Atribución de Autoría utilizando distintos tipos de características a través de una nueva representación

Por

Adrián Pastor López Monroy

Tesis sometida como requisito parcial para obtener
el grado de

MAESTRO EN CIENCIAS EN LA ESPECIALIDAD DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica

Tonantzintla, Puebla

Supervisada por:

Dr. Manuel Montes y Gómez
Investigador del INAOE
Dr. Luis Villaseñor Pineda
Investigador del INAOE

Resumen

Hoy en día la inmensa cantidad de información disponible a través de internet se encuentra en constante crecimiento. Gran parte de ésta es texto escrito por usuarios bajo distintos contextos, por ejemplo: redes sociales, foros, bitácoras, correos electrónicos, etc. En este sentido, surge la necesidad de contar con mecanismos automáticos para facilitar el análisis de dicha información. Una de las situaciones que en recientes años ha estado ganando interés es la Atribución de Autoría (AA). De forma general, la AA consiste en lograr identificar automáticamente los documentos de uno o más autores. Por ejemplo, existe interés en el desarrollo de métodos para hacer frente a situaciones de: verificación de mensajes terroristas, filtrado de *spam*, disputas por derechos de autor, etc. Hoy en día se han propuesto diferentes algoritmos y estrategias para llevar a cabo la AA; en especial enfoques de aprendizaje automático. Con este enfoque se pretende construir clasificadores utilizando un conjunto de documentos de entrenamiento. Desafortunadamente, no siempre se tiene disponible un conjunto de documentos ideal, es decir existen escenarios donde los datos son escasos o desbalanceados. Considerando las situaciones anteriores, los atributos textuales que mejor representen el estilo de cada autor así como la representación de los documentos, juegan un papel fundamental para el buen desempeño de los algoritmos de aprendizaje. En esta tesis se propone un método alternativo para AA que aproveche el uso de

distintos tipos de atributos, por medio de una nueva representación. Se sigue la idea de que distintos tipos de atributos (e.g., n-gramas de caracteres, signos de puntuación) proporcionan distintas perspectivas del estilo de los documentos y por consiguiente de los autores. En particular, proponemos: i) utilizar conjuntos de atributos que puedan retener el estilo de los autores, ii) caracterizarlos con una representación que considere las relaciones entre documentos y autores, y iii) proponer alternativas para la integración de la representación de distintos tipos de atributos en un modelo de clasificación. La evaluación se realiza sobre el corpus c50, el cual ha sido utilizado en distintos trabajos de AA. Durante la evaluación utilizamos la exactitud para medir la clasificación, considerando escenarios con pocos datos de entrenamiento y desbalanceados. Los resultados experimentales demostraron que la representación y el método propuesto en esta tesis son una buena alternativa para AA, incluso en los escenarios difíciles.

Palabras clave: Atribución de Autoría, Clasificación no-temática, Estilometría, Aprendizaje Automático, Ensambles

Abstract

Nowadays, the huge amount of information available in the Web is constantly growing. Much of this information is in plain text written by users under different contexts, for example: social networks, forums, blogs, emails, etc. In this regard, it is important to have automated tools in order to assist the analysis of such information. One situation that has gained interest in recent years is the Authorship Attribution (AA) task. In general the main goal of AA is to identify automatically documents belonging to one or more authors. For example, building methods to deal with situations such as: terrorist message verification, spam filtering, copyright disputes, etc. Currently, different algorithms and strategies for addressing AA have been proposed; especially machine learning approaches. The idea of this approach is to build classifiers using a set of training documents. Unfortunately, the available document set is not always ideal, the latter is because there are scenarios where the instances are few, imbalanced, or both. Considering the above situations, textual features that best represent the style of each author and documents representation, play a key role in the performance of machine learning algorithms. This thesis proposes an alternative method for AA that takes advantage of using different types of attributes, through a new representation. It follows the idea that different types of attributes (e.g., character n-grams, punctuation marks) provide different perspectives of the style of documents and therefore of

authors. In particular, we propose: i) using sets of attributes that can retain the style of the authors, ii) characterizing textual features with a representation that considers the relationships between documents and authors, and iii) proposing alternatives to integrate representations of different types of attributes in a classification model. The evaluation is performed on the c50 corpus, which has been used in different AA works. In our experiments we measure the classification accuracy, considering scenarios with few training data and imbalanced clases for a set of authors. The experimental results showed that the proposed method and our representation is a good alternative to AA, even in settings where the training data is limited or imbalanced.

Keywords: Authorship Attribution, Non-thematic Classification, Stylistic Features, Machine Learning, Ensembles

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el apoyo otorgado a través de la beca no. 243957. Así como al INAOE por todas las facilidades prestadas durante mi estancia académica.

A mis asesores, Dr. Manuel Montes y Gómez y Dr. Luis Villaseñor Pineda quienes con su conocimiento, experiencia y buen carácter me acompañaron a lo largo de mis estudios.

A mis sinodales, Dr. Jesús Ariel Carrasco Ochoa, Dr. Aurelio López López y Dr. Saúl Eduardo Pomares Hernández, por sus observaciones y comentarios. En especial, expreso mi gratitud al Dr. Ariel por su apoyo, consejos y el seguimiento que me brindó.

A mis compañeros de la maestría, en especial a Aarón, Adrián y Ale, gracias por su amistad, por las discusiones sin sentido y por todos los momentos de alegría.

A mi familia, por su apoyo constante e incondicional, en especial a mi papá, por siempre creer en mí apoyándome, animándome, escuchándome, gracias papá eres mi héroe.

Dedicatoria

Para Dios,
porque siempre está conmigo
brindándome fortaleza y entendimiento.

Para mis padres, José Pastor y Ma. Guadalupe,
por su amor y sus palabras de aliento
para alcanzar mis sueños.

Para mis hermanos, Marco y Lupita,
por su apoyo incondicional.

Para Clau,
por su cariño, apoyo, comprensión y motivación.

Para todos los que creen en mí.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Hoy en día existe una enorme cantidad de información disponible a través de internet, gran parte de ésta se encuentra en formato de texto dentro de redes sociales, correo electrónico, bitácoras, foros, diarios, códigos fuente, etc. Dado este contexto, en muchas situaciones toda esta información es poco útil si no se cuenta con herramientas apropiadas para su análisis. Por ejemplo, una de las tareas que ha despertado más interés en recientes años es la Atribución de Autoría (AA). En la AA se pretende construir un algoritmo capaz de aprender el estilo de escritura de uno o más autores, para identificar automáticamente sus futuros documentos (Stamatatos, 2009). Algunas de las problemáticas relacionadas con AA involucran: identificación de acoso sexual, atribución de mensajes terroristas, verificación de autenticidad de notas suicidas, disputas por derechos de autor, identificación de autores de código fuente malicioso, detección de *spam*, detección de plagio, y la atribución anónima o disputas de trabajos literarios para autores conocidos (Stamatatos, 2009).

Para hacer frente a las necesidades en AA, distintos enfoques han sido propuestos. Por ejemplo, algunos trabajos abordan el problema utilizando clasificación

supervisada (de Vel *et al.*, 2001; Pavelec *et al.*, 2008), sobre todo cuando existen unas cuantas decenas de autores. Otros enfoques se encaminan por el lado de la recuperación de información, en especial cuando se tienen miles de autores (Schler *et al.*, 2009) . De manera general, la mayoría de los enfoques involucran dos tareas elementales; la extracción de características y la representación de documentos. En cuanto a estas dos tareas cabe añadir que, aunque algunos procedimientos en AA son similares a los de otras tareas como la clasificación temática, existen algunas diferencias a considerar. Por ejemplo, las características textuales más importantes en AA son no-temáticas (Stamatatos, 2009), debido a que el principal objetivo es modelar el estilo de escritura de cada autor (Schler *et al.*, 2009); para luego determinar la autoría de documentos incluso en el mismo contexto temático. Por otro lado, para la representación de documentos es común utilizar métodos basados en el modelo vectorial; pero variando aspectos en los atributos tales como el pesado y los valores que toman.

Algunos trabajos (Stamatatos, 2009; Solorio *et al.*, 2011) en AA han proporcionado pistas de que la combinación adecuada de algunos tipos de atributos podrían ayudar en la identificación de los documentos del autor. Por lo tanto, es común que algunos trabajos consideren dos o tres tipos de características textuales. Sin embargo, para explotar esta idea y obtener el máximo beneficio es necesario explorar más a detalle los atributos que pueden retener el estilo, formas nuevas de caracterizarlos para ayudar a las representaciones tradicionales y la forma de integrar lo anterior en un contexto de clasificación. Esto es precisamente la idea general de esta tesis.

1.1. Planteamiento del problema

De acuerdo con el estado del arte, gran parte de las investigaciones en AA se abordan utilizando:

- **Representaciones convencionales de clasificación temática (Abbasi y Chen, 2008):** La mayoría de los trabajos abordan AA utilizando algún tipo de Bolsa de Términos (BoT, *Bag of Terms*[1], por sus siglas en inglés), en donde se extrae un conjunto de características textuales y luego se llevan al modelo vectorial. No obstante, en algunas situaciones este enfoque no es suficiente, ya que cuenta con el problema de la alta dimensionalidad y alta dispersión en la representación. Estos inconvenientes afectan la calidad de la representación y dificultan la tarea de los algoritmos de *aprendizaje automático*. Otro de los inconvenientes de BoT es que, al obtener la representación vectorial se pierde cualquier información de orden o relación que exista entre los términos y clases. Estas desventajas hacen difícil la utilización de BoT en escenarios realistas y distintos dominios (foros, correo electrónico, etc.), donde surgen algunas situaciones importantes tales como: escasos datos de entrenamiento, clases altamente desbalanceadas o textos de entrenamiento cortos (Frantzeskou *et al.*, 2007). Dado este contexto, estamos interesados en una representación alternativa para AA, en dónde se mejore la representación de los documentos y se minimizen los problemas clásicos de BoT.

- **Pocos tipos de atributos:** Existen pocos trabajos que se dediquen a explorar la forma de utilizar distintos tipos de atributos para la identificación

[1] Utilizamos las siglas BoT para referirnos a una Bolsa de Palabras (BoW, *Bag of Words*, por sus siglas en inglés), el que los atributos además de palabras pueden ser: n-gramas de caracter, signos de puntuación, una etiqueta que represente la longitud de la palabra, etc.

de autores. Entre los atributos que más han sido utilizados se encuentran; las palabras y los n-gramas a nivel de caracter (Argamon y Juola, 2011). No obstante, existen otros tipos de atributos de estilo que al ser considerados conjuntamente podrían mejorar las tasas de clasificación. Por ejemplo, considere AA sobre correos electrónicos usando como atributos los errores gramaticales y las palabras vacías. En esta situación, podríamos decir que ambos atributos son de diferente tipo (e.g., de idiosincrasia y de contenido). En el presente trabajo se propone explorar estrategias con ensambles de clasificadores para explotar diferentes tipos de características textuales en AA; esto tiene la idea de que, al utilizar distintos tipos de atributos, éstos se complementen entre sí para mejorar la identificación del autor.

1.2. Objetivos

Objetivo general:

Diseñar e implementar un método de Atribución de Autoría basado en el uso conjunto de distintos tipos de atributos, caracterizándolos a través de una nueva representación.

Objetivos específicos:

- Determinar tipos de atributos que conjuntamente puedan ayudar a retener el estilo de escritura de los autores.
- Definir una representación especial para documentos en AA, con baja dimensionalidad y dispersión.
- Proponer un método para AA que aproveche la combinación de distintos conjuntos de atributos, caracterizados a través de alguna representación.

1.3. Organización de la tesis

A continuación se presenta la forma en la que se estructuró esta tesis. Dentro del Capítulo 2 se presentan los conceptos relacionados con AA y ensambles de clasificadores que son utilizados a lo largo del presente trabajo. Posteriormente, en el Capítulo 3 se realiza una revisión del trabajo relacionado con la presente tesis; entre ellos métodos en AA y tipos de representaciones empleadas. El Capítulo 4 explica detalladamente el método de representación y clasificación utilizado. En el Capítulo 5 se muestra la evaluación del método, experimentos relevantes y sus resultados. Por último, en el Capítulo 6 se exponen las conclusiones obtenidas y el trabajo futuro.

Capítulo 2

Fundamento teórico

En este capítulo se introducen los conceptos básicos que fundamentan el presente trabajo. Principalmente se explican dos temas: i) Atribución de Autoría (AA) y ii) Ensambles de clasificadores. La Sección 2.1 presenta la AA exponiendo sus tareas principales, familias de características textuales y algunos de los métodos convencionales. Por otro lado, la Sección 2.2 introduce algunos conceptos de los ensambles de clasificadores tal como los métodos convencionales, formas de combinar predicciones y la importancia de la generación de diversidad. Cabe añadir que, la relevancia de los algoritmos de ensambles de clasificadores para esta tesis, radica en que la combinación de espacios de atributos se aborda tomando en cuenta varios clasificadores en un sistema de votación.

2.1. Atribución de Autoría

En este trabajo nos enfocamos en la Atribución de Autoría basada en métodos estadísticos y computacionales. Esto consiste en estudiar características textuales que al medirlas nos permitan discriminar entre documentos escritos por distintos

autores (Stamatatos, 2009). La AA comparte procesos similares con otras tareas relacionadas con el tratamiento automático de texto (e.g., la clasificación temática). Sin embargo, existen importantes diferencias entre la AA y otros problemas de clasificación de documentos, sobre todo en el tipo de características textuales que se extraen de los documentos; por ejemplo, en AA son más relevantes los atributos de estilo de escritura que los de contenido.

2.1.1. Tareas principales

En general la AA comprende dos enfoques principales, ambos con el objetivo de conocer la autoría de un cierto documento, pero en contextos un tanto distintos. A continuación se explica cada uno de estos enfoques:

- **Identificación de Autoría (IA):** Consiste en predecir el autor de un documento, dado un conjunto de autores candidatos para los cuales se tienen disponibles textos de su autoría (Stamatatos, 2009). En este contexto, la IA se puede establecer como un problema de clasificación multiclase de una etiqueta; donde cada documento pertenece a un autor, y los autores representan las clases a discriminar. Dentro de este enfoque existen dos situaciones:
  - **Clase Cerrada:** Se puede asumir que el documento a predecir pertenece a alguno de los autores candidatos.
  - **Clase Abierta:** En esta subtarea el documento a predecir puede no pertenecer a ninguno de los autores candidatos.
- **Verificación de Autoría (VA):** En este enfoque sólo se cuenta con un autor y sus documentos, el objetivo es determinar si los documentos de prueba pertenecen o no a dicho autor (Argamon y Juola, 2011). En este

sentido, este caso se puede abordar como un problema de clasificación de una clase (Koppel y Schler, 2004).

2.1.2. Familias de características textuales

De acuerdo a recientes foros de AA (Argamon y Juola, 2011), entre los atributos más útiles se encuentran la selección de ciertas palabras y los n-gramas a nivel de caracter. Por ejemplo, tomar en cuenta la frecuencia y distribución de las palabras vacías (e.g., *el, de* y *sobre*) a través del texto, podría contribuir a identificar al autor. Por otro lado, n-gramas a nivel de caracter podrían descubrir ciertas preferencias de estilo. Para ilustrar lo anterior, considere un espacio de características de 3-gramas, en el que una alta frecuencia de los términos *ing* y *ed_* podrían revelar la identidad de autores que tienden a escribir en tiempo progresivo o pasado respectivamente.

Además de las palabras vacías y los n-gramas, existen otros atributos de estilo que pueden hacer más efectiva la AA, por ejemplo algunos de ellos son: marcadores léxicos (e.g. *riqueza del vocabulario* y *longitud de las oraciones*), secuenciales (e.g., *n-gramas* de palabras), estructurales (e.g., *organización del texto*), de contenido, y de idiosincrasia (e.g., *errores gramaticales*) (Abbasi y Chen, 2008). A continuación se describen de manera general algunas de las características textuales más relevantes para la AA.

Características léxicas

Este tipo de características toman en cuenta al texto como una secuencia de tokens. Los tokens podrían ser palabras, números, signos de puntuación o abreviaturas. En este contexto, es posible definir distintas características léxicas basadas

en estas secuencias de tokens. Por ejemplo, medir la longitud de las oraciones, las palabras o los párrafos. También existen medidas para calcular la riqueza del vocabulario o el índice de repetitividad léxica de los documentos basados en el número de tokens (Miranda-García y Calle-Martín, 2005). Incluso extraer, palabras, puntuación, o cualquier otro tipo de token léxico que sea frecuente. La Tabla 2.1 detalla algunos de los atributos léxicos más importantes en AA.

Las características léxicas tienen la ventaja de que muchas de ellas pueden ser extraídas de igual forma para distintos idiomas, con un nivel de análisis relativamente sencillo utilizando herramientas existentes, como los Tokenizers, salvo algunas excepciones como en el Chino, donde esta tarea es un tanto más complicada (Stamatatos, 2009). No obstante, como ya se mencionó, algunas otras características léxicas, requieren de algoritmos más específicos para el idioma, tales como Lematizadores, divisores de oraciones, diccionarios o correctores ortográficos.

Características basadas en caracteres

Desde un punto de vista general, este tipo de característica considera al texto como una secuencia de caracteres. En este sentido, es posible definir características textuales que se basen en estadísticas de los caracteres o en secuencias selectas de estos en el texto.

Utilizar los N más frecuentes n-gramas a nivel de caracter, ha resultado ser de los atributos más efectivos para AA (Houvardas y Stamatatos, 2006). Los n-gramas a nivel de caracter son meras secuencias de caracteres de tamaño n. Éstos son una característica textual con las que es posible mantener información de contenido, al mismo tiempo que información contextual (Stamatatos, 2009). Los n-gramas suelen ser un tanto más tolerantes a errores gramaticales y de puntuación, que un simple enfoque léxico de obtener palabras como tokens. Por ejemplo,

Atributos léxicos en AA	
Atributo	**Descripción**
Palabras	Son conjuntos de palabras (e.g., artículos, preposiciones, adjetivos o adverbios) o simplemente extraer las n palabras más frecuentes (Pavelec *et al.*, 2008). En general, este tipo de características son llevadas a una representación tradicional tal como la BoT, para después clasificar con algún algoritmo de aprendizaje automático.
n-gramas de palabra	Son secuencias de palabras de tamaño n. Éstas mantienen más información contextual que los tokens aislados. Sin embargo, su sola utilización no siempre han resultado ser un buen atributo. Se ha demostrado que su exactitud en la clasificación no siempre es mejor que simplemente palabras aisladas. Lo anterior debido a que la dimensionalidad puede incrementar con el tamaño de n y dificulta el aprendizaje. Además, la representación vectorial suele resultar con alta dispersión, debido a que la combinación de tokens de tamaño n no siempre es encontrada en cada uno de los documentos a clasificar. Otro inconveniente, es que es bastante probable quedarse con n-gramas que representan contenido específico en lugar de información de estilo (Gamon, 2004).
Errores de escritura	Consiste en tomar en cuenta los errores de escritura con el objetivo de capturar cuestiones de idiosincrasia del autor (Koppel y Schler, 2003). Sin embargo, no siempre resulta ser un buen atributo sobre todo si se trata de documentos revisados (e.g., noticias, libros, artículos, etc.), además de que suelen ser dependientes del lenguaje y no siempre es posible conseguir un buen corrector ortográfico.

Tabla 2.1: Algunos tipos de características léxicas que pueden aportar información de estilo y contenido útil para Atribución de Autoría.

considere un documento con las palabras *Brasil* y *Brazil*; un enfoque de palabras los considerará como dos atributos distintos, cuando en realidad representan el mismo concepto, además de un error ortográfico o cierta idiosincrasia del autor. Por otro lado, con el enfoque basado en 3-gramas de carácter obtendremos

los siguiente atributos; *Bra*, *ras*, *raz*, *asi*, *azi*, *sil* y *zil*. Lo anterior significa que mantenemos la información de contenido en atributos como *Bra*, mientras por otro lado mantenemos esos sutiles errores gramaticales o preferencias en atributos como *sil* y *zil*. A pesar de los puntos a favor que tiene utilizar atributos basados en caracteres tales como los n-gramas, existen desventajas en comparación a un enfoque léxico basado en palabras. Una es el aumento en la dimensionalidad y dispersión en la representación. Por ejemplo, al obtener los N n-gramas más frecuentes puede ocasionar que para representar una palabra se necesiten varios n-gramas (e.g., *de_* , *_de*). Otro inconveniente en los n-gramas, consiste en cómo determinar el mejor valor para n, una n grande podría capturar información léxica y contextual, pero también información temática (Houvardas y Stamatatos, 2006). En relación a esto último, existen trabajos para elegir la mejor n o utilizar distintos tamaños de n (Sanderson y Guenter, 2006).

Características sintácticas

Obtener estas características tiene la idea de detectar elementos sintácticos comunes en la escritura del autor. En este sentido, este tipo de atributo es una idea más natural para capturar el estilo. Dos ejemplos de este tipo de característica son:

- Etiquetas de partes de la oración (POS, por sus siglas en inglés de *Part of Speech*). Por ejemplo, para enfocarnos en cómo el autor utiliza palabras que pueden ser empleadas como sustantivos o como adjetivos.

- Árboles sintácticos de las oraciones. Por ejemplo, para enfocarnos en la complejidad de las oraciones del autor (e.g., midiendo la profundidad del árbol sintáctico).

Es importante mencionar que, para extraer características sintácticas normalmente se requiere la utilización de herramientas de Procesamiento de Lenguaje Natural (PLN) más elaboradas, robustas y precisas, las cuales pudiesen no estar disponibles, ya que por lo regular son específicas para cada idioma. Otro inconveniente de estas características es que, siempre existen errores cometidos por las herramientas (no son precisas al 100 %), obteniendo inevitablemente cierto *ruido* del conjunto de datos.

La utilización de estos atributos en el estado del arte ha obtenido buenos resultados, aunque no tan buenos como la utilización de solo características léxicas (Stamatatos, 2009). Sin embargo, la combinación de ambas características ha mejorado los resultados (Gamon, 2004). Algunas de las herramientas para obtener este tipo de atributos son: etiquetadores de partes de la oración, analizadores sintácticos y algunos correctores ortográficos.

Características semánticas

Las características semánticas hacen referencia al significado, sentido, interpretación o coherencia de los diferentes elementos textuales. Extraer características semánticas del texto libre de restricciones quizá sea una de las tareas más complicadas del procesamiento automático de texto. En este tipo de texto no se tiene certeza acerca de la calidad de la escritura (semántica, sintáctica, ortográfica). Además, normalmente se carece de etiquetas o marcadores que proporcionen información acerca de los elementos textuales; por ejemplo, en *e-mails*, el saludo, contenido, firma, tipo de palabras, etc.

Extraer características semánticas puede requerir de un nivel profundo de análisis en el texto, que puede llegar a ser bastante impreciso. En general, las actuales herramientas para Procesamiento de Lenguaje Natural (PLN) no han logrado ma-

nipular apropiadamente tareas complejas como el análisis sintáctico de documentos enteros o el análisis semántico (Stamatatos, 2009). Además, las herramientas que realizan análisis de texto en este nivel suelen ser dependientes del idioma y muy sensibles a los errores gramaticales. Debido a todos estos inconvenientes, existen pocos trabajos que estudien características semánticas con propósitos de extraer elementos de estilo. Uno de los trabajos que se enfoca en estas características es el de Argamon *et al.* (2007). En este trabajo se define un conjunto de características que se asocian a palabras o frases, para identificar qué papel juegan los elementos textuales (e.g., palabras, frases) según su contexto. De esta forma, según el contexto anterior, identificar si se tiene una *aclaración, un complemento de información o un contraste.*

Características específicas de la aplicación

Estas características textuales, a diferencia de todas las anteriores, son dependientes del dominio y tipo de documentos (e.g., *emails*, *foros*, *chats*). En este sentido, podemos decir que al no ser tan generales, no pueden ser extraídas de cualquier conjunto de datos. Un ejemplo de lo anterior son los *emails*, dónde se pueden extraer características estructurales relacionadas con el estilo; por ejemplo, centrar la atención en la parte del saludo, indentado del contenido o firma. Otro ejemplo es en páginas en lenguaje HTML donde se cuenta con etiquetas de la estructura del documento (de Vel *et al.*, 2001). Sin embargo, como ya se ha mencionado, para generalizar el uso en las características textuales que representan el estilo, es importante evitar la dependencia del dominio de los datos.

2.1.3. Métodos convencionales

Una de las clasificaciones más conocidas de los métodos en AA ha sido propuesta en (Stamatatos, 2009). En general los métodos para obtener un modelo de atribución son tres:

1. **Basados en el perfil:** Fue de los primeros métodos en utilizarse (Mosteller y Wallace, 1964). La idea consiste en modelar el estilo de escritura basándose en una cantidad de texto representativa del autor. Éste podía ser obtenido a partir de la concatenación de todos sus documentos. La idea principal es ignorar las pequeñas diferencias entre sus documentos, y extraer características del estilo general (perfil) de escritura.

2. **Basados en instancias:** Se basan en la utilización de múltiples instancias de texto del autor. La idea es extraer características de estilo comunes a nivel documento. Los textos se representan como vectores de atributos, para luego utilizar algún algoritmo de clasificación. Los métodos más modernos normalmente utilizan este enfoque (Solorio *et al.*, 2011; de Vel *et al.*, 2001; Abbasi y Chen, 2008; Plakias y Stamatatos, 2008).

3. **Híbridos:** Éstos combinan características de los dos anteriores. Por ejemplo, representar de manera individual cada documento, pero utilizando características obtenidas a nivel clase. Es decir, se aplica algún algoritmo de clasificación tal como en los métodos basados en instancias, pero sobre vectores de documentos cuyas características textuales fueron extraídas a partir del perfil de escritura de cada autor, tal como en los métodos basados en perfil.

2.2. Algoritmos de Ensambles

La idea principal detrás de estos algoritmos consiste en construir un esquema de predicción integrando múltiples modelos de clasificación. El objetivo es que el modelo de clasificación del ensamble sea mejor que cada uno de sus clasificadores individuales (Rokach, 2009). Una de las preguntas más importantes en el estudio de aprendizaje por Ensambles es, ¿Realmente un conjunto de clasificadores pueden crear uno más fuerte? En este sentido, la respuesta depende de las condiciones del problema y de cómo se sigan las condiciones para construir el ensamble.

De acuerdo al estado del arte relacionado con los métodos de ensambles, la idea de combinar clasificadores no ha sido una tarea sencilla (Rokach, 2009); es por ello que existen criterios importantes a considerar. Un ejemplo es que, entre los miembros del ensamble es importante que exista diversidad en las predicciones. Otro consideración es que exista independencia, es decir que cada uno pueda especializarse y establecer sus propias opiniones basadas en su conocimiento privado. Así también, es necesario algún mecanismo que convierta todas estas opiniones privadas en una decisión colectiva.

2.2.1. Métodos de ensambles

En cuanto a métodos de ensambles se refiere, existen dos grandes categorías (Rokach, 2009) i) los métodos dependientes, en los que cada nuevo clasificador se enfoca en aprender los errores del clasificador anterior, y ii) los independientes, que no se ven influenciados por el desempeño de otros clasificadores. En las siguientes secciones explicaremos las ventajas y desventajas de cada método. Así como su esquema de trabajo, el cual puede ser un factor importante según el problema a resolver. Sin embargo, antes de describirlos cabe mencionar que, existen

4 elementos fundamentales en la construcción de un ensamble (Rokach, 2009), los cuales son mostrados a detalle en la Tabla 2.2.

Elementos para ensambles	
Elemento	**Descripción**
Conjunto de datos de entrenamiento	Normalmente un conjunto de instancias representadas como vectores de n atributos, con un atributo objetivo llamado clase.
Constructor del modelo de clasificación	Es un algoritmo I que a partir de un conjunto de entrenamiento S, obtiene un clasificador M, es decir $M=I(S)$.
Generador de diversidad	Es el componente encargado generar diversidad entre los clasificadores creados. Normalmente, selecciona las instancias o atributos con las que se construye cada clasificador.
Combinador	Es el encargado de combinar las predicciones, con el objetivo de convertirlas en una decisión comunitaria.

Tabla 2.2: Elementos normalmente utilizados para la construcción de ensambles de clasificadores.

Métodos Dependientes

En estos métodos se construyen los miembros del ensamble utilizando información de los clasificadores construidos en iteraciones anteriores (Provost y Kolluri, 1999). La idea principal es que los clasificadores construidos, en iteraciones posteriores logren ser más especializados en elementos definidos como importantes según el problema.

Un ejemplo de este tipo de método son los guiados por selección de instancias, también conocidos como *Boosting*. En este enfoque cada clasificador entrena con el conjunto de instancias que su predecesor clasificó incorrectamente. En este sentido,

cada nuevo clasificador se va especializando en aquellas instancias difíciles. Como ejemplo está el algoritmo *AdaBoost* (*Adaptive Boosting*), éste fue presentado por primera vez por Freund y Schapire en (Freund y Schapire, 1996).

Algoritmo 2.1 El algoritmo *AdaBoost*

Entrada: **I** (constructor del modelo de clasificación), **T** (número de iteraciones), **S** = $\{x_1, x_2, \ldots, x_m\}$ (conjunto de ejemplos etiquetados)
Salida: $M_t, \alpha_t; t = 1, \ldots, T$

1: $t = 1$
2: $D_1(i) = 1/m; i = 1, \ldots, m$
3: **Repite**
4: Construir el clasificador M_t utilizando I y la distribución de instancias D_t
5: $\varepsilon_t = \sum_{i:M_i(x_i)\neq y_i} D_t(i)$
6: **Si** $\varepsilon > .5$ **entonces**
7: $T = t - 1$
8: salir del ciclo.
9: **Fin Si**
10: $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
11: $D_{t+1}(i) = D_t(i) \cdot e^{-\alpha_t y_t M_t(x_i)}$
12: Normaliza D_{t+1}
13: $t++$
14: **Hasta** $t < T$

El Algoritmo 2.1 muestra una versión del *AdaBoost* para un conjunto de m instancias, etiquetadas como -1 y $+1$. La idea central es asignar un peso a cada instancia y cada clasificador. Para esto, en un inicio todas las instancias y clasificadores son igual de importantes (línea 2). En seguida, cada nuevo clasificador realiza una prueba de clasificación que registra su desempeño (línea 5). Posteriormente, según cierto umbral, si el desempeño es bueno se continua la construcción de clasificadores (línea 6), para luego ajustar el peso del clasificador (línea 10) y de las instancias en las que se equivocó (línea 11). Posteriormente, el clasificador de la siguiente iteración se enfoca en aprender las instancias con más peso, es

decir las más difíciles (línea 4). Para la clasificación de instancias de prueba, cada clasificador vota con su peso α, la ecuación 2.1 muestra la idea de la predicción. En pocas palabras, se pretende que el proceso iterativo logre obtener una serie de clasificadores que se complementen entre sí.

$$H(x) = signo\left(\sum_{t=1}^{T} \alpha_t \cdot M_t(x)\right) \tag{2.1}$$

Métodos Independientes

En este enfoque los datos de entrenamiento son transformados en varios subconjuntos, a partir de los cuales los clasificadores son entrenados (Rokach, 2009). Estos subconjuntos de datos pueden ser disjuntos o no. En cuanto a la predicción final se utiliza algún método de combinación (e.g., voto mayoritario). Una de las ventajas de este tipo de sistemas es que pueden ser fácilmente paralelizados, o utilizar fácilmente distintos tipos de clasificadores.

Uno de los métodos más conocidos es el *Bagging* (por sus siglas en inglés de *bootstrap aggregating*). Éste es mostrado en el Algoritmo 2.2, dónde la idea central es obtener un clasificador compuesto I^*, que para obtener buena diversidad entrena cada miembro del ensamble con una submuestra S_t de un tamaño μ con reemplazo (línea 3). La predicción del clasificador compuesto I^* se obtiene con la clase más veces obtenida (método de votación simple).

Una característica importante de los clasificadores *Bagging*, es que a menudo obtienen un clasificador compuesto mejor que cualquiera de sus miembros construido con el conjunto de datos original. La cual es especialmente cierta para los algoritmos que al construir el modelo obtienen clasificadores con cambios significativos si el conjunto de datos fue alterado (Breiman, 1996). Por ejemplo, en

los árboles de decisión, modelos específicos sin poda, construidos sobre distintos subconjuntos de datos, difieren significativamente (Rokach, 2009).

Algoritmo 2.2 El algoritmo *Bagging*

Entrada: **I** (constructor del modelo de clasificación), **T** (número de iteraciones), **S** = $\{x_1, x_2, \ldots, x_m\}$ (conjunto de ejemplos etiquetados), μ (tamaño de la submuestra)
Salida: $M_t; t = 1, \ldots, T$
1: $t = 1$
2: **Repite**
3: S_t= Muestra de μ instancias de S con reemplazo.
4: Construir clasificador M_t utilizando $I(S_t)$
5: $t++$
6: **Hasta** $t > T$

Otro de los algoritmos *Bagging* más conocidos es el *Random Forest* (ver Algoritmo 2.3), el cual emplea un gran número de árboles de decisión sin poda (Breiman, 2001). Éste utiliza N atributos aleatorios y sobre ellos, según su criterio, determina el que mejor discrimine (línea 4).

Algoritmo 2.3 El algoritmo *Random Forest*

Entrada: **IDT** (constructor de árboles de decisión), **T** (número de iteraciones), **S** = $\{x_1, x_2, \ldots, x_m\}$ (conjunto de ejemplos etiquetados) y **N** (número de atributos utilizados en cada nodo)
Salida: $M_t; t = 1, \ldots, T$
1: $t = 1$
2: **Repite**
3: S_t = Muestra de μ instancias de S con reemplazo.
4: Construir un clasificador M_t utilizando $IDT(S_t, N)$.
5: $t++$
6: **Hasta** $t > T$

2.2.2. Métodos de combinación

La combinación de predicciones es uno de los pasos más importantes para el éxito de un ensamble. Dentro de la literatura, podemos clasificar en dos ramas los métodos de combinación de decisiones: métodos de pesado y métodos de meta aprendizaje. Las siguientes Secciones introducen la idea principal acerca de cada método de combinación.

Métodos de pesado

Son muy utilizados cuando los clasificadores tienen un rendimiento comparable al realizar la misma tarea (Rokach, 2009). Un ejemplo de método de pesado es el voto mayoritario, también conocido como el método básico de ensambles (*BEM, basic ensemble method*, por sus siglas en inglés) que simplemente obtiene la clase más elegida por los clasificadores. Muy a menudo éste es utilizado como *baseline* a nuevos métodos de votación. La Expresión 2.2 es una aproximación de (Rokach, 2009) a las ideas anteriores.

$$clase(x) = argmax_{c_i \epsilon dom(y)} \left(\sum_k g(y_k(x), c_i) \right) \tag{2.2}$$

donde $y_k(x)$ es la clasificación del k-ésimo clasificador y $g(y, c)$ es un indicador de función definido como:

$$g(y, c) = \begin{cases} 1 & y = c \\ 0 & y \neq c \end{cases} \tag{2.3}$$

Algunos métodos de votación consideran otros aspectos de la elección, tal como los votos para los otros candidatos. Por ejemplo, en la votación de Condorcet

gana el candidato más preferido ante el resto de los candidatos al compararse de uno en uno (Montague y Aslam, 2002). En el sistema de votación Condorcet básico cada votante ordena a los candidatos en función de sus preferencias, es decir del candidato favorito hasta el menos preferido. En el momento de votar, se compara por parejas a cada candidato contra el resto de los candidatos. Es decir, enfrentar a los candidatos uno a uno en muchas elecciones. Para conocer al ganador entre dos candidatos A y B se cuenta el número de votantes que prefieren a A sobre B y viceversa. El ganador de la elección es el que ganó la mayoría de los enfrentamientos por parejas. En caso de empate, se utiliza un mecanismo alternativo; por ejemplo, votación mayoritaria entre los empatados, o un voto simple para todos los candidatos. En caso de presentar nuevamente empate, otro criterio de selección (e.g., una selección aleatoria) se emplea hasta encontrar un ganador.

Una forma sencilla de hacer la votación Condorcet se explica a continuación. Considere una elección entre 4 candidatos; A, B, C y D. En este contexto, suponga que un votante tiene su lista de candidatos como:B, C, A, D. En esta lista B es el candidato favorito y D es el menos preferido. En seguida, cada lista de votación se convierte en una matriz de voto, donde un 1 significa preferencia al candidato que se está comparando contra todos, y 0 significa que prefiere al candidato rival. Así pues, para el resultado final de la elección, solo hay que sumar las matrices de cada votante. A continuación la Tabla 2.3 muestra la matriz de voto para la lista anterior.

	A	**B**	**C**	**D**
A	-	0	0	1
B	1	-	1	1
C	1	0	-	1
D	0	0	0	-

Tabla 2.3: Matriz de votación Condorcet de un votante con lista: A, B, C, D.

Métodos de meta aprendizaje

Suelen ser empleados cuando los clasificadores muestran un patrón de aciertos o errores en la clasificación de ciertas instancias.

En cuanto a los métodos de meta aprendizaje, el *Stacking* es uno de los mejor conocidos. El *Stacking* es una técnica para obtener la mayor generalización a partir de la exactitud de los modelos (Wolpert, 1992). Normalmente es utilizado en ensambles a base de distintos tipos de clasificadores (e.g., SVM, Naïve Bayes). La idea es centrar la atención en el patrón de las decisiones que toman los distintos clasificadores. En este contexto, la decisión final es tomada por un meta clasificador, llamado así debido a que tuvo una fase de entrenamiento en la que las instancias tienen como atributos las decisiones de los clasificadores miembros.

2.2.3. Generación de diversidad

La generación de diversidad es uno de los elementos más importantes a la hora de construir ensambles con buen desempeño (Tumer y Ghosh, 1996; Krogh y Vedelsby, 1995; Kuncheva, 2005; Maimon y Rokach, 2002). Es interesante mencionar que, aunque es uno de los aspectos fundamentales en casi todos los métodos de

ensambles de la actualidad, en el contexto de clasificación no existe una teoría ampliamente aceptada y definitiva que explique cómo y por qué la diversidad entre distintos modelos contribuye positivamente en el desempeño del clasificador (Brown *et al.*, 2005) . En este sentido, Rokach (2009) clasifica las distintas formas de obtener esta diversidad que son utilizadas en el estudio de ensambles, las cuales son:

- **Manipulación del conjunto de entrenamiento:** cada miembro del ensamble entrena con diferentes muestras o proyecciones del conjunto de datos.
- **Manipulación del constructor del modelo de clasificación:** por ejemplo, ajustar los parámetros con los que se construye el clasificador.
- **Cambiar la representación del atributo objetivo (clase):** cada clasificador resolverá un concepto distinto. Normalmente se remplaza el atributo clase con una función, tal que el dominio del nuevo atributo clase es más pequeño que el original.
- **Particionar el espacio de búsqueda:** La idea es que cada miembro explore en un subespacio distinto, del espacio de búsqueda total. Un ejemplo de ello es entrenar los clasificadores utilizando solo ciertos subcojuntos de características de las instancias. En este sentido, cada clasificador contará con distintas vistas del conjunto de datos.
- **Híbridos:** la idea es obtener diversidad combinando cualquiera de las estrategias anteriores. Por ejemplo, utilizando varios tipos de clasificadores en conjunto con la manipulación del conjunto de entrenamiento.

Capítulo 3

Trabajo relacionado

En este capítulo se presenta el trabajo relacionado más relevante para esta tesis. La Sección 3.1 inicia con una descripción de las representaciones tradicionales en AA, posteriormente en la Sección 3.2 presenta trabajos que han utilizado distintos tipos de atributos, luego la Sección 3.3 presenta algunos enfoques de AA relevantes para este trabajo, y finalmente la Sección 3.4 presenta otros enfoques no tan convencionales en AA y termina por introducir algo de la idea de nuestro trabajo.

3.1. Representaciones tradicionales

Una manera de abordar la AA es considerarla como un problema estándar de clasificación. De esta forma, se puede establecer como un problema multiclase de una etiqueta, donde los autores representan la clase a discriminar. Por lo tanto, distintos enfoques tradicionales pueden ser utilizados para hacer frente a la iden-

tificación de autores. Por ejemplo, la Bolsa de Términos (*BoT, Bag of Terms*[1], por sus siglas en inglés) clasificando con Máquinas de Vectores de Soporte (*SVM, Support Vector Machines*, por sus siglas en inglés) ha sido un enfoque ampliamente utilizado para AA (Houvardas y Stamatatos, 2006). Las representaciones del tipo BoT construyen vectores utilizando características textuales; por ejemplo, tomar cada palabra del vocabulario como atributo. De esta manera, la BoT representa documentos con vectores de características, asignando un valor a cada una de ellas (Pavelec *et al.*, 2008). Este valor podría ser desde valores Booleanos (e.g., 1 o 0) hasta complejos valores calculados a partir del análisis del corpus.

Las representaciones BoT han sido muy utilizadas para identificar autores de correos electrónicos, filtración de *spam* y detección de plagio (Stamatatos, 2009). Sin embargo, uno de los principales problemas de las representaciones BoT es que no mantienen ningún orden o relación entre los términos o clases; lo cual podría proporcionar información valiosa para mejorar la representatividad de los documentos. Un segundo problema con las representaciones del tipo BoT ocurre en escenarios realistas de AA donde existen grandes vocabularios, pero pocos datos de entrenamiento y clases desbalanceadas para los autores candidatos (Stamatatos, 2008). En consecuencia, las representaciones BoT tienden a favorecer las clases mayoritarias, cuando de hecho cada documento puede pertenecer a cualquiera de los autores. (e.g., en cómputo forense donde se requiere discriminar entre un conjunto de presuntos culpables) (Stamatatos, 2008). Un tercer problema con las representaciones BoT es que normalmente tienen alta dimensionalidad, lo que requiere de un gran número de recursos computacionales para llevar a cabo la

[1]Utilizamos las siglas BoT para referirnos a una Bolsa de Palabras (BoW, *Bag of Words*, por sus siglas en inglés), el que los atributos además de palabras pueden ser: n-gramas de caracter, signos de puntuación, una etiqueta que represente la longitud de la palabra, etc.

clasificación de grandes conjuntos de documentos, lo cual podría ser poco práctico en algunas situaciones (e.g. AA en foros, donde se pueden llegar a tener cientos de documentos para algunos autores) (Solorio *et al.*, 2011).

3.2. Utilización de múltiples tipos de atributos

En AA realizar una selección de qué atributos representan el estilo y cómo combinarlos no ha resultado ser una tarea trivial. Existen algunos trabajos en los que se combinan dos o más conjuntos de atributos textuales distintos. Sin embargo, esto implica que normalmente se tiene que considerar el problema de la dimensionalidad; el cual afecta la calidad de la representación, y dificulta la tarea de los algoritmos de aprendizaje. Para afrontar esta situación, algunos trabajos han utilizado algoritmos de selección de características (Forman, 2003). Desafortunadamente, el uso de este tipo de algoritmos puede obtener demasiados atributos de contenido temático, en lugar de atributos que representen el estilo del autor (Stamatatos, 2009) . Por lo tanto, la alternativa más común ha sido seleccionar conjuntos de características textuales enfocados a retener mayor información de estilo. En este sentido, la selección de características en AA suele diferir con la de otras tareas tales como la Clasificación Temática. Por ejemplo, distintos trabajos en AA han demostrado que uno de los criterios más importantes consiste en seleccionar aquellos elementos más frecuentes (Koppel, Akiva, y Dagan, 2006) (Forman, 2003) (éstos normalmente se eliminan en otras tareas). Es decir, a partir de un conjunto de atributos definido, seleccionar los que más ocurrencias tengan dentro del conjunto de documentos. Lo anterior, con la idea de que entre más frecuente sea el atributo a través del corpus, más variación de estilo podría ser capturada. Otro punto importante está en la combinación de atributos. Por ejem-

plo, en muchas ocasiones las características textuales que parecen ser irrelevantes de manera individual, pueden llegar a ser útiles en conjunto con otras (Gamon, 2004).

Desde hace algunos años se han realizado distintos trabajos en AA que contemplan la utilización de conjuntos diferentes de atributos con enfoques de aprendizaje automático. Por ejemplo, en (de Vel *et al.*, 2001) tomaron distintos marcadores de estilo (*e.g., total de palabras cortas, total de líneas en blanco, palabras vacías, etc.*), y representaron correos electrónicos como vectores de 170 atributos; posteriormente, clasificaron con SVM. En otro trabajo, Pavelec *et al.* en (Pavelec *et al.*, 2008) utilizaron adverbios y adjetivos para identificar a los autores de documentos cortos en Portugués a través de SVM. Por otra parte, algunas investigaciones proponen un uso más extenso de tipos de atributos, por ejemplo Abbasi y Chen en (Abbasi y Chen, 2008) proponen una técnica para identificación de autores y perfiles empleando un conjunto de cinco distintos tipos atributos con métodos de Análisis de Componentes Principales (*PCA, Principal Component Analisys* por sus siglas en inglés). Otros trabajos proponen el uso de clasificadores ensamblados que utilicen distintos tipos de atributos. Por ejemplo, Stamatatos y Widmer (Stamatatos y Widmer, 2005) utilizaron un ensamble basado en SVM donde cada clasificador es entrenado con un conjunto diferente de atributos. De manera similar, Cherkauer utilizó un ensamble de redes neuronales entrenadas sobre distintos conjuntos de atributos (Abbasi y Chen, 2008). La idea principal detrás del uso de ensambles es obtener un conjunto de clasificadores que, mediante una decisión colectiva, mejoren la predicción de autores en la clasificación final (Stamatatos y Widmer, 2005)

3.3. Enfoques para identificación de autores

La Tabla 3.1 muestra un pequeño historial de algunos de los trabajos en AA relevantes para esta tesis.

Trabajo	**Múltiples atributos**	**Ensambles**	**Representación especial para atributos**	**Ponderación por atributos**
(Stamatatos y Widmer, 2005)	Sí	Sí	No	No
(Pavelec *et al.*, 2008)	Sí	No	No	No
(Plakias y Stamatatos, 2008)	Sí	No	No	No
(Koppel, Schler, *et al.*, 2006)	Sí	No	No	No
(Abbasi y Chen, 2008)	Sí	No	Sí	No
(Escalante *et al.*, 2011)	No	Sí	No	Sí
(de Vel *et al.*, 2001)	Sí	No	No	No
(Frantzeskou *et al.*, 2007)	Sí	No	No	No
(Solorio *et al.*, 2011)	Sí	No	Sí	No
(Kern *et al.*, 2011)	Sí	Sí	No	Sí
Propuesta	Sí	Sí	Sí	Sí

Tabla 3.1: Algunos trabajos relevantes para esta tesis en AA.

En la Tabla 3.1 se considera si el trabajo utilizó los siguientes criterios para llevar a cabo la AA:

- **Múltiples tipos de atributos:** considera si utilizaron más de un tipo atributo para hacer la identificación de autor.
- **Ensambles:** considera si de alguna forma emplearon ensambles de clasificadores que tomaran en cuenta los tipos de atributos.

- **Representación individual por atributo:** es decir, si se realizó algún tipo de consideración para representar por separado a cada tipo de atributo.
- **Ponderación por atributo:** es decir, si se tomó en cuenta algún tipo de ponderación de estilo para intentar beneficiar a la AA.

Por otro lado, la Tabla 3.2 complementa la información presentada en la Tabla 3.1. La primer columna muestra el máximo número de autores y la segunda columna la máxima exactitud alcanzada en la clasificación.

Trabajo	**Número de autores**	**Exactitud alcanzada**
(Stamatatos y Widmer, 2005)	22	70 %
(Pavelec *et al.*, 2008)	20	83.2 %
(Plakias y Stamatatos, 2008)	10	78 %
(Koppel, Schler, *et al.*, 2006)	10000	88.2 %
(Abbasi y Chen, 2008)	100	91.3 %
(Escalante *et al.*, 2011)	10	86.4 %
(de Vel *et al.*, 2001)	3	92.5 %
(Frantzeskou *et al.*, 2007)	8	100 %
(Solorio *et al.*, 2011)	100	62.1 %
(Kern *et al.*, 2011)	66	67.3 % %

Tabla 3.2: Máxima exactitud alcanzada de algunos trabajos relevantes para esta tesis.

Es importante señalar que la Tabla 3.2 es solo para tener una idea muy general de los resultados alcanzados por el estado del arte relacionado con esta tesis. Esto principalmente porque la mayoría de los trabajos utilizan un conjunto de datos distinto. Es decir, los experimentos son en dominios diferentes (*e-mails, blogs*, foros, noticias, etc.), con documentos de distinta longitud o utilizando un número

de autores tan grande que puede requerir abordar la AA con otro enfoque (e.g., con *recuperación de información*).

Algunos de los trabajos de la Tabla 3.1 fueron mencionados en secciones previas de este capítulo. Sin embargo, el trabajo de Kern *et al.* (2011) sobresale debido que cumple más características (ver columnas de la Tabla 3.1) con respecto a la propuesta de esta tesis. En el trabajo de Kern *et al.* (2011) se realiza AA utilizando un amplio número de atributos textuales y ensambles de clasificadores. Éste trabajo tiene valor para AA por el lado de su método de votación en los ensambles. En la votación se propone un esquema de voto/veto pesado (vea explicación de métodos de pesado en Sección 2.2.2). La idea básica es que cada clasificador además de votar positivamente, también puede votar negativamente de acuerdo a ciertos umbrales alcanzados en una etapa previa de entrenamiento. Sin embargo, aunque se utilizaron ensambles, éstos son utilizados en su forma tradicional tomando un vector de características como si todas fueran de un mismo tipo (clasificando con *Random Forest*). Es decir no aprovechan el uso de los clasificadores ensamblados para sacar ventaja de aspectos importantes en AA. Por ejemplo, Plakias y Stamatatos (2008) utilizan ensambles de SVM para especializar el aprendizaje en cada tipo de atributo. En otro ejemplo de la Tabla 3.1, Escalante *et al.* (2011) utiliza ensambles SVM para aprender vectores de características que son locales a ciertas partes del texto. Por ejemplo, vectores de características de la parte inicial o final de los documentos, con la idea de mantener el estilo de cómo cada autor inicia o termina de escribir.

Aún cuando distintas investigaciones han proporcionado pistas de que utilizar más de un tipo de atributo beneficia la tasa de clasificación, aún no existen investigaciones que den respuestas concisas de cómo combinar distintos tipos de atributos en AA. Es por ello que en esta tesis proponemos un par de alternativas

utilizando algoritmos de ensambles y el uso de distintos espacios de atributos para llevar a cabo la identificación de autores.

3.4. Otras representaciones y enfoques en AA

Existen otros tipos de representaciones no tan convencionales en AA. Por ejemplo, Plakias y Stamatatos (2008) propusieron el uso de Tensores de Segundo Orden para representar las propiedades de estilo de los textos. La idea principal detrás de esta representación consiste en ubicar a las características relevantes tomando en cuenta el contexto de cada término y su frecuencia. Lo último se logra debido a que el modelo basado en tensores toma en cuenta las asociaciones entre características que se encuentran en la misma vecindad (Plakias y Stamatatos, 2008). De esta forma, cada característica es asociada con otras dentro del mismo renglón y columna. Para manejar tensores en vez de vectores utilizan una generalización de SVM llamada *Support Tensor Machines* (STM) (Cai *et al.*, 2006). Para la evaluación utilizaron los 2500 n-gramas más frecuentes, y utilizaron la exactitud para medir la clasificación. Esta representación con tensores toma en cuenta cierta relación entre los términos. No obstante, ésta no garantiza resolver el problema de la dispersión de la información y la alta dimensionalidad.

En otras áreas del tratamiento automático de textos, existen algunas técnicas para construir relaciones semánticas produciendo vectores de baja dimensionalidad. Por ejemplo, el Análisis Semántico Latente (*LSA, Latent Semantic Analysis*, por sus siglas en inglés) (Deerwester, 1990) y el Análisis Semántico Explicito (ESA, *Explicit Semantic Analysis*, por sus siglas en inglés) (Gabrilovich y Markovitch, 2009) que interpretan elementos del texto y sus relaciones en un conjunto predefinido de conceptos. Este tipo de técnicas hacen frente al problema de la

dimensionalidad, debido a que ésta es limitada por el número de elementos semánticos (conceptos). Sin embargo, el problema con estas técnicas es que usualmente es necesario interpretar los términos en un complejo espacio de conceptos (Zhixing *et al.*, 2010), lo cual resulta en un alto costo computacional; además, tal como ya se mencionó estas técnicas fueron pensadas para tareas de *recuperación de información* o *clasificación temática* (Zhixing *et al.*, 2010).

Considerando las situaciones anteriores y dado que la representación de documentos es un procedimiento clave; nuestro interés radica en un método para llevar a cabo un simple pero efectivo análisis semántico enfocado en la tarea de AA. En nuestra propuesta seguimos algunas de las idea del Análisis Semántico Conciso (*CSA, Concise Semantic Analysis* (Zhixing *et al.*, 2010), por sus siglas en inglés), la cual es una técnica independiente del lenguaje diseñada para clasificación temática que extrae algunos conceptos de las etiquetas de las clases del corpus. CSA ha sido exitosamente utilizada en clasificación temática (Zhixing *et al.*, 2010) empleando solamente las palabras como términos, sin embargo no ha sido utilizada para la tarea de AA. En nuestra propuesta seguimos algunas de las ideas de CSA con el objetivo de obtener relaciones entre términos, documentos y autores. Sin embargo, se han introducido funciones diferentes para pesar los vectores de términos y documentos con el objetivo de favorecer la tarea de AA. Nuestra idea es conseguir una representación especial para AA que ayude a hacer frente a los principales problemas de la representación convencional BoT. A continuación presentamos concretamente las desventajas a las que nos referimos:

- **No preservan ningún tipo de relación entre los términos y las clases:** En este contexto, información valiosa está siendo ignorada, principalmente porque creemos que, para características de estilo, podría ser útil tomar en cuenta las relaciones entre los autores y sus vocabularios más allá

de frecuencias de palabras aisladas.

- **Producen alta dimensionalidad y una alta dispersión de la información:** Ambas afectan la calidad de la representación y el rendimiento de la mayoría de los algoritmos de Aprendizaje Automático; especialmente cuando existen vocabularios grandes, pero datos de entrenamiento escasos y desbalanceados.

En este documento introducimos la Representación Documento Autor para caracterizar documentos, con el objetivo de superar esas desventajas en AA y ayudar a nuestro método de combinación de atributos con ensambles. Con relación a la primera desventaja de BoT, proponemos utilizar la riqueza léxica de los documentos y relaciones entre los términos, documentos y autores para mejorar la representatividad. De esta forma, nos interesamos en las relaciones que los autores mantienen con sus términos, para después definir cómo un documento está relacionado con su autor. En este contexto, a estos atributos de relaciones les llamaremos atributos de segundo orden, debido a que son calculados a partir de los atributos que fueron extraídos para BoT. Estos atributos de segundo orden son pocos, pero también son ricos en representatividad; lo cual hace frente a la segunda desventaja.

Capítulo 4

Método propuesto

En este capítulo se presenta un método alternativo para la representación de textos en AA; la idea principal es obtener un nuevo conjunto de atributos que relacionan a los documentos con cada autor. Además, proponemos el uso de la riqueza del vocabulario; siguiendo la idea de que los autores tienden a escribir sus documentos con tasas similares de repetición para sus términos. Por otro lado, para probar la idea de que la combinación de atributos beneficia a la AA se presenta: i) una combinación sencilla de atributos a la que nos referiremos como Vista General y ii) una combinación que los considere por separado a la cual llamamos Vista Individual. Para la explicación de estos métodos se inicia por una descripción general del método cuyos elementos específicos se presentan a través de las siguientes secciones.

4.1. Representación de atributos

En el Capítulo 3 ya se ha mencionado que la representación BoT es el enfoque tradicional para representar los documentos en AA. Así también se han expuesto

sus desventajas tales como su alta dimensionalidad y dispersión, las cuales dificultan la tarea de los algoritmos de aprendizaje. En este contexto, proponemos el uso de una nueva representación que aborde los principales problemas de BoT y que ayude en la tarea de AA. Se pretende obtener un nuevo conjunto de atributos (a los cuales nos referimos como de segundo orden) que puedan ser utilizados por si solos o en complemento con los atributos de las representaciones convencionales.

4.1.1. Representación Documento Autor (RDA)

La Representación Documento Autor (RDA) está motivada por algunas de las ideas del Análisis Semántico Conciso (Zhixing *et al.*, 2010), pero transportándolas al contexto de la AA. En este sentido, hacemos un pesado de los términos considerando la riqueza del vocabulario y frecuencias de los términos, lo que nos permite obtener de manera sencilla un análisis semántico para AA. La RDA almacena características textuales de los documentos en un vector, donde el problema de la dimensionalidad está limitado por el número de autores a clasificar. Para lograr esto, la RDA es construida en dos pasos; primero se construyen vectores de términos en un espacio de autores, y luego se construyen vectores de documentos en un espacio de autores. Las siguentes dos secciones explican estos pasos con detalle.

Representación de los Términos

La representación de los términos es el primer paso para obtener la RDA. Para esta etapa, es necesario construir una representación en el modelo vectorial para cada término. Recuerde que, los términos son cualquier unidad textual utilizada como característica del documento, por ejemplo: palabras, n-gramas, frases, puntuación, etc.

La idea principal detrás de este primer paso es capturar la relación que cada término mantiene con cada autor. En otras palabras, la intención es calcular valores que muestren cómo un término t_j es utilizado por cada autor a_i. En este contexto, sea $\{t_1, \ldots, t_m\}$ el vocabulario en la colección y $\{a_1, \ldots, a_n\}$ el conjunto de autores a ser identificados. Para cada término t_j en el vocabulario, construimos un vector $\mathbf{t_j} = \langle ta_{1j}, \ldots, ta_{nj} \rangle$, donde ta_{ij} es un valor real que representa la relación del término t_j con el autor a_i. Para calcular ta_{ij} solo tomamos en cuenta aquellos documentos que pertenecen al autor a_i, es decir obtenemos un valor a nivel clase. La relación de un término con un autor toma en cuenta la frecuencia del término en los documentos de ese autor. De esta forma, frecuencias altas denotan cierta preferencia por el término. La Expresión 4.1 expresa la idea anterior calculando un peso relativo.

$$w_{ij} = \sum_{k:d_k \in A_i} \log_2 \left(1 + \frac{tf_{kj}}{len(d_k)}\right) \tag{4.1}$$

Donde A_i es el conjunto de documentos que pertenecen al autor a_i, tf_{kj} es el número de ocurrencias del término t_j en el documento d_k, y $len(d_k)$ es la longitud del documento d_k. El objetivo de la función logarítmica en la Expresión 4.1 es suavizar las frecuencias altas de los términos.

Como se puede observar, debido a la sumatoria de estas frecuencias, los pesos pueden variar demasiado entre los términos. Por lo tanto, es conveniente aplicar la normalización de la Expresión 4.2 para obtener el valor final de ta_{ij}. Note que, esta normalización toma en cuenta los pesos que fueron calculados para otros autores, consiguiendo que cada peso sea relativo a todos los autores.

$$ta_{ij} = \frac{w_{ij}}{\sum_{i=1} w_{ij}} \tag{4.2}$$

De las expresiones anteriores se puede observar que el método obtiene información a nivel clase. Es decir, a partir de los documentos conocidos de cada autor. En este sentido, los vectores de término se calculan solamente a partir del conjunto de entrenamiento. De tal forma que al hacer la representación para el conjunto de prueba, se utilizan los vectores de término calculados en el conjunto de entrenamiento.

Representación de los Documentos

En el paso previo calculamos los vectores de términos que representan las relaciones entre los términos y los autores. La idea principal en este segundo paso es construir relaciones entre los documentos y los autores; éstos son, en cierto modo nuestros atributos de segundo orden. Estos los calculamos a partir de los vectores de los términos contenidos en el documento. Para ello, obtenemos los términos de cada documento y sumamos sus vectores. De esta forma, tendremos documentos representados como $\mathbf{d_k} = \langle da_{1k}, \ldots, da_{nk} \rangle$, donde n es el número total de autores, y da_{ik} es un valor real que representa la relación entre el documento d_k con el autor a_i. Cabe señalar que, cada vector término $\vec{t}_j$ antes de ser sumado, es pesado por la frecuencia del término t_j en el documento d_k, normalizado por la longitud de d_k. Finalmente, con el objetivo de tomar en cuenta la tasa de repetitividad del contexto, multiplicamos por la riqueza léxica del documento d_k (ver explicación de la Expresión 4.4). La Expresión 4.3 muestra las ideas anteriores.

$$\vec{d_k} = riqueza(d_k) \sum_{t_j \epsilon D_k} \frac{tf_{kj}}{len(d_k)} \times \vec{t_j} \tag{4.3}$$

donde D_k es el conjunto de términos que pertenecen al documento d_k. Adicionalmente definimos:

$$riqueza(d_k) = \frac{1}{repetitividad(d_k)} \tag{4.4}$$

La Expresión 4.4 intenta capturar mas información acerca de la riqueza léxica; siguiendo la idea de que los autores tienden a mantener tasas de repetitividad similares de sus términos a través de sus documentos. Además, la riqueza léxica nos permite hacer frente a la siguiente situación; documentos con alta riqueza léxica normalmente tienen muchos términos con frecuencias bajas (esto es relativo a la longitud del documento). De ello intuimos que, estas bajas frecuencias no reflejan la importancia adecuada de los términos del autor. Lo anterior se basa en la hipótesis de que, un autor con documentos léxicamente ricos presta más atención en seleccionar sus términos para transmitir el mensaje; esto quiere decir que a través del texto existen términos muy importantes con una tasa de repetición relativamente baja. Por lo tanto, para la relevancia de los términos, consideramos la riqueza léxica de el documento que los contiene. De esta forma, si el contexto es rico, entonces los términos fueron cuidadosamente seleccionados y por lo tanto su relevancia será mayor aunque su frecuencia sea baja.

Para calcular la repetitividad de un documento necesitamos una medida independiente de la longitud del texto. Por esta razón, hemos utilizado la K de Yule, calculada tal como sugiere (Miranda-García y Calle-Martín, 2005). La Expresión 4.5 muestra cómo la K de Yule es calculada para cada documento:

$$repetitividad(d_k) = 10^4 \left(\sum_{i=1}^{N} \frac{i^2 V(i,N)}{N^2} \right) - \frac{1}{N} \tag{4.5}$$

donde la N representa la longitud del documento y $V(i,N)$ es el número de palabras que ocurren i veces en el documento.

Por último, cabe añadir que este tipo de representación "semántica" favorece a los algoritmos de aprendizaje automático basados en prototipos (Zhixing *et al.*, 2010), debido a que produce vectores densos (prácticamente sin ceros) y con baja dimensionalidad. Así pues, la clasificación es llevada a cabo de forma muy rápida comparada con la tradicional BoT. Para realizar la clasificación, optamos por hacerlo buscando el vector más parecido. Por esta razón, hemos escogido el algoritmo de 1 vecino más cercano (*1NN, 1 Nearest Neighbor*, por sus siglas en inglés) utilizando la distancia Euclideana.

Análisis de complejidad de RDA

La construcción de los vectores de términos en la Sección 4.1.1 es una sumatoria de las frecuencias del término en cada documento, con respecto a cada autor. De esta forma, su complejidad es $O(dta)$ donde d es el número de total de documentos, t es el máximo número de términos diferentes en un documento y a es el número de autores. Para la representación de un documento, cada vector de término es sumado. Debido a que cada término es representado por a autores, la complejidad de representar un documento está dada por $O(at)$. De esta forma $O(dta)$ es la complejidad de representar todos los documentos en el conjunto de datos. En conclusión, la complejidad total de la representación RDA esta dada por $O(2dta)$, la cual queda como $O(dta)$ dónde por lo regular tendremos que $a < 100$.

4.2. Enfoques para la identificación de autor

En esta sección se explican los dos enfoques para llevar a cabo la combinación de atributos. El primer enfoque es referido como *Vista General*, en éste la representación es un vector por documento utilizando todos los atributos textuales que contienen los espacios. El segundo enfoque es el de *Vista Individual*, en éste se tienen distintas perspectivas (vectores) del documento para cada espacio de atributos. Las siguientes secciones muestran los detalles específicos de cada enfoque.

La idea fundamental es construir distintas perspectivas para un conjunto de documentos. Es decir, cada instancia se representa utilizando distintos espacios de atributos (e.g., n-gramas a nivel de caracter, palabras vacías, y signos de puntuación). Intuitivamente, se persigue la idea de que al tener más vistas de los documentos, exista al menos un espacio (o alguna combinación de espacios) en el que la discriminación entre autores sea mejor. Por ejemplo, en el conjunto de n-gramas de caracter, el estilo de un autor a_1 podría ser muy similar al del autor a_2, sin embargo su estilo podría variar en el espacio de palabras vacías. Las siguientes secciones muestran dos alternativas para desarrollar las ideas anteriores.

4.3. Espacios de atributos

Para facilitar la explicación del resto de este capítulo, primeramente establecemos lo que queremos decir con el término *espacio de atributos*. Cuando se menciona *espacio de atributos*, nos referimos a un conjunto de tipos de características textuales que tienen algo en común. Por ejemplo, el espacio "léxico" podría contener un subconjuntos de palabras vacías, bigramas de palabras y signos de

puntuación (comunes porque pertenecen a la misma familia de características). La idea central es utilizar un esquema de distintos espacios en donde internamente, cada espacio contenga conjuntos de atributos similares entre sí, pero distintos con respecto a los conjuntos de atributos de otros espacios; esto es para seguir la idea de complementarlos entre sí al representar los documentos.

Para determinar si dos conjuntos de atributos A y B comparten algo o son similares en cierta forma, proponemos hacerlo sobre la base del Apéndice B, el cual explica algunos criterios útiles para usar en conjunto dos o más tipos de atributos. Con estos criterios intentamos beneficiar el esquema de clasificación, a través de espacios de atributos que se complementen entre sí.

4.3.1. Enfoque de Vista General

Consiste en representar cada instancia como un solo vector. En este enfoque los atributos son la unión de los distintos vocabularios (conjuntos de atributos) contenidos en cada espacio. La idea principal de este enfoque es comprobar de la forma más simple que, usar distintos tipos de atributos es mejor que utilizarlos de forma individual. De esta forma, se cuenta con un solo vector que contiene los distintos tipos de atributos y se entrena un clasificador. La Figura 4.1 muestra las ideas anteriores.

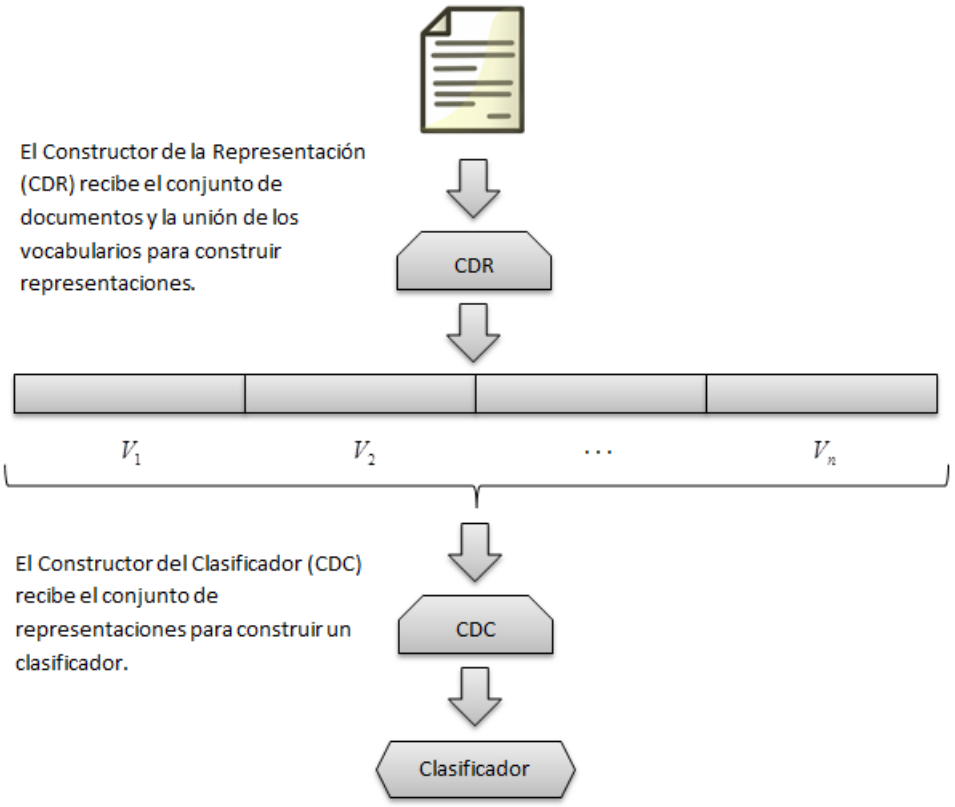

Figura 4.1: Enfoque de Vista General.

El Algoritmo 4.1 presenta el enfoque sencillo para construir un clasificador c_g empleando Vista General de atributos. En este enfoque, sea $\mathcal{D} = \{d_1, \ldots, d_k\}$ el universo de documentos del problema y $\mathcal{P}(\mathcal{D})$ el conjunto potencia, considere los siguientes elementos:

- Sea D_m el conjunto de documentos etiquetados disponibles para entrenar, tal que $D_m \subset \mathcal{D}$.
- El conjunto de distintos vocabularios $A_m = \{V_1, \ldots, V_n\}$ que se utilizarán para representar D_m.
- La función constructor de alguna representación CDR: $\mathcal{P}(\mathcal{D}) \times V \to S$. La cual con $CDR\Big(D_m, \bigcup_{V_j \in A_m} V_j\Big) = S_m$, donde S_m es la representación correspondiente a los documentos D_m utilizando como atributos los elementos de los vocabularios de A_m.
- La función constructor de algún clasificador CDC: $S \to C$. La cual con $CDC(S_m) = c_m$, dónde c_m es un clasificador construido a partir de las instancias S_m.

En el Algoritmo 4.1, la función CDR se utiliza para obtener una representación general S_m (línea 1). Posteriormente, CDC utiliza S_m para obtener un clasificador c_g (línea 2).

Algoritmo 4.1 Algoritmo simple de *Vista General*

Entrada: **CDC**, $\mathbf{V_m} = \bigcup_{V_j \in A_m} V_j$, **CDR**, $\mathbf{D_m}$

Salida: c_g

1: $S_m = CDR(D_m, V_m)$
2: Construir un clasificador general c_g utilizando $CDC(S_m)$.

En este algoritmo, para un conjunto de prueba, las instancias son representadas de la misma forma que las de entrenamiento y el clasificador c_g es el encargado de predecir la clase.

4.3.2. Enfoque de Vista Individual

Consiste en representar cada instancia como un conjunto de m vectores, donde cada vector representa una perspectiva del documento en un espacio de atributos específico. Posteriormente, se entrena un clasificador por cada vector i, la idea es que éste se especialice en ese tipo de características. La Figura 4.2 muestra las ideas del enfoque individual.

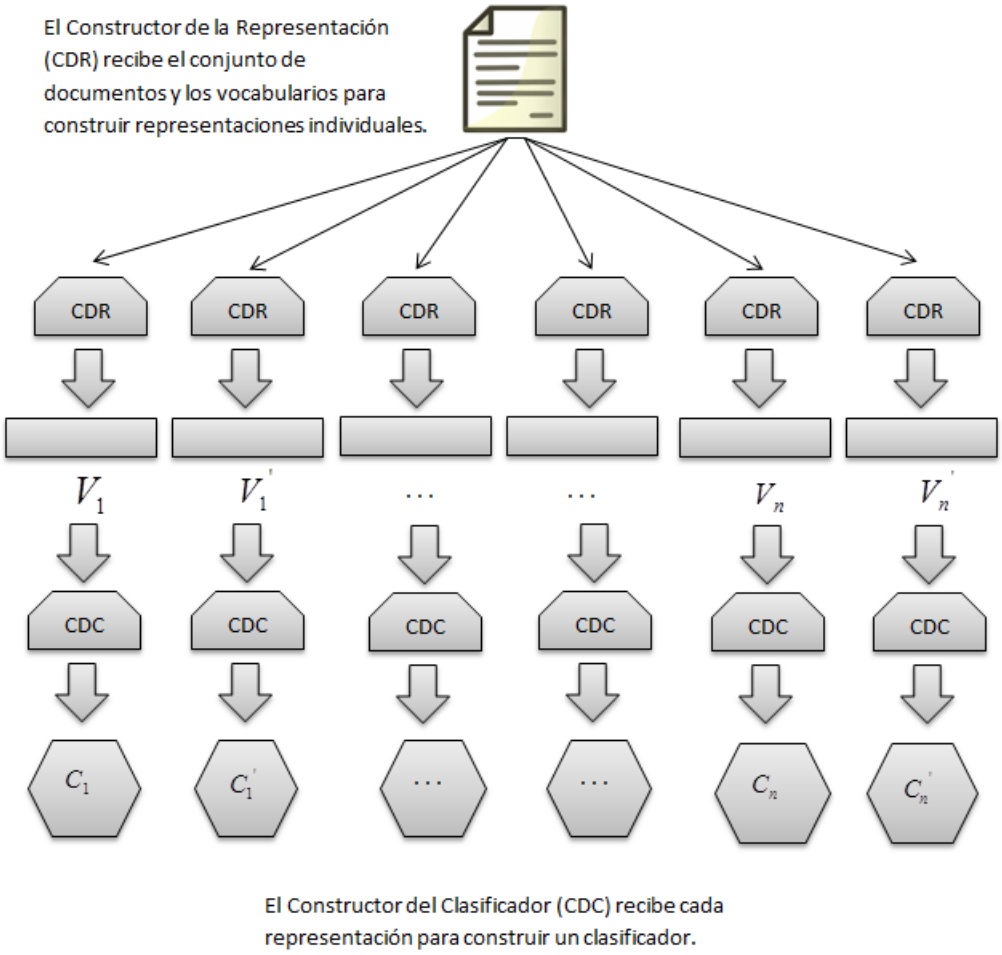

Figura 4.2: Enfoque de Vista Individual.

El Algoritmo 4.2 presenta el enfoque de ensambles considerando Vista Individual por tipo de atributo. En este enfoque, sea $\mathcal{D} = \{d_1, \ldots, d_k\}$ el universo de documentos del problema y $\mathcal{P}(\mathcal{D})$ el conjunto potencia, considere los siguientes elementos:

- Sea D_m el conjunto de documentos etiquetados disponibles para entrenar, tal que $D_m \subset \mathcal{D}$.
- El conjunto de distintos vocabularios $A_m = \{V_1, \ldots, V_n\}$ que se utilizarán para representar D_m.
- Un conjunto de distintos vocabularios ${A_m}' = \{{V_1}', \ldots, {V_n}'\}$ para incrementar diversidad, dónde:

$$\forall({V_i}' \in {A_m}'), \exists(V_i \in A_m) \left[\left(|{V_i}'| = |V_i| \right) \right.$$
$$\left. \wedge \left({V_i}' = \left\{ (X \cup Y) : (X \subset V_i) \wedge \left(Y \subset \bigcup_{V_j \in (A_m - V_i)} V_j \right) \right\} \right) \right] \quad (4.6)$$

- La función Constructor de Representación CDR: $\mathcal{P}(\mathcal{D}) \times A \rightarrow S$. La cual con $CDR(D_m, V_i) = S_{mi}$, dónde S_{mi} es la representación correspondiente a los documentos D_m utilizando como atributos los elementos de V_i.
- La función Constructor de Clasificador CDC: $S \rightarrow C$. La cual con $CDC(S_{mi}) = c_{mi}$, donde c_{mi} es un clasificador construido a partir de las instancias S_{mi}.

De forma simple, en el Algoritmo 4.2 el CDR se utiliza para obtener dos representaciones de D_m por cada tipo de atributo i: S_{mi} y ${S_{mi}}'$ (línea 3 y 4). La

Algoritmo 4.2 Algoritmo Propuesto de *Vista Individual*

Entrada: $\mathbf{CDC}$, $\mathbf{A_m}, \mathbf{A_{m\prime}}$, $\mathbf{CDR}$, $\mathbf{D_m}$
Salida: $\{c_{mi}, {c_{mi}}' | i = 1, \ldots, n\}$
1: **Para** $i = 1$ **hasta** n **hacer**
2: $S_{mi} = CDR(S_m, V_i)$
3: ${S_{mi}}' = CDR(S_m, {V_i}')$
4: Construir c_{mi} y ${c_{mi}}'$ utilizando $CDC(S_{mi})$ y $CDC({S_{mi}}')$ respectivamente.
5: $i++$
6: **Fin Para**

representación S_{mi} es construida con base en el vocabulario V_i. Mientras tanto, la representación ${S_{mi}}'$ es construida utilizando el vocabulario ${V_i}'$. Por último, CMD construye un clasificador c_{mi} para S_{mi} y otro ${c_{mi}}'$ para ${S_{mi}}'$.

Combinación de predicciones

Para llevar a cabo la clasificación se utiliza un método de votación Condorcet (vea explicación en Sección 2.2.2). La razón de utilizar una votación Condorcet radica en la forma de construir los clasificadores en el Algoritmo 4.2. En éste se puede observar que un clasificador c_{mi} es entrenado con las instancias representadas con el 100 % de los atributos del espacio, mientras otro clasificador ${c_{mi}}'$ es entrenado con las instancias representadas solo asegurando un cierto porcentaje del total de los atributos (60 % para nuestros experimentos), y el resto es tomado aleatoriamente de los otros espacios. Dada esta situación tenemos dos clasificadores especializados en un espacio de atributos ligeramente distinto. En este sentido, es más evidente que, en distintas predicciones emitan como candidato favorito al mismo autor. Dado este contexto, nos interesa no solo el candidato favorito, sino también la forma en que posicionó al resto de los candidatos. Esto también es importante para las instancias que para un cierto espacio suelen confundirse entre dos o más autores. Principalmente, debido a que es probable que dos clasificadores

emitan como candidato favorito a un cierto autor incorrecto, pero es más difícil que se hayan equivocado exactamente en toda una lista de predicciones.

Dada la situación anterior, es necesario adaptar al clasificador para proporcionar la lista de preferencias, para después construir la matriz de votación. La adaptación depende del clasificador, por ejemplo en un clasificador binario como SVM bajo un esquema de uno-contra-todos la matriz de voto puede ser obtenida de forma prácticamente directa (vea matriz Condorcet en Sección 2.2.2). Sin embargo, en un clasificador multi-clase como Naïve Bayes, la matriz de voto se podría construir a partir de una lista de autores ordenada en función de la probabilidad de pertenecer a cada clase. Para propósitos de esta tesis, utilizamos Weka (Hall *et al.*, 2009), que proporciona la funcionalidad anterior para todos sus clasificadores.

Capítulo 5

Experimentos y Resultados

En este capítulo presentamos la evaluación de los métodos propuestos. La Sección 5.1 explica el conjunto de datos utilizado y las pruebas de evaluación que se llevaron a cabo. Posteriormente, en las Secciones 5.2 y 5.3, mostramos a detalle la evaluación para la representación RDA y los métodos de ensambles respectivamente.

5.1. Metodología experimental

Para evaluar los métodos propuestos, hemos utilizado un subconjunto de 10 autores del corpus c50. Los autores utilizados son: *Alan Crosby, Alexander Smith, Benjamin KangLim, David Lawder, Jane Macartney, Jim Gilchrist, Marcel Michelson, Mure Dickie, Robin Sidel y Todd Nissen.* Este subconjunto de autores del corpus fue originalmente utilizado por Plakias y Stamatatos (2008) en (Plakias y Stamatatos, 2008). El corpus c50 está conformado por textos del Corpus Reuters Volumen 1 (Lewis *et al.*, 2004). El corpus c50 consta de 50 autores con documentos que pertenecen a la categoría CCAT (la cual trata noticias acerca de la industria).

Se utiliza la misma categoría en aras de reducir el factor temático, y centrar la evaluación en AA. Por último, cada autor tiene 50 documentos para entrenar y 50 documentos para probar.

Los experimentos que se han realizado son similares a los reportados en (Plakias y Stamatatos, 2008). Primeramente y con el propósito de simular escenarios realistas (Stamatatos, 2009), hemos construido diferentes conjuntos de entrenamiento. Tres de ellos están balanceados, tomando aleatoriamente 50, 20 y 10 documentos de entrenamiento por autor, y los otros tres están desbalanceados tomando aleatoriamente 2:10, 5:10 y 10:20 (donde $a : b$ significa, mínimo a y máximo b documentos por autor). De este modo, el desempeño de cada método es medido por la exactitud en la clasificación en todo el conjunto de prueba.

La representación RDA y los algoritmos de ensambles fueron construidos tal como se describe en el Capítulo 5. Además, cada experimento es el promedio de diez corridas, con el objetivo de tener suficientes datos para llevar a cabo pruebas estadísticas. Con respecto a esto último, hemos aplicado la prueba de los signos de Wilcoxon para cada resultado, obteniendo un 95 % de confianza estadística para los resultados de nuestros experimentos. En nuestras pruebas denotamos en negritas los resultados significativamente mejores. Cabe añadir que se han seleccionado los experimentos más relevantes que muestran las propiedades interesantes de los métodos propuestos. No obstante, en el Apéndice D están algunos otros experimentos que contribuyeron con la investigación.

5.2. Evaluación de la Representación Documento Autor

Para la evaluación de RDA se han llevado a cabo tres diferentes experimentos mostrados en la Sección 5.2.1. En el primero y en el segundo comparamos RDA contra BoT, el cual es un enfoque tradicional. Para ello utilizamos dos de los tipos de términos más efectivos en AA; palabras y n-gramas a nivel de caracter (Stamatatos, 2009). Cabe añadir que también comparamos RDA contra el Modelo de Espacio de Tensores (MET) (Plakias y Stamatatos, 2008), el cual ha sido evaluado utilizando el corpus c50. Por último, en el tercer experimento RDA es construido basado en una simple selección de atributos, para conseguir mejores resultados. En resumen, comparamos RDA contra los siguientes métodos:

- Bolsa de Términos (utilizando palabras y n-gramas de caracteres) clasificando con SVM y 1NN. Hemos utilizado SVM debido a que se ha mostrado que es efectivo para AA (Pavelec *et al.*, 2008) (Plakias y Stamatatos, 2008). También utilizamos 1NN debido a que nos permite mostrar cómo mejora su rendimiento cuando RDA es utilizado (ver argumento al final de la Sección 4.1.1).

- Modelo de Espacio de Tensores (utilizando n-gramas de caracteres), clasificando con Máquinas de Tensores de Soporte (MTS) (Plakias y Stamatatos, 2008). Nos comparamos contra este método por dos razones. La primera es que es un método que se centra en evaluar cómo la sola representación de tensores beneficia la AA. La segunda razón es que la evaluación la tiene bien establecida desde trabajos anteriores (Stamatatos, 2008) que se enfocan en el desbalance de los datos.

5.2.1. Experimentos

Experimento 1. RDA utilizando palabras

La Tabla 5.1 muestra los resultados del primer experimento utilizando 2500 palabras, truncadas con el algoritmo de M. F. Porter (Porter, 1980) (e.g., *playing*, *played*, *plays* son tomadas como *play*). Es importante mencionar que las palabras vacías se mantienen, con la idea de capturar información de estilo acerca de cómo los autores las distribuyen en sus documentos. Se puede observar que RDA supera la representación BoT cuando los datos están desbalanceados (un escenario realista). Creemos que esto es gracias a las relaciones capturadas en RDA, las cuales están representando a los documentos desde una perspectiva más allá de las palabras independientes.

Modelo	Instancias por autor					
	Balanceado			Desbalanceado		
	50	10	5	10:20	5:10	2:10
BoT - SVM	**79.6**	**71.6**	**65.8**	55.2	56.4	42.8
RDA - SVM	70.0	62.3	57.7	**61.2**	**59.6**	**46.1**
BoT - 1NN	37.0	49.6	36.6	30.8	39.4	34.2
RDA - 1NN	**70.8**	**65.5**	**61.1**	**66.2**	**62.0**	**53.3**

Tabla 5.1: RDA contra SVM utilizando las 2500 palabras más frecuentes.

Experimento 2. RDA utilizando n-gramas a nivel de caracter

La Tabla 5.2 muestra los resultados del segundo experimento, en este experimento comparamos BoT y RDA utilizando los 2500 3-gramas más frecuentes a

nivel de caracter. Además, comparamos RDA contra MTS utilizando la misma metodología que los autores de (Plakias y Stamatatos, 2008) siguieron en sus experimentos (igual desbalance de datos y mismo tipo y cantidad de términos); por ello, podemos decir que los resultados son directamente comparables.

Los resultados en Tabla 5.2 muestran como RDA supera a BoT en la mayoría de los conjuntos de datos desbalanceados. También puede observarse que al utilizar 3-gramas se mejora la exactitud respecto al experimento anterior, esto es debido a que los 3-gramas de caracter son características que pueden retener más información de estilo. Este experimento también nos permite mostrar que en la mayoría de los conjuntos de datos, RDA (especialmente RDA-1NN) es mejor que BoT-SVM y MET-MTS cuando se ejecutan bajo las mismas condiciones.

Modelo	Instancias por autor					
	Balanceado			Desbalanceado		
	50	10	5	10:20	5:10	2:10
BoT - SVM	**80.8**	**64.4**	48.8	**64.2**	62.4	51.0
RDA - SVM	72.1	63.1	**56.6**	62.1	**63.9**	**53.2**
BoT - 1NN	36.4	50.3	38.6	33.8	41.4	36.2
RDA - 1NN	**76.0**	**67.3**	**62.7**	**66.9**	**65.6**	**55.1**
MET - MTS	78.0	67.8	53.4	63.0	62.6	50.0

Tabla 5.2: Se compara RDA contra BoT y MET utilizando los 2500 3-gramas más frecuentes.

Experimento 3. RDA utilizando un umbral de frecuencia

Las Figuras 5.1 y 5.2 muestran los resultados del tercer experimento, así como una propiedad interesante de RDA; la cantidad de términos con la que se construye. En este experimento podemos observar que RDA puede ser mejorada con una sencilla selección de atributos.

En estos experimentos seleccionamos aquellos atributos con frecuencia igual o mayor que 3, 5, 7 y 10, en el conjunto de entrenamiento de cada experimento. La idea principal es explorar el comportamiento de RDA cuando se incrementa o se reduce la cantidad de términos, con el objetivo de mejorar la calidad de la representación. La Figura 5.1 y 5.2 muestra qué RDA puede ser notablemente mejorada por esta simple selección. En general, la mejor configuración de RDA fue con un umbral de frecuencia de 5.

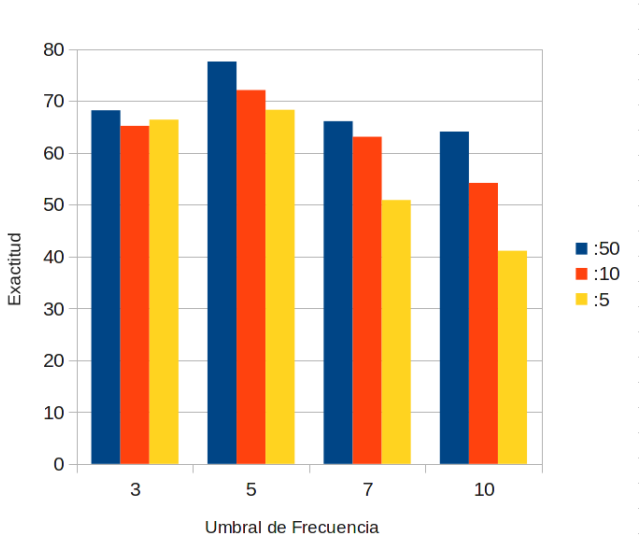

Figura 5.1: RDA con diferentes umbrales de frecuencia en los datos balanceados. Cada barra representa la exactitud de un experimento y una configuración.

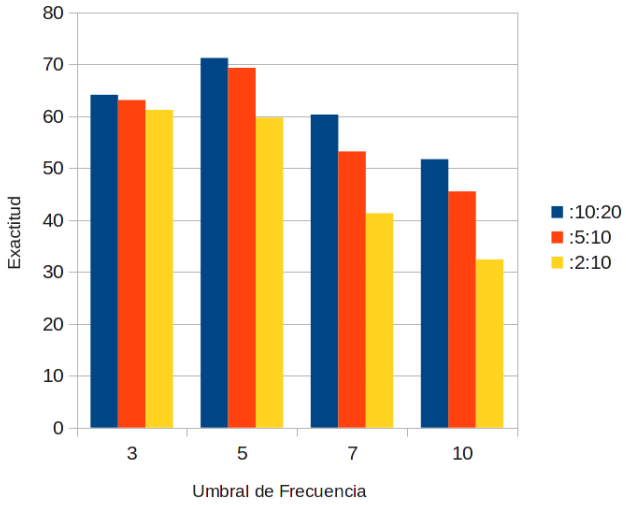

Figura 5.2: RDA con diferentes umbrales de frecuencia en los datos desbalanceados. Cada barra representa la exactitud de un experimento y una configuración.

5.2.2. Discusión de los resultados

Estos resultados muestran el buen desempeño de nuestra propuesta. Note que, especialmente cuando el corpus está desbalanceado o con pocos datos de entrenamiento, RDA supera a los demás métodos. También se ha mostrado que RDA proporciona un mejor rendimiento que los enfoques tradicionales y la representación MET, cuyos resultados están reportados en el estado del arte. También, analizando las Tablas 5.1 y 5.2 podemos observar como BoT reduce sus tasas de exactitud cuando los datos son escasos o cuando las clases son desbalanceadas; por otro lado, RDA parece ser menos sensible a conjuntos de datos pequeños y desbalanceados. Además, en contraste con un número fijo de términos para los experimentos, hemos mostrado como definir un umbral de frecuencia puede mejorar notablemente el rendimiento de RDA.

5.3. Evaluación del método de ensambles

En esta Sección presentamos una serie de experimentos utilizando la representación y los métodos de ensambles propuestos. Esto con el objetivo de mostrar que la combinación de distintos tipos de atributos es mejor que utilizarlos de manera individual. Para ello consideramos los espacios de atributos definidos en la Sección 5.3.1, y los utilizamos en los experimentos de la Sección 5.3.2 ; además, al igual que en la sección anterior, realizamos experimentos con 2500 atributos para cada espacio, primero de forma individual, y luego de forma conjunta mediante los algoritmos propuestos.

La Sección 5.3.2 presenta tres experimentos para evaluar los métodos de ensambles. En el primer experimento, se utilizan los algoritmos propuestos utilizando como base una representación de atributos convencional (BoT); esto para probar la idea de la combinación de atributos en la forma más simple y tradicional. En el segundo experimento, probamos los métodos utilizando la representación propuesta RDA; para apreciar el desempeño cuando se proporciona ayuda en la parte de la representación. Por último, el tercer experimento muestra la integración de la representación BoT y RDA; con el objetivo de proporcionarle a cada instancia dos tipos de perspectivas por atributo (una convencional y una de "segundo orden").

5.3.1. Espacios de atributos utilizados

Motivados por el primer criterio del Apéndice B proponemos el uso conjunto de cuatro distintos espacios de atributos: palabras, n-gramas a nivel de carácter, n-gramas a nivel de palabra, y estilo. Cabe mencionar que debido a su efectividad, los 3-gramas y las palabras son muy utilizados individualmente en trabajos de la literatura (Stamatatos y Widmer, 2005; Rokach, 2009; de Vel *et al.*, 2001). Es por

ello que también los hemos incluido dentro de los espacios, para complementarlos con otros tipos de atributos y mejorar la clasificación.

Cada uno de los espacios utilizados está constituido por los β elementos más frecuentes en los datos de entrenamiento. Lo último, para seguir la idea de que, entre más frecuente un elemento más variación de estilo puede capturar (Stamatatos, 2009). Cabe señalar que la mayoría de estos tipos de atributos no requieren de herramientas críticamente dependientes del lenguaje. En consecuencia, los atributos se extraen fácilmente para muchos tipos de documentos, dominios y lenguajes (salvo algunas excepciones como el Chino).

A continuación se presenta cada uno de los espacios utilizados, y los conjuntos de atributos que consideran (para una descripción a detalle vea el Apéndice A):

Espacios de características textuales utilizados	
Espacio	**Conjuntos de atributos que incluye**
Palabras	palabras, contracciones, abreviaturas, y palabras compuestas a través de guiones.
n-gramas nivel de caracter	3-gramas y 5-gramas
n-gramas nivel de palabra	2-gramas y 3-gramas considerando al texto como una secuencia de: palabras, contracciones, abreviaturas, y palabras compuestas a través de guiones.
Estilo	Signos de puntuación, longitud de palabras, colocaciones, palabras con más de una etiqueta POS y longitud de las oraciones según la cantidad de: elementos léxicos, palabras vacías, palabras no vacías y puntuación.

Tabla 5.3: Principales espacios de atributos considerados, cada uno conteniendo los $\beta = 2500$ términos más frecuentes

Tenga en cuenta que para elegir los atributos de cada espacio, además del criterio 1 se utilizó el criterio 3 (ver Apéndice B). Por ejemplo, para determinar los atributos del espacio de n-gramas de caracter se realizaron pruebas con $n = 1...15$. Con esto, se tomaron decisiones tales como: no usar 4-gramas debido a que discriminan de forma muy similar a los 3-gramas, o no usar 10-gramas por que ya no aportan información relevante.

5.3.2. Experimentos

Experimento 1. Método utilizando BoT

La Tabla 5.4 muestra los resultados del primer experimento, en éste nuestro objetivo es observar una situación particular; explorar si utilizar conjuntamente espacios de atributos ayuda en la identificación de autor. De la Tabla 5.4 se puede observar que los algoritmos propuestos obtienen resultados superiores en casi todos los casos.

El experimento Vista-General-1 utiliza como vocabulario los 2500 elementos más frecuentes por espacio. Este muestra resultados superiores a utilizar cualquier espacio de atributos de forma individual, aun cuando la dimensionalidad y la dispersión en la representación son mayores.

En el experimento de Vista-General-2 se emplea como vocabulario los 625 elementos más frecuentes por espacio (un total de 2500 atributos). Esto para evaluar utilizando vectores con la misma dimesionalidad que los utilizados por los otros espacios individuales. Note que, no se utiliza ningún algoritmo de selección de características debido a lo expuesto en la Sección 3.2. Cabe añadir que, seleccionar los N atributos más frecuentes no significa seleccionar a los mejores, o que todos estén capturando el estilo (Stamatatos, 2009). Por lo tanto, el enfoque de

Vista-General-2 con un vocabulario de 2500 elementos, muestra que una combinación simple de los espacios de atributos es mejor que utilizar cada uno de forma individual.

El experimento de Vista-Individual muestra también mejora notable con respecto a utilizar cualquier espacio de atributos de forma individual. Éste esperimento utiliza como vocabulario los 2500 términos más frecuentes por espacio. Sin embargo, el entrenar un clasificador para cada espacio de atributos ayuda a reducir el problema de la dimensionalidad. Además, pensamos que este experimento sigue obteniendo buenos resultados debido a que la mayoría de sus clasificadores son competitivos entre sí y cuentan con diversidad en sus predicciones individuales, lo cual mejora la predicción conjunta.

Modelo	Experimentos con diferentes atributos					
	Balanceado			Desbalanceado		
	50	10	5	10:20	5:10	2:10
	Exp 1. Usando SVM a 10 iteraciones					
Palabras	78.2	69.1	63.2	68.0	64.0	48.0
n-gramas de caracter	78.3	69.1	62.0	68.3	62.1	49.0
n-gramas de palabra	77.1	69.3	61.1	68.2	64.1	50.4
Estilo	47.2	34.1	27.2	36.4	32.5	30.1
Vista-General-1	**81.2**	**71.3**	**64.3**	**72.1**	**66.2**	50.3
Vista-General-2	**80.8**	**71.1**	**63.9**	**72.2**	**66.8**	**52.1**
Vista-Individual	**83.1**	**73.2**	**67.4**	**73.1**	**67.2**	**53.3**

Tabla 5.4: Utilización individual y conjunta de atributos utilizando BoT como representación base.

Experimento 2. Método utilizando RDA

La Tabla 5.5 muestra los resultados del segundo experimento. En este experimento se explora la combinación de espacios de atributos representados con RDA.

Modelo	Experimentos con diferentes atributos					
	Balanceado			Desbalanceado		
	50	10	5	10:20	5:10	2:10
	Exp 2. Usando 1NN a 10 iteraciones					
Palabras	76.0	70.1	65.7	69.6	65.9	54.8
n-gramas de caracter	76.0	67.3	62.7	69.0	65.6	55.1
n-grams de palabra	74.0	69.1	59.4	68.0	65.0	52.0
Estilo	51.6	42.1	36.7	41.8	37.6	34.8
Vista-General	75.8	66.2	56.6	69.0	58.9	50.7
Vista-Individual	**78.0**	**72.0**	**66.9**	**73.4**	**68.0**	**56.1**

Tabla 5.5: Utilización individual y conjunta de atributos utilizando RDA como representación base.

De la Tabla 5.5 se puede observar que para el algoritmo de Vista-General los resultados no son estadísticamente superiores que utilizar individualmente cada uno de los espacios. Esto es debido a que en RDA los vectores, que representan la relación de términos con autores, son construidos bajo un espacio de atributos específico. Es decir, estos vectores utilizan información a nivel clase (ver ecuación 4.1 de la Sección 4.1.1), para luego ser normalizados con respecto a los demás atributos (ver ecuación 4.2 de la Sección 4.1.1). Por lo tanto, los vectores RDA ya vienen muy concretos para un espacio específico, con escalas de valores muy diferentes de un espacio a otro. Además, la representación RDA comprime la

información de miles de atributos textuales en unos pocos atributos de relación, en este sentido una vez que son obtenidos, son muy sensibles a cualquier manipulación que se les aplique (e.g., normalizaciones), perjudicando la clasificación.

Por otro lado, en el experimento anterior el algoritmo Vista-General funcionó con BoT debido a que no existe ninguna compresión de atributos, es decir los valores de cada atributo están explícitos. Si bien es cierto que también algunos atributos están fuera de escala, cada uno representa una y solo una característica textual.

En cuanto al experimento de Vista-Individual, éste mantiene un buen desempeño frente a la utilización individual de cada espacio de atributos. Esto es debido a que las representaciones RDA no son mezcladas para un mismo clasificador. Sino más bien tenemos un clasificador especializado en cada espacio representado por RDA. Esto evita problemas de escala en los valores de un espacio a otro, y cada clasificador puede enfocarse mejor en su problema.

Experimento 3. Método utilizando BoT + RDA

La Tabla 5.6 muestra los resultados del tercer experimento, en el que se utiliza una combinación de ambas representaciones: BoT y RDA. El objetivo es explorar si proporcionarle a cada documento perspectivas de atributos específicos (BoT) y atributos de relaciones (RDA) ayuda en la identificación de autores. A partir de la Tabla 5.6 podemos observar que ambos enfoques de combinación ayudan en la identificación de autor. En particular el algoritmo de Vista-Individual obtiene resultados mejores que utilizar individualmente cada espacio. Parte de la explicación es similar a la del Experimento 2 (clasificadores especializados). Sin embargo, en este caso la inclusión de la representación BoT beneficio aún más. Esto es debido a que, utilizar BoT y RDA incrementa la diversidad al tener los

espacios representados de distinta forma, y además se mejora la diversidad de opinión al incrementar el número de clasificadores totales.

Modelo	Experimentos con diferentes atributos					
	Balanceado			Desbalanceado		
	50	10	5	10:20	5:10	2:10
	Exp 3. Usando SVM-1NN a 10 iteraciones					
Palabras	78.6	68.6	62.8	70.9	63.0	49.4
n-gramas de caracter	78.8	69.0	63.8	70.2	61.9	48.3
n-gramas de palabra	77.2	68.7	62.6	71.5	63.4	51.0
Estilo	47.6	36.9	32.2	38.6	34.4	30.8
Vista-General	**81.6**	**72.0**	**65.9**	**72.2**	**66.3**	**53.9**
Vista-Individual	**83.4**	**72.9**	**67.6**	**72.9**	**68.5**	**56.6**

Tabla 5.6: Utilización individual y conjunta de atributos utilizando BoT y RDA como representación base.

5.3.3. Discusión de los resultados

La Tabla 5.7 muestra los resultados de los enfoques utilizados para la combinación de atributos. En ésta se puede apreciar que en general combinar atributos beneficia a la AA. No obstante, con respecto a la exactitud, los enfoques propuestos para combinarlos siguen una tendencia similar que su representación base. Es decir, con una representación BoT tiene buen rendimiento sobre datos balanceados. Mientras que con una representación RDA se consigue mejorar al tener datos desbalanceados. En este contexto, el experimento $BoT + RDA$ obtiene resultados

que muestran que combinar ambas representaciones es mejor que utilizarlas individualmente; no en el sentido de obtener resultados superiores, sino en el sentido de conseguir un método que logra juntar las buenas propiedades de cada representación y no es muy afectado por las malas. Es decir, un método con un desempeño similar a BoT en la parte balanceada, y similar a RDA en la parte desbalanceada. Por último, concluimos que se incrementa la exactitud en $BoT + RDA$ debido a que la diversidad en el Ensamble 4.2 se incrementa con la manipulación de más espacios de atributos y más votos de los clasificadores.

Representación	Modelo	Experimentos con diferentes atributos					
		Balanceado			Desbalanceado		
		50	10	5	10:20	5:10	2:10
BoT	Vista-General	81.2	71.3	64.3	72.1	66.2	50.3
	Vista-General-2	80.8	71.1	63.9	72.2	66.8	52.1
	Vista-Individual	**83.1**	**73.2**	**67.4**	**73.1**	67.2	53.3
RDA	Vista-General	75.8	66.2	56.6	69.0	58.9	50.7
	Vista-Individual	78.0	72.0	66.9	**73.4**	**68.0**	**56.1**
BoT + RDA	Vista-General	**81.6**	**72.0**	**65.9**	**72.2**	**66.3**	**53.9**
	Vista-Individual	**83.4**	**72.9**	**67.6**	**72.9**	**68.5**	**56.6**

Tabla 5.7: Resultados generales.

5.4. Consideraciones adicionales

Durante la evaluación encontramos que es posible configurar parámetros de cada representación y cada sistema de clasificación para producir mejores resul-

tados. Sin embargo, en aras de probar y exponer claramente las ideas presentadas en esta tesis decidimos utilizar un conjunto de parámetros fijos para construir las representaciones y clasificadores. Con esto logramos i) evitar presentar resultados demasiado sobreajustados a los datos, y ii) mantener tan simple como sea posible, un sistema ya de por si complejo.

En esta tesis comparamos RDA contra la representación MET del trabajo (Plakias y Stamatatos, 2008). Se utilizó a MET y solo el corpus c50 como referencia debido a que, la forma de evaluarse en AA se tiene bien definida desde trabajos anteriores (Stamatatos, 2008). Posterioremente, probamos dos alternativas para explorar la idea de la combinación de atributos; esto es, evaluando espacios de forma individual y conjunta.

Existe otro trabajo en AA (Escalante *et al.*, 2011), el cual se evalúa sobre el mismo corpus. Con respecto a este trabajo, algunos de nuestros resultados llegan a ser similares (ver Tabla 5.8). No obstante, no lo consideramos como referencia en nuestras tablas anteriores debido a dos inconvenientes: i) su alto coste computacional y la falta pruebas estadísticas que respalden los resultados, y ii) para alcanzar buenos resultados el trabajo ajusta una serie de parámetros dependientes de su sistema (Lebanon *et al.*, 2007). Con respecto al punto (i), obtener las corridas necesarias es uno de los problemas más importantes, ya que una ejecución requiere de varias horas y considerables recursos computacionales. Lo anterior, implica que para tener el promedio de 10 corridas con palabras y 10 corridas con n-gramas en cada conjunto de entrenamiento, se necesita programar una plataforma para ejecutar consecutivamente varias instancias del método de Escalante *et al.* (2011), además de meses de experimentación con dicha plataforma. En este contexto, realizar los experimentos para compararnos directamente está fuera de los alcances de esta tesis. Con respecto al punto (ii), se necesitan experimentos adicionales

dedicados solo a explorar los parámetros de $BoT + RDA$ para encontrar ajustes óptimos de: umbrales de frecuencia para cada representación, parámetros de clasificadores con distintos *kernels* para SVM, y distintos valores de k para el algoritmo kNN. En este sentido, pensamos que esto sobre ajustaría demasiado nuestro método a los datos (distanciándonos del objetivo de la tesis). Es por ello que nos parece más interesante explorar el beneficio de utilizar RDA como representación base del método de (Escalante *et al.*, 2011), lo cual es considerado a realizarse para trabajo futuro.

Representación	Modelo	Experimentos con diferentes atributos					
		Balanceado			Desbalanceado		
		50	10	5	10:20	5:10	2:10
	Exp 3. Usando SVM (1 corrida)						
LOWBOW	Palabras	82.0	72.8	69.2	74.1	70.7	66.6
(Escalante *et al.*, 2011)	n-gramas	86.4	82.2	80.6	82.2	80.5	77.8
	Exp 3. Usando nuestros ensambles (1 corrida)						
BoT + RDA	Vista-General	87.6	81.7	75.9	78.2	75.8	70.9
	Vista-Individual	87.2	78.9	79.1	73.4	78.3	74.1

Tabla 5.8: Resultados sin pruebas de significancia estadística de BoT+RDA contra los resultados reportados para LOWBOW

Capítulo 6

Conclusiones y trabajo futuro

En la tarea de AA la selección de los atributos de estilo y la representación de los documentos son dos procedimientos fundamentales que influyen en el desarrollo y desempeño de los métodos. Sin embargo, la tarea de seleccionar y representar los atributos no ha sido una tarea sencilla. En este sentido, diversas soluciones se han planteado, una de las más comunes es la utilizacion de los N términos más frecuentes. En cuanto a la representación, la mayoría utiliza BoT (o variantes), para después clasificar con SVM. Algunos otros, van más allá de la BoT y utilizan representaciones para capturar relaciones entre los términos (Plakias y Stamatatos, 2008). Dado este contexto, en esta tesis presentamos una solución alternativa al problema de la AA.

En este Capítulo se expone un resumen de las ideas presentadas, las conclusiones obtenidas, las principales aportaciones, y algunas ideas que pudieran desarrollarse en el trabajo futuro.

6.1. Conclusiones

En la presente tesis se desarrolló una investigación para AA en la que destaca lo siguiente: la representación RDA, la combinación de distintos tipos de atributos y su integración en un sistema de clasificación.

En cuanto a la representación RDA, hemos explorado una nueva alternativa para caracterizar documentos en AA. Ésta representa a los documentos con un vector de atributos de segundo orden, para determinar cómo un texto está relacionado con los autores bajo un cierto espacio de características textuales. Al menos en lo que conocemos, ésta es la primera vez que una técnica similar a RDA es explorada para la tarea de AA. En este sentido, encontramos que representaciones como RDA pueden conservar información relevante para AA. Esto ayuda a mantener buenas tasas de clasificación, incluso cuando el corpus está desbalanceado, lo cual es un escenario realista. Pensamos que esto es debido a que, las relaciones entre términos, autores y la riqueza del vocabulario del contexto, logra preservar el estilo de escritura en la representación final. También reportamos mejores resultados que los enfoques convencionales en datos desbalanceados, así como resultados superiores contra el método MTS (Plakias y Stamatatos, 2008). De esta forma, hemos demostrado la alta calidad que RDA puede conseguir en la representación de los documentos. Con todo ello, concluimos que RDA es una representación factible y afectada en menor medida por el desbalance de los datos. Además, RDA puede ser utilizada en AA para descubrir un nuevo conjunto de atributos (atributos de segundo orden) que representan relaciones entre documentos y autores.

En cuanto a los espacios de atributos, utilizamos algunos que han sido empleados individualmente (e.g., 3-gramas de caracter y palabras). Además, definimos otro espacio de atributos de estilo, dónde incluimos algunos atributos clásicos

(e.g., longitud de las palabras) y definimos algunos otros tales como medir la longitud de las oraciones basándose en diferentes criterios (e.g., signos de puntuación, palabras vacías, etc.). Para integrar esto en un sistema identificación de autores, construimos representaciones de documentos que consideren los distintos tipos de atributos. En cuanto a la representación de los documentos utilizamos BoT y RDA, individual y conjuntamente. Posteriormente, para probar la mejora en la identificación presentamos dos alternativas. La primera es *Vista-General*, la cual consiste en mezclar los vocabularios de los espacios para construir una representación sencilla. La segunda alternativa es *Vista-Individual*, ésta tiene la idea de construir un ensamble con un clasificador para cada espacio de atributos. Presentamos experimentos con ambos enfoques y encontramos que en general combinar los espacios de atributos adecuados ayuda en la AA. Pensamos que esto se debe principalmente a que las diferentes perspectivas de los documentos y clasificadores especializados logran capturar con más detalle el estilo de los documentos de los autores.

En conclusión, hemos estudiado una representación exitosa para AA que tiene el potencial de ser utilizada en diferentes maneras, especialmente porque produce atributos con un alto nivel de representatividad en vectores densos con baja dimensionalidad y un bajo coste computacional. También encontramos que ir más allá de la sola selección de n-gramas de caracter y palabras beneficia la AA, ya que se pueden conseguir conjuntos de atributos que se complementen entre sí. De esta forma, los documentos pueden ser discriminados en uno o más espacios de atributos. Además, incrementar los espacios de atributos, las representaciones utilizadas y el número de clasificadores, beneficia la diversidad en el aprendizaje y la votación de los ensambles.

6.2. Trabajo futuro

A continuación se plantean las siguientes ideas como trabajo futuro, con el fin de que se pueda continuar investigando aspectos relacionados con la presente tesis.

- Explorar la utilización de RDA como representación base de otros métodos de clasificación de AA. Así como probar los atributos construidos por RDA en conjunto con otras representaciones del estado del arte.
- Explorar el uso de RDA y los métodos de combinación de atributos bajo otras condiciones; por ejemplo, evaluar RDA incrementando el número de autores, utilizando documentos de distintas longitudes (e.g., con documentos cortos), o en dominios distintos (e.g., AA en blogs, foros, etc.)
- Explorar el uso de estos atributos de segundo orden en la AA *crosslingüe*. En este contexto, sería interesante estudiar si la RDA produce vectores similares para los documentos pertenecientes a un mismo autor en distintos idiomas.

Apéndices

Apéndice A

Espacios de atributos

A continuación se presentan los espacios de atributos utilizados para la construcción de los ensambles de clasificadores. En la siguiente lista, el * indica que el atributo puede ser extraído fácilmente en textos de distintos idiomas.

1. **Conjunto de palabras***: palabras, contracciones, abreviaturas, y palabras compuestas a través de guiones
2. **n-gramas nivel de caracter***: 3-gramas y 5-gramas
3. **n-gramas nivel de palabra***: bi-gramas y tri-gramas
4. **Atributos de estilo**:
   - **Signos de puntuación***
   - **Longitud de palabras***: consiste en extraer como atributos etiquetas que representen la longitud de las palabras. Por ejemplo, el texto "La investigación es muy divertida!!", produce la cadena "token_len{2} token_len{13} token_len{2} token_len{3} token_len{9}".
   - **Longitud de oraciones**: consiste en extraer etiquetas que representen la longitud de las oraciones. En donde para cada oración se extrae su longitud en términos de cada uno de los siguientes elementos:

  - **elementos léxicos***: palabras, puntuación, o abreviaturas.
  - **palabras vacías***: artículos, preposiciones, adverbios, etc.
  - **palabras no vacías**
  - **puntuación***: puntos, comas, signos de interrogación, comillas, etc.

- **Colocaciones de (Bigramas y Trigramas)***: consiste en extraer pares y tercias de colocaciones utilizando como elemento base cada uno de los siguientes atributos.
  - **palabras***
  - **palabras vacías***
  - **puntuación***
- **Palabras con más de una etiqueta POS:** todas aquellas palabras que pueden ser utilizadas en más de una forma al escribir. Por ejemplo, la palabra en inglés *lay* puede ser utilizada al menos como adjetivo, o verbo.

Apéndice B

Criterios para formar espacios de atributos

Para efectos de esta tesis proponemos los siguientes tres:

1. **Criterio de familias de características textuales:** Es decir, usar los elementos de A y B en un mismo espacio, sólo si A y B pertenecen a la misma familia de características textuales. Por ejemplo, agrupar los conjuntos de 1-gramas y 3-gramas en un mismo espacio llamado "Conjunto de características basadas en caracteres".

2. **Criterio específico del dominio:** Es decir, agrupar las características de acuerdo a una característica específica del dominio de documentos. Por ejemplo, los errores de escritura pueden ser muy valiosos en un dominio de mensajes de correo electrónico, *foros*, o redes sociales. En este contexto, un "Conjunto de errores gramaticales" agruparía los conjuntos de atributos denoten uso incorrecto de mayúsculas, errores ortográficos, o errores de estructura, por mencionar algunos.

3. **Criterio cuantificable**: Este consiste en definir un método, o prueba que proporcione valores específicos para determinar si un conjunto de atributos A es similar a un conjunto de atributos B. Por ejemplo una prueba simple es que, sobre el conjunto de entrenamiento, se evalúen dos pruebas de clasificación. La primera y segunda prueba utilizan como atributos (vocabulario) los elementos de A y B respectivamente. Si a partir de las pruebas de clasificación al menos un α porcentaje de los aciertos son idénticos, entonces se determina que los conjuntos de atributos A y B son similares, y por lo tanto se agrupan en un mismo espacio. Note que, en este ejemplo el valor α sirve para controlar que tan similares deben ser los atributos, de tal forma que continuen contribuyendo a la diversidad.

Apéndice C

Artículo publicado

El siguiente es un artículo derivado de esta tesis:

- *A New Document Author Representation for Authorship Attribution.* Adrián Pastor López-Monroy, Manuel Montes-y-Gómez, Luis Villaseñor-Pineda, J. Ariel Carrasco-Ochoa and José Fco. Martínez-Trinidad Lecture Notes in Computer Science, Volume 7329, pp. 283-292, 2012.

Apéndice D

Experimentos adicionales

A continuación se presentan algunos experimentos con BoT y RDA utilizando los clasificadores Naïve Bayes y Random Forest. La evaluación se realizó igual que en (Plakias y Stamatatos, 2008) utilizando el corpus c50, los 2500 términos más frecuentes por espacio, y utilizando la exactitud como medida de clasificación. Note que la tendencia de combinar atributos se mantiene en casi todos los casos de los algoritmos Vista General y Vista Individual.

Modelo	Experimentos con diferentes atributos					
	Balanceado			Desbalanceado		
	50	10	5	10:20	5:10	2:10
	Exp 1. Usando Random Forest-BoT					
Palabras	71.6	65.5	57.8	62.6	55.6	41.3
n-gramas de palabra	74.0	62.5	51.5	60.8	54.8	41.7
n-gramas de caracter	74.4	61.8	51.9	65.0	56.5	41.3
Estilo	50.8	39.2	34.0	41.0	35.1	30.8
Vista-General	**76.6**	**67.8**	**60.0**	**66.1**	**55.2**	**41.9**
Vista-Individual	**77.8**	**69.5**	**61.7**	**67.9**	**56.4**	**42.4**
	Exp 2. Random Forest-RDA					
Palabras	72.4	51.9	29.1	59.3	39.4	33.2
n-gramas de palabra	73.6	53.4	30.1	59.2	42.1	31.7
n-gramas de caracter	74.6	55.4	31.1	60.2	43.1	31.7
Estilo	53.2	41.7	33.5	43.5	38.5	35.0
Vista-General	75.6	56.2	29.5	**62.0**	42.0	33.9
Vista-Individual	**76.9**	**60.1**	**36.5**	**64.7**	**46.3**	**38.7**
	Exp 3. Usando Random Forest-BoT+RDA					
Palabras	70.4	64.3	58.0	65.5	59.3	42.5
n-gramas de palabra	74.2	61.9	52.3	65.0	52.1	40.1
n-gramas de caracter	73.8	62.5	53.3	64.8	53.8	41.5
Estilo	54.6	44.9	34.2	44.8	40.2	32.2
Vista-General	**76.8**	**66.3**	**60.0**	**68.1**	**63.5**	**45.1**
Vista-Individual	**78.6**	**68.7**	**62.9**	**69.4**	**65.1**	**47.8**

Tabla D.1: Algunos experimentos con uso individual y conjunta de atributos utilizando BoT y RDA como representación base y Random Forest como clasificador.

Modelo	Experimentos con diferentes atributos					
	Balanceado			Desbalanceado		
	50	10	5	10:20	5:10	2:10
	Exp 1. Usando Naïve Bayes-BoT					
Palabras	72.0	62.0	49.5	57.5	50.9	39.1
n-gramas de palabra	72.4	57.4	46.4	54.8	44.3	32.3
n-gramas de caracter	72.2	59.8	47.9	56.4	48.6	40.0
Estilo	45.2	29.8	22.6	34.0	26.0	22.8
Vista-General	**74.4**	**64.6**	**55.5**	**59.5**	**52.9**	**35.9**
Vista-Individual	**75.6**	**56.4**	**55.0**	**61.4**	**55.0**	**40.4**
	Exp 2. Usando Naïve-Bayes-RDA					
Palabras	71.0	64.2	49.4	66.7	55.9	46.1
n-gramas de palabra	70.2	64.4	48.1	65.5	57.7	43.9
n-gramas de caracter	72.6	63.0	49.0	66.1	56.6	46.3
Estilo	52.2	42.7	32.6	46.8	39.0	31.6
Vista-General	72.6	63.5	43.8	**68.9**	**59.2**	**49.5**
Vista-Individual	**74.8**	**66.5**	50.6	**68.4**	**61.2**	**51.7**
	Exp 3. Usando Naïve-Bayes-BoT+RDA					
Palabras	72.0	61.4	51.2	58.2	51.3	38.1
n-gramas de palabra	72.4	55.7	46.1	52.6	43.5	32.2
n-gramas de caracter	72.2	56.3	47.1	56.9	50.4	40.4
Estilo	48.0	29.7	22.6	34.5	25.6	24.9
Vista-General	**73.4**	**63.9**	**53.6**	**61.7**	**55.4**	**44.3**
Vista-Individual	**76.0**	**65.2**	**54.8**	**63.1**	**56.1**	**44.4**

Tabla D.2: Algunos experimentos con uso individual y conjunta de atributos utilizando BoT y RDA como representación base y Naïve Bayes como clasificador.

Referencias

Abbasi, A., y Chen, H. (2008). Writeprints: a stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems*, *26*, Article 7, 29p.

Argamon, S., y Juola, P. (2011). Overview of the international authorship identification competition at pan-2011. *Notebook for PAN at CLEF*.

Argamon, S., Whitelaw, C., Chase, P., Hota, S. R., Garg, N., y Levitan, S. (2007). Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, *58*, 802–822.

Breiman, L. (1996). Bagging predictors. *Mach Learn*, *24*, 123–140.

Breiman, L. (2001). Random forests. *Mach Learn*, *45*, 5–32.

Brown, G., Wyatt, J., Harris, R., y Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Inf. Fusion*, *6*, 5–20.

Cai, D., He, X., Wen, J., Han, J., y Ma, W. (2006). Support tensor machines for text categorization. *Technical report, UIUCDCS-R-2006-2714, University of Illinois at Urbana-Champaign*.

de Vel, O., Anderson, A., Corney, M., y Mohay, G. (2001). Mining e-mail content for author identification forensics. *ACM SIGMOD Record*, *30*, 55–64.

Deerwester, S. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*, 391–407.

Escalante, H. J., Solorio, T., y Gómez, M. Montes-y. (2011). Local histograms of character n-grams for authorship attribution. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 288–298.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, *3*, 1289–1305.

Frantzeskou, G., Stamatatos, E., Gritzalis, S., Chaski, C. E., y Howald, B. (2007). Identifying authorship by byte-level n-grams: the source code author profile (scap). *Int. Journal of Digital Evidence*, *6*, 18p.

Freund, Y., y Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Machine learning: proceedings of the thirteenth international conference*, 325–332.

Gabrilovich, E., y Markovitch, S. (2009). Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, *34*, 443–498.

Gamon, M. (2004). Linguistic correlates of style: Authorship classification with deep linguistic analysis features. *Proceedings of the 20th International Conference on Computational Linguistics*, 611–617.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., y Witten, I. H. (2009). The weka data mining software: An update. *ACM SIGKDD Explorations*, *11*, 10–18.

Houvardas, J., y Stamatatos, E. (2006). N-gram feature selection for author identification. *In Proceedings of the 12th International Conference on Artificial Intelligence. LNCS*, *4183*, 77–86.

Kern, R., Seifert, C., Zechner, M., y Granitzer, M. (2011). Vote/veto meta-classifier for authorship identification. *Notebook for PAN at CLEF 2011*.

Koppel, M., Akiva, N., y Dagan, I. (2006). Feature instability as a criterion

for selecting potential style markers. *Journal of the American Society for Information Science and Technology*, *57*, 1519–1525.

Koppel, M., y Schler, J. (2003). Exploiting stylistic idiosyncrasies for authorship attribution. *Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*, 69–72.

Koppel, M., y Schler, J. (2004). Authorship verification as a one-class classfication problem. *Proceedings of the 21st International Conference on Machine Learning*, 7p.

Koppel, M., Schler, J., Argamon, S., y Messeri, E. (2006). Authorship attribution with thousands of candidate authors. *SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 659–660.

Krogh, A., y Vedelsby, J. (1995). Neural network ensembles, cross validation and active learning. *Adv. Neural Inf. Process Syst.*, *7*, 231–238.

Kuncheva, L. (2005). Combining pattern classifiers. *Wiley Press, New York*, 241–259.

Lebanon, G., Mao, Y., y Dillon, J. (2007). The locally wighted bag of words framework for document representation. *Journal of Machine Learning Research*, *8*, 2405–2441.

Lewis, D., Yang, Y., Rose, T., y Li., F. (2004). Rcv1: a new benchmark collection for text categorization research. *Journal of Machine Learning Research*, *5*, 361–397.

Maimon, O., y Rokach, L. (2002). Improving supervised learning by feature decomposition. *Proceedings of foundations of information and knowledge systems, Salzan Castle, Germany*, 178–196.

Miranda-García, A., y Calle-Martín, J. (2005). Yule's k characteristic k revisited.

Language Resources and Evaluation, *39*, 287–294.

Montague, M., y Aslam, J. A. (2002). Condorcet fusion for improved retrieval. *Proceedings of the eleventh international conference on Information and knowledge management, ACM*, 538–548.

Mosteller, F., y Wallace, D. L. (1964). Inference and disputed authorship: The federalist. *Addison-Wesley*.

Pavelec, D., Justino, E., Batista, L. V., y Oliveira, L. S. (2008). Author identification using writer-dependent and writer-independent strategies. *In Proceedings of the 2008 ACM Symposium on Applied Computing - SAC08*, 414–418.

Plakias, S., y Stamatatos, E. (2008). Tensor space models for authorship attribution. *In Proc. of the 5th Hellenic Conference on Artificial Intelligence (SETN'08), LNCS*, *5138*, 239–249.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*, 130–137.

Provost, F. J., y Kolluri, V. (1999). A survey of methods for scaling up inductive learning algorithms. *Proceeding of 3rd international conference on knowledge discovery and data mining*, *3*, 131–139.

Rokach, L. (2009). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*, 1–39.

Sanderson, C., y Guenter, S. (2006). Short text authorship attribution via sequence kernels, markov chains and author unmasking: An investigation. *Proceedings of the International Conference on Empirical Methods in Natural Language Engineering*, 482–491.

Schler, J., Koppel, M., y Argamon, S. (2009). Computational methods in authorship attribution. *Journal of the American Society for Information Science*, *60*, 9–26.

Solorio, T., Pillay, S., Raghavan, S., y Gómez, M. Montes-y. (2011). Modality specific meta features for authorship attribution in web forum posts. *In Proceedings of the 5th International Joint Conference on Natural Language Processing*, 156–164.

Stamatatos, E. (2008). Author identification: Using text sampling to handle the class imbalance problem. *Information Processing and Management*, *44*, 790–799.

Stamatatos, E. (2009). A survey on modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, *60*, 538–556.

Stamatatos, E., y Widmer, W. (2005). Automatic identification of music performers with learning ensembles. *Artificial Intelligence*, *165*, 37–56.

Tumer, K., y Ghosh, J. (1996). Error correlation and error reduction in ensemble classifiers. *Connection science, special issue on combining artificial neural networks: ensemble approaches*, *8*, 385–404.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*, 241–259.

Zhixing, L., Zhongyang, X., Yufang, Z., Chunyong, L., y Kuan, L. (2010). Fast text categorization using concise semantic analysis. *Pattern Recognition Letters*, *32*, 441–448.