grupo C_i (se hace la división entre $|C_i| - 1$ ya que no se incluye la distancia $d(i, i)$ en la suma). Entonces a_i es una medida determina que tan bien esta i asignado a su grupo.

Posteriormente se calcula la distancia promedio mínima del punto i a todos los demás puntos en otros grupos utilizando la ecuación (2.18):

$$b(i) = min_{i \neq j} \frac{1}{|C_j|} \sum_{j \in C_j} d(i, j), \tag{2.18}$$

donde, b(i) la distancia promedio mínima del punto i a todos los demás puntos en otros grupos donde i no es un miembro. El grupo con la menor diferencia promedio es el "grupo vecino"de i ya que es el siguiente grupo adecuado para el punto i.

Para calcular el valor silhouette de un punto i se considera la ecuación (2.19):

$$\begin{aligned} s(i) &= \frac{b(i) - a(i)}{max\{a(i), b(i)\}}, \quad \text{si} \quad |C_i| > 1, \\ s(i) &= 0, \qquad\qquad\qquad\quad \text{si} \quad |C_i| = 1, \end{aligned} \tag{2.19}$$

Este valor también puede ser escrito como en la ecuación (2.20), en donde podemos ver que el valor de $s(i)$ comprende los rangos $-1 \leq s(i) \leq 1$, donde un alto valor índica que i está en el grupo correcto y no está muy relacionado con los grupos vecinos:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & \text{si } a(i) < b(i) \\ 0, & \text{si } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1, & \text{si } a(i) > b(i), \end{cases} \tag{2.20}$$

donde se obtiene el valor silhouette del punto i; el valor total de silhouette es el promedio de los valores $s(i)$ de todos los puntos.

Validación externa de la agrupación

Consiste en comparar los resultados de un análisis de grupos con un resultado externo conocido, como son las etiquetas de las clases. Las medidas de evaluación pueden ser intrínsecas y extrínsecas. Las medidas intrínsecas están basadas en que tan cerca están todos los elementos en un grupo y que tan distantes son estos elementos de otros grupos. Las medidas extrínsecas están basadas en las comparaciones entre la salida del sistema de agrupación y un estándar dorado (gold standar), usualmente construido manualmente por una persona [25]. Usualmente las medidas extrínsecas son las más utilizadas para la evaluación de desempeño de un sistema de agrupación.

Validación relativa de la agrupación

Evalúa la estructura de los grupos haciendo una variación en el valor de diferentes parámetros de un solo algoritmo (por ejemplo, variar el número de grupos k). Generalmente esta validación es utilizada para determinar el número de grupos.

Bcubed Recall, Bcubed Precision y Bcubed F-score introducidas por [25] son un conjunto de medidas que de acuerdo a [26] cumplen con el criterio suficiente para evaluar la formación de los grupos.

2.5.5. Medidas de similitud y distancia

Una medida de similitud es una función de valor real que cuantifica la similitud entre dos objetos, que representa la inversa de la distancia entre dichos elementos. Se han realizado una gran investigación en este campo, puesto que las medidas de similitud y distancia son de gran importancia para los problemas de clasificación de patrones, agrupación y recuperación de información [27].

Presencia/ausencia de datos

En [28] Gower utilizó el concepto de presencia/ausencia de datos. El autor estableció que para dos vectores binarios x y y de tamaño n, si n es el número de atributos que describen los objetos (1 para la presencia del atributo y 0 para la ausencia) de manera que $x = (x_1, x_2, ..., x_n)$ y $y = (y_1, y_2, ..., y_n)$, se pueden definir los siguientes cuatro valores:

- a como el número de atributos cuyo valor es 1 en ambos vectores, x y y.
- b como el número de atributos cuyo valor es uno en el vector x pero cero en el vector y.
- c como el número de atributos cuyo valor es cero en el vector x pero uno en el vector y.
- d como el número de atributos cuyo valor es cero en ambos vectores, x y y.

Similitud del coseno

La similitud del coseno se basa en la medida del coseno de los ángulos entre los vectores, por lo tanto, si el ángulo es pequeño, el coseno es grande, lo que significa que la similitud entre los vectores es grande. En la ecuación (2.21) se muestra esta medida para los vectores x y y:

$$cos(x, y) = \frac{x \cdot y}{||x|| \;\; ||y||} = \frac{\sum_{i=1}^{n} x_i \; y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \;\; \sqrt{\sum_{i=1}^{n} y_i^2}}, \tag{2.21}$$

donde, $x \cdot y$ es el producto punto de los vectores, $||x||$ es la magnitud del vector x y n es la dimensión de los vectores.

Capítulo 3

Estado del arte

En este capítulo se realiza un estudio de los principales modelos propuestos relacionados con la atribución de autoría no supervisada. Está dividido es dos secciones, la primera está relacionada con aquellos trabajos basados en el mismo corpora que el utilizado en este trabajo (enfoques presentados en la edición del PAN 2017), los resultados expresados en esta sección serán con los que se comparará directamente este trabajo; la segunda sección está relacionada con trabajos realizados correspondientes a la agrupación.

3.1. Modelos propuestos para la edición de PAN 2017

El **PAN**[1], en su edición del 2017, propuso la tarea de *author clustering*. La definición formal de la tarea es: Dada una colección de documentos cortos, todos los documentos deben ser asignados a grupos, donde cada grupo corresponde a un autor diferente. Tenemos que cada documentos $d \in D$ debe ser asignado a exactamente un grupo k, siendo que k no está dado [7].

Varios enfoques fueron presentados para resolver la tarea, los tres mejores enfoques (los resultados pueden observarse en la tabla (3.1)) son presentados a continuación

Tabla 3.1: Resultados de la evaluación de la tarea *author clustering* del PAN 2017 (promedio de los resultados de todos los problemas de agrupación). Los participantes están ordenados de acuerdo al B^3 F-score.

Participante	B^3 **F-score**	B^3 **Recall**	B^3 **Precision**
Gómez-Adorno et al.	**0.573**	**0.639**	0.607
García et al.	0.565	0.518	**0.692**
Kocher y Savoy	0.552	0.517	0.677

Gómez-Adorno et al.

En [1], los autores utilizan una combinación de diferentes tipos y tamaños de n-gramas.

- N-gramas de caracteres tipados de tamaño 3 (de acuerdo a la definición de [10]).
- N-gramas de caracteres (con una variación de n entre 2 y 8).
- N-gramas de palabras (con una variación de n entre 1 y 3).

Posteriormente, hacen un filtrado de características basados en la ocurrencia de estas, después de hacer un análisis variando el número de características a conservar y la repercusión en el valor del F-Bcubed, mantienen las 20,000 características con mayor ocurrencia. Los autores utilizan el esquema de pesado *Log Entropy*.

[1] Véase https://pan.webis.de/

El análisis de agrupación jerárquica aglomerativa es el algoritmo utilizado por Gómez-Adorno et al para la agrupación. Para la unión de los grupos usan un algoritmo de enlace de pares, es decir, tomaron de dos en dos documentos para determinar la unión; donde la distancia coseno promedio de todos los documentos en los dos grupos considerados fue utilizada para determinar si se unirían o no.

Para la evaluación de los grupos y determinación del número de grupos (k) utilizan el índice Calinski-Harabaz. Considerando que un mayor valor del índice Caliński-Harabaz define grupos mejor formados realizaron pruebas para la agrupación con una variación de k desde 2 hasta $n-1$, donde n es el número de muestras. Así, seleccionaron el valor de k donde se maximiza el valor del índice Caliński-Harabaz.

García-Mondeja et al.

En [29], consideraron diferentes características, por lenguaje, para representar los textos:

- N-gramas de caracteres, con una variación de n entre 1 y 5 (inglés, holandés y griego).
- N-gramas de palabras, con una variación de n ente 1 y 5 (inglés, holandés y griego).
- N-gramas de lemas, con una variación de n ente 1 y 5 (inglés).
- N-gramas de PoS-Tagging, con una variación de n ente 1 y 5 (inglés).

A diferencia de [1], determinaron un conjunto de características diferente de acuerdo de acuerdo al lenguaje y género de los documentos, además no utilizaron la frecuencia de los términos, sino una representación binaria. En la tabla (3.2) se especifican las características en específico que utilizaron por lenguaje y género.

García-Mondeja el al. utilizaron el algoritmo β-compact para la agrupación de documentos, el cuál esta descrito en el pseudocódigo (Pseudocódigo1)

Algorithm 1 Algoritmo β-compact

Input: U - universe of documents
Output: Cluster – Several groups of documents

1: Cluster = Ø
2: G = BuildGraphMaximum_β_Similarity(U)
3: Cluster = SearchConexedComponentsIgnoringOrientation(G)

donde *Graph of Maximum β similarity* es una gráfica orientada en donde los vértices son los objetos; existe una arista entre dos vértices O_i y O_j si O_j es β-similar a O_i y O_j es más similar que el resto de los objetos. *SearchConexedComponentsIgnoringOrientation* se hizo basado en tres diferentes medidas de similitud: Dice, Jaccard y Coseno. Las variaciones del valor de β fueron de 0.05 a 0.5. Los valores finales para el sistema están especificados en la tabla (3.2)

Kocher y Savoy

En [30], los autores utilizaron en la versión final del sistema n-gramas de caracteres de longitud 6, considerando únicamente los 200 n-gramas más frecuentes (para la parte de pruebas, BoW fueron consideradas). Para la agrupación de los documentos, los autores se basaron en la medida de distancia *SPATIUM*.

Para calcular la distancia entre dos documentos, $Texto_A$ y $Texto_B$ *SPATIUM* utiliza una variante de la norma L1 llamada *Canberra*. Esta distancia sugiere que las distancias absolutas de las características individuales son normalizadas basadas en la suma de estas, como se indica en la ecuación (3.1):

$$\Delta(A,B) = \Delta_{AB} = \sum_{i=1}^{m} \frac{|P_A[f_i] - P_B[f_i]|}{P_A[f_i] + P_B[f_i]}, \tag{3.1}$$

Tabla 3.2: Parámetros de configuración finales del enfoque de García-Mondeja et al. para la tarea de agrupación del PAN 2017

Lenguaje	**Género**	**Tipo de N-grama**	**Valor de B**	**Medida de similitud**
Inglés	artículos	Lemas (n=1)	0.235	Dice
	reseñas	Caracteres (n=3)	0.210	Dice
Griego	artículos	Caracteres (n=3)	0.175	Dice
	reseñas	Caracteres (n=2)	0.455	Dice
Holandés	artículos	Caracteres (n=3)	0.230	Dice
	reseñas	Palabras (n=1)	0.270	Dice

donde m indica el número de características (palabras y símbolos de puntuación o n-gramas de caracteres), P_A y P_B representan la probabilidad estimada de ocurrencia de la característica f_i en el $Texto_A$ y el $Texto_B$ respectivamente. Para estimar esta probabilidad se utiliza la ecuación (3.2):

$$P[f_i] = \frac{ff_i}{n}, \tag{3.2}$$

donde ff_i es la frecuencia de ocurrencia de la característica f_i y n es la suma de todas las características que corresponden al texto. Esta ecuación permite una probabilidad de 0, ya que no existe una suavización sobre la misma.

Un valor pequeño de Δ_{AB} indica que dos documentos fueron escritos por el mismo autor, mientras que un valor grande de Δ_{AB} indica lo opuesto. Para determinar cuándo Δ_{AB} tiene un valor pequeño, para esto se deben considerar los siguientes indicios:

- *Indicio 1:* $\Delta(A, j) < \phi(A, .) = m(A, .) - \delta * std(A, .)$.
- *Indicio 2:* $\Delta(j, B) < \phi(., B) = m(., B) - \delta * std(., B)$.
- *Indicio 3:* $\Delta(B, j) < \phi(B, .) = m(B, .) - \delta * std(B, .)$.
- *Indicio 4:* $\Delta(j, A) < \phi(., A) = m(., A) - \delta * std(., A)$.

donde δ es una constante, $\Delta(A, j)$ es la distancia del $Texto_A$ a todos los otros textos, $m(A, .)$ y $std(A, .)$ son la media y la desviación estándar de esta distribución. $\Delta(., B)$ es la distribución de las distancias al $Texto_B$, de donde se obtienen $m(., B)$ y $std(., B)$ que son la media y la desviación estándar, respectivamente, de las distancias al $Texto_B$.

Se dice que el $Texto_A$ y el $Texto_B$ fueron escritos por el mismo autor cuando se cumplen dos de los custro indicios. Para la agrupación, los autores utilizan una técnica de unión simple, en donde se crea una lista ordena con el valor de cada par de textos basados en la certeza que se tiene (por los indicios) de una autoría compartida. Además, incluyen el número de indicios que se cumplieron h y la distancia absoluta entre los dos textos. Posteriormente, se obtiene la probabilidad de cada elemento en la lista con $\frac{h}{5}$ y $\frac{(h+1)}{5}$. El puntaje final depende de los otros pares de textos que cumplieron satisfactoriamente h indicios.

3.2. Investigación relacionada con la agrupación (*clustering*)

Existen varias consideraciones a tomar cuando se habla de agrupación. En un enfoque tradicional de aprendizaje de máquina los pasos a seguir para la agrupación son los siguientes:

1. Pre-procesamiento del texto y extracción de características.

2. Selección y/o reducción de características.

3. Agrupación.

4. Validación de la agrupación.

Existen varias investigaciones que se han realizado enfocadas en los pasos anteriores. A continuación se muestran algunas de ellas:

El estilo de un autor puede ser identificado seleccionando características léxicas, sintácticas, semánticas, basadas en caracteres, entre otras. Actualmente no existe un conjunto de características específico que funcione de manera universal en la tarea atribución de autoría. Existen varios trabajos que se han realizado con diferentes características y pasos de pre-procesamiento para la atribución de autoría.

En [31] se hizo un estudio del desempeño de diferentes características utilizadas para la tarea de atribución de autoría. Las características léxicas que utilizó son: longitud promedio de párrafos, longitud promedio de las oraciones, sílabas promedio por palabra, oración y párrafo, longitud máxima de oraciones (por palabra), longitud mínima de oraciones (por palabra), legibilidad (utilizando el índice Gunning-Fog [32]), entre otras; las características basadas en caracteres son: apóstrofes por párrafo, signos de puntuación por párrafo, comillas por párrafo, entre otras.

Un análisis sobre características léxicas (longitud de palabras, longitud de oraciones, n-gramas de palabras, ...), sintácticas (Part-of-Specch PoS, estructuras en oraciones, ...), semánticas (errores, sinónimos, dependencias semánticas), basadas en caracteres (diccionario de caracteres, n-gramas de caracteres, ...) y de aplicación específica (contenido, lenguaje específico, contenido específico) se realizó en [8]. En él, determinan que hay varios factores determinantes en la atribución de autoría, como son la longitud del texto, distribución de los documentos por autor, número de candidatos, ...

En [9] propusieron una clasificación de n-gramas de caracteres que mejora los resultados en cuanto a la tarea de atribución de autoría con respecto a los obtenidos con n-gramas de caracteres normales. En [10] proponen una modificación a esta clasificación para considerar signos de puntuación continuos (en la propuesta original se descartan). Demuestran como existe una mejora en el resultado con respecto al modelo original de [9] y además como el utilizar subconjuntos de estas categorías puede mejorar la clasificación con respecto a los resultados obtenidos usando todas las categorías.

La necesidad de la selección de características (SC) y/o reducción de características (RC) surge a menudo en problemas de aprendizaje de máquina. SC y RC puede mejorar la precisión con la que un clasificador aprende de los datos [15].

Al tener datos con dimensión alta pueden tenerse características que son irrelevantes y/o redundantes que pueden tener una influencia dañina para el resultado de la tarea de clasificación o agrupación, además de los costos computacionales que esta implica. Es por esto que la SC y RC tiene un papel muy importante en las tareas de aprendizaje.

En el aprendizaje de máquina no supervisado la SC resulta una tarea compleja, puesto que el único determinante que se tiene para la SC son las mismas características. Diversos estudios proponen ciertas medidas que ayudan a determinar la redundancia o irrelevancia de las características de forma no supervisada.

En [16] propusieron la medida TV, basándose en la varianza de las características (TV) hacen un ordenamiento de las mismas, posteriormente determinan que porcentaje de las características logra una agrupación más uniforme. En [15] la medida MAD es utilizada de forma similar. La diferencia entre la MAD y la TV es la ausencia del cuadrado en la MAD, en su lugar se utiliza el valor absoluto.

En [33] propusieron un método basado en optimización de enjambre de partículas (*Binary Particle Swarm Optimization*- BPSO) para hacer la SC. El PSO es aplicado por documento y utilizan como función *fitness*, un mayor valor de MAD en las características seleccionadas es un mejor conjunto de características de acuerdo a este enfoque.

En [34] propusieron un filtro basado en la correlación que existe entre las características para determinar el subconjunto de las mismas cuya relevancia es mayor y no son redundantes. El único parámetro que necesita ser establecido es el valor de alfa, que determina el umbral para establecer la redundancia de las características.

En [35] hicieron un estudio de diversos algoritmos de agrupación aplicados a diferentes datos. En este trabajo concluyen cuán importante son los pasos de pre-procesamiento y selección/extracción de características para obtener mejores resultados en la agrupación. Además, especifican como es necesario comprobar a través de varios algoritmos de agrupación para determinar aquel que se acople mejor a la tarea específica.

Las medidas de similitud en general son de gran importancia para la tarea de agrupación. En lingüística, Sahu et al. propuso una distancia de coseno modificada para agrupar documentos usando Mahout con Hadoop [36]; para la tarea de agrupar en el problema de la identificación de autoría Gómez-Adorno et al. [1] usó un análisis de agrupamiento jerárquico basado en un algoritmo de enlace promedio, para unir a los grupos una similitud de coseno se utilizó; García-Mondeja et al. evaluaron diferentes funciones de similitud para realizar la agrupación basada en un umbral [29]; en [30] Kocher et al. usaron la medida SPATIUM (palabra latina que significa distancia) para determinar los grupos en función de las reglas.

Capítulo 4

Método propuesto

En este trabajo se propone un método para la atribución de autoría no supervisada multi-lenguaje y multi-género. Este método está desarrollado utilizando como base el corpora de la tarea del PAN 2017 agrupación de documentos por autor *author clustering*.

La propuesta está divida en cinco secciones: pre-procesamiento y extracción de características, selección de características, reducción de características, agrupación de documentos y similitud del coseno con pesos. Esta propuesta toma como punto de partida el enfoque propuesto en [1] (mejor resultado en el estado del arte, B^3 *F-score*, para el corpus utilizado).

4.1. Preprocesamiento y extracción de características

Debido a la característica de longitud de los documentos del corpus y la característica multi-lenguaje de los mismos, los únicos pasos de pre-procesamiento fueron los siguientes:

- Eliminación de saltos de línea.
- Reducción de espacios consecutivos a un solo espacio.
- Modificación de la representación de número a su equivalente en ceros (por ejemplo: '1024' a '0000').

Para hacer la extracción de características se utilizó un enfoque similar a [1], en donde se obtuvieron tres tipos diferentes de representaciones del texto, posteriormente, fueron concatenados para así obtener un conjunto variado de características por documentos. Las características consideradas son:

- N-gramas de palabras, utilizando n con una variación de 1 a 3.
- N-gramas de caracteres, con una variación de n de 2 a 7.
- N-gramas de palabras tipados, con un valor de $n = 3$.

El procedimiento para obtener estas representaciones puede observarse en las siguientes sub-secciones; el código se puede ver en el Anexo (A). Al final de cada *token* obtenido se propuso concatenar un sufijo, especificando el tipo de n-grama y el valor de n, esto para asegurar que cada una de las *tokens* quedara representado por solo una categoría, puesto que se hizo la concatenación de todos los *tokens* obtenidos por representación.

Los *tokens* (concatenados) se procesaron mediante la librería *CountVectorizer* del paquete *scikit-learn*[1] para obtener la representación vectorial de los mismos.

[1] Véase https://scikit-learn.org/stable/

4.1.1. *N*-gramas de palabras

Para la extracción de los n-gramas de palabras se utilizaron dos enfoques diferentes, uno para $n = 1$ y otro $n \neq 1$. En ambos enfoques se utilizaron las palabras en minúsculas. Se propuso la utilización de caracteres especiales como son (... “ ” „ ’ ‘ ’ _ – – « ») como signos de puntuación extras a los contenidos en la librería *string.punctuation* (librería utilizada por [1] para identificar puntuación); estos caracteres especiales fueron obtenidos del mismo corpus de entrenamiento.

N igual a uno

Para $n = 1$ (mejor conocida como bolsa de palabras BoW, por sus siglas en inglés *Bag of Words*) se usó el método *word_tokenize* del paquete *nltk* (no se descartaron *stop words* ni puntuación) especificando la opción *language* de acuerdo a cada uno de los idiomas. Al final de cada 1-grama se agregó la cadena *_bw*, esto para asegurar que cada uno de los n-gramas extraídos tengan una única categoría. Un ejemplo de la representación en 1-gramas está en la figura (4.1):

Figura 4.1: Representación 1-grama de palabras (Bow) del problem013, document0001 (fragmento)

```
These subjects may seem ripped from the headlines, but they are not unusual for Enard...
--------------------------------------------------------------------------------------
these_bw, subjects_bw, may_bw, seem_bw, ripped_bw, from_bw, the_bw, headlines_bw
,_bw, but_bw, they_bw, are_bw, not_bw, unusual_bw, for_bw, enard_bw
```

N diferente de uno

Para $n \neq 1$ se utilizó en paquete *sent_tokenize* de *nltk* para obtener los enunciados especificando la opción *language* para cada uno de los idiomas diferentes; posteriormente se eliminaron los signos de puntuación obtenidos de la librería *string.punctuation* y caracteres especiales de puntuación (... “ ” „ ’ ‘ ’_ – – « »), luego se formaron los n-gramas manualmente. Al final de cada n-grama se concateno la cadena *_wg_#*, donde # especifica el tamaño del n-grama extraído. La figura (4.3) la representación en 2-gramas de palabras.

Figura 4.2: Representación n-grama de palabras del problem027, document0008 (fragmento)

```
Anderzijds is een veilige warme trein veel comfortabeler, en moeten de mensen bij wie de honger...
-----------------------------------------------------------------------------------------------
anderzijds_is_wg_2, is_een_wg_2, een_veilige_wg_2, veilige_warme_wg_2, warme_trein_wg_2
trein_veel_wg_2, veel_comfortabeler_wg_2, comfortabeler_en_wg_2, en_moeten_wg_2, moeten_de_wg_2
de_mensen_wg_2, mensen_bij_wg_2, bij_wie_wg_2, wie_de_wg_2, de_honger_wg_2
```

4.1.2. *N*-gramas de caracteres

Para la extracción de los n-gramas de caracteres se hizo un pre-procesamiento sencillo, haciendo la transformación de números a su equivalente en ceros, es decir 548 → 000. También se redujeron saltos de línea y espacios acumulados. Posteriormente se hizo la separación por párrafos y se extrajeron los n-gramas de caracteres. Al final de cada n-grama se concatenó la cadena _np_#, donde # especifica el tamaño del n-grama.

Figura 4.3: Representación n-grama de caracteres del problem054, document0003 (fragmento)

```
Επίσης στα θετικότατα...
-------------------------------------------------------------------------------------------
επ_np_2, πί_np_2, ίσ_np_2, ση_np_2, ης_np_2, ς _np_2,  σ_np_2, στ_np_2, τα_np_2, α _np_2
 θ_np_2, θε_np_2, ετ_np_2, τι_np_2, ικ_np_2, κό_np_2, ότ_np_2, τα_np_2, ατ_np_2, τα_np_2
```

4.1.3. *N*-gramas de caracteres tipados

Los n-gramas de caracteres tipados se extrajeron de acuerdo a su categoría, para esto se implementó una función con el pre-procesamiento específico y el método de extracción de cada una (para todas las categorías

se hizo la transformación de números a su equivalente en ceros; al igual que la reducción de saltos de línea y espacios acumulados). En la tabla (4.2) se muestra la representación del texto 4.1 en n-gramas de caracteres tipados. A continuación se describe el procedimiento para obtener esta representación:

***Affix* n-gramas**

- **prefix** (Ver *func_prefix* en el Anexo: A)

  1. Se agrega un espacio a cada lado de los signos de puntuación.
  2. Se utiliza la librería *word_tokenize* de *nltk* para obtener los *tokens* utilizados.
  3. Para cada *token* se verifica que el tamaño sea mayor que n, si es así se agrega a la representación los primeros n caracteres del *token*, en otro caso es descartado.
  4. Al final de cada *token* se concatena la cadena _pf_#, donde # es el valor de n.

- **suffix** (Ver *func_sufix* en el Anexo A): Se siguen los pasos 1 y 2 del proceso de extracción de *prefix*, posteriormente

  3. Para cada *token* se verifica que el tamaño sea mayor que n, si es así se agrega a la representación los últimos n caracteres del *token*, en otro caso es descartado.
  4. Se hace una concatenación de _sf_# al final del *token*, # es el valor de n.

- **space-prefix** (Ver *func_space_prefix* en el Anexo A):

  1. Se hace la separación del texto en párrafos.
  2. Obtención de los *tokens* de cada párrafo mediante el uso de un *split* de espacios.
  3. Se verifica que el tamaño de cada *token* sea mayor que n - 2 y no contenga signos de puntuación en los primeros dos caracteres, si no cumple es descartado, de otro modo es agregado a la representación.
  4. El valor _sp_# es concatenado al final de cada *token*.

- **space-suffix** (Ver *func_space_sufix* en el Anexo A): Los pasos 1 y 2 del proceso de extracción de *space_prefix* son seguidos, luego

  3. Se verifica que el tamaño de cada *token* sea mayor que $n-2$ y no contenga signos de puntuación en los últimos dos caracteres, si no cumple es descartado, de otro modo es agregado a la representación.
  4. El valor _ss_# es concatenado al final de cada *token*.

***Word* n-gramas**

- **whole-word** (Ver *func_whole_word* en el Anexo A): Se siguen los pasos 1 y 2 del proceso de extracción de *prefix*, posteriormente

  1. Se agrega un espacio a cada lado de los signos de puntuación.
  2. Se utiliza la librería *word_tokenize* de *nltk* para obtener los *tokens* utilizados.
  3. Se verifica que la longitud del *token* sea igual a n, si lo es se agrega a la representación, en otro caso es descartado.
  4. Se concatena la cadena _ww_# al final de cada *token*.

- **mid-word** (Ver *func_mid_word* en el Anexo A): Se siguen los pasos 1 y 2 del proceso de extracción de *whole-word*, después

  3. Se verifica que la longitud del *token* sea $n \geq 2$, si lo es se agrega a la representación, en otro caso es descartado.

4. La cadena _mw_# es concatenada al final de cada *token*.

- **multi-word** (Ver *func_multi_word* en el Anexo A):

1. Se obtienen los *tokens* aplicando un *split* de espacios.
2. Se toman de dos en dos *tokens*, si el último caracter el primer *token* (*token_1*) y el primer caracter del segundo *token* (*token_2*) no son signos de puntuación, se crea un nuevo *token* (*token_3*) que comprende el último caracter del *token_1* al final de concatena _ y luego (de nuevo al final) se concatena el primer caracter del *token_2*.
3. A cada *token_3* de la representación se le concatena al final la cadena _lw_#.

***Punctuation* n-gramas**

- **beg-punct** (Ver *func_beg_punct* en el Anexo A):

1. Se hace la separación por párrafos del texto.
2. Por cada párrafo, se recorren los caracteres desde la posición 0 hasta (longitud_del_párrafo - $n-1$). Considerando m como la posición actual, se toman los caracteres desde m hasta $m+2$; se verifica que m sea un signo de puntuación y que $m+1$ y $m+2$ no lo sean, si se cumple esta condición se agregan los caracteres en las posiciones m, $m+1$ y $m+2$ concatenados es ese orden como un *token* en la representación, de otro modo se descartan.
3. Se concatena la cadena _bp_# a la al final de cada *token* en la representación.

- **mid-punct** (Ver *func_mid_punct* en el Anexo A): Dos métodos para obtener las características son utilizados, el resultado de estos métodos se une en uno solo para obtener la representación de *mid-punct*.
  Primer método, se apega a la representación propuesta en [10]:

1. Se agrega un espacio a cada lado de los signos de puntuación.
2. Se obtienen los *tokens* aplicando un *split* de espacios.
3. Para cada *token*, si este es un signo de puntuación se agrega a la representación, si no es descartado.
4. A cada *token* en la representación se le concatena la cadena _mp_#.

  Segundo método, se apega a la representación propuesta en [9]:

1. Se utiliza el método *word_tokenize* de *nltk* para obtener los *tokens*.
2. Por cada *token* se verifica que no exista una puntuación en el último y el primer caracter y que exista un caracter en alguna otra posición en el *token*, si se cumple la condición el *token* es agregado a la representación, es descartado de otro modo.
3. Se concatena la cadena _mp_# al final de cada *token* en la representación.

- **end-punct** (Ver *func_end_punct* en el Anexo A)

1. Se hace la separación por párrafos del texto.
2. Por cada párrafo, se recorren los caracteres desde la posición 0 hasta (longitud_del_párrafo - $n-1$). Considerando m como la posición actual, se toman los caracteres desde m hasta $m+2$; se verifica que $m+2$ sea un signo de puntuación y que m y $m+1$ no lo sean, si se cumple esta condición se agregan los caracteres en las posiciones m, $m+1$ y $m+2$ concatenados es ese orden como un *token* en la representación, de otro modo se descartan.
3. A cada *token* en la representación se le concatena la cadena _ep_#.

Tabla 4.1: Fragmento de texto tomado del problem001, document0005 para visualizar el resultado de la representación de n-gramas tipados

> *Lunches apart, the centre offers a daycare system for people with dementia as well as bingo, pedicure, advice clinics, geri-exercise, even baths*

Tabla 4.2: Resultado de la representación en n-gramas tipados de la oración en la tabla (4.1)

SC	**Categoría**	***N*-gramas de caracteres tipados**
affix	*prefix*	Lun_pf_3, apa_pf_3, cen_pf_3, off_pf_3, day_pf_3, sys_pf_3, peo_pf_3, wit_pf_3, dem_pf_3, wel_pf_3, bin_pf_3, ped_pf_3, adv_pf_3, cli_pf_3, ger_pf_3, exe_pf_3, eve_pf_3, bat_pf_3
	sufix	hes_sf_3, art_sf_3, tre_sf_3, ers_sf_3, are_sf_3, tem_sf_3, ple_sf_3, ith_sf_3, tia_sf_3, ell_sf_3, ngo_sf_3, ure_sf_3, ice_sf_3, ics_sf_3, eri_sf_3, ise_sf_3, ven_sf_3, ths_sf_3
	space-prefix	_ap_sp_3, _th_sp_3, _ce_sp_3, _of_sp_3, _da_sp_3, _sy_sp_3, _fo_sp_3, _pe_sp_3, _wi_sp_3, _de_sp_3, _as_sp_3, _we_sp_3, _as_sp_3, _bi_sp_3, _pe_sp_3, _ad_sp_3, _cl_sp_3, _ge_sp_3, _ev_sp_3, _ba_sp_3
	space-sufix	es__ss_3, he__ss_3, re__ss_3, rs__ss_3, re__ss_3, em__ss_3, or__ss_3, le__ss_3, th__ss_3, ia__ss_3, as__ss_3, ll__ss_3, as__ss_3, ce__ss_3, en__ss_3
word	*whole-word*	the_ww_3, for_ww_3
	mid-word	unc_mw_3, nch_mw_3, che_mw_3, par_mw_3, ent_mw_3, ntr_mw_3, ffe_mw_3, fer_mw_3, ayc_mw_3, yca_mw_3, car_mw_3, yst_mw_3, ste_mw_3, eop_mw_3, opl_mw_3, eme_mw_3, men_mw_3, ent_mw_3, nti_mw_3, ing_mw_3, edi_mw_3, dic_mw_3, icu_mw_3, cur_mw_3, dvi_mw_3, vic_mw_3, lin_mw_3, ini_mw_3, nic_mw_3, xer_mw_3, erc_mw_3, rci_mw_3, cis_mw_3, ath_mw_3
	multi-word	s_a_lw_3, e_c_lw_3, e_o_lw_3, s_a_lw_3, a_d_lw_3, e_s_lw_3, m_f_lw_3, r_p_lw_3, e_w_lw_3, h_d_lw_3, a_a_lw_3, s_w_lw_3, l_a_lw_3, s_b_lw_3, e_c_lw_3, n_b_lw_3
punct	*beg-punct*	, t_bp_3, , p_bp_3, , a_bp_3, , g_bp_3, -ex_bp_3, , e_bp_3
	mid-punct	,_mp_3, ,_mp_3, ,_mp_3, ,_mp_3, -_mp_3, ,_mp_3, i-e_mp_3
	end-punct	rt,_ep_3, go,_ep_3, re,_ep_3, cs,_ep_3, ri-_ep_3, se,_ep_3

4.2. Selección de características

Una vez extraídas las características como se indica en la sección4.1 las características son pesadas con *Log-entropy*, posteriormente se analizan diferentes métodos de selección de características (SC) no supervisada: filtros, BPSO y correlación.

4.2.1. Filtros

Para realizar la SC basada en filtros dos medidas son consideradas TV y MAD, para esto se hace un ordenado de las características basadas en su valor (TV y MAD, respectivamente) y se considera un porcentaje de características (con mayor valor) a conservar; este porcentaje se establece a través de las pruebas en el corpus de entrenamiento, el valor queda definido por lenguaje y género de los documentos para ser aplicado en el corpus de pruebas.

4.2.2. Filtro para selección de características utilizando BPSO

La implementación del BPSO está basada en la representación propuesta en [33]. Básicamente se hace una selección de características por documento mediante BPSO, la representación final de los documentos queda dada por la tabla (4.3).

Tabla 4.3: Representación de una solución para la selección de características con BPSO

X	0	1	1	-1	-1	1	0	-1	1	0

Para el BPSO cada solución (partícula) denota un subconjunto de características, como se pude observar en 4.3, en donde un valor de cero indica que la característica no fue seleccionada, un valor de -1 indica que la característica no está en la representación original del documento y un valor de 1 indica que la característica fue seleccionada.
La función *fitness* es el MAD, como se muestra en la ecuación (4.1):

$$\begin{aligned} MAD_{(X_i)} &= \frac{1}{a}\sum_{j=1}^{t}|X_{ij} - \overline{X}_i|, \\ \overline{X}_i &= \frac{1}{a}\sum_{j=1}^{t}X_{ij}, \end{aligned} \tag{4.1}$$

donde $MAD_{(X_i)}$ representa la función *fitness* de la solución i, x_{ij} es el valor de la característica j en el documento i. El valor de x_{ij} (a diferencia del pesado *tf-idf* propuesto por [33]) será el pesado de la característica aplicando *Log-entropy*. a_i es el número de características seleccionadas en el documento i. $\overline{X}_i$ es el valor promedio del vector **i**.

El algoritmo de selección de características usando BPSO esta descrito en el algoritmo (2).

Algorithm 2 Algoritmo para la selección de características usando BPSO de acuerdo a lo propuesto en [33]

Entrada: Generar aleatoriamente las partículas iniciales
Salida: Partícula óptima

1: Inicializar el enjambre y los parámetros del algoritmo PSO, c_1, c_2, etc.
2: Evaluar todas las partículas con la función *fitness* de la ecuación (4.1)..
3: **while** criterio de terminación **do**
4: Actualizar la velocidad
5: Actualizar cada posición
6: Evaluar la función *fitness*
7: Reemplazar la peor partícula con la mejor partícula
8: Actualizar los valores de LB (mejor posición local) y GB (mejor posición global)
9: **end while**

Return Un nuevo subconjunto de características D^1

4.2.3. Filtro para selección de características basado en la correlación

La implementación de la selección de características está basada en la propuesta de [34], en donde utilizan dos enfoques: medir la relevancia de las características basándose en el *score* para mantener las características más relevantes; medir la redundancia basándose en la correlación de las características para identificar aquellas que son redundantes. El algoritmo está especificado en (Algoritmo 3).

Algorithm 3 Algoritmo para la selección de características usando un filtro basado en correlación [34]

Entrada: Conjunto original de características (F)
Datos de entrenamiento (D)
Parámetro de umbral (α)
Salida: Subconjunto óptimo de características (Opt)

for each características f_i en F **do** //*score de todas las características (pesado con Log-entropy)*
  s(f_i) ← (f_i, D)
ST ← característicasOrdenanasDescendente(s(f_i), F) //ordenamiento descendente de las características en F por *score*
f_j ← obtenerPrimerElemento(ST)
while $f_j \neq NULL$ **do**
  F^1 ← obtenerSiguientesCaracterísticas(f_j, ST)
  for each característica f_i en F^1
    sim(f_i, f_j) ← obtenerSimilitud(f_i, f_j)
  $thres_{redundant}(f_i)$ ← obtenerUmbralDeRedundancia(α)
  $thres_{remove}(f_i)$ ← obtenerUmbralDeEliminacion(α)
  ST^1 ← característicasOrdenanasAscendente(s(f_i), F^1) //ordenamiento ascendente de las características en F^1 por *score*
  cr ← calcularRelevanciaAcumulada(ST^1)
  for each característica f_i en ST^1 //identificar redundancia y eliminar
    if (sim(f_i, f_j) > $thres_{redundant}$)
      if (cr - s(f_i) > $thres_{remove}$)
        eliminar f_i de ST
  f_j ← obtenerSiguienteElemento(f_j, ST)
end
$Opt = ST$
Return Opt

La fórmula para calcular el $thres_{redudant}$ se especifica en la ecuación (4.2):

$$Thres_{redundant} = \overline{sim_{f_i}} + (1 - \frac{\alpha}{2}) * \sigma_{f_j}, \tag{4.2}$$

donde α es el coeficiente de confianza (valor entre 0 y 1 definido por el usuario), $\overline{sim_{f_i}}$ es el promedio de las similitudes entre las características f_i y f_j tal que $f_i \in ST^1$, así como σ es su desviación estándar. La característica f_i que tiene un que tiene un valor sim(f_i, f_j) mayor que $Thres_{redundant}$ es identificada como redundante con respecto a la característica f_j.

La relevancia acumulada (cr) especificada en la ecuación (4.3) es el criterio que determinará cuando una característica redundante debe ser eliminada:

$$cr = \sum_{i=1}^{|ST^1|} s(f_i), \tag{4.3}$$

así se tiene que la característica f_i será eliminada cuando $cr - s(f_i)$ es mayor que el $Thres_{remove}$, el cual está definido en la ecuación (4.4):

$$Thres_{remove} = (1 - \frac{\alpha}{2}) * cr, \tag{4.4}$$

donde la eliminación de la característica redundante tendrá un pequeño impacto en el valor de cr, por lo cual no se considera relevante.

4.3. Reducción de características

Para la reducción de características se propone el uso de PCA con diversas entradas, primero con las características reducidas de los métodos propuestos y luego con el total de las características. La función PCA de *scikit-learn* es utilizada para la reducción.

Para el PCA cada componente principal es una proyección de los datos en un *eigenvector* de la matriz de covarianza, si se tienen menos muestras que características, la matriz de covarianza solo tiene n *eigenvector* con un valor diferente de cero, entonces hay solo n *eigenvector* (componentes) que se pueden obtener. Debido a esto, al aplicar directamente el PCA se obtienen solo entre 18 y 20 dimensiones (número de documentos por problema), por lo que se propone realizar copias de cada uno de los documentos para así poder obtener una reducción con un mayor número de dimensiones.

4.4. Agrupación de documentos

Para la agrupación de documentos se propone utilizar tres algoritmos base: agrupación *K-means*, agrupación aglomerativa y agrupación espectral. Para la determinación del número de grupos k se considerarán cinco índices de validación: índice Calinski-Harabaz, índice Davies-Bouldin, índice SD, índice S_Dbw e índice Silhouette. Utilizando valores desde $k = 2$ hasta $(n - 1)/2$ se comprobará el desempeño de cada índice basado en cada algoritmo de agrupación.

4.5. Similitud del coseno con pesos

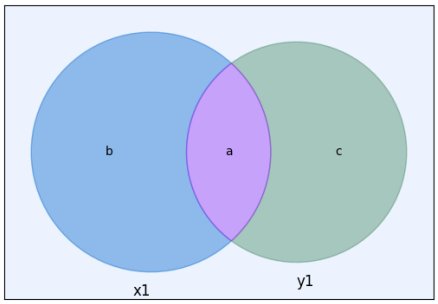
Una vez obtenido el algoritmo de agrupación a utilizar, así como el índice, se utiliza la medida propuesta en [37] para la establecer las similitudes de entrada al algoritmo. Esta medida se explica a continuación:

La idea básica detrás de la representación de presencia/ausencia de datos para vectores binarios podría aplicarse a vectores de valor real. Considerando los vectores con valores reales x y y de tamaño n, los valores de los datos de presencia/ausencia se definen como:

- a como el número de atributos cuyo valor es mayor que cero en ambos vectores, x y y.
- b como el número de atributos cuyo valor es mayor que cero en el vector x pero cero en el vector y.
- c como el número de atributos cuyo valor es cero en el vector x pero es mayor que cero en el vector y.
- d como el número de atributos cuyo valor es cero en ambos vectores, x y y.

Estos valores pueden visualizarse en la figura (4.4) en donde el cuadro representa todas las características, $x1$ representa aquellas dimensiones en donde el vector x tiene un valor diferente de cero y $y1$ representata las dimensiones en el vector y con un valor diferente de cero. Como puede observarse el vector x queda descrito por la unión de los subconjuntos a y b, mientras que el vector y queda definido por la unión de los subconjuntos a y c entonces, la intersección de $x1$ y $y1$ queda descrita por el subconjunto a.

Figura 4.4: Representación de la presencia/ausencia de datos para vectores de valor real.



Usando la redefinición de los valores de presencia/ausencia de datos, podemos reescribir los índices de la suma en la ecuación (2.21) considerando solo aquellos índices donde el valor no es cero como se muestra en la ecuación (4.5):

$$\begin{aligned} cos(x,y) &= \frac{\sum_{i \epsilon I_a} x_i \; y_i}{\sqrt{\sum_{i \epsilon I_{a \cup b}} x_i^2} \quad \sqrt{\sum_{i \epsilon I_{a \cup c}} y_i^2}} \\ &= \frac{\sum_{i \epsilon I_a} x_i \; y_i}{\sqrt{\sum_{i \epsilon I_a} x_i^2 + \sum_{i \epsilon I_b} x_i^2} \quad \sqrt{\sum_{i \epsilon I_a} y_i^2 + \sum_{i \epsilon I_c} y_i^2}}, \end{aligned} \tag{4.5}$$

donde i es el índice de x y y de tal forma que $I_a \Rightarrow x_i > 0$ y $y_i > 0$, $I_b \Rightarrow x_i > 0$ and $y_i = 0$, $I_c \Rightarrow x_i = 0, y_i > 0$.

Por lo tanto, de acuerdo con la idea de visualización utilizada por [38], proponemos una constante w tal que los valores de x_i en I_b y I_c se modificarán. Por lo tanto, las dimensiones donde los valores de los vectores son exclusivos (I_b y I_c) se reducirán por el factor w y la similitud del coseno estará más directamente relacionada con los valores no exclusivos (I_a) . La ecuación (4.6) muestra esta relación.

$$cos_w(x,y) = \frac{\sum_{i \epsilon I_a} x_i \; y_i}{\sqrt{\sum_{i \epsilon I_a} x_i^2 + w \sum_{i \epsilon I_b} x_i^2} \quad \sqrt{\sum_{i \epsilon I_a} y_i^2 + w \sum_{i \epsilon I_c} y_i^2}}, \tag{4.6}$$

donde w es una constante con valor $[0, 1]$. Como puede ser observado en la ecuación (4.6) la similitud cos_w es aún reflexiva y simétrica. La ecuación (4.6) puede ser escrita como una medida de distancia, como se muestra en la ecuación (4.7).

$$dist_w(x,y) = 1 - \frac{\sum_{i \epsilon I_a} x_i \; y_i}{\sqrt{\sum_{i \epsilon I_a} x_i^2 + w \sum_{i \epsilon I_b} x_i^2} \quad \sqrt{\sum_{i \epsilon I_a} y_i^2 + w \sum_{i \epsilon I_c} y_i^2}}, \tag{4.7}$$

Capítulo 5

Pruebas y resultados

Este capítulo describe las pruebas que se realizaron en cuanto a cada una de las etapas de la agrupación. El capítulo se divide en 6 secciones: corpus de pruebas, evaluación, pre-procesamiento y extracción de características, selección de características, reducción de características y agrupación.

5.1. Corpus de pruebas

El corpora utilizado para el desarrollo y evaluación del sistema es el del PAN 2017. Este corpora está dividido en dos corpus, uno de entrenamiento y uno de pruebas. El corpus está compuesto de dos géneros (artículos y reseñas) y tres idiomas (inglés, holandés y griego). Ambos corpus están divididos en problemas, cada problema es un conjunto de documentos (de un mismo género y un mismo idioma) y un archivo especificando el idioma y género del problema. Los textos son cortos (longitud de un párrafo), con una longitud de entre 100 y 500 caracteres. En la tabla (5.1) se especifica cada uno de los problemas que componen los corpus de entrenamiento y pruebas [26].

Tabla 5.1: Corpus de entrenamiento y pruebas proporcionados por el PAN 2017. Número de documentos (N), número de autores (k), palabras por documentos (*Palabras*)

	Lenguage	**Género**	**Problemas**	***N***	***k***	***Palabras***
Entrenamiento	Inglés	artículos	10	20	5.6	52.6
	Inglés	reseñas	10	19.4	6.1	62.2
	Holandés	artículos	10	20	5.3	51.8
	Holandés	reseñas	10	18.2	6.5	140.6
	Griego	artículos	10	20	6.0	48.2
	Griego	reseñas	10	20	6.1	39.4
Pruebas	Inglés	artículos	20	20	5.7	52.5
	Inglés	reseñas	20	20	6.4	65.3
	Holandés	artículos	20	20	5.7	49.3
	Holandés	reseñas	20	18.4	7.1	152.0
	Griego	artículos	20	19.9	5.2	46.6
	Griego	reseñas	20	20	6.0	37.1

5.2. Evaluación

Las medidas utilizadas en [26] son *Bcubed Recall* (B^3 *Recall*), *Bcubed Precision* (B^3 *Precision*) y *Bcubed F-score* (B^3 *F-score*), puesto que cumplen con varias restricciones formales importantes, como son homoge-

neidad de grupo, integridad de grupos, entre otras [25]. Estas medidas son las utilizadas para la validación para tener una base en la comparación con el estado de arte.

5.3. Pre-procesamiento y extracción de características

Como se explicó en la sección (4.1), existe una diferencia en cuanto al método utilizado para la representación del texto presentado en [1] y la propuesta aquí; básicamente la diferencia está en cómo se consideran las palabras al momento de obtener los *tokens*; mientras en [1] utilizan un *split* de espacios para considerar las palabras, nosotros propusimos utilizar la librearía *word_tokenize* de *nltk*, que hace la separación de ciertas palabras por lenguaje, como: *it's* → *it, 's*, *hadn't* → *had, n't*; en elfoque de [1] so representadas como: *it's* → *it, ', s*, *hadn't* → *had, n, ', t*. Además, para obtener los signos de puntuación se hizo una evaluación sobre el corpus para determinar todos los caracteres utilizados (incluyendo caracteres especiales); posteriormente también se utilizaron sufijos para identificar cada *token*, es decir, para cada n-grama se agregó _sf_#, donde sf especifica el tipo de n-grama extraído y # especifica el tamaño del n-grama (valor de n). Esto permitió asignar a cada *token* una sola categoría por ejemplo, considerando el *token* extraído de un texto en inglés *the*: tiene la representación en n-gramas de tamaño 1 *the_bw*; y la representación en n-gramas de caracteres *the_np_3*

Para la obtención de la categoría *mid_punct* de los n-gramas de caracteres tipados, decidimos usar los enfoques en [1] (que está basado en [10]) y el enfoque en [9]. El primer enfoque se concentra en el conteo de la ocurrencia de los signos de puntuación, mientras que el segundo se concentra en determinar las palabras que tiene un signo de puntuación en la parte media, como son: geri-exercise, geri-zumba, ... (palabras obtenidas de textos en inglés del corpus de entrenamiento).

Basado en lo anterior, se realizaron pruebas para determinar cómo funcionaba la propuesta hecha de extracción de características para la formación de los grupos, para esto se implementó el enfoque propuesto en [1], y haciendo la modificación de características se obtuvo una mejoría en el promedio total del B^3 *F-score*. En las tablas (5.2) y (5.3) se muestra una comparativa entre la salida obtenida (para el corpus de entrenamiento y el corpus de pruebas, respectivamente) con las características originales y el enfoque propuesto.

Tabla 5.2: Comparación de los resultados obtenidos con la extracción de características propuesta por Gómez-Adorno et al. y el método propuesto (corpus de entrenamiento PAN 2017).

Idioma	**Género**	**Gómez-Adorno et al.**			**Propuesta**		
		B^3 ***F-score***	B^3 ***Presicion***	B^3 ***Recall***	B^3 ***F-score***	B^3 ***Presicion***	B^3 ***Recall***
Inglés	artículos	0.5288	0.6571	0.5018	**0.5396**	0.6493	0.5303
	reseñas	**0.6304**	0.6225	0.7029	0.6090	0.6030	0.6735
Holandés	artículos	0.6011	0.7777	0.5085	**0.6238**	0.7625	0.5505
	reseñas	**0.5463**	0.5507	0.6247	0.5155	0.5061	0.6310
Griego	artículos	0.5910	0.6599	0.5888	**0.5997**	0.6167	0.6268
	reseñas	**0.5480**	0.5985	0.5426	0.5286	0.5593	0.5905
Promedio		**0.5743**	0.6444	0.5782	0.5694	0.6162	0.6004

Tabla 5.3: Comparación de los resultados obtenidos con la extracción de características propuesta por Gómez-Adorno et al. y el método propuesto (corpus de pruebas PAN 2017). Las medidas utilizadas son las *Bcubed*

Idioma	**Género**	**Gómez-Adorno et al.**			**Propuesta**		
		B^3 ***F-score***	B^3 ***Presicion***	B^3 ***Recall***	B^3 ***F-score***	B^3 ***Presicion***	B^3 ***Recall***
Inglés	artículos	**0.6179**	0.7080	0.6027	0.6106	0.6760	0.6327
	reseñas	0.5646	0.5885	0.6322	**0.5763**	0.5897	0.6543
Holandés	artículos	0.6789	0.7042	0.6894	**0.6833**	0.7026	0.7057
	reseñas	0.4741	0.3973	0.7514	**0.5481**	0.5418	0.6170
Griego	artículos	0.5436	0.6616	0.5348	**0.5568**	0.6229	0.5641
	reseñas	**0.5599**	0.5828	0.6138	0.5404	0.5048	0.6842
Promedio		0.5732	0.6071	0.6374	**0.5859**	0.6063	0.6430

Considerando los resultados (en el corpus de entrenamiento y el de pruebas) se decidió utilizar la modificación en la extracción de características para continuar con los experimentos posteriores.

5.4. Selección de Características

Utilizando la extracción de características y preprocesamiento especificado en la sección 5.3, las características fueron pesadas utilizando *Log-entropy*. Los métodos de selección de características fueron probados usando el siguiente enfoque: el algoritmo de agrupación jerárquica aglomerativa, con una función de promedio (*average*) para determinar los grupos y similitud del coseno. El índice Caliński-Harabaz se utilizó para determinar el número de grupos.

5.4.1. Filtros

Para la selección de características se hicieron diversas pruebas para determinar el porcentaje que proporcionaba una mejor agrupación, el B^3 *F-score* es la medida utilizada para determinar la mejor agrupación. Para cada medida, TV y MAD, se hicieron pruebas variando el porcentaje de diferentes maneras: para todos los problemas, por lenguaje, por género y por género y lenguaje. A continuación, se presentan los resultados:

TV

Como se menciona en la sección 2, en la subsección 2.4.2, el TV es una medida que se utiliza para evaluar las características de tal forma que las características quedan ordenadas en base a su relevancia; un umbral (o porcentaje) es establecido para determinar las características a conservar durante el filtrado.

La tabla (5.4) muestra los resultados de las variaciones de porcentaje de características a conservar utilizando como ordenamiento de características el valor de TV. Como se puede observar, la selección de características por lenguaje y género tiene un mejor B^3 *F-score* que el resto.

Basándose en los resultados de la tabla (5.4) se obtuvieron los resultados en el corpus de pruebas del PAN 2017 señalados en la tabla (5.7).

Considerando los valores obtenidos con el filtro basado en el valor TV en la tablas (5.4) y (5.7), con respecto a los valores del B^3 *F-score*:

- **corpus de entrenamiento**: se obtuvo un valor de **0.5988** que supera en 0.0294 al valor obtenido sin una SC (**0.5694**). Además supera en 0.0245 al valor del estado del arte (**0.5743**).
- **corpus de pruebas**: Se obtuvo un valor de **0.5851** que esta 0.0008 por debajo del obtenido sin una SC (**0.5859**), pero esta **0.0119** por arriba del estado del arte (**0.5732**).

Tabla 5.4: Comparación de resultados, en base al valor del B^3 *F-score*, de las variaciones del porcentaje de características conservadas al ordenar por el valor TV (corpus de entrenamiento PAN 2017). El valor obtenido de B^3 *F-score* sin utilizar una SC es de 0.5694. *TV* es el porcentaje seleccionado de características ordenadas por valor TV.

Características	***TV***	B^3 ***F-score***
Ninguna	90	**0.5760**
Lenguaje		
inglés	100	0.5743
holandés	70	0.5895
griego	90	0.5840
		0.5826
Género		
artículos	100	0.5877
reseñas	70	0.5805
		0.5841
Lenguaje y género		
inglés, artículos	30	0.5553
inglés, reseñas	90	0.6182
holandés, artículos	100	0.6238
holandés, reseñas	70	0.6009
griego, artículos	90	0.6246
griego, reseñas	60	0.5699
		0.5988

Tabla 5.5: Resultados obtenidos al aplicar los valores de porcentaje de TV por lenguaje y género de la tabla 5.4 (corpus de pruebas PAN 2017). *tv* es el porcentaje seleccionado de características ordenadas por valor TV, y *Orig.* son lo valores del enfoque original de [1].

Lenguaje, género	**Filtro TV**		**Sin Filtro**	***Orig***
	tv	B^3 ***F-score***	B^3 ***F-score***	B^3 ***F-score***
inglés, artículos	30	0.5950	0.6106	**0.6179**
inglés, reseñas	90	**0.5772**	0.5763	0.5646
holandés, artículos	100	**0.6833**	**0.6833**	0.6789
holandés, reseñas	70	**0.5608**	0.5481	0.4741
griego, artículos	90	0.5376	**0.5568**	0.5436
griego, reseñas	60	0.5565	0.5404	**0.5599**
Promedio		0.5851	**0.5859**	0.5732

MAD

Como se menciona en la sección 2, en la subsección 2.4.2, el MAD es una medida que se utiliza para evaluar las características de tal forma que, las características quedan ordenadas en base a su relevancia; un umbral (o porcentaje) es establecido para determinar las características a conservar durante el filtrado.

La tabla (5.6) muestra los resultados de las variaciones de porcentaje de características a conservar utilizando el ordenamiento de características el valor de MAD.

Considerando los valores obtenidos con el filtro basado en el valor MAD en la tablas (5.6) y (**??**), con respecto a los valores del B^3 *F-score*:

- **corpus de entrenamiento**: se obtuvo un valor de **0.5974** que supera en 0.0280 al valor obtenido sin una SC (**0.5694**). Además supera en 0.0231 al valor del estado del arte (**0.5743**).
- **corpus de pruebas**: Se obtuvo un valor de **0.5897** que esta 0.0038 por arriba del obtenido sin una SC (**0.5859**), además esta **0.0165** por arriba del estado del arte (**0.5732**).

Tabla 5.6: Comparación de resultados, en el valor de B^3 *F-score*, de la variaciones del porcentaje de características conservadas al ordenar por el valor MAD (corpus de entrenamiento PAN 2017). El valor obtenido de B^3 *F-score* sin utilizar una SC es de 0.5694. *MAD* es el porcentaje seleccionado de características ordenadas por valor MAD.

Características	***MAD***	B^3 ***F-score***
Ninguna	80	**0.5787**
Lenguaje		
inglés	100	0.5743
holandés	80	0.5972
griego	80	0.5848
		0.5854
Género		
artículos	100	0.5877
reseñas	80	0.5852
		0.5856
Lenguaje y género		
inglés, artículos	30	0.5627
inglés, reseñas	90	0.6205
holandés, artículos	100	0.6238
holandés, reseñas	50	0.6082
griego, artículos	80	0.5998
griego, reseñas	80	0.5697
		0.5974

Tabla 5.7: Resultados obtenidos al aplicar los valores de porcentaje de MAD por lenguaje y género de la tabla 5.6 (corpus de pruebas PAN 2017). *mad* es el porcentaje seleccionado de características ordenadas por valor MAD, y *Orig.* son los valores del enfoque original de [1].

Lenguaje, género	**Filtro MAD**		**Sin Filtro**	***Orig***
	mad	B^3 ***F-score***	B^3 ***F-score***	B^3 ***F-score***
inglés, artículos	30	0.5613	0.6106	**0.6179**
inglés, reseñas	90	**0.5906**	0.5763	0.5646
holandés, artículos	100	**0.6833**	**0.6833**	0.6789
holandés, reseñas	50	**0.6007**	0.5481	0.4741
griego, artículos	80	0.5453	**0.5568**	0.5436
griego, reseñas	80	0.5571	0.5404	**0.5599**
Promedio		**0.5897**	0.5859	0.5732

5.4.2. BPSO

Los valores de entrada utilizados para el BPSO para la selección de características son: $c_1 = 2$, $c_2 = 2$, $w_{max} = 0{,}9$ y $w_{min} = 0{,}5$; el tamaño de la población (número de partículas) fue establecido a 15 con un número de generaciones (iteraciones) de 500. El algoritmo se ejecutó 5 veces en cada corpus (entrenamiento

y pruebas). Los resultados obtenidos de estas ejecuciones se encuentras en el anexo (B).

Considerando los valores de las cinco pruebas del BPSO se tiene (con respecto a los valores del B^3 *F-score*):

- **corpus de entrenamiento**: promedio de **0.5327** (con una desviación estándar de 0.006), un valor de 0.0367 por debajo del obtenido sin una SC (**0.5694**) y de 0.0416 por debajo del estado del arte (**0.5743**).
- **corpus de pruebas**: promedio de **0.5734** (con una desviación estándar de 0.0054), un valor de 0.0125 por debajo del obtenido sin una SC (**0.5859**) y de 0.0002 por arriba del estado del arte (**0.5732**).

Debido a las características exploratorias del algoritmo, este no genera un óptimo global de las características seleccionadas, por lo que se hicieron experimentos considerando las características seleccionadas en todos los experimentos del BPSO. La unión e intersección de las características, tanto para el corpus de pruebas como para el de entrenamiento, fueron consideradas. Los resultados obtenidos de estas pueden observarse en el anexo (C).

Considerando los valores resultantes de la intersección y unión de características de las pruebas del BPSO, con respecto a los valores del B^3 *F-score*, se tiene:

- **corpus de entrenamiento**:
  - intersección: se obtuvo un valor de **0.5199**, 0.0495 por debajo del obtenido sin una SC (**0.5694**) y de 0.5199 por debajo del estado del arte (**0.5743**).
  - unión: se obtuvo un valor de **0.5113**, 0.0581 por debajo del obtenido sin una SC (**0.5694**) y de 0.0630 por debajo del estado del arte (**0.5743**).
- **corpus de pruebas**:
  - intersección: se obtuvo un valor de **0.5293**, 0.0566 por debajo del obtenido sin una SC (**0.5859**) y de 0.0439 por debajo del estado del arte (**0.5732**).
  - unión: se obtuvo un valor de **0.5184**, 0.0675 por debajo del obtenido sin una SC (**0.5859**) y de 0.0548 por debajo del estado del arte (**0.5732**).

En comparación con los resultados de la selección de características con filtros, las características resultantes de la selección mediante BPSO resultaron tener una agrupación con un B^3 *F-score* muy por debajo del presentado sin una selección de características tanto para el entrenamiento como para las pruebas.

5.4.3. Correlación

Utilizando el algoritmo propuesto por [34] se hicieron pruebas considerando variaciones del valor de entrada α. Las variaciones que se consideraron cuyos resultados (por lenguaje-género) son mejores están en el anexo (D); del resultado de estas variaciones de α, con respecto a los valores del B^3 *F-score*, se tiene:

- **corpus de entrenamiento**: se obtuvo un valor de **0.4941**, 0.0753 por debajo del obtenido sin una SC (**0.5694**) y de 0.0802 por debajo del estado del arte (**0.5743**).
- **corpus de pruebas**: se obtuvo un valor de **0.4820**, 0.1039 por debajo del obtenido sin una SC (**0.5859**) y de 0.0912 por debajo del estado del arte (**0.5732**).

De lo anterior se concluye que no existe un valor por categoría que mejore el resultado en cuanto al valor del B^3 *F-score* (**0.5694** para el corpus de entrenamiento y **0.5859** en el corpus de pruebas).

5.5. Reducción de Características

Utilizando la extracción de características y preprocesamiento especificado en la sección 5.3, las características fueron pesadas utilizando *Log-entropy*. El método de reducción de características fue probado usando el siguiente enfoque: el algoritmo de agrupación jerárquica aglomerativa, con una función de promedio (*average*) para determinar los grupos y similitud del coseno. El índice Caliński-Harabaz se utilizó para determinar el número de grupos.

Para realizar la reducción de características mediante PCA primero se probó con las muestras disponibles (14-20), posteriormente se hicieron copias de cada uno de los documentos de entrada. Para esto se utilizó la representación vectorial de cada documento. Por documento se tomaron las dimensiones en donde el valor es diferente de cero y se aplicó una modificación considerando un valor aleatorio entre 0.001 y 0.009, con una probabilidad uniforme de modificación de 0.45. Se obtuvieron 40 copias por cada documento. Posteriormente se realizó la reducción de dimensiones a 500 mediante PCA. Considerando que para hacer la expansión de dimensiones se usaron valores aleatorios, cada una de las pruebas de expansión y PCA fue realizada 5 veces.

Cuatro diferentes perspectivas fueron consideradas para la reducción de características: características completas (sin reducción), características seleccionadas con el filtro TV, características seleccionadas con el filtro MAD y características seleccionadas con el filtro basado en BPSO.

5.5.1. Características completas

La representación vectorial original (sin selección de características) con un pesado *Log-entropy* fue considerada para las primeras pruebas. Los resultados de las pruebas pueden verse en el anexo (E).

Considerando los valores de la reducción con PCA de las características sin selección, se tiene (con respecto a los valores del B^3 *F-score*):

- **corpus de entrenamiento**: se obtuvo un promedio de **0.5083**, 0.0611 por debajo del obtenido sin una SC (**0.5694**) y de 0.0660 por debajo del estado del arte (**0.5743**).
- **corpus de pruebas**: se obtuvo un promedio de **0.5254**, 0.0605 por debajo del obtenido sin una SC (**0.5859**) y de 0.0478 por debajo del estado del arte (**0.5732**).

Características completas con expansión de muestras

La representación vectorial original (sin selección de características) con un pesado *Log-entropy* fue considerada para hacer la expansión de muestras, obteniendo así la nueva representación vectorial de tamaño (n, m), donde n es el número de documentos ($n = n_1 * 40$, donde n_1 es el número de documentos y 40 es el número de copias por documento) y m es el número original de dimensiones (sin selección). Los resultados obtenidos al aplicar PCA sobre las muestras expandidas se pueden ver en el anexo (E).

Considerando los valores de las cinco pruebas del PCA co muestras extendidas para las características sin selección, se tiene (con respecto a los valores del B^3 *F-score*):

- **corpus de entrenamiento:** promedio de 0.5866 (con una desviación estándar de 0.001), un valor de 0.0172 por arriba del obtenido sin una SC (0.5694) y de 0.0123 por arriba del estado del arte (**0.5743**).
- **corpus de pruebas:** promedio de 0.5670 (con una desviación estándar de 0.0006), un valor de 0.0189 por debajo del obtenido sin una SC (0.5859) y de 0.0062 por debajo del estado del arte (**0.5732**).

5.5.2. Características resultantes del filtro TV

La representación vectorial resultantes del filtro TV con un pesado *Log-entropy* fue considerada para las primeras pruebas. Los resultados pueden verse en el anexo (F).
Considerando los valores de la reducción con PCA de las características sin selección, se tiene (con respecto a los valores del B^3 *F-score*):

- **corpus de entrenamiento:** promedio de 0.5583, un valor de 0.0405 por debajo del obtenido sin una reducción (0.5988) y un valor de 0.0160 por debajo del estado del arte (**0.5743**).
- **corpus de pruebas:** promedio de 0.5503, un valor de 0.0348 por debajo del obtenido sin SC (0.5851) y un valor de 0.0229 por debajo del estado del arte (**0.5732**).

Características resultantes del filtro TV con expansión de muestras

En el anexo (F) se muestra el resultado de la aplicación de PCA a las características seleccionadas con TV con la expansión de muestras.
Considerando los valores de las cinco pruebas del PCA como muestras extendidas para las características con selección TV, se tiene (con respecto a los valores del B^3 *F-score*):

- **corpus de entrenamiento:** promedio de 0.5745 (con una desviación estándar de 0.0004), un valor de 0.0243 por debajo del obtenido sin una reducción (filtro TV: 0.5988) y un valor de 0.0002 por arriba del estado del arte (**0.5743**).
- **corpus de pruebas:** promedio de 0.5739 (con una desviación estándar de 0.0003), un valor de 0.0112 por debajo del obtenido sin una reducción (filtro TV: 0.5851) y un valor de 0.0007 por arriba del estado del arte (**0.5732**).

5.5.3. Características resultantes del filtro MAD

La representación vectorial resultantes del filtro MAD con un pesado *Log-entropy* fue considerada para las primeras pruebas. En el anexo (G) se pueden observar los resultados.

Considerando los valores de la reducción con PCA de las características sin selección, se tiene (con respecto a los valores del B^3 *F-score*):

- **corpus de entrenamiento:** promedio de **0.5724**, un valor de 0.0250 por debajo del obtenido sin una reducción (filtro MAD **0.5974**) y un valor de 0.0019 por debajo del estado del arte (**0.5743**).
- **corpus de pruebas:** promedio de **0.5584**, un valor de 0.0313 por debajo del obtenido sin una reducción (filtro MAD **0.5897**) y un valor de 0.0148 por debajo del estado del arte (**0.5732**).

Características resultantes del filtro MAD con expansión de muestras

La representación vectorial resultante del filtro MAD con un pesado Log-entropy fue considerada para las pruebas. El resultado puede observarse en el anexo (G).

Considerando los valores de las cinco pruebas del PCA como muestras extendidas para las características con selección MAD, se tiene (con respecto a los valores del B^3 *F-score*):

- **corpus de entrenamiento:** promedio de **0.5796** (con una desviación estándar de 0.0001), un valor de 0.0178 por debajo del obtenido sin una reducción (filtro MAD: **0.5974**) y un valor de 0.0053 por arriba del estado del arte (**0.5743**).
- **corpus de pruebas:** promedio de **0.5795** (con una desviación estándar de 0.0001), un valor de 0.0102 por debajo del obtenido sin una reducción (filtro MAD: **0.5897**) y un valor de 0.0063 por arriba del estado del arte (**0.5732**).

5.5.4. Características resultantes de la selección mediante el algoritmo BPSO

Considerando los resultados obtenidos de la unión y la intersección de las ejecuciones del BPSO, se realizó la expansión de muestras de la misma forma que para las características completas.

Intersección

Considerando los valores de las cinco pruebas del PCA como muestras extendidas para las características resultantes de la intersección, se tiene (con respecto a los valores del B^3 *F-score*):

- **corpus de entrenamiento:** promedio de **0.4577** (con una desviación estándar de 0.0039), un valor de 0.0622 por abajo del obtenido sin una reducción (selección BPSO -intersección-: **0.5199**).
- **corpus de pruebas:** promedio de **0.4603** (con una desviación estándar de 0.0056), un valor de **0.0690** por debajo del obtenido sin una reducción (selección BPSO -intersección-: 0.5293).

Unión

Considerando los valores de las cinco pruebas del PCA como muestras extendidas para las características resultantes de la unión, se tiene (con respecto a los valores del B^3 *F-score*):

- **corpus de entrenamiento:** promedio de **0.4594** (con una desviación estándar de 0.0055), un valor de 0.0519 por debajo del obtenido sin una reducción (selección BPSO-unión-: **0.5113**).
- **corpus de pruebas:** promedio de **0.4623** (con una desviación estándar de 0.0081), un valor de 0.0561 por debajo del obtenido sin una reducción (selección BPSO-unión-: **0.5184**).

5.6. Agrupación

Considerando las características establecidas en 2.1 y utilizando el pesado *Log-entropy*, se consideraron cada uno de los métodos de selección de características establecidos en la sección 4.2 y el método de reducción de características especificado en 4.3 para determinar el algoritmo de agrupación a utilizar.

Se realizaron pruebas para tres algoritmos de agrupación: agrupación jerárquica aglomerativa, agrupación k-means y agrupación espectral; cada uno con cinco índices de validación para determinar el número k de grupos: índice Calinski-Harabaz, índice Davies-Bouldin, índice SD, índice S_Dbw e índice Silhouette. En las tablas (5.8), (5.9) y (5.10) se hace una comparación de los índices de validación para la agrupación jerárquica aglomerativa, k-means y espectral, respectivamente.

Agrupación jerárquica

Tabla 5.8: Comparación de B^3 *F-score* resultantes de los diferentes índices de validación para la agrupación jerárquica, los valores están en función de la selección de características (diferentes métodos) y reducción de características (corpus de entrenamiento PAN 2017). *SC* es el método utilizado para selección de características, *PCA* indica si se hizo una reducción de características (reducción a 500 dimensiones), *BPSO_U* indica la unión de la selección por BPSO, *BPSO_I* indica la intersección de la selección por BPSO, *Corr.* indica la selección por correlación (a un valor de α de 0.9)

Corpus	***SC***	***PCA***	**Índice de validación (valor B^3 *F-score*)**				
			Calinski-Harabaz	**Davies-Bouldin**	**SD**	**S_Dbw**	**Silhouette**
Entrenamiento	-	No	0.5694	0.5019	**0.5807**	0.5078	0.5605
	-	Si	0.5864	0.5157	**0.5899**	0.5197	0.5777
	TV	No	**0.5988**	0.5205	0.5786	0.5318	0.4987
	TV	Si	**0.5745**	0.5703	0.5720	0.5625	0.5274
	MAD	No	**0.5974**	0.5224	0.5767	0.5264	0.5083
	MAD	Si	**0.5794**	0.5725	0.5727	0.5690	0.5276
	BPSO_I	No	0.5199	0.5198	**0.5602**	0.4908	0.5297
	BPSO_U	No	0.5113	0.5144	**0.5354**	0.4713	0.5057
	Corr.	No	0.4733	0.4755	0.4665	**0.4887**	0.4667
Pruebas	-	No	**0.5859**	0.5152	0.5840	0.5175	0.5695
	-	Si	0.5660	0.5389	**0.5896**	0.5493	0.5645
	TV	No	**0.5851**	0.5188	0.5733	0.5207	0.5237
	TV	Si	0.5742	0.5757	**0.5862**	0.5702	0.5429
	MAD	No	**0.5897**	0.5343	0.5817	0.5346	0.5282
	MAD	Si	0.5797	0.5747	**0.5858**	0.5754	0.5502
	BPSO_I	No	0.5293	0.5243	**0.5548**	0.4939	0.5334
	BPSO_U	No	0.5184	0.5165	**0.5530**	0.4972	0.5165
	Corr.	No	0.4647	**0.4832**	0.4625	0.4788	0.4664

Agrupamiento K-means

Tabla 5.9: Comparación de B^3 *F-score* resultantes de los diferentes índices de validación para la agrupación k-means, los valores están en función de la selección de características (diferentes métodos) y reducción de características (corpus de entrenamiento PAN 2017). *SC* es el método utilizado para selección de características, *PCA* indica si se hizo una reducción de características (reducción a 500 dimensiones), *BPSO_U* indica la unión de la selección por BPSO, *BPSO_I* indica la intersección de la selección por BPSO, *Corr.* indica la selección por correlación (a un valor de α de 0.9).

Corpus	***SC***	***PCA***	**Índice de validación (valor B^3 *F-score*)**				
			Calinski-Harabaz	**Davies-Bouldin**	**SD**	**S_Dbw**	**Silhouette**
Entrenamiento	-	No	0.5325	0.5431	**0.5599**	0.5577	0.5557
	-	Si	0.5375	0.5535	**0.5566**	0.5555	0.5553
	TV	No	0.5574	0.5614	**0.5596**	0.5640	0.5274
	TV	Si	0.5615	0.5611	0.5647	**0.5719**	0.5369
	MAD	No	**0.5701**	0.5630	0.5667	0.5602	0.5326
	MAD	Si	**0.5700**	0.5609	0.5573	0.5576	0.5462
	BPSO_I	No	0.4738	0.5343	0.5410	**0.5458**	0.4901
	BPSO_U	No	0.4624	**0.5339**	0.5265	0.5312	0.4751
	Corr.	No	0.4517	0.4692	0.4599	**0.4722**	0.4535
Pruebas	-	No	0.5248	**0.5559**	0.5541	0.5529	0.5543
	-	Si	0.5169	0.5526	**0.5605**	0.5576	0.5480
	TV	No	0.5560	0.5803	**0.5832**	0.5681	0.5471
	TV	Si	0.5672	**0.5782**	0.5750	0.5747	0.5452
	MAD	No	0.5766	0.5732	0.5775	**0.5799**	0.5515
	MAD	Si	0.5755	0.5765	0.5814	**0.5857**	0.5521
	BPSO_I	No	0.4742	0.5331	**0.5359**	0.5335	0.4810
	BPSO_U	No	0.4715	**0.5342**	0.5329	0.5298	0.4863
	Corr.	No	0.4505	0.4618	0.4501	**0.4740**	0.4423

Agrupamiento espectral

Tabla 5.10: Comparación de B^3 *F-score* resultantes de los diferentes índices de validación para la agrupación espectral, los valores están en función de la selección de características (diferentes métodos) y reducción de características (corpus de entrenamiento PAN 2017). *SC* es el método utilizado para selección de características, *PCA* indica si se hizo una reducción de características (reducción a 500 dimensiones), *BPSO_U* indica la unión de la selección por BPSO, *BPSO_I* indica la intersección de la selección por BPSO, *Corr.* indica la selección por correlación (a un valor de α de 0.9).

Corpus	***SC***	***PCA***	**Índice de validación (valor B^3 *F-score*)**				
			Calinski-Harabaz	**Davies-Bouldin**	**SD**	**S_Dbw**	**Silhouette**
Entrenamiento	-	No	0.5317	0.5259	0.5337	0.5186	**0.5449**
	-	Si	0.5324	0.5273	0.5331	0.5284	**0.5486**
	TV	No	0.5488	0.5515	**0.5518**	0.5477	0.5162
	TV	Si	0.5508	0.5508	0.5471	**0.5567**	0.5175
	MAD	No	**0.5680**	0.5507	0.5521	0.5594	0.5396
	MAD	Si	**0.5687**	0.5524	0.5542	0.5541	0.5396
	BPSO_I	No	0.5297	0.5281	**0.5402**	0.5313	0.5200
	BPSO_U	No	0.5165	0.5271	**0.5425**	0.5330	0.5074
	Corr.	No	**0.4591**	0.4579	0.4508	0.4728	0.4538
Pruebas	-	No	0.5270	0.5364	0.5457	0.5393	**0.5594**
	-	Si	0.5269	0.5341	0.5488	0.5354	**0.5625**
	TV	No	0.5528	0.5496	**0.5529**	0.5512	0.5334
	TV	Si	**0.5549**	0.5534	0.5529	0.5513	0.5325
	MAD	No	**0.5563**	0.5549	0.5538	0.5545	0.5525
	MAD	Si	0.5565	0.5571	0.5555	**0.5576**	0.5508
	BPSO_I	No	0.5247	0.5229	**0.5430**	0.5360	0.5236
	BPSO_U	No	0.5296	0.5212	**0.5444**	0.5164	0.5215
	Corr.	No	0.4423	0.4553	0.4428	**0.4707**	0.4426

5.6.1. Comparación entre los enfoques

En la tabla (5.11) se muestra una comparativa de loa mejores enfoques con el estado del arte. Como se puede observar existe una mejora en el promedio de B^3 *F-score*

Tabla 5.11: Comparación de los mejores resutados basados en el B^3 *F-score*. *Orig.* es el enfoque original de [1]; *Modif.* es el resultado obtenido con la características extraidas en 4.1; *Enfoque_1* utiliza las características seleccionadas en 4.1, una reducción de carcterísticas mediante PCA, agrupamiento jerárquico aglomerativo utilizando SD como medida de validación de grupos; *Enfoque_2* utiliza las características seleccionadas en 4.1, hace una selección de características con el filtro basado en MAD, agrupamiento jerárquico aglomerativo utilizando Calinski-Harabaz como medida de validación de grupos

Corpus	***Características***	B^3 ***F-score***			
		Orig.	***Modif.***	***Enfoque_1***	***Enfoque_2***
Entrenamiento	inglés, artículos	0.5288	0.5396	0.5575	**0.5627**
	inglés, reseñas	0.6304	0.6090	**0.6469**	0.6205
	holandés, artículos	0.6011	**0.6238**	0.592	**0.6238**
	holandés, reseñas	0.5463	0.5155	0.5753	**0.6082**
	griego, artículos	0.5910	0.5997	**0.6017**	0.5998
	griego, reseñas	0.5480	0.5286	0.5659	**0.5697**
	Promedio	0.5743	0.5694	0.5899	**0.5974**
Pruebas	inglés, artículos	0.6179	0.6106	**0.6223**	0.5613
	inglés, reseñas	0.5646	0.5763	**0.6052**	0.5906
	holandés, artículos	0.6789	**0.6833**	0.6436	**0.6833**
	holandés, reseñas	0.4741	0.5481	0.5716	**0.6007**
	griego, artículos	0.5436	**0.5568**	0.5502	0.5453
	griego, reseñas	**0.5599**	0.5404	0.5448	0.5571
	Promedio	0.5732	0.5859	0.5896	**0.5897**

5.6.2. Distancia del coseno con pesos

Considerando los valores de la tabla 5.11, los mejores valores obtenidos son con la configuración: selección de características utilizando MAD (sin reducción de características), agrupación jerárquica aglomerativa con el índice Calinski-Harabaz como validador. Los valores obtenidos fueron **0.5974** para el corpus de pruebas y **0.5897** para el corpus de entrenamiento.

La distancia de coseno con pesos fue utilizada para hacer la agrupación, obtenido valores que maximizan el entrenamiento y posteriormente aplicándolos a las pruebas. Como se puede observar en la tabla (5.12), los valores específicos de w aplicados por categoría mejoran el resultado con respecto a los obtenidos con la distancia del coseno sin modificar:

- **corpus de entrenamiento:** Se obtuvo un valor de **0.6030**, superior en 0.0056 con respecto al obtenido con la distancia del coseno normal **0.5974** y superior en 0.0287 con respecto al reportado en el estado del arte(**0.5743**).
- **corpus de pruebas:** Se obtuvo un valor de **0.5911**, superior en 0.0014 con respecto al obtenido con la distancia del coseno normal **0.5897** y superior en 0.0179 con respecto al reportado en el estado del arte(**0.5732**).

Tabla 5.12: Resultado de utilizar la distancia del coseno con pesos x_w y y_w iguales. Se utiliza el mismo valor de w por categoría en el entrenamiento y en las pruebas.

Características	**Entrenamiento**		**Pruebas**	
	w	B^3 ***F-score***	**w**	B^3 ***F-score***
inglés, artículos	0	0.5716	0	0.5867
inglés, reseñas	1	0.6205	1	0.5906
holandés, artículos	0.1	0.6314	0.1	0.6620
holandés, reseñas	0.2	0.6093	0.2	0.5986
griego, artículos	0.2	0.6108	0.2	0.5455
griego, reseñas	0.2	0.5742	0.2	0.5630
Promedio		**0.6030**		**0.5911**

Capítulo 6

Conclusiones y trabajo futuro

Se propuso una extracción de características basada en el idioma para la extracción de los *tokens*, así como la utilización de caracteres especiales como puntuación. Se utilizó la concatenación de diversos tipos de n-gramas (de palabras, de caracteres y de caracteres tipados) a los cuales se les agrego un sufijo (especificando su tipo y tamaño) para asegurar que cada n-grama perteneciera a solo una categoría. Se utilizó un pesado *log-entropy* para las características. Para la selección de características se usó un filtro basado en la medida MAD considerando diferentes porcentajes, por idioma, de las características a conservar. El algoritmo utulizado para hacer la agrupación fue el de agrupación jerárquica aglomerativa, utilizando el enfoque de promedio (*average*) para la unión de los grupos; la distancia del coseno con pesos propuesta fue utilizada para calcular la distancia entre los documentos.

La estrategia propuesta para la tarea de atribución de autoría representó una mejora al resolver el problema de atribución de autoría no supervisada para el corpora del PAN 2017. El tamaño de los textos (con longitud de entre 100 y 500 caracteres) no fue favorable para mejorar el valor máximo obtenido (**0.6030** para el corpus de entrenamiento y **0.5911** para el corpus de pruebas); sin embargo se logró superar el mejor enfoque del PAN 2017 (**0.5743** para el corpus de entrenamiento y **0.5732** para el corpus de pruebas).

La propuesta realizada de separar los *tokens* por tipos de n-gramas, la utilización de símbolos especiales como *tokens* y la extracción de *tokens* por idioma representaron el mayor aumento en cuanto al valor del B^3 *F-score* para superar el estado del arte.

La selección de características aportó a la agrupación, ya que permitió mejorar los resultados comparados con los que se tenía de la extracción. Cabe resaltar el valor de la reducción de características, ya que se obtuvieron resultados bastante similares utilizando otro índice para la validación de los grupos.

La similitud del coseno con pesos propuesta mejoró el resultado, tanto para el corpus de entrenamiento (de **0.5974** a **0.6030**) como para el corpus de pruebas (de **0.5897** a **0.5911**).

Trabajo a futuro

Como trabajo a futuro se plantea la idea de utilizar palabras embebidas como representación de los textos, así como usar aprendizaje profundo para la selección de características y agrupación.

De igual forma se continuará la investigación en cuanto a la distancia del coseno con pesos, puesto que existen valores de w (cuando los valores de w son diferentes para x y para y) que mostraron mejor agrupación (en cuanto a B^3 *Fscore*).

Participación en congresos y publicaciones

Como parte de este trabajo de investigación se participó en la tarea de *Cross-domain Authorship Attribution* del PAN 2018 [2] en donde nuestro sistema quedó dentro de los primeros 6 lugares, superando el *baseline*. Se publicó el artículo *"CIC-GIL Approach to Cross-domain Authorship Attribution"* [37] presentado el enfoque tomado para resolver la tarea de atribución de autoría. La primera página del artículo se muestra en la figura (I.1) del anexo (I).

Se participó en el congreso internacional CLEF (*Conference and Labs of the Evaluation Forum -Information Access Evaluation meets Multilinguality, Multimodality, and Visualization-*), en donde se presentó el poster relacionado con el artículo [37]. Se colaboró para la realización del artículo *"Hierarchical Clustering Analysis: The Best-Performing Approach at PAN 2017 Author Clustering Task"* [39]. La primera página del artículo se muestra en la figura (I.2) del anexo (I).

También se participó en la conferencia internacional MICAI (*Mexican International Conference on Artificial Intelligence*) con el artículo *"Enhancement of Performance of Document Clustering in the Authorship Identification Problem with a Weighted Cosine Similarity"* [40]. La primera página del artículo se muestra en la figura (I.3) del anexo (I).

Además, se participó en la tarea *Cross-domain Authorship Attribution* del PAN 2019 [41], en donde nuestro sistema quedó en el quinto lugar, quedando solo 4.8 porciento por debajo del primer lugar.

Bibliografía

[1] H. Gómez-Adorno, Y. Aleman, D. Vilariño, M. A. Sanchez-Perez, D. Pinto, and G. Sidorov, "Author clustering using hierarchical clustering analysis," in *CLEF 2017 Working Notes*, CEUR Workshop Proceedings, 2017.

[2] M. Kestemont, M. Tschugnall, E. Stamatatos, W. Daelemans, G. Specht, B. Stein, and M. Potthast, "Overview of the Author Identification Task at PAN-2018: Cross-domain Authorship Attribution and Style Change Detection," in *Working Notes Papers of the CLEF 2018 Evaluation Labs* (L. Cappellato, N. Ferro, J.-Y. Nie, and L. Soulier, eds.), CEUR Workshop Proceedings, CLEF and CEUR-WS.org, Sept. 2018.

[3] M. Potthast, M. Hagen, F. Schremmer, and B. Stein, "Overview of the Author Obfuscation Task at PAN 2018: A New Approach to Measuring Safety," in *Working Notes Papers of the CLEF 2018 Evaluation Labs* (L. Cappellato, N. Ferro, J.-Y. Nie, and L. Soulier, eds.), CEUR Workshop Proceedings, CLEF and CEUR-WS.org, Sept. 2018.

[4] P. Rosso, F. Rangel, M. Potthast, E. Stamatatos, M. Tschuggnall, and B. Stein, "Overview of PAN'16—New Challenges for Authorship Analysis: Cross-genre Profiling, Clustering, Diarization, and Obfuscation," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 7th International Conference of the CLEF Initiative (CLEF 16)* (N. Fuhr, P. Quaresma, B. Larsen, T. Gonçalves, K. Balog, C. Macdonald, L. Cappellato, and N. Ferro, eds.), 2016.

[5] M. A. Sanchez-Perez, A. Gelbukh, and G. Sidorov, "Adaptive algorithm for plagiarism detection: The best-performing approach at PAN 2014 text alignment competition," in *Proceedings of the 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8–11, 2015*, vol. 9283, pp. 402–413, Springer, 2015.

[6] F. Rangel, P. Rosso, M. Montes-y-Gómez, M. Potthast, and B. Stein, "Overview of the 6th author profiling task at pan 2018: Multimodal gender identification in twitter," in *Working Notes Papers of the CLEF 2018 Evaluation Labs* (L. Cappellato, N. Ferro, J.-Y. Nie, and L. Soulier, eds.), CEUR Workshop Proceedings, CLEF and CEUR-WS.org, Sept. 2018.

[7] M. Tschuggnall, E. Stamatatos, B. Verhoeven, W. Daelemans, G. Specht, B. Stein, and M. Potthast, "Overview of the author identification task at PAN-2017: style breach detection and author clustering," in *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum, Dublin, Ireland, September 11-14, 2017.*, 2017.

[8] E. Stamatatos, "A survey of modern authorship attribution methods," *Journal of the American Society for Information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[9] U. Sapkota, S. Bethard, M. Montes-y-Gómez, and T. Solorio, "Not all character n-grams are created equal: A study in authorship attribution," in *Proceedings of the 2015 Annual Conference of the North American Chapter of the ACL: Human Language Technologies.*, NAACL-HLT'15, pp. 93–102, Association for Computational Linguistics, 2015.

[10] I. Markov, E. Stamatatos, and G. Sidorov, "Improving cross-topic authorship attribution: The role of pre-processing," in *Proceedings of the 18th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing 2017, Springer, 2017.

[11] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernández, "Syntactic n-grams as machine learning features for natural language processing," *Expert Systems with Applications*, vol. 41, no. 3, pp. 853 – 860, 2014.

[12] M. D. Lee, D. J. Navarro, and H. Nikkerud, "An empirical evaluation of models of text document similarity," in *Proceedings of the Cognitive Science Society*, vol. 27, 2005.

[13] B. Pincombe, "Comparison of human and latent semantic analysis (lsa) judgements of pairwise document similarities for a news corpus," tech. rep., Defence Science and Technology Organization Salisbury (Australia) Info Sciences Lab, 2004.

[14] C. Li and B. Wang, "Principal components analysis," Mar 2019. Recuperado el 2019, de http://www.ccs.neu.edu/home/vip/teach/MLcourse/5_features_dimensions/lecture_notes/PCA/PCA.pdf.

[15] A. J. Ferreira and M. A. Figueiredo, "Efficient feature selection filters for high-dimensional data," *Pattern Recognition Letters*, vol. 33, no. 13, pp. 1794 – 1804, 2012.

[16] L. Liu, J. Kang, J. Yu, and Z. Wang, "A comparative study on unsupervised feature selection methods for text clustering," in *2005 International Conference on Natural Language Processing and Knowledge Engineering*, pp. 597–601, 2005.

[17] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of high dimensional data," *In: Wille L.T. (eds) New Directions in Statistical Physics*, no. 4, pp. 273–309, 2004.

[18] I. Dabbura, "K-means clustering: Algorithm, applications, evaluation methods, and drawbacks," 2018. Recuperado el 2019, de https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.

[19] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pp. 849–856, MIT Press, 2001.

[20] A. Kassambara, "Cluster validation statistics: Must know methods," 2018. Recuperado el 2019, de https://www.datanovia.com/en/lessons/cluster-validation-statistics-must-know-methods/.

[21] C. Legány, S. Juhász, and A. Babos, "Cluster validity measurement techniques," in *Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, pp. 388–393, 2006.

[22] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Clustering validity checking methods: part II," *SIGMOD Record (ACM Special Interest Group on Management of Data)*, vol. 31, no. 3, pp. 19–27, 2002.

[23] F. Kovács, C. Legány, and A. Babos, "Cluster validity measurement techniques," *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, 02 2006.

[24] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53 – 65, 1987.

[25] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information retrieval*, vol. 12, no. 4, pp. 461–486, 2009.

[26] M. Tschuggnall, E. Stamatatos, B. Verhoeven, W. Daelemans, G. Specht, B. Stein, and M. Potthast, "Overview of the Author Identification Task at PAN 2017: Style Breach Detection and Author Clustering," in *Working Notes Papers of the CLEF 2017 Evaluation Labs* (L. Cappellato, N. Ferro, L. Goeuriot, and T. Mandl, eds.), CEUR Workshop Proceedings, 2017.

[27] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2Nd Edition)*. New York, NY, USA: Wiley-Interscience, 2000.

[28] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, vol. 27, no. 4, pp. 857–871, 1971.

[29] Y. García-Mondeja, D. Castro-Castro, V. Lavielle-Castro, and R. Munoz, "Discovering author groups using a -compact graph-based clustering," in *CLEF 2017 Working Notes*, CEUR Workshop Proceedings, 2017.

[30] M. Kocher and J. Savoy, "Unine at clef 2017: Author clustering," in *CLEF 2017 Working Notes*, CEUR Workshop Proceedings, 2017.

[31] U. Mahor and S. Das, "Performance evaluation of various feature extraction and classification techniques for authorship attribution," *International Journal of Innovation and Scientific Research*, vol. 16, pp. 252–259, 2015.

[32] R. Gunning, "The fog index after twenty years," *Journal of Business Communication*, vol. 6, no. 2, pp. 3–13, 1969.

[33] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A new feature selection method to improve the document clustering using particle swarm optimization algorithm," *Journal of Computational Science*, vol. 25, pp. 456–466, 2018.

[34] P. Pramokchon and P. Piamsa-nga, "An unsupervised, fast correlation-based filter for feature selection for data clustering," in *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, pp. 87–94, Springer Singapore, 2014.

[35] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[36] L. Sahu and B. R. Mohan, "An improved k-means algorithm using modified cosine distance measure for document clustering using mahout with hadoop," in *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, pp. 1–5, 2014.

[37] C. Martín-del-Campo-Rodríguez, H. Gómez-Adorno, G. Sidorov, and I. Z. Batyrshin, "CIC-GIL approach to cross-domain authorship attribution: Notebook for PAN at CLEF 2018," in *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018.* (L. Cappellato, N. Ferro, J. Nie, and L. Soulier, eds.), vol. 2125 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018.

[38] I. Batyrshin, N. Kubysheva, V. Solovyev, and L. Villa-Vargas, "Visualization of similarity measures for binary data and 2 x 2 tables," *Computación y Sistemas*, vol. 20, no. 3, pp. 345–353, 2016.

[39] H. Gómez-Adorno, C. Martín-Del-Campo-Rodríguez, G. Sidorov, Y. Alemán, D. Vilariño, and D. Pinto, "Hierarchical clustering analysis: The best-performing approach at PAN 2017 author clustering task," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 9th International Conference of the CLEF Association, CLEF 2018, Avignon, France, September 10-14, 2018, Proceedings* (P. Bellot, C. Trabelsi, J. Mothe, F. Murtagh, J. Nie, L. Soulier, E. SanJuan, L. Cappellato, and N. Ferro, eds.), vol. 11018 of *Lecture Notes in Computer Science*, pp. 216–223, Springer, 2017.

[40] C. Martín-del-Campo-Rodríguez, G. Sidorov, and I. Z. Batyrshin, "Enhancement of performance of document clustering in the authorship identification problem with a weighted cosine similarity," in *Advances in Computational Intelligence - 17th Mexican International Conference on Artificial Intelligence, MICAI 2018, Guadalajara, Mexico, October 22-27, 2018, Proceedings, Part II* (I. Z. Batyrshin, M. de Lourdes Martínez-Villaseñor, and H. E. P. Espinosa, eds.), vol. 11289 of *Lecture Notes in Computer Science*, pp. 49–56, Springer, 2018.

[41] M. Kestemont, E. Stamatatos, E. Manjavacas, W. Daelemans, M. Potthast, and B. Stein, "Overview of the Cross-domain Authorship Attribution Task at PAN 2019," in *CLEF 2019 Labs and Workshops, Notebook Papers* (L. Cappellato, N. Ferro, D. Losada, and H. Müller, eds.), CEUR-WS.org, Sept. 2019.

Anexo A

Código fuente para el pre-procesamiento y la extracción de características (Python).

```
#!/usr/bin/env python
"""
Author clustering system (feature extraction).
"""
__author__ = 'Carolina Martin del Campo Rodriguez'
__email__ = 'cm.del.cr@gmail.com'
__version__ = '1.0'

import re
from string import punctuation

from nltk import word_tokenize, sent_tokenize

punctuation = punctuation + "                                          "#

################################################################################
## Funciones para preprocesamiento
def pre_proc_text_es(text):
    text = re.subn(r"\d", "0",
                   re.sub(r"\t", " ", re.sub(r"(\n|\r)+", "\n", text)))[0]

    for c in punctuation:
        text = text.replace(c, ' %s '%c)
    text = re.sub(r"\s+", " ", text)

    return text

def pre_proc_text_dig(text):
    return re.subn(r"\d", "0",
                   re.sub(r"\t", " ", re.sub(r"(\n|\r)+", "\n", text)))[0]

def cleaning_text(text):
    for c in punctuation:
        text = text.replace(c, ' %s '%c)
        text = re.sub('\s{2,}', ' ', text)

    return text

################################################################################
## Bolsa de palabras
```

```
41 def bag_of_words(text, language):
42     wordlist = []
43 
44     for token in word_tokenize(text, language):
45         wordlist.append(token.lower() + "_bw")
46 
47     return wordlist
48 
49 ##########################################################################
50 ## N-gramas de palabras
51 def words_n_grams(text, sizes, language):
52     gramslist = []
53 
54     for c in punctuation:
55         text = text.replace(c, ' %s ' %c)
56     regex = re.compile('[%s]' %re.escape(punctuation))
57 
58     for size in sizes:
59         for line in sent_tokenize(text, language):
60             line = regex.sub('', line)
61 
62             for i in range(0, len(line.split()) - (size - 1)):
63                 gramslist.append('_'.join(
64                     [t.lower() for t in line.split()[i:i + size]])
65                                 + "_wg_" + str(size))
66 
67     return gramslist
68 
69 ##########################################################################
70 ## N-gramas de caracteres
71 def untyped_character_n_grams(text, sizes):
72     untyped = []
73     text = pre_proc_text_dig(text)
74 
75     for size in sizes:
76         for paragraph in text.split('\n'):
77             for i in range(0, len(paragraph) - (size - 1)):
78                 untyped.append(
79                     paragraph[i:i + size].lower() + "_np_" + str(size))
80 
81     return untyped
82 
83 ##########################################################################
84 ## N-gramas de caracteres tipados
85 def typed_character_n_grams(text, sizes, language):
86     tokens = []
87     for size_grams in sizes:
88         tokens.extend(func_prefix(size_grams, text, language))
89         tokens.extend(func_sufix(size_grams, text, language))
90         tokens.extend(func_space_prefix(size_grams, text))
91         tokens.extend(func_space_sufix(size_grams, text))
92         tokens.extend(func_whole_word(size_grams, text, language))
93         tokens.extend(func_mid_word(size_grams, text, language))
94         tokens.extend(func_multi_word(size_grams, text, language))
95         tokens.extend(func_beg_punct(size_grams, text))
96         tokens.extend(func_mid_punct(size_grams, text, language))
97         tokens.extend(func_end_punct(size_grams, text))
98 
99     return tokens
100 
101 # ------------------------------------------------------------------------
102 # affix
103 # ------------------------------------------------------------------------
104 def func_prefix(size_grams, text, language):
105     tokens = []
```

```
    text = re.sub(r"\s+", " ", re.sub(r"\n", " ", pre_proc_text_es(text)))
    text = word_tokenize(text, language)

    for i in range(len(text)):
        if len(text[i]) > size_grams:
            tokens.append(text[i][:size_grams] + "_pf_" + str(size_grams))

    return tokens

# ------------------------------------------------------------------------
def func_sufix(size_grams, text, language):
    tokens = []
    text = re.sub(r"\s+", " ", re.sub(r"\n", " ", pre_proc_text_es(text)))
    text = word_tokenize(text, language)

    for i in range(len(text)):
        if len(text[i]) > size_grams:
            tokens.append(
                text[i][-size_grams:] + "_sf_" + str(size_grams))

    return tokens

# ------------------------------------------------------------------------
def func_space_prefix(size_grams, text):
    tokens = []

    paragraphs = pre_proc_text_dig(text).split("\n")
    for paraph in paragraphs:
        values = paraph.split(" ")[1:]
        for token in values:
            if len(token) > size_grams - 2 and \
                    re.search("[" + punctuation + "]",
                              token[:(size_grams - 1)]) is None:
                tokens.append("_" + token[:(size_grams - 1)]
                              + "_sp_" + str(size_grams))

    return tokens

# ------------------------------------------------------------------------
def func_space_sufix(size_grams, text):
    tokens = []
    paragraphs = pre_proc_text_dig(text).split("\n")
    for paraph in paragraphs:
        values = paraph.split(" ")[:-1]
        for token in values:
            if len(token) > size_grams - 2 and \
                    re.search("[" + punctuation + "]",
                              token[-(size_grams - 1):]) is None:
                tokens.append(token[-(size_grams - 1):]
                              + "_" + "_ss_" + str(size_grams))

    return tokens

# ------------------------------------------------------------------------
# word
# ------------------------------------------------------------------------
def func_whole_word(size_grams, text, language):
    tokens = []
    text = re.sub(r"\s+", " ", re.sub(r"\n", " ", pre_proc_text_es(text)))
    text = word_tokenize(text, language)

    for word in text:
        if len(word) == size_grams:
            tokens.append(word + "_ww_" + str(size_grams))

```

```
171     return tokens
172
173 # ------------------------------------------------------------------------
174 def func_mid_word(size_grams, text, language):
175     tokens = []
176     text = re.sub(r"\s+", " ", re.sub(r"\n", " ", pre_proc_text_es(text)))
177     text = word_tokenize(text, language)
178     for word in text:
179         if len(word) >= size_grams + 2:
180             for j in range(1, len(word) - size_grams, 1):
181                 tokens.append(
182                     word[j: j + size_grams] + "_mw_" + str(size_grams))
183
184     return tokens
185
186 # ------------------------------------------------------------------------
187 def func_multi_word(size_grams, text, language):
188     tokens = []
189     words = re.sub(r"\s+", " ",
190                    re.sub(r"\n", " ", pre_proc_text_dig(text))).split(" ")
191
192     for i in range(len(words) - 1):
193         if re.search("(.*)[" + punctuation + "]+$", words[i]) is None \
194                 and re.search("^[" + punctuation + "]+(.*)",
195                               words[i + 1]) is None:
196             j = len(words[i]) - (size_grams - 2) \
197                 if len(words[i]) - (size_grams - 2) >= 0 else 0
198             m = 1 + ((len(words[i]) - (size_grams - 2)) * -1
199                      if len(words[i]) - (size_grams - 2) < 0 else 0)
200             while True:
201                 if j >= len(words[i]) or m > len(words[i + 1]):
202                     break
203
204                 if re.search("[" + punctuation + "]",
205                              words[i][j: len(words[i])]) is None and \
206                         re.search("[" + punctuation + "]",
207                                   words[i + 1][: m]) is None:
208                     word = words[i][j: len(words[i])] + "_" \
209                            + words[i + 1][: m]
210                     tokens.append(word + "_lw_" + str(size_grams))
211                 j += 1
212                 m += 1
213
214     return tokens
215
216 # ------------------------------------------------------------------------
217 # punct
218 # ------------------------------------------------------------------------
219 def func_beg_punct(size_grams, text):
220     tokens = []
221     paragraphs = pre_proc_text_dig(text).split("\n")
222     for paraph in paragraphs:
223         for k in range(len(paraph) - (size_grams - 1)):
224             if paraph[k] in punctuation and \
225                     re.search("[" + punctuation + "]",
226                               paraph[k + 1: k + size_grams]) is None:
227                 tokens.append(
228                     paraph[k: k + size_grams] + "_bp_" + str(size_grams))
229
230     return tokens
231
232 # ------------------------------------------------------------------------
233 def func_mid_punct(size_grams, text, language):
234     tokens = []
235     for token in cleaning_text(text).split():
```

```
236         if token in punctuation:
237             tokens.append(token + '_mp_' + str(size_grams))
238
239     text = re.subn(r"\d", "0", text)[0]
240     values = word_tokenize(text, language)
241
242     for i in range(len(values)):
243         if re.search("^[" + punctuation + "]*([^" + punctuation +
244                      "])+[" + punctuation + "]+(([^" + punctuation
245                      + "])+[" + punctuation + "]*)$",
246                      values[i]) is not None:
247             val = False
248             m = -1
249             for k in range(len(punctuation)):
250                 m = values[i].find(punctuation[k])
251                 if m > 0 and m < len(values[i]) - 1:
252                     val = True
253                     break
254             if val:
255                 k = m if size_grams - 2 > m else size_grams - 2
256                 w = size_grams - m if size_grams - 2 > m else 2
257                 while True:
258                     word = values[i][(m - k): (m + w)]
259                     w += 1
260                     k -= 1
261                     tokens.append(word + "_mp_" + str(size_grams))
262                     if k <= 0 or len(values[i][(m - k): (m + w)]) \
263                             != size_grams:
264                         break
265
266     return tokens
267
268 # ----------------------------------------------------------------------
269 def func_end_punct(size_grams, text):
270     tokens = []
271     paragraphs = pre_proc_text_dig(text).split("\n")
272     for paraph in paragraphs:
273         for k in range(len(paraph) - (size_grams - 1)):
274             if paraph[k + (size_grams - 1)] in punctuation and \
275                     re.search("[" + punctuation + "]",
276                               paraph[k: k + (size_grams - 1)]) is None:
277                 tokens.append(paraph[k: k + size_grams] + "_ep_"
278                               + str(size_grams))
279
280     return tokens
```

Anexo B

Resultados de las ejecuciones de las pruebas de la selección de características mediante BPSO

Tabla B.1: Resultados obtenidos de las salidas de las ejecuciones del BPSO para la selección de características por lenguaje y género (corpus de entrenamiento PAN 2017). *Carac* es el número promedio de características seleccionadas.

Lenguaje, género	**Prueba 1**		**Prueba 2**		**Prueba 3**		**Prueba 4**		**Prueba 5**	
	Carac	B^3 ***F-score***	***Carac***	B^3 ***F-score***	***Carac***	B^3 ***F-score***	***Carac***	B^3 ***F-score***	***Carac***	B^3 ***F-score***
inglés, artículos	2332.80	0.5405	2330.90	0.5229	2300.00	0.5065	2469.60	0.5151	2366.90	0.5413
inglés, reseñas	2785.30	0.5389	2731.20	0.5305	2825.30	0.5509	2816.00	0.5153	2636.50	0.5306
holandés, artículos	2632.60	0.5548	2635.60	0.5665	2486.90	0.5194	2487.70	0.5422	2568.20	0.5635
holandés, reseñas	5185.10	0.5030	5480.30	0.4707	5650.50	0.4771	5705.90	0.4486	5206.70	0.4755
griego, artículos	2601.20	0.5888	2512.50	0.5869	2525.50	0.6159	2452.40	0.5880	2591.20	0.5746
griego, reseñas	2225.10	0.5172	2237.50	0.5259	2101.70	0.5225	2258.60	0.5232	2215.40	0.5236
Promedio	2960.35	**0.5405**	2988.00	**0.5339**	2981.65	**0.5320**	3031.70	**0.5221**	2930.82	**0.5348**

Tabla B.2: Resultados obtenidos de las salidas de las ejecuciones del BPSO para la selección de características por lenguaje y género (corpus de pruebas PAN 2017). *Carac* es el número promedio de características seleccionadas.

Lenguaje, género	**Prueba 1**		**Prueba 2**		**Prueba 3**		**Prueba 4**		**Prueba 5**	
	Carac	B^3 ***F-score***	***Carac***	B^3 ***F-score***	***Carac***	B^3 ***F-score***	***Carac***	B^3 ***F-score***	***Carac***	B^3 ***F-score***
inglés, artículos	2365.90	0.5718	2331.90	0.6176	2358.30	0.5962	2361.00	0.5536	2355.90	0.6110
inglés, reseñas	2412.00	0.5537	2458.60	0.5745	2405.70	0.5618	2459.90	0.5766	2341.40	0.5329
holandés, artículos	2846.90	0.5609	2793.00	0.5528	2751.10	0.5555	2836.30	0.5841	2754.90	0.5538
holandés, reseñas	2816.20	0.5830	2836.10	0.6163	2772.90	0.5742	2834.30	0.5890	2882.90	0.5990
griego, artículos	2480.70	0.5720	2511.90	0.5868	2492.90	0.5854	2475.70	0.6229	2456.20	0.5952
griego, reseñas	2507.50	0.5501	2479.30	0.5416	2535.90	0.5512	2369.10	0.5256	2249.30	0.5524
Promedio	2571.53	**0.5652**	2568.47	**0.5816**	2552.80	**0.5707**	2556.05	**0.5753**	2506.77	**0.5740**

Anexo C

Resultados de utilizar la intersección/unión de las características seleccionadas en las ejecuciones del BPSO

Tabla C.1: Intersección y unión de los resultados obtenidos de las salidas de las ejecuciones del BPSO para la selección de características por lenguaje y género (corpus de entrenamiento PAN 2017). *Carac* es el número promedio de características seleccionadas.

Lenguaje, género	**Intersección**		**Unión**	
	Carac	B^3 ***F-score***	***Carac***	B^3 ***F-score***
inglés, artículos	2721.20	0.4792	1874.90	0.4683
inglés, reseñas	3012.60	0.5505	2381.20	0.5221
holandés, artículos	2863.40	0.5540	2093.50	0.5495
holandés, reseñas	6642.80	0.5359	4177.00	0.5481
griego, artículos	2852.30	0.4844	2038.00	0.5058
griego, reseñas	2608.90	0.5152	1628.80	0.4741
Promedio	3450.20	**0.5199**	2365.57	**0.5113**

Tabla C.2: Intersección y unión de los resultados obtenidos de las salidas de las ejecuciones del BPSO para la selección de características por lenguaje y género (corpus de pruebas PAN 2017). *Carac* es el número promedio de características seleccionadas.

Lenguaje, género	**Intersección**		**Unión**	
	Carac	B^3 ***F-score***	***Carac***	B^3 ***F-score***
inglés, artículos	2670.75	0.5721	1938.95	0.5370
inglés, reseñas	2984.25	0.4860	2545.65	0.5156
holandés, artículos	2736.00	0.5693	2019.95	0.5565
holandés, reseñas	7110.55	0.5298	4379.45	0.4826
griego, artículos	2787.55	0.5421	1888.10	0.5131
griego, reseñas	2514.10	0.4769	1414.65	0.5053
Promedio	3467.20	**0.5293**	2364.46	**0.5184**

Anexo D

Resultados por categoría del mejor resultado de la selección de características con el filtro-correlación

Tabla D.1: Comparación de resultados, en el valor de B^3 *F-score*, de la variaciones de α como parámetro del algorítmo basado en correlación para selección de características (corpus de entrenamiento PAN 2017).

Características	α	B^3 ***F-score***
Ninguna	0.9	**0.4733**
Lenguaje		
inglés	0.1	0.4869
holandés	0.5	0.5011
griego	0.7	0.4646
		0.4842
Género		
artículos	0.5	0.4810
reseñas	0.9	0.4865
		0.4837
Lenguaje y género		
inglés, artículos	0.1	0.4811
inglés, reseñas	0.2	0.4983
holandés, artículos	0.5	0.5451
holandés, reseñas	0.9	0.4855
griego, artículos	0.6	0.4592
griego, reseñas	0.7	0.4952
		0.4941

Tabla D.2: Comparación de resultados, en el valor de B^3 *F-score*, de la variaciones de α como parámetro del algorítmo basado en correlación para selección de características (corpus de pruebas PAN 2017).

Características	α	B^3 ***F-score***
Ninguna	0.4	**0.4692**
Lenguaje		
inglés	0.8	0.4963
holandés	0.4	0.4487
griego	0.1	0.4856
		0.4769
Género		
artículos	0.4	0.5035
reseñas	0.8	0.4508
		0.4771
Lenguaje y género		
inglés, artículos	0.8	0.5215
inglés, reseñas	0.8	0.4712
holandés, artículos	0.4	0.4854
holandés, reseñas	0.8	0.4428
griego, artículos	0.1	0.5307
griego, reseñas	0.1	0.4405
		0.4820

Anexo E

Tablas de resultados de la aplicación de PCA sobre el conjunto de muestras originales

Tabla E.1: Resultado de aplicar PCA con las características originales. *Orig.* es el número orignal de características, *Reduc.* es el número reducido de características obtenidas con el PCA

Lenguaje, género	**Corpus de entrenamiento**			**Corpus de pruebas**		
	Orig.	***Reduc.***	B^3 **F-score**	***Orig.***	***Reduc.***	B^3 **F-score**
inglés, artículos	25979.90	20.00	0.5065	25442.65	20.00	0.5854
inglés, reseñas	29748.10	19.40	0.5822	32131.70	20.00	0.5129
holandés, artículos	26626.50	20.00	0.5639	25252.70	20.00	0.5683
holandés, reseñas	47416.40	18.20	0.4269	52160.45	18.35	0.4364
griego, artículos	27171.90	20.00	0.4848	26192.70	19.90	0.5465
griego, reseñas	22624.00	20.00	0.4857	21478.90	20.00	0.5028
Promedio	29927.80	19.60	**0.5083**	30443.18	19.71	**0.5254**

Tabla E.2: Resultado de aplicar PCA sobre la expansión de muestras de las características originales (corpus de entrenamiento PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.5853	0.5982	0.5853	0.5853	0.5853
inglés, reseñas	0.6110	0.6090	0.6143	0.6090	0.6163
holandés, artículos	0.6206	0.6189	0.6189	0.6206	0.6206
holandés, reseñas	0.5406	0.5406	0.5406	0.5406	0.5406
griego, artículos	0.6354	0.6354	0.6294	0.6354	0.6354
griego, reseñas	0.5259	0.5262	0.5257	0.5211	0.5259
Promedio	**0.5865**	**0.5880**	**0.5857**	**0.5853**	**0.5873**

Tabla E.3: Resultado de aplicar PCA sobre la expansión de muestras de las características originales (corpus de pruebas PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.6554	0.6554	0.6618	0.6480	0.6554
inglés, reseñas	0.5730	0.5743	0.5719	0.5749	0.5721
holandés, artículos	0.6228	0.6187	0.6215	0.6187	0.6225
holandés, reseñas	0.4954	0.4954	0.4885	0.4954	0.4885
griego, artículos	0.5288	0.5261	0.5289	0.5288	0.5274
griego, reseñas	0.5320	0.5328	0.5320	0.5328	0.5320
Promedio	**0.5679**	**0.5671**	**0.5674**	**0.5664**	**0.5663**

Anexo F

Tablas de resultados de la aplicación de PCA sobre las características seleccionadas con TV

Tabla F.1: Resultado de aplicar PCA con las características resultantes del filtro TV. *Orig.* es el número original de características, *Reduc.* es el número reducido de características obtenidas con el PCA

Lenguaje, género	**Corpus de entrenamiento**			**Corpus de pruebas**		
	Orig.	***Reduc.***	B^3 ***F-score***	***Orig.***	***Reduc.***	B^3 ***F-score***
inglés, artículos	25979.90	20.00	0.5686	25442.65	20.00	0.5895
inglés, reseñas	29748.10	19.40	0.6276	32131.70	20.00	0.5575
holandés, artículos	26626.50	20.00	0.5639	25252.70	20.00	0.5683
holandés, reseñas	47416.40	18.20	0.5178	52160.45	18.35	0.5345
griego, artículos	27171.90	20.00	0.5345	26192.70	19.90	0.5218
griego, reseñas	22624.00	20.00	0.5375	21478.90	20.00	0.5303
Promedio	29927.80	19.60	**0.5583**	30443.18	19.71	**0.5503**

Tabla F.2: Resultado de aplicar PCA sobre la expansión de muestras de las características resultantes del filtro TV (corpus de entrenamiento PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.5637	0.5637	0.5637	0.5637	0.5637
inglés, reseñas	0.5942	0.5942	0.5891	0.5942	0.5942
holandés, artículos	0.6206	0.6206	0.6206	0.6206	0.6206
holandés, reseñas	0.5519	0.5519	0.5500	0.5519	0.5500
griego, artículos	0.5733	0.5733	0.5733	0.5733	0.5733
griego, reseñas	0.5453	0.5453	0.5453	0.5453	0.5453
Promedio	**0.5748**	**0.5748**	**0.5737**	**0.5748**	**0.5745**

Tabla F.3: Resultado de aplicar PCA sobre la expansión de muestras de las características resultantes del filtro TV (corpus de pruebas PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.5713	0.5713	0.5713	0.5713	0.5713
inglés, reseñas	0.5916	0.5927	0.5927	0.5927	0.5927
holandés, artículos	0.6187	0.6187	0.6215	0.6179	0.6215
holandés, reseñas	0.5498	0.5498	0.5495	0.5498	0.5498
griego, artículos	0.5567	0.5567	0.5567	0.5567	0.5567
griego, reseñas	0.5532	0.5532	0.5532	0.5532	0.5532
Promedio	**0.5736**	**0.5737**	**0.5742**	**0.5736**	**0.5742**

Anexo G

Tablas de resultados de la aplicación de PCA sobre las características seleccionadas con MAD

Tabla G.1: Resultado de aplicar PCA con las características resultantes del filtro MAD. *Orig.* es el número original de características, *Reduc.* es el número reducido de características obtenidas con el PCA

Lenguaje, género	**Corpus de entrenamiento**			**Corpus de pruebas**		
	Orig.	***Reduc.***	B^3 ***F-score***	***Orig.***	***Reduc.***	B^3 ***F-score***
inglés, artículos	25979.90	20.00	0.5599	25442.65	20.00	0.5611
inglés, reseñas	29748.10	19.40	0.6376	32131.70	20.00	0.5817
holandés, artículos	26626.50	20.00	0.5639	25252.70	20.00	0.5683
holandés, reseñas	47416.40	18.20	0.5784	52160.45	18.35	0.5882
griego, artículos	27171.90	20.00	0.5591	26192.70	19.90	0.5130
griego, reseñas	22624.00	20.00	0.5357	21478.90	20.00	0.5385
Promedio	29927.80	19.60	**0.5724**	30443.18	19.71	**0.5584**

Tabla G.2: Resultado de aplicar PCA sobre la expansión de muestras de las características resultantes del filtro MAD (corpus de entrenamiento PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.5530	0.5530	0.5530	0.5530	0.5530
inglés, reseñas	0.5759	0.5759	0.5759	0.5759	0.5759
holandés, artículos	0.6206	0.6189	0.6206	0.6206	0.6189
holandés, reseñas	0.6128	0.6128	0.6128	0.6128	0.6128
griego, artículos	0.5479	0.5479	0.5479	0.5479	0.5479
griego, reseñas	0.5678	0.5678	0.5678	0.5678	0.5678
Promedio	**0.5797**	**0.5794**	**0.5797**	**0.5797**	**0.5794**

Tabla G.3: Resultado de aplicar PCA sobre la expansión de muestras de las características resultantes del filtro MAD (corpus de pruebas PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.5645	0.5645	0.5645	0.5645	0.5645
inglés, reseñas	0.5769	0.5769	0.5769	0.5769	0.5769
holandés, artículos	0.6181	0.6181	0.6200	0.6200	0.6203
holandés, reseñas	0.6189	0.6189	0.6189	0.6189	0.6189
griego, artículos	0.5458	0.5458	0.5458	0.5458	0.5458
griego, reseñas	0.5519	0.5519	0.5516	0.5504	0.5519
Promedio	**0.5794**	**0.5794**	**0.5796**	**0.5794**	**0.5797**

Anexo H

Tablas de resultados de la aplicación de PCA sobre las características seleccionadas con BPSO (intersección/unión)

Tabla H.1: Resultado de aplicar PCA sobre la expansión de muestras de las características resultantes de la selección mediante BPSO -Intersección- (corpus de entrenamiento PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.4397	0.4653	0.4496	0.4496	0.4549
inglés, reseñas	0.4829	0.4547	0.4736	0.4736	0.4774
holandés, artículos	0.4400	0.4574	0.4575	0.4575	0.4326
holandés, reseñas	0.4866	0.4769	0.4519	0.4519	0.4890
griego, artículos	0.4420	0.4262	0.4528	0.4528	0.4741
griego, reseñas	0.4459	0.4567	0.4468	0.4468	0.4642
Promedio	**0.4562**	**0.4562**	**0.4554**	**0.4554**	**0.4654**

Tabla H.2: Resultado de aplicar PCA sobre la expansión de muestras de las características resultantes de la selección mediante BPSO -Intersección- (corpus de pruebas PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.4547	0.4602	0.4497	0.4544	0.4535
inglés, reseñas	0.4524	0.4784	0.4706	0.4726	0.4820
holandés, artículos	0.4562	0.4553	0.4423	0.4535	0.4403
holandés, reseñas	0.4965	0.4916	0.4750	0.4830	0.4549
griego, artículos	0.4491	0.4759	0.4345	0.4394	0.4590
griego, reseñas	0.4591	0.4596	0.4476	0.4520	0.4566
Promedio	**0.4613**	**0.4702**	**0.4533**	**0.4592**	**0.4577**

Tabla H.3: Resultado de aplicar PCA sobre la expansión de muestras de las características resultantes de la selección mediante BPSO -Unión- (corpus de entrenamiento PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.4662	0.4578	0.4577	0.4520	0.4552
inglés, reseñas	0.4657	0.4656	0.4637	0.4671	0.4659
holandés, artículos	0.4487	0.4463	0.4518	0.4404	0.4442
holandés, reseñas	0.5041	0.4821	0.4663	0.4844	0.4737
griego, artículos	0.4512	0.4443	0.4146	0.4522	0.4593
griego, reseñas	0.4647	0.4601	0.4463	0.4556	0.4738
Promedio	**0.4668**	**0.4594**	**0.4501**	**0.4586**	**0.4620**

Tabla H.4: Resultado de aplicar PCA sobre la expansión de muestras de las características resultantes de la selección mediante BPSO -Unión- (corpus de pruebas PAN 2017).

Lenguaje, género	B^3 ***F-score***				
	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
inglés, artículos	0.4583	0.4576	0.4516	0.4577	0.4731
inglés, reseñas	0.4709	0.4572	0.4807	0.4748	0.4810
holandés, artículos	0.4576	0.4521	0.4413	0.4426	0.4570
holandés, reseñas	0.4799	0.4436	0.4902	0.4792	0.4964
griego, artículos	0.4379	0.4557	0.4496	0.4406	0.4820
griego, reseñas	0.4537	0.4522	0.4597	0.4599	0.4746
Promedio	**0.4598**	**0.4531**	**0.4622**	**0.4591**	**0.4773**

Anexo I

Publicaciones

Figura I.1: Artículo *"CIC-GIL Approach to Cross-domain Authorship Attribution"*

CIC-GIL Approach to Cross-domain Authorship Attribution

Notebook for PAN at CLEF 2018

Carolina Martín-del-Campo-Rodríguez[1], Helena Gómez-Adorno[1,2], Grigori Sidorov[1], and Ildar Batyrshin[1]

[1] Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico City, Mexico
cm.del.cr@gmail.com, sidorov@cic.ipn.mx, batyr1@gmail.com
[2] Universidad Nacional Autónoma de México (UNAM), Engeneering Institute (II), Mexico City, Mexico
hgomeza@iingen.unam.mx

Abstract We present the CIC-GIL approach to the cross-domain authorship attribution task at PAN 2018. This year's evaluation lab focuses on the closed-set attribution task applied to a Fanfiction corpus in five languages: English, French, Italian, Polish, and Spanish. We followed a traditional machine learning approach and selected different feature sets depending on the language. We evaluated document features such as typed and untyped character n-grams, word n-grams, and function word n-grams. Our final system uses the log-entropy weighting scheme and SVM as classifier.

1 Introduction

The authorship attribution (AA) task consists in identifying the author of a given document among a list of candidates. There are several subtasks within the authorship attribution field such as author identification [4], author obfuscation [11] and author profiling [12]. The AA methods are used for many practical applications like electronic commerce, forensics, and humanities research [2,5]. The Authorship Attribution task is viewed as a multi-class, single-label classification problem, i.e. an automatic method has to assign a single class label (the author) to the unknown authorship documents.

Character n-grams are considered among the best feature representation for authorship attribution problems [16]. In [14], the authors introduced a categorization of character n-grams and showed that some categories have better performance than others in an AA task. Furthermore, several studies indicate that the combination of different types of n-grams introduces useful information to the classification algorithm, providing a robust model [13].

This paper describes our approach to the cross-domain authorship attribution task at PAN 2018 [4,17]. We examined different document features (typed and untyped character n-grams, word n-grams, and function word n-grams), weighting schemes (tf-idf and log-entropy), and machine learning algorithms (support vector machines, multinomial naive Bayes, and multi-layer perceptron).

Figura I.2: Artículo *"Hierarchical Clustering Analysis: The Best-Performing Approach at PAN 2017 Author Clustering Task"*

Hierarchical Clustering Analysis: The Best-Performing Approach at PAN 2017 Author Clustering Task

Helena Gómez-Adorno[1,2(✉)], Carolina Martín-del-Campo-Rodríguez[2], Grigori Sidorov[2], Yuridiana Alemán[3], Darnes Vilariño[3], and David Pinto[3]

[1] Engeneering Institute (II), Universidad Nacional Autónoma de México (UNAM), Mexico City, Mexico
hgomeza@iingen.unam.mx
[2] Instituto Politécnico Nacional (IPN), Center for Computing Research (CIC), Mexico City, Mexico
cm.del.cr@gmail.com, sidorov@cic.ipn.mx
[3] Faculty of Computer Science (FCC), Benemérita Universidad Autónoma de Puebla (BUAP), Puebla, Mexico
yuridiana.aleman@gmail.com, dvilarinoayala@gmail.com, dpinto@cs.buap.mx

Abstract. The author clustering problem consists in grouping documents written by the same author so that each group corresponds to a different author. We described our approach to the author clustering task at PAN 2017, which resulted in the best-performing system at the aforementioned task. Our method performs a hierarchical clustering analysis using document features such as typed and untyped character n-grams, word n-grams, and stylometric features. We experimented with two feature representation methods, log-entropy model, and TF-IDF, while tuning minimum frequency threshold values to reduce the feature dimensionality. We identified the optimal number of different clusters (authors) dynamically for each collection using the Caliński Harabasz score. The implementation of our system is available open source (https://github.com/helenpy/clusterPAN2017).

Keywords: Author clustering · Hierarchical clustering
Authorship-link ranking

1 Introduction

Authorship Attribution consists in identifying the author of a given document in a collection. There are several subtasks within the Authorship Attribution field such as author verification [18], author clustering [15], and plagiarism detection [16]. This paper focuses on the author clustering task, which is defined as follows: given a document collection, the task is to group documents written by the same author so that each group corresponds to a different author. Applications of this problem include automatic text processing in repositories (Web), retrieval of documents written by the same author, among others.

P. Bellot et al. (Eds.): CLEF 2018, LNCS 11018, pp. 216–223, 2018.
https://doi.org/10.1007/978-3-319-98932-7_20

Figura I.3: Artículo *"Enhancement of Performance of Document Clustering in the Authorship Identification Problem with a Weighted Cosine Similarity"*

Enhancement of Performance of Document Clustering in the Authorship Identification Problem with a Weighted Cosine Similarity

Carolina Martín-del-Campo-Rodríguez(✉), Grigori Sidorov, and Ildar Batyrshin

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Mexico City, Mexico
cm.del.cr@gmail.com,sidorov@cic.ipn.mx,batyr1@gmail.com

Abstract. Distance and similarity measures are essential to solve many pattern recognition problems such as classification, information retrieval and clustering, where the use of a specific distance could led to a better performance than others. A weighted cosine distance is proposed considering a variation in the weights of exclusive attributes of the input vectors. An agglomerative hierarchical clustering of documents was used for the comparison between the traditional cosine similarity and the one proposed in this paper. This modified measure has outcome in an improvement in the formation of clusters.

1 Introduction

A similarity measure is a real-valued function that quantifies the similarity between two objects, representing the inverse of the distance between such elements.

There are many distance/similarity measures encountered in different fields. In ecology Forbes proposed in [1] a coefficient for clustering ecological related species. In biology Jaccard applied in [2] a measure of similarity to compare the distribution of flora in different areas. In chemistry different similarity measures were applied in [3] and [4] for searching in chemical databases. Overview and general methods of construction of similarity measures are considered in [5].

In linguistics, Sahu et al. proposed a modified cosine distance to cluster documents using Mahout with Hadoop [6]; for the task of grouping in the problem of authorship identification Gómez-Adorno et al. [7] used a hierarchical clustering analysis based on an average linkage algorithm, to join the clusters a cosine similarity was used; García-Mondeja et al. evaluated different similarity functions to perform clustering based on a threshold [8]; in [9] Kocher et al. used the measure SPATIUM (Latin word that means distance) to determine the clusters based on rules.

I. Batyrshin et al. (Eds.): MICAI 2018, LNAI 11289, pp. 49–56, 2018.
https://doi.org/10.1007/978-3-030-04497-8_4