UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN, MENCIÓN EN TELEMÁTICA

Desarrollo e implementación de un algoritmo basado en *text mining* aplicado a la atribución de autoría

Autor:
D. Alejandro Pedrosa Antón
Tutor:
D. Jesús Poza Crespo
Valladolid, 9 de Septiembre de 2014

TÍTULO:	**Desarrollo e implementación de un algoritmo basado en *text mining* aplicado a la atribución de autoría**
	Development and implementation of an algorithm based in text mining techniques for authorship attribution
AUTOR:	**D. Alejandro Pedrosa Antón**
TUTOR:	**D. Jesús Poza Crespo**
DEPARTAMENTO:	**Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL

PRESIDENTE:	**Dña. María García Gadañón**
VOCAL:	**D. Jesús Poza Crespo**
SECRETARIO:	**D. Carlos Gómez Peña**
SUPLENTE:	**D. Roberto Hornero Sánchez**
SUPLENTE:	**D. Miguel López Coronado**

FECHA:	**9 de Septiembre de 2014**
CALIFICACIÓN:	

A mis padres, mi hermano Pedro y mis abuelos Leopoldo y Manolita

Resumen

El objetivo del presente Trabajo Fin de Grado es la implementación de un algoritmo basado en métodos de *text mining* que permita solventar el problema de la decisión de la autoría en textos españoles.

La principal motivación que se persigue es el uso de dos elementos estilométricos, los *TAGS* y las palabras funcionales (*function words*, FW) que permitan identificar correctamente el estilo intrínseco de cada autor. De este modo, se pretende desarrollar una plataforma que dé solución al problema que se quiere resolver: la atribución de autoría.

Para su desarrollo se han utilizado tres modelos basados en inteligencia artificial: máquinas de vectores de soporte (*support vector machine,* SVM), una red neuronal de tipo perceptrón multicapa (*multilayer perceptron*, MLP) y *Näive Bayes.* Estos algoritmos se han entrenado mediante la combinación de dos indicadores de estilo: las palabras funcionales y los *TAGS*. Dichos elementos se caracterizan por no presentar una naturaleza dependiente del contexto y por ello escapan de la manipulación por parte de la consciencia humana. Asimismo, se ha estudiado el efecto que ocasiona el uso del análisis de componentes principales (PCA) que permite la reducción de la dimensionalidad del espacio de características del problema. Por último, se ha utilizado un banco de 6 autores españoles contemporáneos para el conjunto de las pruebas, consiguiendo unos resultados de un 92.65% de precisión para experimentos realizados con dos sujetos.

Con el presente Trabajo de Fin de Grado se concluye que efectivamente las características analizadas son útiles para el contexto del problema de la atribución de autoría y que además su combinación permite obtener mejores precisiones. Asimismo, el algoritmo implementado consigue precisiones de hasta el 92.65%, mejorando resultados de estudios previos en el ámbito español.

Palabras Clave

Atribución de autoría, minería de textos, inteligencia artificial, análisis de componentes principales, validación cruzada, clasificación de autores, palabras funcionales, TAGS, normalización, combinación de modelos.

Abstract

The aim of this project is the implementation of an algorithm based on text mining methods, useful to address the problem of authorship attribution in the context of Spanish documents.

The main motivation is the analysis of two style indicators, TAGS and function words, which can provide accurate and reliable results about the intrinsic author's style. Thereby, this study is aimed at developing a software tool, helpful to address the authorship attribution.

In order to address this problem, three artificial intelligence models are implemented: support vector machine (SVM), a multilayer perceptron (MLP) neural network and Näive Bayes. Those algorithms are trained by the merging of two stylometry indicators: function words and TAGS. These indicators are characterized by being not context-dependent. Therefore, human conscience cannot influence on those markers. Additionally, the effect produced by the analysis of principal components, a technique to reduce the characteristics-space dimensionality, is studied. Finally, 6 contemporary Spanish authors will be used as benchmark. Comparisons between two authors yield results with an effective accuracy of 92.65 %.

The research carried out in this Final Grade Project, suggests that the use of stylometric characteristics can be useful for the authorship attribution's problem. Furthermore, the combination of parameters leads to increasing accuracies. Hence, the implemented algorithm obtains accuracies of 92.65%, improving the results of previous studies focused on Spanish texts.

Keywords

Authorship Attribution, text mining, artificial intelligence, principal components analysis, cross-validation, author classification, function words, TAGS, normalization, models' combination.

Agradecimientos

En primer lugar quería agradecer todo el esfuerzo y apoyo que me ha proporcionado mi tutor, Jesús Poza Crespo, ya que sin su ayuda y consejos no hubiera sido capaz de llevar adelante este Trabajo Fin de Grado.

También, cómo no, a todos mis amigos que han estado a mi lado en todas las experiencias buenas y malas que me ha aportado este proyecto. En particular a Evelyn y Sonia, mis chicas de los cafés y las llamadas a cualquier hora; a Luis Miguel, Guillermo, Josu y Miguel, grandes amigos del colegio mayor Santa Cruz.

Agradecer la amistad que me llevo con toda la tropa de ahijados: Rodrigo, Héctor, Pedro, Eduardo, Ryohei y Óscar cuyas charlas interminables acerca de temas variopintos me permitieron desconectar en aquellos momentos que más lo necesitaba. Muchas gracias.

Tampoco puedo olvidar de mencionar a mi madre, Blanca, cuyo conocimiento sobre los puntos y las comas me ha aportado una nueva visión de la escritura, gracias.

Hacer una mención a mis dos tutores y amigos Pedro y Carolina quienes estuvieron en mi etapa de becario en la empresa TID, Telefónica Investigación + Desarrollo, cuyos consejos sobre la vida y el trabajo durante los *breaks* del café siempre serán recordados. Gracias a vosotros también.

Por último dar las gracias a toda mi familia que ha sido un soporte fundamental tanto económico como emocional ya que sin ellos no habría sido posible realizar todos los resultados obtenidos hasta la fecha. Muchas gracias familia Pedrosa Antón.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1 Introducción

El problema de la atribución de la autoría en la literatura es una cuestión latente desde nuestros comienzos con la escritura. Es fácil pensar que antiguamente era más sencilla esta labor, ya que con una simple firma o sello el documento quedaba autenticado. Por supuesto, en aquellas épocas ese método gozaba de una credibilidad casi divina y nadie o casi nadie dudaba de la poca fiabilidad del mismo. Según fueron transcurriendo los años, los expertos de la literatura comenzaron a estudiar el *arte de escribir* a lo que posteriormente se denominará *estilometría.* Holmes (1994) escribió sobre esta ciencia y dijo que su finalidad era la búsqueda de un *estilo* inherente en los textos. Holmes entendía que este *ente* denominado *estilo* agruparía a un conjunto de patrones medibles, únicos para cada autor. Así, encontrando dichas características, podríamos identificar de forma razonada quién fue escribió el documento a estudiar. Entonces, surge un nuevo problema, cómo encontrar e identificar aquellos elementos que hacen posible la identificación unívoca de un autor. Bailey (1979), adelantándose a Holmes, en el año 1979 propuso dentro de un contexto forense una serie de características que se adaptan perfectamente al problema en cuestión. Estos elementos, decía, debían ser salientes, estructurales, frecuentes, fácilmente cuantificables y relativamente inmunes a un control consciente de la persona. Así, la mayoría de los estudios referentes al campo de la atribución de la autoría se fundamentan en encontrar estas características, independientes del proceso mental humano, que hagan posible su cometido: establecer razonadamente la potestad del documento (Stamatatos 2009).

Este Trabajo de Fin de Grado tiene como objetivo el desarrollo de un algoritmo que permita la distinción de un autor de la literatura española actual dentro de un conjunto limitado de ellos. Para realizar tal fin, se seguirá el esquema que se muestra en la Figura 1.1.

La primera etapa es la de preprocesado. En ella el documento plano (DP), es decir, sin ningún tipo de alteración, se someterá a una serie de adaptaciones para que su formato se adecúe a las siguientes fases. A partir de este punto pasará a llamarse documento adaptado (DA). Posteriormente, el DA prosigue por la etapa de extracción de características que seleccionará aquéllas que considere más representativas.

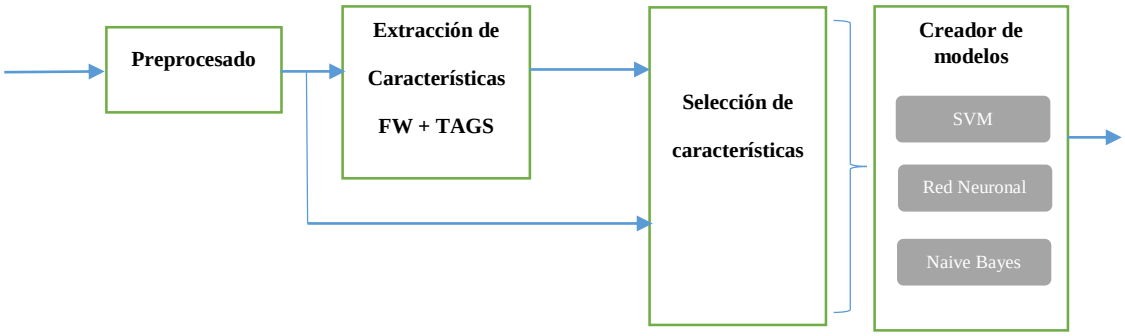


Figura 1.1: Esquema del algoritmo propuesto para realizar la atribución de autoría

Una vez obtenidas, se pasará al extractor de características, encargado de obtener del DA una representación numérica (frecuencia de aparición) de las características elegidas. Luego, se crearán los modelos y se entrenarán con los datos obtenidos en el período anterior. Este algoritmo utilizará tres modelos: máquina de vectores de soporte (*support vector machine*, SVM), red neuronal basada en el perceptrón multicapa y *Näive Bayes*. Una vez creados los modelos, se utilizarán una serie de documentos de test para analizar la precisión del algoritmo.

Este capítulo ofrece una visión preliminar del problema planteado. Para comenzar, se explicará someramente el concepto de minería de datos y las repercusiones que tiene, haciendo un mayor hincapié en el siguiente apartado sobre el tema de la minería de textos, que es lo que afecta al estudio del trabajo. Posteriormente, se introducirán conceptos acerca de los analizadores del lenguaje que harán posible la manipulación de la información almacenada como texto. En la sección siguiente, se hará una referencia a las características independientes del contexto, que se utilizarán en los distintos modelos de inteligencia artificial (IA). Por último, se indicarán cuáles son los objetivos que se pretenden alcanzar.

1.2 Problema de la atribución de autoría

La ciencia computacional y estadística comenzó a apoyar el estudio de la atribución de autoría en el siglo XIX. El artículo escrito por Mosteller y Wallace (1963) acerca de los *Federalist Papers* es considerado el más influyente. Esta publicación, en el *Journal of the American Statistical Association*, marcó el nacimiento de una línea de

investigación centrada en la búsqueda de métodos estadísticos que permitiera desentrañar el origen de un conjunto de palabras que unidas representaban un texto. Gracias a los avances en investigación en campos como *machine learning* (ML) y sobre procesado del lenguaje natural (NLP), ha sido posible avanzar en esta área (Stamatatos, 2009). Además, debido a la ingente información que uno tiene disponible de forma electrónica, como son los correos electrónicos, las redes sociales, los libros electrónicos, etc., la tarea del investigador para encontrar posibles textos de estudio es trivial y sencilla. La idea que se esconde detrás de todas estas investigaciones, es la de encontrar alguna característica dentro de los textos, que sea cuantificable para, de este modo, poder distinguir entre distintos autores. Para Stamatatos (2009), el criterio más importante para la elección de características dentro del campo de la atribución de autoría es la frecuencia. Asimismo, cuanto más frecuente es dicha característica mayor será la variación estilística que recoja.

El Capítulo 2 se dedica a analizar la situación actual del problema de la atribución de autoría. En él, se recorrerá la historia desde el siglo XIX hasta la actualidad, indicando el proceso evolutivo que ha tenido lugar.

1.3 Minería de datos

La finalidad de la minería de datos es la obtención de patrones predictivos a partir de un conjunto de datos, los cuáles por sí solos no predicen nada. Mediante el uso de estos patrones se pretende pronosticar, con una cierta efectividad, cómo será el próximo acontecimiento. Estos datos pueden ser de dos tipos: estructurados y no estructurados. Los datos estructurados son aquellos que permiten establecer relaciones comparativas por ejemplo los datos numéricos. Un ejemplo muy utilizado es el conjunto de datos *Iris flower*, popularizado en el campo de *machine learning* por Ronald Fisher (1963). Este conjunto de datos engloba cuatro características (longitud del pétalo, anchura del pétalo, longitud del sépalo y anchura del sépalo), de tres especies de flores: Setosa, Versicolor y Virginica. En la Figura 1.2 se muestra una distribución de dichas características conforme a cada una de las especies diferenciadas por colores: rojos para la clase Setosa, verde para la Versicolor y azul para la Virgínica. La idea principal consiste en obtener a partir de esta información unos patrones que permitan identificar la especie de la flor, utilizando

estas cuatro características. Para su obtención existen diversos métodos que proporcionan distintos grados de fiabilidad. En este escenario se realizará un análisis supervisado, es decir, conocemos la salida de nuestro problema que en este caso sería la especie de la flor. En otros contextos, es posible no conocer la salida y lo que se pretende buscar son grupos dentro del conjunto de datos a estudiar, que se relacionen por alguna circunstancia. Ese análisis sería no supervisado.

Por otra parte, los datos no estructurados son aquellos que no se puede establecer esa relación directamente y requieren de un proceso de adaptación para poder ser analizados. Es el caso de los textos y será el campo de la minería de textos (*text mining*) la encargada de extraer de esta información datos con los cuales se puedan establecer comparaciones.

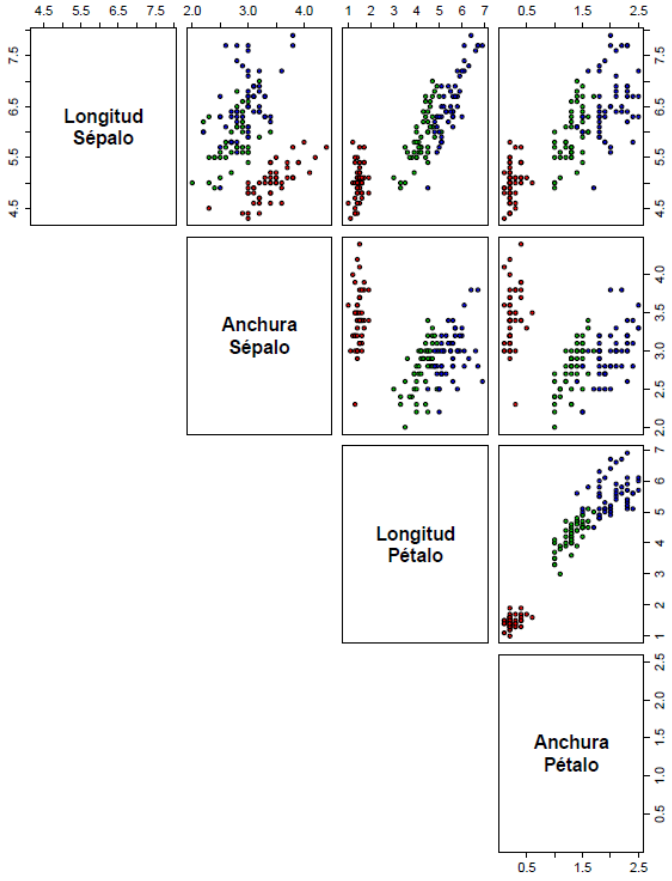


Figura 1.2: Diagrama de dispersión que representa la distribución de cada clase: rojo para Setosa verde para Versicolor y azul para Virgínica para cada par de características obtenida con el programa Rstudio a partir del conjunto de datos *Iris flower*.

1.4 Text Mining

En la actualidad la población está abordada por la gran cantidad de información que tiene disponible de manera casi inmediata. Debido a los avances en las comunicaciones, cualquier persona es capaz de expresar cualquiera de sus pensamientos en la red para que instantáneamente sea compartido por miles de usuarios. Por ello, la ciencia que se encarga de obtener información a partir de los textos está creciendo mucho en estos últimos años (Feldman & Sandger 2006).

Como ya se ha explicado anteriormente, las técnicas de *text mining* se basan en la obtención de patrones dentro de un conjunto de textos seleccionado. La gran diferencia con el campo de *data mining,* es el tipo de datos para analizar. Los textos son algo amorfo y difícil de tratar. Por ello, son necesarios unos mecanismos que hagan de ellos algo compatible para ser analizados computacionalmente. De ese modo, los métodos de selección y extracción de características son herramientas cruciales para la transformación de estos datos no estructurados en información útil para los siguientes procesos que tienen como finalidad la obtención de patrones predictivos (Jiang 2006).

Las aplicaciones que dan sentido a este área de la ciencia son aquellas en las que grandes masas de texto deben ser analizadas para alcanzar algún propósito (Feldman & Sandger 2006). Son muchas las posibilidades y abarcan muchos ámbitos de la sociedad. Para ser más precisos, técnicas de *text mining* son utilizadas para obtener tendencias por ejemplo en el campo de la *web* (Cooley *et ál*. 1997). Analizando las distintas opiniones de los usuarios se pueden obtener resultados para predecir futuros acontecimientos (Bing & Zhang 2006). En el área de las finanzas, banqueros analistas y consultoras han desarrollado sistemas basados en *text mining* que permitan alcanzar lo que se denomina *business intelligence*, inteligencia en el negocio (Spenke & Beilken, 1999; IntertekGroup, 2002; Kloptchenko *et ál.* 2004). También llega su repercusión a los campos de la biomedicina, investigadores exploran grandes masas de informes médicos con el propósito de identificar patrones complejos acerca de las interacciones entre las proteínas (Fukuda *et ál*. 1999; Craven & Kumliean, 1999; Rindflesch *et ál*. 1999; Zhou & Cui 2004). Su presencia en el campo de la seguridad nacional es muy relevante puesto que

se analizan grandes masas de información electrónica (emails, foros, redes sociales, sms, etc.), con el objetivo de detectar próximos sucesos contra la sociedad (Zanasi 2007).

1.5 Analizador del lenguaje. Freeling

La concepción de que un ordenador sea capaz de comprender un texto al mismo grado que el nivel humano es una tarea muy compleja. Ésta es precisamente el área de investigación del *text mining*. Este dilema es debido a la naturaleza propia del lenguaje, puesto que su comportamiento abarca más de lo que una computadora por sí sola puede analizar y comprender (Jiang 2006). Sucede justo al contrario en el mundo de los datos estructurados, donde un ordenador expresaría todo su potencial mediante cálculos rápidos y complejos, y donde una persona no sería capaz de evaluar. En cambio, los textos son diferentes. En ellos no sólo afecta aquello que está escrito, que es visible sino que hay una serie de variables implícitas que una máquina no es capaz de distinguir. Estas características son el contexto del texto, el nivel cultural del autor, el sentido y relación de las palabras, etc. De este modo, en la frase: *Yo bajo con el hombre bajo a tocar el bajo bajo la escalera*, la palabra "bajo" adquiere distintas funciones según el lugar dónde se encuentre y también dependiendo de las palabras que le acompañen. Así la primera aparición del término bajo correspondería sintácticamente a un verbo, pero la siguiente sería un adjetivo ya que acompaña al sustantivo *hombre.* De esta manera, es necesario el desarrollo de una herramienta con cierto grado de inteligencia, que sea capaz de discriminar estas características que residen dentro de los textos y son las que le dan el sentido que el autor quiere expresar.

Los analizadores del lenguaje (*Natural Language Processing,* NLP) serán las herramientas encargadas de adaptar el conjunto de textos para la posible compresión de una computadora. Además, se dividen en distintos módulos que se complementan, pudiendo trabajar de forma conjunta. Dos aplicaciones libres que realizan estas tareas son: *Standford NLP* (Standford NLP Group 2014) y *Freeling NLP* (Padró & Stanilovsky 2012). La primera de ellas se desarrolló en la universidad de Standford y tiene una gran influencia en el análisis de textos ingleses. *Freeling,* en cambio, es una herramienta desarrollada en Barcelona y su mayor aplicación está en el análisis de textos españoles.

Aun así, ambas herramientas permiten analizar textos en otros idiomas. Para este trabajo se ha preferido utilizar la herramienta *Freeling*, debido a que aporta mayor facilidad de integración en otros proyectos y su potencial con el procesamiento de textos españoles.

El Capítulo 3 está dedicado a los analizadores de lenguaje. En él se desarrollará con mayor precisión los conceptos introducidos anteriormente y se explicará más detalladamente las dos herramientas previamente presentadas.

1.6 Características independientes del contexto

En el momento que un investigador se plantea el reto de obtener información a partir de datos no estructurados, como es el caso de los textos, lo primero que tiene que tener claro es el conjunto de características que quiere obtener de ellos. Por ello, centrándonos en nuestro problema, hay mucha información que podemos obtener: correlaciones entre palabras, términos, frecuencias de aparición, *bags-of-words*, fonemas, funciones gramaticales, etc.

Antes de decidir y utilizar una herramienta NLP que permita obtener las anteriores características, hay que establecer con criterio aquellas que mejor se adapten a nuestro escenario. Como ya se introdujo en el Apartado 1.1, el problema que se plantea trata sobre la atribución de la autoría y por tanto se requiere de un conjunto de características que identifiquen unívocamente a un autor. Así, a primera vista, aquellas que dependan del contexto parecen no ser las idóneas para ser sujeto de estudio, ya que si un autor escribiera sobre temas muy distintos entonces se obtendrían valores muy diferentes, provocando un fallo en el algoritmo. Con ello, lo que hay que buscar son aquellos elementos que carezcan de relación con el contexto inherente. Así se consigue que las características seleccionadas no sean sensibles ante cambios importantes en la temática del texto y no generen resultados falsos que debiliten al algoritmo a desarrollar (Holmes 1994).

De este modo, introducimos los conceptos de las palabras funcionales (FW) (Argamon & Levitan 2005) y los *TAGS* (Stamatatos 2009). Ambos son dos características obtenidas mediante un análisis sintáctico de los textos. Las FW son aquellas palabras sin significado aparente por sí solas. Ejemplo de ello son las preposiciones, las conjunciones,

los artículos, etc. Luego están los *TAGS*, concepto utilizado para distinguir asociaciones sintácticas de grupos de 2 a 6 palabras. Así, por ejemplo, podemos distinguir entre un grupo formado por un artículo y un sustantivo de un grupo compuesto de un adverbio y un sustantivo. Como se dijo previamente, ambas características no dependen del contexto, debido a que las FW son comunes en todos los textos y los *TAGS* no hacen referencia al significado de la palabra sino a su relación sintáctica con las palabras que le rodean. Este es el caso de: *el perro come* y *el gato maulla.* Ambas oraciones corresponden al mismo *TAG*: *determinante + verbo + sustantivo.*

El Capítulo 4 trata sobre las características independientes del contexto. En él se explicará con mayor detalle las variables que se pueden obtener, haciendo un análisis más profundo de las seleccionadas en el algoritmo propuesto: las FW y los *TAGS*.

1.7 Técnicas avanzadas de clasificación

Durante los puntos anteriores se ha hecho referencia al concepto de IA pero en ninguna ocasión se ha explicado en qué consiste. Es fácil llegar a una idea más o menos acertada del significado de este concepto. Consiste en la integración en una máquina de lo que nos diferencia de ellas, el intelecto. Sin llegar a la concepción de *máquinas pensantes,* si no de herramientas artificiales capaces de establecer decisiones en base a unas características del entorno (Russel & Norvig 1995). A partir de estas características se establecen reglas con las que se toman las decisiones del tipo *condición-acción.* Si se cumple el requisito, se ejecutará la operación configurada. Por ejemplo la regla: (condición) **Si** el coche de delante está frenando (acción) **entonces** iniciar frenado (Russel & Norvig 1995).

El fundamento de todos los algoritmos de IA supervisados es muy simple, está dividido en dos etapas. La primera de ellas se denomina la etapa de entrenamiento. En ella se seleccionan un conjunto de observaciones, por ejemplo una parte del conjunto de datos *Iris flower*, introducido en el Apartado 1.3. Con ese grupo de observaciones se entrenará el algoritmo indicando el resultado final que debería escoger. Una vez finalizada esta etapa, con el resto de observaciones que no se han utilizado se pasaría a la fase de

test. Esta parte se emplea para comprobar la bondad del modelo y así conocer a priori cómo va a funcionar. Se utilizaría el modelo creado para decidir sobre las nuevas observaciones sin decirle cuál sería la solución correcta. Una vez recopilados todos los resultados habría que establecer la relación de aciertos entre el número total de observaciones, con ello conseguiríamos el porcentaje de acierto o precisión. La Figura 1.3 muestra un esquema del procedimiento.

Para el desarrollo de este trabajo se han utilizado varias herramientas de IA, que permiten decidir de manera automática sobre la autoría del documento. Concretamente, se han empleado tres modelos de IA: SVM, una red neuronal basada en el perceptrón multicapa (MLP) y el modelo de *Näive Bayes*.

El Capítulo 5 se centra en IA aplicada. En él se ofrecerá una visión más detallada sobre la IA y los tres modelos que se han utilizado.

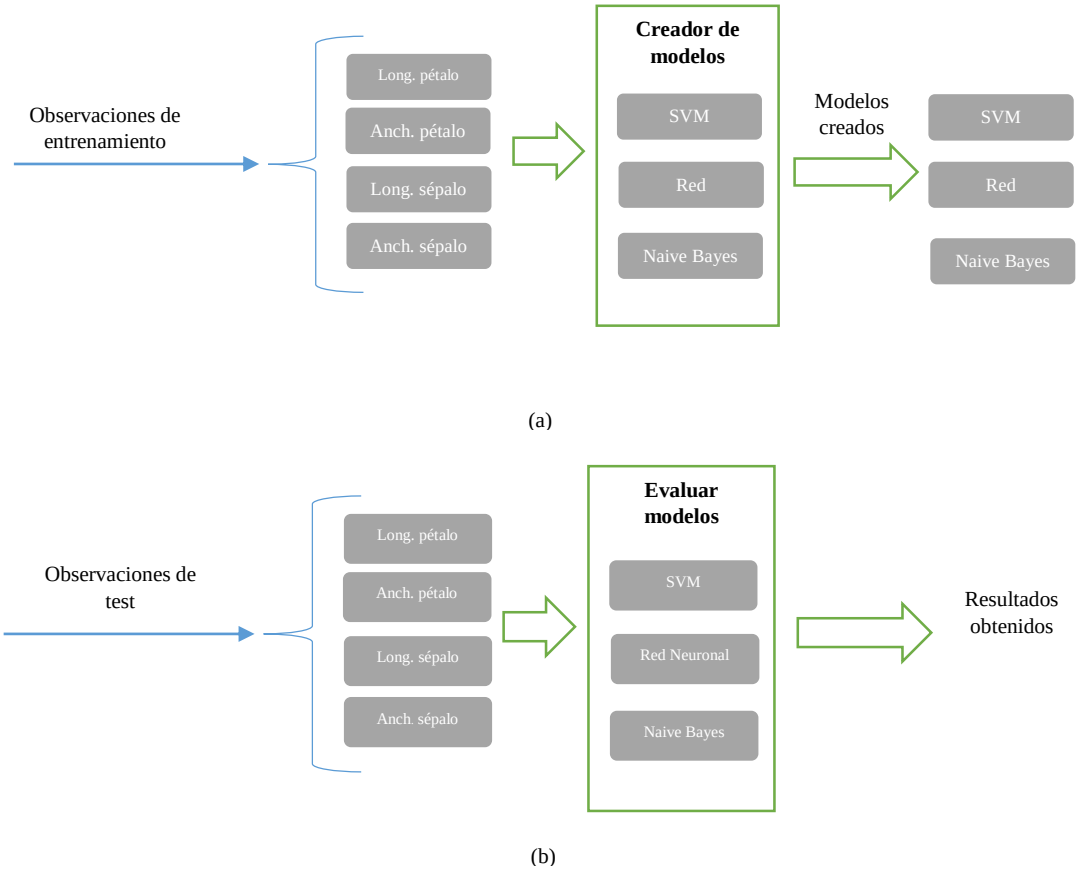


Figura 1.3: Esquema funcionamiento de los algoritmos de IA: (a) Esquema de entrenamiento; (b) Esquema de test

1.8 Objetivos del proyecto

Este trabajo fin de grado tiene como motivación principal **la aplicación de técnicas de IA y *text mining* sobre documentos de texto para poder resolver el problema de la atribución de autoría**.

Una vez introducida la meta que se pretende alcanzar, para llegar a ella hay que conseguir unos objetivos intermedios:

- Modificar el formato del texto para que la herramienta NLP sea capaz de analizar correctamente la información.

- Identificar y obtener el conjunto de características (FW y *TAGS*) que permitan diferenciar el estilo de los autores que dependerá de diversos factores.

- Proponer una metodología basada en técnicas avanzadas de clasificación que permita maximizar el rendimiento de las características estilísticas seleccionadas.

- Aplicar una serie de técnicas para ajustar lo más preciso posible los distintos modelos de IA para optimizar/maximizar los resultados de la fase de test.

- Comparar los resultados obtenidos con estudios previos dentro del campo de la atribución de autoría.

1.9 Estructura del proyecto

En lo que se refiere a la estructura del proyecto, presenta una división en 8 Capítulos y 4 Apéndices los cuáles se detallan a continuación:

- Capítulo 1: *'Introducción'*

Se trata del capítulo introductorio. Ofrece una primera visión sobre los conceptos que se van a desarrollar más detenidamente en las siguientes secciones. Además, en él se incluyen los objetivos que se pretenden alcanzar con el trabajo, así como los pasos que se tienen que dar para llegar a él.

- Capítulo 2: *'Atribución de autoría en textos'*

A lo largo del capítulo se presentará la historia que acompaña al problema de la atribución de autoría, se hará referencia a los estudios más relevantes e influyentes dentro de este campo y se profundizará en aquellos que se consideran más relevantes. Por último, se concluirá el capítulo analizando el estado actual del problema de la autoría de textos.

- Capítulo 3: *'Text mining. Freeling'*

Como el título del capítulo indica, se examinará el campo del *text mining*, explicando detalladamente en qué consiste, cuáles son las principales técnicas para la obtención de información a partir de textos y qué aplicaciones tiene en la actualidad. Se finalizará explicando *Freeling,* una de las herramientas NLP que permiten analizar el lenguaje y transformar datos sin estructura a información manejable computacionalmente.

- Capítulo 4: *'Características independientes del contexto. Las palabras funcionales y los TAGS'*

Este capítulo analiza la importancia que tiene el contexto en el campo de la atribución de autoría, indicando cuáles son los inconvenientes de seleccionar características que presenten una gran dependencia con el entorno. Asimismo, se describirán las ventajas de escoger las características que no dependan de él, haciendo un mayor hincapié en las seleccionadas para el algoritmo: las FW y los *TAGS.*

- Capítulo 5: *'Técnicas avanzadas de clasificación'*

En este capítulo se analiza el uso de IA en el campo de *machine learning.* Se explicarán los conceptos más importantes sobre los tres modelos que se utilizarán en el algoritmo: SVM, MLP y *Näive Bayes.*

- Capítulo 6: *'Desarrollo software'*

El contenido de este capítulo describe el conjunto de tecnologías y herramientas que se han utilizado para el desarrollo del trabajo. Igualmente, para facilitar la tarea de comprensión, se ilustrará mediante esquemas la metodología que se propone en el presente TFG.

- Capítulo 7: *'Resultados obtenidos'*

En este capítulo se muestra el funcionamiento del algoritmo utilizando textos de la literatura española actual. En él se mostrará la precisión del algoritmo, usando un banco de pruebas de 6 autores españoles, Miguel Mora, Fernando J. Pérez y Miguel Jiménez; y de 3 autores de la literatura española: Miguel Delibes Setién, Carlos Ruiz Zafón y Arturo Pérez Reverte.

- Capítulo 8: *'Conclusiones'*

El último capítulo analizará los objetivos alcanzados con el trabajo, realizando a su vez una comparativa con aquellos que se pretendían conseguir. Asimismo, se presentarán las nuevas líneas de trabajo y se discutirán las limitaciones que ha habido durante el desarrollo del trabajo.

- Bibliografía

Recoge el conjunto de recursos bibliográficos que se han utilizado durante el desarrollo del TFG.

- Apéndice A: *'Glosario de Abreviaturas y Acrónimos'*

En este apéndice se listan todas las abreviaturas y acrónimos que se han utilizado a lo largo de la memoria.

- Apéndice B: *'Ficheros de propiedades'*

En este apéndice se encuentran los dos ficheros que utiliza la plataforma para el correcto funcionamiento del flujo de operaciones del algoritmo.

- Apéndice C: *'Pliego de Condiciones'*

  Recoge el conjunto de recursos que han sido necesarios para la realización del trabajo fin de grado.

- Apéndice D: *'Presupuesto'*

  Se indica el coste del presente Trabajo de Fin de Grado, desglosándolo en los diferentes gastos que han surgido.

Capítulo 2

Problema de la atribución de autoría

2.1 Introducción

Este capítulo analizará el contexto en el que se enmarca el problema de la atribución de autoría, desde sus inicios hasta la actualidad. Asimismo, analizará los distintos sucesos que han ido surgiendo a lo largo de los años y que han servido de motivación para el problema.

El enigma de la asignación de la potestad de un documento es y ha sido una cuestión presente desde que las personas concibieron el arte de la escritura. Este sistema, consiste en el uso de un soporte en el cual un elemento es utilizado para dibujar grafos, siguiendo una serie de pautas y reglas que hacen posible la posterior lectura y comprensión de aquello que se quiere conservar. De esta manera, información de muchas de las áreas del conocimiento, como el caso de la historia, la filosofía, la literatura, etc. ha perdurado. Gracias a ello, ha sido posible avanzar a lo que actualmente llamamos sociedad del siglo XXI. Ahora bien, lo que nosotros entendemos como conocimiento escrito, son relatos de personas que en la gran mayoría de ocasiones no hemos llegado a conocer. En el siglo XVI, Descartes aseguraba que *el hombre moderno sólo confía en sí y desconfía de la realidad* (Corazón 2008). Por ello, siguiendo el pensamiento propuesto por el filósofo y dada nuestra naturaleza a desconfiar de todo aquello que no podamos ver o probar, surge la necesidad de preguntarse si aquello que estamos leyendo ha sido redactado por la persona que dice ser y no estamos atribuyendo una autoría equívocamente.

La estructura del capítulo se dividirá en tres partes principales. La primera de ellas explicará el origen del problema dentro del ámbito computacional y explicará el ejemplo más influyente: los *Federalist Papers.* La siguiente parte, mostrará la evolución que ha tenido lugar hasta los años noventa. La tercera parte analiza el estado actual en el que se encuentra la solución al problema que se plantea (Stamatatos 2009).

2.2 Los inicios del problema. “Federalist Papers”

La aplicación de la ciencia en el campo de la atribución de la autoría tuvo lugar a partir del siglo XIX (Stamatatos 2009). Posteriormente, el estudio de Mosteller y Wallace

(1963) acerca de la disputa de la autoría de los *Federalist Papers* se consideró el trabajo más influyente dentro de este área que impulsó a muchos investigadores a encontrar una solución al problema.

2.2.1 El siglo XIX. Mendenhall y las curvas características

Los primeros intentos de obtener características cuantificables a partir de textos, que hicieran posible la identificación inequívoca del autor, se remontan a finales del siglo XIX. Fue Mendenhall (1887) y su estudio publicado en la revista *Science* lo que impulsó a los campos de la ciencia matemática y estadística a dar un soporte al problema de la atribución de la autoría.

Augustus de Morgan fue un matemático Inglés que en una carta enviada el 18 de Agosto de 1851 a uno de sus amigos, le explicaba la idea de estudiar la longitud de las palabras como un indicador del estilo individual de un escritor *(*Grzybek 2006). Así, lo consideró como un posible factor para determinar la autoría del documento. Esta suposición la recopiló Mendenhall (1887) y la trasladó al plano empírico, desarrollando lo que él mismo llamaba como *espectro de palabras* o *curva característica* (CC) (Figura 2.1). Para llevar a cabo estas CC, Mendenhall seleccionaba grupos de un número determinado de palabras dentro de un documento de texto y contaba el número de letras que constituían cada una de ellas. Así, conseguía un gráfico donde plasmaba las distintas curvas obtenidas a partir de cada uno de los grupos de palabras.

Mendenhall observaba que a primera vista las gráficas parecían muy similares, aunque haciendo un análisis más exhaustivo se encontraba con zonas en las que había diferencias importantes. La siguiente prueba que realizó, fue incrementar el número de palabras dentro de cada uno de los grupos, consiguiendo que las curvas se suavizaran considerablemente. En la Figura 2.2 se muestra cómo las curvas tienden a establecer lo que intentaba buscar Mendenhall, la CC propia del autor. Después comparó las CC de dos autores diferentes, uno de ellos fue Dickens con la obra *Oliver Twist* y el otro, Thackeray, con la obra *Vanity fair*. Para ello utilizó dos grupos de 10000 palabras, uno para cada autor. En la Figura 2.3a se muestran las CC obtenidas. Con los resultados llegó a la conclusión de que aunque efectivamente había diferencias en las CC, si bien observó

que había a su vez cierto grado de similitud. En la Figura 2.3b se muestra la cantidad de palabras encontradas para los casos de vocablos compuestos por 11, 12 y 13 letras, llegando en dos de las tres opciones a coincidir en los resultados (Mendenhall 1887).

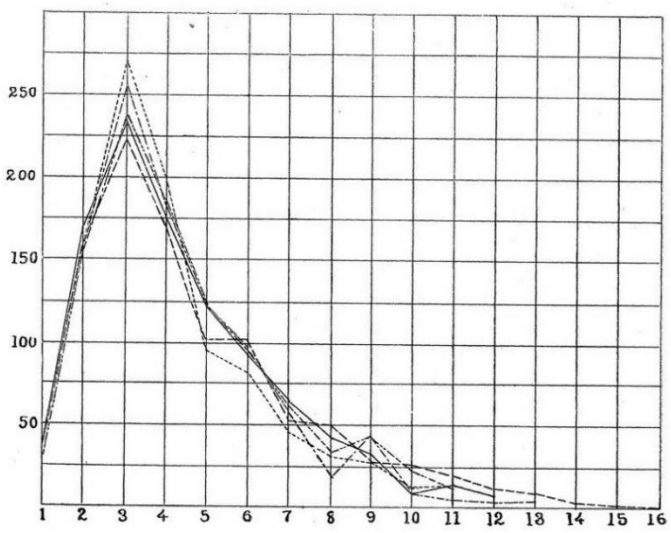
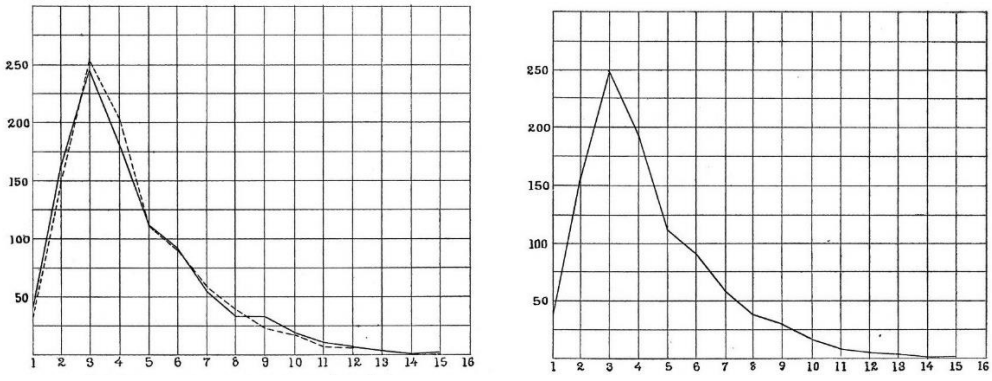
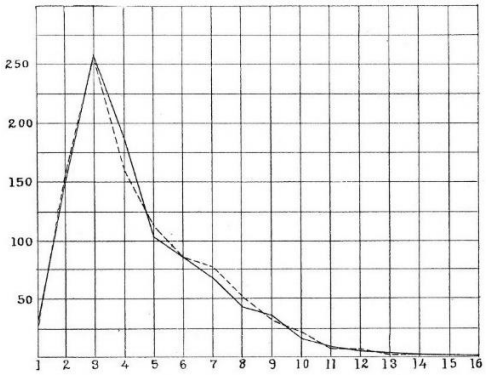


Figura 2.1: Curva característica de 5 grupos de 1000 palabras de la obra *Oliver Twist* (Mendenhall



(a) (b)

Figura 2.2: CC de la obra *Oliver Twist*: (a) Dos grupos de 5000 palabras; (b) Un grupo de 10000 palabras (Mendenhall 1887).



(a)

Number of letters.....	11	12	13
Number of words in Dickens.....	85	57	29
Number of words in Thackeray....	85	53	29

(b)

Figura 2.3: Comparativa entre dos obras de distintos autores, *Oliver Twist* y *Vanity Fair*: (a) Gráfica de las CC (línea contínua: *Oliver Twist*, línea discontínua: *Vanity Fair*); (b) Resultados para palabras con 11, 12 y 13 letras (Mendenhall 1887).

De esta manera, Mendenhall vio claramente la posibilidad de aplicaciones futuras utilizando esta técnica. Además, añadió que no sólo había que limitarse al uso del indicador de la longitud de las palabras, sino que también podía ser aplicado al mismo problema, el estudio de las sílabas y el número de palabras que había en las frases, entre otras. En 1901 publicó un nuevo estudio continuando con su idea de las CC, utilizando únicamente el indicador de la longitud de las palabras, en el que realizó un análisis de las obras de Bacon y Shakespeare dejando al lector la libertad de establecer sus propias conclusiones (Mendenhall 1901).

Los resultados de Medenhall iniciaron una etapa de nuevos estudios que fueron apareciendo en el siglo XX, como los de Yule (1983, 1944) y Zipf (1932). Estos estudios fueron construyendo el camino para uno de los trabajos más influyentes en el campo de la atribución de autoría, el de Mosteller y Wallace (1963) acerca de la potestad de los *Federalist Papers.*

2.2.2 Mosteller y Wallace. "Federalist Papers"

Los *Federalist Papers* son unos artículos publicados en los años 1787-1788 por Alexander Hamilton, John Jay y James Madison con la intención de convencer a los habitantes de Nueva York de la ratificación de la constitución. Estos ensayos aparecieron

en los periódicos y de la gran mayoría de ellos se conoce su autor. A Jay se le atribuyen cinco (los números 2, 3, 4, 5 y el 64), siendo la única aportación del autor al conjunto de ellos. A Hamilton, en cambio, le asignaron la autoridad de 43 informes y a Madison de otros 14. Sin embargo, la autoría de otros 12 informes queda en el aire entre Madison y Hamilton, además faltarían otros tres en los que se considera que existe una aportación conjunta de los autores (Wallace & Mosteller 1963).

Estos documentos sin identificar provocaron una gran controversia y se promulgaron diversas conclusiones sobre quién podría ser el autor de cada uno de ellos. La duda de quién fue el autor de cada publicación no pudo resolverse debido a que los autores Hamilton y Madison esperaron demasiado a establecer sus reclamaciones. En el momento en que aparecieron las listas que atribuían la potestad de los documentos a Hamilton, éste había fallecido y por ello era imposible obtener su testimonio. Por otra parte, Madison tardó alrededor de una década en dar su versión debido a que su postura política del momento no coincidía con la escrita en los documentos que pretendía reclamar (Wallace & Mosteller 1963).

El problema de distinguir los estilos de Madison y Hamilton se complicó cuando en 1941 Williams y Mosteller realizaron un estudio de la longitud de las frases de ambos autores (Mosteller *et ál*. 2010). En sus conclusiones obtuvieron datos muy afines: en el caso de la media de palabras en cada frase, 34.55 fue el valor para Hamilton y 34.59 el obtenido para Madison; en cuanto a la desviación media estándar recogieron datos muy semejantes también, 19.2 y 20.3 para Madison y Hamilton, respectivamente. Esto era debido a que estos personajes eran maestros en el estilo *espectador* de la época, que se caracterizaba por ser complicado y dirigido a la oratoria.

De esta manera, Wallace y Mosteller (1963) comprendieron que debían de explotar otros indicadores de estilo. Así, su método se basó en el análisis estadístico Bayesiano de la frecuencia de un conjunto reducido de palabras comunes. En la Figura 2.4 se observa un análisis de las palabras *by, from* y *to.* Para el experimento se seleccionó un documento de cada autor, el número 48 para Hamilton y el 50 para

Rate	*by*		Rate	*from*		Rate	*to*	
	H	M		H	M		H	M
1– 3	2		1– 3	3	3	20–25		3
3– 5	7		3– 5	15	19	25–30	2	5
5– 7	12	5	5– 7	21	17	30–35	6	19
7– 9	18	7	7– 9	9	6	35–40	14	12
9–11	4	8	9–11		1	40–45	15	9
11–13	5	16	11–13		3	45–50	8	2
13–15		6	13–15		1	50–55	2	
15–17		5		—	—	55–60	1	
17–19		3	Totals	48	50		—	—
	—	—				Totals	48	50
Totals	48	50						

Figura 2.4: Análisis de la frecuencia de aparición de las palabas *by*, *from* y *to* en grupos de 1000 palabras (Wallace & Mosteller 1963).

Madison. Posteriormente se dividieron los documentos en grupos de 1000 palabras y se calculó el número de apariciones de las palabras seleccionadas. Los resultados fueron bastante descriptivos. La palabra *by* indicaba claramente una preferencia por ratios de frecuencia bajos en Hamilton y altos en Madison. En el caso de *to* sucede al contrario aunque con menor acentuación. Respecto al término *from*, las frecuencias bajas no muestran grandes diferencias en los autores; sin embargo, para las altas sí que lo hacen en el informe de Madison (Wallace & Mosteller 1963).

El motivo de la selección de este tipo de palabras tiene como fundamento su escasa dependencia contextual. De esta manera, lo demostraban con el mismo análisis ilustrado en la Figura 2.4 pero estaba vez utilizando la palabra *war* (guerra) como sujeto de estudio (Wallace & Mosteller 1963). En la Figura 2.5 se muestran los resultados obtenidos. Mosteller y Wallace concluían que el significado que aporta la palabra guerra automáticamente sugería que el ratio de uso dependía del tema bajo discusión y por tanto contenidos que tratasen sobre actos armados tenían una tendencia a que tuvieran mayor número de apariciones. Es por ello que a estas palabras las denominaban palabras contextuales y eran consideradas no aptas para ser elementos discriminadores de estilo.

Con todo ello seleccionaron aquellas palabras que no dependían del contexto y realizaron un análisis estadístico bayesiano. Según su estudio, los resultados mostraron claramente una tendencia a que los documentos en disputa estaban escritos por Madison

Rate/1000	H	M
0 (exactly)	23	15
0+–2	16	13
2– 4	4	5
4– 6	2	4
6– 8	1	3
8–10	1	3
10–12	—	3
12–14	—	2
14–16	1	2
	—	—
Totals	48	50

Figura 2.5: Análisis de la frecuencia de aparición de la palabra *war* en grupos de 1000 palabras (Wallace & Mosteller 1963).

aunque en algunos casos no estaba tan clara esa conclusión. Asimismo, demostraron que lo rutinario, es decir, las palabras que se usan casi inconscientemente y que no dependen del contexto, presentan un gran valor en campos de la predicción. Además, probaron que su trabajo era robusto frente a cambios, realizando estudios con trabajos externos a los *Federalist Papers* (Wallace & Mosteller 1963).

2.3 Evolución

2.3.1 Estilometría. El arte de la escritura

Hasta finales de 1990 las investigaciones en el campo de la atribución de la autoría se centraron exclusivamente en la búsqueda de características que permitieran cuantificar el estilo de escritura, lo que se denominó *estilometría* (Stamatatos 2009).

La *estilometría* es la ciencia que intenta encontrar elementos diferenciadores que permitan discriminar de manera precisa la forma en la que se expresan las personas en un documento. Un personaje influyente en este campo fue Holmes (1994, 1998). Para él, este concepto buscaba una unidad numérica que tradujera de forma precisa el *estilo* de un texto y además proseguía explicando el concepto de *estilo* como un conjunto medible de patrones que fuesen únicos para cada autor.

Con la llegada de los ordenadores la tarea investigadora resultaba más sencilla, gracias a la capacidad computacional que aportaban. Debido a la reducción en la complejidad de obtención de los datos, fueron apareciendo nuevos indicadores de *estilo*. Algunos de los indicadores que se investigaron durante esta etapa fueron:

- **Sílabas**. W. Fucks (1952) fue el pionero en investigar este indicador con autores ingleses y alemanes. Su estudio se basaba en calcular 3 valores: la media de sílabas por palabra, la frecuencia relativa de palabras con un número determinado sílabas y su distribución en el texto. Sus conclusiones sugerían que estos indicadores eran buenos diferenciadores entre géneros (prosa y verso). El estudio posterior de Fucks y Lauter (1965) descubrió que la frecuencia en la distribución de sílabas por palabra era un buen discriminador de idiomas. Posteriores estudios, partiendo de la base de Fucks, siguieron esta rama (Bruno 1974; Brainerd 1974).

- **Longitud de la frase.** Yule (1939) consideró el indicador que consistía en contar el número de palabras en una frase y utilizó esta técnica para estudiar la disputada potestad de la obra *The Imitation of Christ.* En sus conclusiones afirmaba que este elemento de estilo no era demasiado fiable para ser usado como discriminador. Sin embargo, dio lugar a nuevos estudios en los cuales se investigó sobre las posibilidades de este elemento (Williams 1940 ; Wake 1957 & Sichel 1974). Una de las conclusiones más importantes para Holmes (1994), indicaba que la longitud de las palabras es un elemento estilístico que cae dentro de la consciencia humana y que sólo será válido en el caso de que se respete la puntuación del autor o el editor sea únicamente una persona. Por último, Tallentire (1972) estaba de acuerdo con que el indicador estilístico que recogiera la media en la longitud de palabras no resultaba ser muy adecuado para los estudios de la atribución de autoría; sin embargo, la distribución de la longitud de las frases sí que podía llegar a dar buenos resultados incluso por sí sola.

- **Distribución de las partes del habla.** Algunos investigadores consideraron como indicadores de estilo la frecuencia de aparición de sustantivos, adjetivos, pronombres, etc. Es el caso de Somers (1966), quien sugiere que un alto número de sustantivos indica un grado de inteligencia más cultivada mientras que el uso

de una mayor cantidad de verbos implicaría una actitud empática y activa. Otros investigadores han buscado las relaciones entre ciertas clases. Antosch (1969) estudió el comportamiento del par verbo-adjetivo entre distintos géneros literarios y demostró un grado de dependencia con respecto al tema. Brainerd (1973, 1974) investigó la posibilidad de utilizar la frecuencia de artículos y pronombres.

- **Palabras funcionales.** La frecuencia de ciertas palabras como indicador de autoría fue otro de los elementos que se ha estudiado. En este caso no vale cualquier palabra sino aquellas que no dependan del contexto, por ello las denominaron palabras funcionales. Ellegard (1962) utilizó estos elementos estilísticos para su estudio en cuanto a la autoría de las cartas de Junius, una serie de panfletos políticos escritos entre 1769-1772. El objetivo de su trabajo era el de encontrar un *ratio de distinción* para cada FW. Su obtención se basó en la relación entre la aparición relativa de la palabra a estudiar en las cartas de Junios con respecto a la aparición de la misma en otros textos ajenos a él. Otros autores que utilizaron esta técnica fueron Mosteller y Wallace (1963) con el estudio de los *Federalist Papers*.

2.3.2 Soluciones asistidas por ordenador

Hasta finales de los 90 la gran mayoría de soluciones que se plantearon referente al problema de la atribución de la autoría eran asistidas por un ordenador, es decir, el objetivo no era el de desarrollar una herramienta completa (Stamatatos 2009). Algunas herramientas llegaron a desarrollarse y utilizarse como la técnica *CUSUM* (suma acumulativa) (Morton & Michaelson 1990) la cual fue aceptada para su uso en los tribunales. Aunque debido a la falta de evaluaciones objetivas fue muy criticada y considerada como dudosa (Holmes & Tweedie 1995). Uno de los grandes problemas del momento, era la falta de un banco de pruebas fiable (Stamatatos 2009). Por ello, no era posible obtener estimaciones objetivas de precisión de las técnicas de atribución de autoría. Algunas otras de las limitaciones que imponía la época eran las siguientes (Stamatatos 2009):

- Los textos que se investigaban eran demasiado largos y probablemente estilísticamente no homogéneos
- El número de autores a discriminar era pequeño, normalmente dos o tres.
- La evaluación de los distintos métodos propuestos era meramente intuitivo, solían basarse en la inspección visual de un diagrama de dispersión.
- La comparación de los distintos métodos era complicado debido a la escasez de un banco de datos de pruebas.

2.3.3 Finales de los 90. La revolución electrónica

Los últimos años de la década de los 90 supusieron un cambio muy importante para el campo de la atribución de la autoría (Stamatatos 2009). Esto es debido a la aparición de grandes cantidades de textos en formato electrónico. Así, aparecen los correos electrónicos, los blogs, los foros *online,* etc., fuentes de textos que están disponibles gracias a Internet. Además, supuso un impacto en muchas áreas científicas como la de obtención de información (*information retrieval), machine learning*, y NLP (Stamatatos 2009).

El campo de *information retrieval* desarrolló técnicas eficientes para la representación y clasificación de grandes volúmenes de texto. A su vez, en el área de *machine learning* aparecieron algoritmos capaces de trabajar con datos dispersos y multidimensionales y se consiguieron nuevos métodos para comparar distintas técnicas utilizando el mismo banco de datos. Por último, en el área de *natural language processing* se desarrollaron nuevas herramientas NLP, capaces de analizar eficientemente los textos y proporcionar nuevas características cuantificables a partir de los textos (Stamatatos 2009).

Las consecuencias de estos avances en el campo de la atribución de autoría se hicieron presentes en muchas áreas en forma de aplicaciones. En el mundo de la ley criminal se desarrollaron herramientas que permitían identificar la veracidad de notas de suicidio (Abbasi & Chen 2005). En la ley civil, por ejemplo, se crearon utilidades para temas de disputas de *copyright* (Chaski 2005; Grant 2007) y en la ciencia computacional

forense para identificar el autor del código de software malicioso (Frantzeskou *et ál*. 2006).

Así, esta época está dominada por el esfuerzo de obtener herramientas prácticas que den solución a problemáticas que impliquen el uso de textos del mundo real, más que encontrar una respuesta acerca de las disputas sobre la potestad de artículos literarios.

2.4 Estado actual

La tendencia actual que siguen el conjunto de trabajos en el área de la atribución de autoría se aleja de la búsqueda de elementos estilísticos que sean capaces de distinguir la personalidad literaria del autor. Actualmente se centran en el desarrollo de métodos que permitan la obtención de estos indicadores con el fin de maximizar el efecto buscado (Stamatatos 2009). Este apartado mostrará una visión general de las distintas características que van a ser objeto de estudio. Las Tablas 2.1 y 2.2 muestran un resumen del conjunto de las mismas que se van a investigar.

Características		***Herramientas y recursos necesarios***
Léxicas	Basadas en *tokens*	Divisor en tokens, [Particionador de frases]
	Riqueza del vocabulario	Divisor en tokens
	Frecuencia de palabras	Divisor en tokens[Divisor de raíces , un divisor de lemas]
	Palabras con n-gramas	Divisor en tokens
	Errores	Divisor en tokens, Corrector ortográfico
Caracteres	Tipos de caracteres (letras, dígitos, etc.)	Diccionario de caracteres
	Caracteres por n-gramas (longitud fija)	-
	Caracteres por n-gramas (longitud variable)	Selector de características
	Métodos de compresión	Herramienta compresora de textos

Tabla 2.1: Tabla de las características léxicas y basadas en caracteres que serán objeto de estudio en el problema de la atribución de autoría (Stamatatos 2009).

Características		***Herramientas y recursos necesarios***
Sintácticas	Parte del habla	Divisor en tokens, divisor de frases, etiquetador de la parte del habla
	Trozos (*Chunks)*	Divisor en tokens, divisor de frases, [etiquetador de la parte del habla], creador de trozos.
	Estructura de las oraciones y frases	Divisor en tokens, divisor de frases, [etiquetador de la parte del habla], creador de trozos. Analizador parcial
	Frecuencia de reglas de escritura	Divisor en tokens, divisor de frases, [etiquetador de la parte del habla], creador de trozos. Analizador completo
	Errores	Divisor en tokens, divisor de frases, corrector ortográfico sintáctico
Semánticas	Sinónimos	Divisor en tokens,[etiquetador de la parte del habla], Theasaurus
	Dependencias semánticas	Divisor en tokens, divisor de frases, etiquetador de la parte del habla, creador de trozos, analizador parcial y analizador semántico
	Función	Divisor en tokens, divisor de frases, etiquetador de la parte del habla, diccionarios especializados
Específicas de la aplicación	Estructura	Analizador HTML, analizadores especiales
	Específicas del contenido	Divisor en tokens, [Creador de raíces y de lemas], diccionarios especiales
	Específicas del lenguaje	Divisor en tokens, [Creador de raíces y de lemas], diccionarios especiales

Tabla 2.2: Tabla de las características sintácticas, semánticas y específicas de la aplicación que serán objeto de estudio en el problema de la atribución de autoría (Stamatatos 2009).

2.4.1 Características léxicas

Para la obtención de estas características los textos son tratados como una estructura que agrupa elementos, palabras, delimitados por signos de puntuación. A este conjunto se le conoce como frase u oración dependiendo de la existencia de un verbo

dentro de esta asociación (Stamatatos 2009). De esta manera, se obtienen los siguientes indicadores:

.

- **Basados en *tokens*.** Un *token* hace referencia a una palabra, número y en ciertas ocasiones incluyen también a los signos de puntuación. De ellos se pueden extraer diversas características. Dos de ellas serían la longitud de palabras (*word length*), y la longitud de la frase (*sentence length)*.

- **Riqueza del vocabulario.** Este indicador hace referencia a la variedad lingüística que compone el conjunto de palabras del texto. Con ello hace referencia al número de palabras distintas en referencia al total del documento. Cabe destacar que este valor dependerá en gran medida de la longitud del texto y por ello serán necesarias técnicas de normalización que suavicen el efecto del tamaño como por ejemplo la técnica K (Yule 1944) y la técnica R (Honore 1979).

- **Frecuencia de palabras y *words-ngrams*.** Es muy habitual como indicador del estilo de un autor representar sus textos mediante un vector de frecuencias de palabras (Stamatatos 2009). De esta manera el texto es considerado como un conjunto de palabras que llevan asociada una frecuencia de ocurrencia (Sebastiani 2002). Como se mencionó en el Apartado 2.3.1 con respecto a las FW, el objetivo es la búsqueda de las palabras que sean independientes del contexto para utilizarlas en escenarios diferentes dentro del mismo autor. Sin embargo, la obtención de estas FW no es algo trivial y sencillo, sino que es dependiente del lenguaje y requiere de un estudio del mismo que permita seleccionar las que considere más representativas para el problema a tratar. De esta manera, existen listas de palabras que presentan buenas propiedades para ser utilizadas como discriminadoras del estilo de escribir de un autor. Para el idioma inglés, Abbasi y Chen (2005) publicaron una lista de 150 FW. Sin embargo, Coyotl-Morales *et ál*. (2006) dijeron que estas *bag-of-words* o listas de palabras no incluían la información contextual de la palabra que representaban, es decir, la palabra *take* en las frases *take on*, *the second take* y *take a bath* era considerada la misma característica (Stamatatos 2009). De este modo, Coyotl-Morales *et ál*. (2006) consideraron la información contextual como un elemento más a tener en cuenta

en estas listas. Así, la dimensionalidad del problema creció en una proporción directa a las posibles combinaciones entre las palabras.

- **Errores.** Koppel y Schler (2003) consideraron los errores en la escritura como una medida más para detectar el estilo del autor. Definieron un conjunto de errores gramaticales, como la inserción o eliminación de caracteres en una palabra, y errores de formato, como por ejemplo palabras con todas sus letras en mayúsculas.

2.4.2 Análisis de caracteres

Otra de las líneas de investigación para la obtención de características que definan el estilo de escribir de un autor consiste en tratar al texto como una secuencia de caracteres, así aparecen nuevos indicadores (Stamatatos 2009):

- **Tipos de caracteres.** Estos elementos hacen referencia a la frecuencia de aparición de caracteres, dígitos, signos de puntuación, minúsculas, mayúsculas, etc. Este tipo de información es fácil de obtener y se ha probado su eficacia para cuantificar el estilo de escribir (Grieve 2007).

- ***Character n-grams.*** Al igual que sucedía con las palabras, estos indicadores se refieren a grupos de *n* caracteres, haciendo que el espacio de variables dependa de este valor. Además, este valor *n* puede ser fijo o puede variar. Por ejemplo en la frase: *El gato de Blanca,* si hiciéramos una extracción de las combinaciones de 3 caracteres tendríamos: [El_], [l_g], [_ga], etc. Estos indicadores tienen mayor resistencia y tolerancia frente a textos con presencia de ruido, es decir, con omisión de caracteres o escritura con faltas ortográficas, que en el caso de los *word-ngrams.* Esto es debido a que en caso de analizar una palabra errónea, el análisis léxico lo contaría como una palabra diferente a la que verdaderamente representa. En cambio, mediante un análisis de caracteres obtendríamos algunos grupos erróneos pero no el conjunto de ellos. Además, estos elementos son claramente más fáciles de obtener, sobre todo en lenguas orientales en las que la

tarea de obtener los *tokens* es un proceso muy complicado (Matsuura & Kanada 2000).

2.4.3 Análisis sintáctico

Este tipo de análisis se centra en la idea de que el autor inconscientemente tiende a utilizar patrones sintácticos similares. Para realizarlo se necesitan herramientas del campo NLP, que introducirán ruido en los datos obtenidos. Baayen *et ál*. (1996) fueron los primeros en utilizar medidas basadas en información sintáctica de los textos para temas de la atribución de autoría. Así características que podemos obtener serían las siguientes (Stamatatos 2009):

- ***Part-of-speech* (POS).** Estos elementos indican cuál es la función sintáctica de cada uno de las palabras dentro de una frase. Son características que servirán para complementar otras técnicas. Por ejemplo, en la frase: *La casa azul*, obtendríamos el siguiente resultado a partir del POS: det + sust + adj.

- **Fragmentos y estructura de las frases.** El análisis de POS se puede usar para obtener la estructura de las frases y además conseguir fragmentos de ellas. Así, Stamatatos *et ál*. (2000, 2001) utilizaron una herramienta NLP que detectaba frases y las fragmentaba. Lo pusieron en práctica con los textos denominados *Modern Greek.* De este modo, la frase: *El gato de Blanca fue querido por toda la familia*, se analizaría de la siguiente manera: NP [*El gato de Blanca*] VP [*fue querido*] PP [*por toda la familia*], siendo NP, VP y PP parte nominal, parte verbal, parte preposicional.

- **Frecuencia en reglas de escritura.** Baayen *et ál*. (1996) introdujeron este tipo de características, cuya finalidad era contar el número de apariciones de ciertas reglas sintácticas. Por ejemplo la regla *A :PP* → *P: PREP + PC : NP* indica que un sintagma adverbial preposicional está constituido por una preposición seguido de un sintagma nominal que funciona como

complemento preposicional. Así, se consigue tener en cuenta la función sintáctica de cada palabra y a su vez del conjunto de ellas.

- **Errores.** Estos elementos tienen que ver con errores a nivel sintáctico que pueden tener lugar en los textos. Koppel y Schler (2003) realizaron un estudio basándose en estos indicadores, utilizando herramientas que detectaran estos errores. Llegaron a la conclusión de que las herramientas de detección necesitaban un mayor grado de madurez debido a que la precisión en los resultados no era la adecuada.

2.4.4 Análisis semántico

Este campo requiere de un análisis más profundo de los textos mediante las ya citadas herramientas NLP y su objetivo es la obtención de características que aporten información semántica. Asimismo, estas técnicas tendrán como implicación un incremento en la aportación de ruido al conjunto de características obtenidas, debido a los errores que puedan introducir el uso de estas utilidades NLP. Es por ello que características ligadas a esta área de análisis no han sido muy explotadas. A continuación, se explicarán dos indicadores de estilo (Stamatatos 2009):

- **Sinónimos.** Estos indicadores se basaban en encontrar similitudes semánticas entre las palabras. Para ello, McCarthy *et ál*. (2006) utilizaron la base de datos de WordNet (Fellbaum 1998) para estimar información acerca de sinónimos e hiperónimos.

- **Dependencias semánticas.** Un ejemplo de estas características estilísticas es el caso de Gamon (2004), quien desarrolló una serie de grafos que mostraban dependencias semánticas entre las palabras. Sin embargo, no pudo proporcionar información acerca de la precisión de la herramienta. Estos esquemas contenían dos tipos de información; la primera estaba relacionada con características semánticas de la propia palabra, número y persona de los sustantivos, tiempo de los verbos, etc. La segunda parte tenía que ver con las relaciones semánticas y sintácticas con los hijos que dependían de él. Los resultados mostraban que

información semántica, combinada con información léxica y sintáctica, incrementaban la precisión de la clasificación.

2.4.5 Características específicas de aplicaciones

Las características desarrolladas anteriormente son todas ellas independientes de la aplicación a la que se vaya a destinar, ya que todas ellas se pueden extraer de un texto siempre y cuando se utilicen las herramientas necesarias. Sin embargo, a su vez existen escenarios concretos de los cuales podemos extraer características específicas que en otros serían imposibles de obtener. Es el caso de la estructura de los textos en dominios como el correo electrónico o los foros, que podría ser utilizada como un indicador de estilo. Igualmente habrá ocasiones en las que el problema de la autoría se centre sólo en un contexto y será en estos casos en los que tengan cabida las características que dependan del mismo. Por último, existe la posibilidad de tomar ventaja de los idiomas y aprovechar ciertos elementos propios de ellos, así Tambouratzis *et ál*. (2004) utilizaron el fenómeno de la diglosia en los textos *Modern Greek* como medida estilística diferenciadora.

Capítulo 3

Text mining. Freeling

3.1 Introducción

El campo de la minería de datos o *data mining* es un área de investigación cuyo objetivo es el de encontrar patrones que permitan predecir acontecimientos futuros. De esta manera, mediante una serie de técnicas asistidas por ordenador se podrán obtener un conjunto de relaciones y reglas mediante las cuales podremos establecer conclusiones futuras (Feldman & Sandger 2006)

Como ya se introdujo en el Capítulo 1, existen dos tipos de datos: aquellos que presentan una naturaleza estructurada y aquellos que en su forma natural no muestran la información que les caracteriza. Dentro de estos últimos se encuentran los textos, los cuales una máquina no es capaz de analizar. Por ello, es necesario que se desarrollen mecanismos, técnicas y herramientas capaces de traducir el conjunto de elementos que conforman un documento a datos estructurados analizables computacionalmente. La parte de la ciencia que se encarga de trabajar con este tipo de contenido se la denomina minería de textos o *text mining* (Aggarwal & Zhai 2012).

3.2 Text Mining

3.2.1 Significado e implicaciones

El incremento de grandes masas de textos disponibles ha supuesto la necesidad de investigar sobre algoritmos avanzados capaces de aprender características y patrones de los textos de un modo dinámico y escalable. La naturaleza de los textos es dispersa y presenta dimensiones muy grandes. Para ser concretos, un texto puede contener por ejemplo 200 palabras diferentes de un total de unas 100.000. Esto supone una dificultad añadida a la hora de obtener aquello que estamos buscando (Aggarwal & Zhai 2012).

Text mining está muy ligado a los conceptos básicos de *data mining* (Feldman & Sandger 2006). Esto es así debido a que ambos campos presentan arquitecturas similares, basadas en sistemas que dan cabida a técnicas de preprocesado, algoritmos de obtención de patrones, métodos de visualización de los resultados, etc. Sin embargo, son campos

diferentes ya que no persiguen los mismos requisitos. Para el caso de *data mining,* en la etapa de preprocesado básicamente se realizan las tareas de normalización del conjunto de datos y la creación de tablas con distintas relaciones. En cambio, en el contexto de *text mining*, a priori estos cometidos no pueden realizarse debido a la naturaleza de los datos. Es por ello, que esta fase adquiere un mayor protagonismo y en ella se desarrollarán mecanismos para la obtención de los distintos elementos que maximicen la precisión de la solución del problema que se quiere resolver. Asimismo, *text mining* también tiene relación con el campo de *information retrieval*. En esta ocasión, la diferencia que existe entre ambos es que mientras que la primera utiliza las técnicas de obtención de características como medio para obtener patrones predictivos, el segundo utiliza dichos mecanismos como la finalidad del campo de estudio. Así, áreas como *text mining*, *information retrieval*, *data mining* y *machine learning*, van a estar muy relacionadas aunque hay que dejar claro que la finalidad que tiene cada uno de los campos es lo que hace que estas ramas de conocimiento adquieran nombres diferentes (Feldman & Sandger 2006).

Las implicaciones que tiene *text mining* abarcan muchos campos. Una de sus aplicaciones más importantes es en el campo de los medios sociales. Debido a la gran cantidad de información textual que se transmite en lapsos de tiempo muy cortos, se crea la necesidad de desarrollar algoritmos basados en *text mining* capaces de procesar todos esos datos y además ser capaces de almacenar aquello que posteriormente se va a utilizar para algún fin (Feldman & Sandger 2006). Por ejemplo, en el campo de la seguridad nacional analizan los textos de correos electrónicos, foros y redes sociales, en busca de patrones que permitan distinguir un posible ataque terrorista (Zanasi 2007). También tiene influencia en la investigación biomédica, ya que en muchas ocasiones la búsqueda de información es a partir de numerosos informes y trabajos de investigación. En este caso, las técnicas de *text mining* permiten agilizar este proceso, ayudando en gran medida en la localización del objeto de búsqueda mediante relaciones entre los significados de los términos dentro de cada trabajo (Aggarwal & Zhai 2012). Por último, destaca el ámbito de Internet, donde tiene gran implicación. Así, está presente en técnicas que se usan para mejorar los motores de búsqueda que hacen más precisos los resultados de búsqueda. (Crescenzi *et ál*. 2001).

3.2.2 Corpus y documento

La unidad básica de información dentro de *text mining* es el documento, un concepto que permite agrupar un conjunto de datos textuales relacionados que comparten una idea e intentan expresarla al sujeto lector (Feldman & Sandger 2006).

Un documento, a priori, no presenta una estructura de manera explícita que haga posible su análisis. En cambio, aunque se le considere un tipo de dato no estructurado para algunas áreas del conocimiento, presenta una estructura bien definida. Es el caso de los campos de la semántica y la sintaxis (Feldman & Sandger 2006). Si observamos los signos de puntuación se hace posible una subdivisión del texto en secciones, párrafos, frases etc. Así mismo, realizando un análisis semántico de las palabras se puede conocer la función que tendría cada una de ellas dentro del documento y así crear una nueva estructura acorde a esta información. Por ejemplo, la frase: *Héctor trabaja en el campo*, mediante este análisis sabemos que la palabra *trabaja* es un verbo y actúa como núcleo de la oración, *Héctor* es un sustantivo y actuaría como sujeto y por último *en el campo* funcionaría como complemento del verbo. De esta manera, tendríamos la estructura Sujeto + Verbo + Complemento. Cabe decir, que es preciso el uso de herramientas NLP que permitan realizar este tipo de análisis y, como se ha dicho previamente, al no ser herramientas con una precisión absoluta, su uso implicará la generación de ruido en el conjunto de los datos.

Con la llegada de los datos electrónicos, aparece el concepto de meta- datos. Este término hace referencia a información adicional no visible por pantalla que se incluye en los textos que presentan este tipo de forma (Feldman & Sandger 2006). El objetivo de este añadido es la especificación de una serie de características informativas acerca del documento, que serán utilizadas posteriormente en procesos de clasificación. Por ejemplo, se puede incluir información del formato, nombre del autor, fecha de publicación, palabras clave, título del documento, etc. La consecuencia de utilizar estos datos es la simplificación de los algoritmos de búsqueda y clasificación, puesto que no es necesario utilizar técnicas de adaptación (Feldman & Sandger 2006).

Otra de las consecuencias que traen estos nuevos datos basados en texto es la división de los documentos en dos tipos claramente diferenciables. Aquellos cuyo formato de escritura es libre, como por ejemplo los textos literarios, trabajos de investigación, informes, etc. Por otra parte, aparecen documentos en los que tienen que seguir unas pautas establecidas. Por ejemplo, aparecen los documentos web, basados en un lenguaje de marcado, HTML, etc. Los documentos que guardan código fuente también siguen una serie de reglas para poder ser creados. Así, este tipo de información va a ser tratado de manera diferente al caso anterior, ya que existen pautas explícitas que permiten obtener información sin utilizar herramientas NLP, eliminando esa aportación de ruido. La Figura 3.1 muestra un ejemplo de cada tipo (Feldman & Sandger 2006).

Cabe destacar que normalmente un documento guarda relación con otros. Será el concepto de corpus el que permita englobar a un conjunto de textos que mantengan una relación coherente. Existen dos tipos de corpus dependiendo de la naturaleza del problema. Aquellos en los que se mantiene constante el número de componentes que le caracterizan; es el caso, por ejemplo, de los problemas de resolución de la potestad de una serie de documentos finitos e invariables. En este escenario el *corpus* no varía en el tiempo. El segundo tipo afecta a problemas en los que el número de documentos sí que varía. Si pensamos en los motores de búsqueda *web* que utilizan técnicas basadas en *text mining*, llegamos a la conclusión de que el número de soluciones disponibles se modificará con la publicación y eliminación de portales *web* (Feldman & Sandger 2006).

Por tanto, *text mining* trabajará en base a un corpus que dependerá del problema a resolver y estará compuesto por un conjunto variable o constante de documentos con un formato que puede ser libre o estar limitado por una serie de pautas definidas.

3.2.3 Arquitectura

Las herramientas desarrolladas para *text mining* siguen la estructura básica que se muestra en el esquema propuesto en la Figura 3.2 (Feldman & Sandger 2006). En él se encuentran tres etapas. La primera de ellas, es el preprocesado, cuyo objetivo es la obtención de las distintas características a partir de los documentos. Posteriormente, se

obtendrán patrones predictivos o clasificatorios, tendencias, relaciones, etc., a partir de los elementos anteriores, concluyendo con la etapa de visualización de los resultados. Este esquema puede variar dependiendo del escenario del problema. Por tanto, puede ocurrir que se añada algún tipo de retroalimentación y las entradas de la primera etapa dependan a su vez de otros factores. A esto hay que añadir que los tres bloques propuestos, a su vez, se podrán dividir en nuevas secciones haciendo el esquema aún más complejo.

```
<html>

<head>
<meta name="description" content="Documento Web">
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width">
<title>Documento Web</title>
</head>

<body>
<p>Hola Mundo</p>
</body>


</html>
```

(a)

Resumen: En este informe se realiza un análisis sobre la problemática generada al proveer de Sistema Operativo a una cantidad media-alta de máquinas pertenecientes a una red. Se pretende estudiar brevemente las alternativas más viables atendiendo a la flexibilidad, escalabilidad y tiempo invertido en la instalación y configuración de sistemas basados en Unix, haciendo un estudio más exhaustivo en el sistema FAI debido a que es la mejor solución de entre las estudiadas

(b)

Figura 3.1: Ejemplos de tipos de documentos. (a) Documento con estructura pautada; (b) Documento con estructura libre

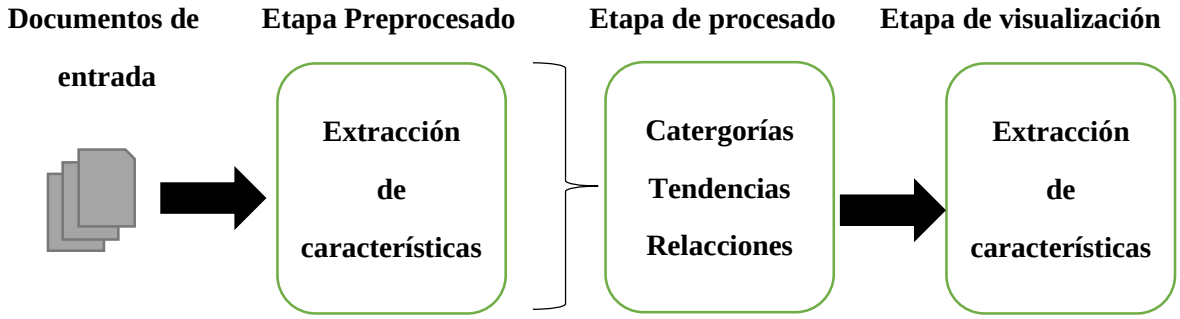


Figura 3.2: Arquitectura básica de un problema de *text mining*

3.3 Técnicas de Text Mining

3.3.1 Extracción de información

El objetivo de estas técnicas es encontrar información estructurada a partir de datos, en este caso textuales no estructurados o poco estructurados (Apartado 3.2.2). Las dos tareas más importantes dentro de este campo son: el reconocimiento del nombre de entidades y la extracción de relaciones (Jiang 2006).

- **Reconocimiento del nombre de entidades (*Name entity recognition*, NER).** La mejor manera de explicar este concepto es mediante el uso de ejemplos, así con la frase *Sonia estudia Grado en Trabajo Social desde 2011*, obtendríamos los siguientes resultados:

  Nombre_Persona (Sonia)
  Fecha (desde 2011)
  Universidad (Grado en Trabajo Social)

  Esta técnica tiene como objetivo la selección de una secuencia de palabras en el texto que describan una entidad del mundo real. De este modo, en la frase anterior podemos distinguir: un nombre de Persona, una Fecha y una Universidad. Cabe decir que los tipos de nombres de entidades vendrán establecidos y existirán una serie de reglas que permitan obtenerlas, que dependerán del contexto que les rodea (Jiang 2006).

- **Extracción de relaciones.** Estos métodos se encargan de buscar relaciones entre las entidades que conforman el texto. Así, para el caso anterior obtendríamos el siguiente resultado:

  Nombre_Estudiante (Sonia, Grado en Trabajo Social)
  Fecha_de_Inicio (2011, Grado en Trabajo Social)

> De esta manera, hemos relacionado las entidades *Nombre_Persona* y *Universidad* creando la clase *Nombre_Estudiante* y *Fecha* con *Universidad* formando *Fecha_de_Inicio*.

Estas técnicas buscan primero las distintas entidades que conforman el texto y posteriormente crean relaciones entre ellos (Jiang 2006). Es muy importante la precisión en la obtención de estas entidades, ya que va a ser la base para la generación del resto de características.

Estos resultados por sí solos ya muestran una estructura y pueden ser visualizados. Otra de las posibilidades es que se utilicen en otros sistemas para completar algún requisito. Es el caso por ejemplo de la inclusión de esta información en una base de datos para que un motor de búsqueda pueda permitir criterios más complejos mejorando la experiencia del usuario (Jiang 2006).

3.3.2 Agrupamiento de textos

El concepto de agrupamiento o *cluster* tiene que ver con la obtención de grupos de objetos que presentan algún tipo de similitud. Es una técnica no supervisada debido a que a priori no se conocen las distintas asociaciones que se van a realizar. Por ello, para su generación es necesaria una función que permita calcular el grado de parecido entre los diferentes objetos para su posterior agrupación. Llevándolo al plano de los textos algunas de las aplicaciones basadas en este concepto de *clustering* son (Aggarwal 2006a):

- **Navegación y organización de documentos.** Una de las posibilidades es la de organizar el conjunto de documentos mediante una estructura jerárquica dividiéndola en diferentes categorías. De esta manera, se facilita la tarea de búsqueda dentro del conjunto de los mismos. Un ejemplo es el método *Scatter/Gather* (Cutting *et al*. 1992).

- **Mejora de motores de búsqueda.** Cuando estamos realizando una búsqueda lo hacemos en base a unos términos. El problema se encuentra cuando aparecen sinónimos. Normalmente estos motores de búsqueda analizan los documentos en

base a la cadena introducida como criterio de búsqueda. Esto implica que aquellos documentos cuyo tema esté relacionado, pero no incluyan esas palabras no aparezcan como resultados (Aggarwal 2006a). Por ello, en este contexto las técnicas de agrupamiento cobran importancia, ya que en la búsqueda no se limitan al texto introducido sino que establecen grupos en torno al tema buscado y mostrarán aquellos que sean más representativos. Esto implica que habrá documentos pertenecientes al mismo grupo que en su totalidad estén relacionados, pero sin tener obligación de utilizar exactamente las mismas palabras.

- **Clasificación de documentos.** Aunque, como se ha dicho anteriormente, este tipo de análisis es no supervisado, ello no implica que no pueda ser utilizado como un paso previo a uno supervisado. De esta manera, *clusters* de palabras y otros métodos pueden ser utilizados para mejorar la precisión de aplicaciones supervisadas (Baker & McCallum 1998).

3.3.3 Clasificación de textos

El objetivo de la clasificación de textos es la categorización de los distintos documentos en clases previamente establecidas, lo que se denomina análisis supervisado. Para llevar a cabo este tipo de proceso existen dos vertientes principales (Feldman & Sandger 2006). La primera de ellas se fundamenta en el conocimiento de expertos en la materia quienes a través de pautas y reglas establecen las condiciones y requerimientos necesarios para clasificar los distintos documentos. La segunda opción tiene que ver con los conceptos de ML que utilizan procesos inductivos a partir de observaciones categorizadas para obtener una serie de patrones. Estos patrones se utilizarán para la clasificación de documentos nuevos. Cabe decir que los sistemas basados en el conocimiento de expertos suelen obtener mejores rendimientos, aunque la tendencia es a utilizar técnicas basadas en ML principales (Feldman & Sandger 2006). Esto es debido a que el primero de los métodos requiere de un gran conocimiento en la materia, en cambio en el campo de ML sólo es necesario la creación de observaciones ya clasificadas, para que el propio algoritmo sea capaz de establecer las reglas que permitan futuras clasificaciones.

Algunas de las aplicaciones que utilizan este tipo de técnicas son las siguientes:

- **Indexación de documentos mediante vocabulario específico.** Normalmente los documentos son etiquetados por una o varias palabras clave cuyo significado aporta una visión general del tema que desarrolla. De esta manera, la tarea de búsqueda se hace más sencilla y rápida utilizando estos términos. Si consideramos como categoría al conjunto finito de estos elementos que pueden ser considerados como palabras clave de cada documento entonces estamos frente a un problema de clasificación de textos. La resolución de esta cuestión puede realizarse automática o semi-automáticamente. En el primero de los casos el algoritmo de clasificación seleccionará la/s categoría/s que mejor represente/n al documento. En cambio, en el segundo escenario, el sistema devolverá aquellas categorías que mejor le caractericen y será el usuario quien decida cuáles seleccionar (Feldman & Sandger 2006).

- **Ordenación de documentos.** Otro de los problemas es el tema de la ordenación de los documentos en distintos grupos establecidos a priori. Por ejemplo, en el caso del correo electrónico consistiría en diferenciar aquellos *emails* publicitarios de los personales, del trabajo, etc. De esta manera, se puede ofrecer una visión del corpus mucho más estructurada y fácil de ver, mejorando la experiencia del usuario (Feldman & Sandger 2006).

- **Filtrado de textos.** Esta cuestión está muy relacionada con el punto anterior, ya que podría considerarse como una ordenación de textos binaria. Este escenario procesará los textos uno a uno y los clasificará como categoría-1 o categoría-0, dependiendo de una serie de requisitos. Para ser más precisos, un ejemplo sería el caso de una aplicación de filtrado de *spams*. Un *spam* es un tipo de correo no deseado que en ocasiones llega a la bandeja de entrada. El objetivo de esta herramienta es clasificar estos mensajes de texto en dos categorías: SPAM o NO-SPAM.

De esta manera, si detecta un correo clasificado como SPAM, lo filtrará y no lo dejará pasar a la bandeja de entrada (Feldman & Sandger 2006).

- **Clasificación jerárquica de páginas *web*.** Estas herramientas se encargan de clasificar las distintas páginas *web* en temas pre-establecidos, para así obtener un catálogo que mejora la experiencia de búsqueda y navegación en Internet (Feldman & Sandger 2006).

3.4 Procesamiento de lenguajes naturales (NLP)

3.4.1 El problema del lenguaje natural

El lenguaje es un mecanismo que permite la comunicación entre las personas. El área de la ciencia denominada *Natural language processing* se dedica al análisis y procesamiento de este tipo de información para poder ser comprendido por un ordenador (Chopra *et ál*. 2013). De esta manera, NLP es un campo de la ciencia computacional, IA y lingüística, enfocada al conjunto de interacciones entre los ordenadores y los lenguajes humanos o propiamente dicho lenguajes naturales (LN) (Chopra *et ál*. 2013).

El problema que plantean los LN es la existencia de la ambigüedad en el conjunto de los datos (Chopra *et ál*. 2013). De este modo, la palabra *banco* tiene distintas acepciones dependiendo del contexto en el que se encuentre. No es lo mismo la frase: *Pedro sacó dinero del banco*, que la frase: *Pedro se sentó en el banco.* En el primer caso el término implica a un edificio donde se guarda dinero; en cambio, el segundo implicaría un asiento para descansar. A esta dificultad habría que añadir la influencia que tiene el contexto en cuanto a las connotaciones de lo comunicado. La frase: *¡Qué contento estoy!*, no tiene el mismo significado para un persona que haya aprobado un examen, de aquella que le haya suspendido. Es por ello que todas estas cuestiones y algunas más hacen que la tarea de *comprender* un lenguaje para un ordenador sea compleja (Chopra *et ál*. 2013).

3.4.2 Etapas en NLP

La tarea del NLP se divide en 5 grandes bloques (Chopra *et ál*. 2013):

- **Análisis morfológico y léxico.** El análisis morfológico implica la obtención de la estructura de las palabras. Dependiendo del escenario del problema, la profundidad del análisis puede presentar un grado u otro. En el caso de la palabra *el* dentro del contexto *el gato*, mediante este análisis se obtendría: determinante artículo masculino singular. Otro de los posibles usos es la obtención de los morfemas, es decir, la unidad más pequeña con significado. Por ejemplo la palabra *libreta* se divide en lo siguiente:
  - **libr- :** lexema, hace referencia a la palabra libro.
  - ***–eta:*** morfema dependiente derivativo sufijo que le da un significado de objeto para tomar una serie de anotaciones.

  Así mismo se pueden crear más términos utilizando el mismo lexema como por ejemplos ***libr****-ero,* ***libr****-ería,* ***libr****-o,* ***libr****-eto,* ***libr****-illo etc.*

  En el otro caso, el análisis léxico se encargará de detectar las distintas fronteras en el texto, es decir, las palabras, frases, párrafos, etc.

- **Análisis sintáctico.** El objetivo de este proceso es comprender la finalidad de las palabras considerándolas como un conjunto, la frase. De esta manera, se extraerán las funciones que desempeñan estos elementos que definen la estructura de la frase. Un ejemplo sencillo sería: *Rodrigo juega al póker*. Con este tipo de análisis se obtendría: Sujeto (*Rodrigo*), Núcleo verbal (*juega*) y Complemento del verbo (*al póker*).

- **Análisis semántico.** La semántica tiene relación con el significado de las palabras y este análisis tiene como objetivo la obtención del concepto de las palabras y su correlación con el resto de términos que le rodean. Así

por ejemplo, la frase: *El niño anciano*, sería incorrecta semánticamente debido a que conceptualmente los niños son jóvenes.

- **Integración del discurso.** Esta fase analiza las relaciones que tienen las distintas frases en el texto. De este modo, se reconocen las apariciones de una frase o parte de ella en otras posteriores. Por ejemplo: *Evelyn vive en la Carcavilla. Allí vive Alba también.* Con este análisis la palabra *Allí* hace referencia a *la Carcavilla*. De esta manera, se integran las connotaciones que tienen unas frases sobre otras completando el significado de aquello que se quiere expresar.

- **Análisis pragmático.** El objetivo de esta etapa es el de entender el sentido que el hablante intenta expresar. Para ello hay que analizar el significado acorde al contexto en el que se encuentre. Por ejemplo la frase: *Cierra la ventana*, podría interpretarse como una sugerencia o como una orden, dependiendo de la situación en la que se posicionase.

Cabe añadir que la dificultad para obtener la información en cada una de las etapas se va incrementando debido a la complejidad y variedad de los lenguajes. Además, las técnicas que se utilicen en unos LN pueden no servir en otros. De esta manera, en el campo de NLP es necesario dividir el problema en los distintos LN para ofrecer resultados lo más cercanos a la realidad.

3.4.3 Aplicaciones de las herramientas NLP

El ámbito de actuación de las herramientas NLP es muy amplio, a continuación se explicarán algunas de las aplicaciones (Chopra *et ál*. 2013):

- **Traducción de textos.** Traducción automática de textos.

- **Segmentación morfológica.** Estas aplicaciones se encargan de subdividir las palabras en morfemas, la dificultad aparece con el tipo de LN a procesar.

- **Reconocimiento de entidades.** Como ya se explicó en el Apartado 3.3.1 se utilizarán estas herramientas para reconocer identidades en los textos.

- **Etiquetado de la parte del hablante.** Etiqueta cada una de las palabras mostrando información de la parte del hablante.

- **Árboles sintácticos.** Estas aplicaciones se encargan de crear gráficos en forma de árbol con las funciones sintácticas de cada palabra. En la Figura 3.3 se muestra un ejemplo obtenido con la aplicación *Freeling* en la que se ha analizado la frase: *Terminé mis estudios en el C.M. Santa Cruz* (Padró & Stanilovsky 2012). En el árbol se muestra información sintáctica y además se incluye la etiqueta de cada palabra (POS) y el *lemma*.

3.5 Herramientas NLP

A las aplicaciones que permiten realizar un análisis de los distintos LN, fundamentándose en los conceptos vistos en los apartados anteriores, se les denomina herramientas NLP. Este apartado introducirá dos: la herramienta *Standford NLP* (Standford NLP Group 2014) y *Freeling NLP* (Padró & Stanilovsky 2012). Se profundizará en la última, ya que ha sido la que se ha utilizado en el presente trabajo.

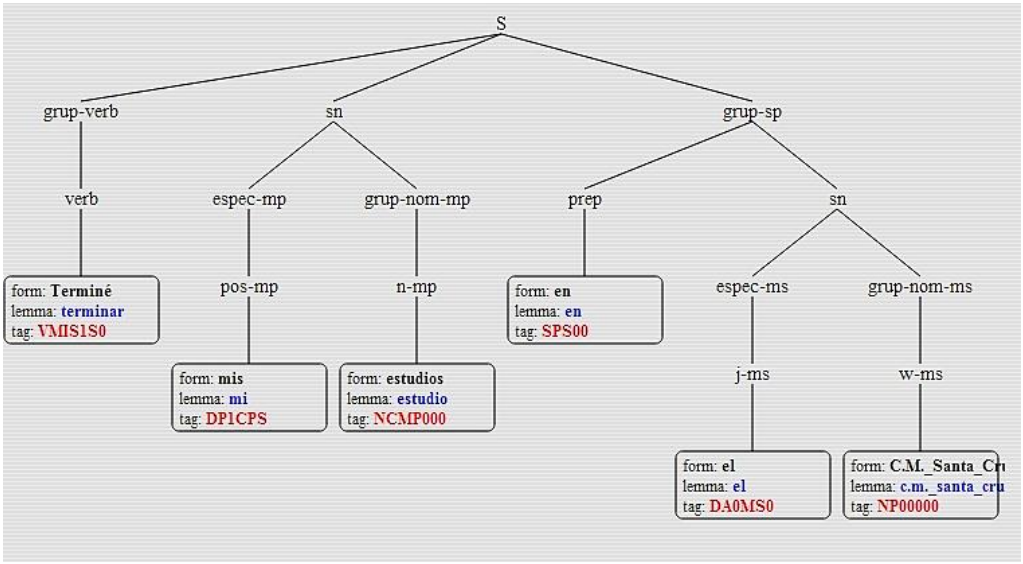


Figura 3.3: Ejemplo de un árbol sintáctico obtenido con la herramienta *Freeling 3.1*

3.5.1 Standford NLP

The *Standford NLP Group* (Standford NLP Group 2014), es un grupo de trabajo de la Universidad de Standford compuesto por un conjunto de investigadores que trabajan juntos para desarrollar algoritmos que permitan a los ordenadores procesar y entender los LN. Algunas de las áreas de trabajo son: comprensión de frases (*sentence understanding*), traducción automática, análisis sintáctico y etiquetado, extracción de información biomédica y elección del sentido de la palabra (*word sense disambiguation)*.

3.5.2 Freeling NLP

La herramienta *Freeling* es una librería de código abierto, capaz de procesar múltiples idiomas y proporciona un amplio número de funcionalidades capaces de analizar los distintos LN compatibles (Padró & Stanilovsky 2012). El proyecto se desarrolló en el centro de investigación TALP (TALP 2014) en España. El objetivo del mismo es la creación de una librería que pueda ser llamada desde una aplicación de usuario para satisfacer las necesidades de análisis del servicio que se esté ofreciendo. La versión actual 3.1 proporciona servicios de: identificación del lenguaje, división en *tokens*, división en frases, análisis morfológico, clasificación y detección de nombres de entidades, reconocimiento de datos, números, magnitudes físicas, monedas y ratios, codificación fonética, etiquetado de la parte del habla, análisis semántico, análisis de dependencia semántica, elección del sentido de la palabra y resolución de correferencias (Padró & Stanilovsky 2012).

El número de idiomas que era capaz de analizar en sus orígenes era tres: inglés, español y catalán. Actualmente soporta once idiomas: asturiano (as), catalán (ca), inglés (en), francés (fr), gallego (gl), italiano (it), ruso (ru), esloveno (sl), español (es) y galés (cy). La Tabla 3.1, *(Freeling 2014b),* muestra los distintos servicios disponibles para cada uno de ellos. Otra de las características de *Freeling* es que aunque ha sido desarrollado mediante el lenguaje c++, también tiene desarrolladas unas *apis* que permiten el uso de estos módulos en otros lenguajes como por ejemplo Java y *python.*

	as	**ca**	**cy**	**en**	**es**	**fr**	**gl**	**it**	**pt**	**ru**	**sl**
División en tokens	X	X	X	X	X	X	X	X	X	X	
División en frases	X	X	X	X	X	X	X	X	X	X	
Detección de números		X		X	X		X	X	X	X	
Detección de fechas		X		X	X		X		X	X	
Diccionario morfológico	X	X	X	X	X	X	X	X	X	X	
Reglas de afijos	X	X	X	X	X	X	X	X	X		
Detección de múltiples palabras	X	X	X	X	X	X	X	X	X		
Detección de nombres de entidades básicas	X	X	X	X	X	X	X	X	X	X	
Detección de nombres de entidades B-I-O		X		X	X		X		X		
Clasificación de nombres de entidades		X		X	X				X		
Detección de cantidades		X		X	X		X		X	X	
Etiquetado de PoS	X	X	X	X	X	X	X	X	X	X	
Codificación fonética				X	X						
WN sense anotation		X		X	X	X	X				X
UKB sense disambiguation		X		X	X	X					X
Shallow parsing	X	X		X	X		X		X		
Análisis semántico	X	X		X	X		X				
Resolución de correferencias					X						

Tabla 3.1: Tabla con los módulos diponibles para cada idioma en la version de *Freeling 3.1*

Capítulo 4

Características independientes del contexto. Palabras funcionales y *TAGS*

4.1 Introducción

Al abordar un problema mediante *machine learning* el primer paso es la elección de las características a utilizar en los modelos (Kaufmann 2005). Este proceso es muy importante, ya que el grado de descripción que aporten estos elementos influirá en la bondad obtenida por el algoritmo. Desde un principio hay que tener claro qué es lo que se está buscando: una metodología que permita la distinción de la autoría de un texto. De esta manera, las características que se necesitan encontrar tienen que resaltar el estilo individual de cada autor y además ser unívocas para no dar a confusión. Asimismo, este problema implica la manipulación de datos no estructurados, es decir, se tendrá que obtener características no explícitas en los textos que se van a analizar. Como ya se ha visto en el Apartado 2.4 son muchos los posibles análisis que se pueden hacer y así es el número de indicadores de estilo que podemos obtener.

Por lo tanto, hay que establecer una serie de requisitos que deben singularizar a las variables que escojamos. Por el momento, el primer requisito ya está establecido; tienen que ser elementos que maximicen la distinción estilística de manera unívoca. El segundo de ellos tiene que ver con el contexto; hay dos tipos de variables: aquellas que dependen de este factor y las que permanecen inmutables ante cambios del mismo. Para el problema en cuestión, las características del primer tipo no encajan con las exigencias, ya que si un texto del mismo autor tratase de temas diferentes estos indicadores mostrarían resultados distintos (Holmes 1994). Por ello, estas variables quedan descartadas. Entonces las variables independientes del contexto son las que mejor se adaptan a la naturaleza del problema. A partir de este punto la elección de qué elementos hacen máxima la diferenciación es algo más relativo y subjetivo, ya que el área de la atribución de la autoría es joven y además dependen del LN que se utilice debido a que cada lenguaje presenta unas propiedades y estructuras diferentes (Stamatatos 2009).

Mediante un análisis de las distintas posibilidades, este estudio se basará en utilizar dos indicadores distintos: las palabras funcionales y los *TAGS*. En las siguientes secciones se profundizará en cada uno de ellos por separado y en el último apartado se explicará la causa de trabajar con ambos.

4.2 TAGS

4.2.1 Definición y obtención

Este tipo de indicadores estilísticos se centra en el área del análisis léxico (Apartado 2.4.1). La idea fundamental es la obtención de reglas de escritura que informen sobre la frecuencia de uso de estructuras morfológicas que caractericen el estilo de escribir de un autor (Peng *et ál* 2004; Sanderson & Guenther 2006; Coyotl Morales *et ál*. 2006). Para dejarlo claro, se analizará la frase: *El vídeo de Pedro fue un éxito*, obteniendo una primera aproximación a este tipo de indicadores de estilo.

1. El primer paso es la obtención de la información morfológica de cada palabra, para simplificar sólo se especificará la categoría gramatical:

    a. ***El*****:** Determinante.
    b. ***vídeo*****:** Sustantivo.
    c. ***de*****:** Preposición.
    d. ***Pedro*****:** Sustantivo
    e. ***fue*****:** Verbo.
    f. ***un*****:** Determinante.
    g. ***éxito:*** Sustantivo.

2. La siguiente tarea es la agrupación de estas categorías formando estructuras (*TAGS*), por ejemplo grupos de 2:

    a. Determinante + Sustantivo
    b. Sustantivo + Preposición
    c. Verbo + Determinante

    Otra de las posibilidades es grupos de 3 o de 4:

    a) Determinante + Sustantivo + Preposición
    b) Sustantivo + Preposición + Sustantivo

c) Determinante + Sustantivo + Preposición + Sustantivo
d) Sustantivo + Preposición + Sustantivo+Verbo

3. Por último, hay que contar la frecuencia de aparición de los distintos *TAGS* en el texto que se está analizando.

De esta manera, hemos obtenido un conjunto de datos estructurados que permitirán diferenciar el estilo del autor. La Tabla 4.1 muestra un ejemplo del resultado que se obtendría. Cada fila hace referencia a un documento y cada columna corresponde a un *TAG,* salvo la última de ellas, que está reservada para guardar el nombre del autor. Las columnas se rellenarán con la frecuencia de aparición del *TAG* en el texto que se está analizando. De esta manera, se obtiene una matriz con información estructurada a partir de un conjunto de datos que carecen de ella. La idea de seleccionar las funciones gramaticales es debido a que no dependen del contexto. Por ejemplo, si hubiéramos elegido combinaciones de palabras, está claro que términos como *neurona* tendrían más apariciones en textos del área de la medicina. Así, centrándonos en su esencia gramatical y no semántica, las frases: *Óscar es estudiante de Informática* y *Alejandro es estudiante de Telecomunicaciones*, aportarían los mismos resultados. De la otra manera los cambios en ambas frases implicarían dos resultados diferentes, haciendo que el algoritmo fuera sensible a cambios de contexto.

4.2.2 Selección supervisada

Cuando estamos seleccionando los *TAGS* que servirán como indicadores estilísticos del autor surgen tres problemas: (i) cuál es el número de *TAGS* que hay que

	Det+Sus	*Verb+Sus*	*Prep+Sus+Verb*	*Sujeto*
Texto(1)	20	15	40	Autor 1
Texto(2)	25	34	12	Autor 2
....				
Texto(N-1)	18	20	36	Autor 1

Tabla 4.1: Tabla con los datos obtenidos a partir de los *TAGS* en diferentes textos, ejemplo con datos rellenados de manera aleatoria

elegir, (ii) cuántos componentes gramaticales compondrán cada *TAG*, y (iii) la elección de los mismos.

- **Número de *TAGS*.** La solución que se propone a este factor dependerá fundamentalmente de dos requisitos. El primero es que el número elegido tiene que ser tal que maximice la descripción que aportan el conjunto de variables sin llegar a la redundancia. El segundo tiene que ver con la selección de una cantidad que permita obtener ese conjunto de *TAGS* para las distintas longitudes de textos que va a procesar el algoritmo. Es decir, una vez obtenida la lista que contiene todos los posibles *TAGS* existentes en los documentos analizados, el valor elegido tiene que ser menor o igual que esa longitud. Cumpliendo estos dos requisitos podemos estar seguros de que el algoritmo va a poder escoger ese número de características y además que aportarán un significado estilístico del autor preciso.

- **Componentes gramaticales.** Este factor hace referencia al número de integrantes que conforman el *TAG*, que también será clave para la obtención de unas características que mejoren la precisión del algoritmo. En este caso, hay diversos métodos que se pueden utilizar. El primero de ellos es la selección de aquellos que están compuestos por un número de componentes fijo, establecido a priori. De esta manera, obtendremos *TAGS* cuyo número de elementos gramaticales no varía. Otra de las posibilidades es permitir una cantidad variable dentro de cada *TAG*. En este caso hay dos posibilidades: (i) que la cantidad de cada grupo esté previamente establecida (de este modo tendremos por ejemplo una configuración de 10 *TAGS* de 2 componentes, 20 *TAGS* de 3 componentes 5 *TAGS* de 4 componentes, etc.); (ii) la solución escogida en el TFG, que esa configuración se haga dinámica en base a una función que estime su precisión estilística. El siguiente apartado explicará una técnica para seleccionar aquellos *TAGS* que mejor diferencien a los distintos autores.

- **Elección de los *TAGS*.** Indudablemente, el mayor de los problemas, la elección de qué grupos gramaticales serán los que mejor distingan al

conjunto de autores. Este proceso puede realizarse principalmente de dos maneras. La primera mediante un estudio exhaustivo del LN correspondiente y la posterior creación de una lista fija de *TAGS*, siguiendo la idea de listas fijas propuesta por Abbasi y Chen (2005). Sus elementos serán los buscados en todos los textos. El otro método es el propuesto en el presente TFG e implica la obtención de la lista de *TAGS* mediante un análisis inductivo. Es decir, a partir de unos documentos cuyos autores son conocidos, se obtiene una lista de *TAGS* de cada sujeto y seguidamente se seleccionan aquellos que mejor diferencien el estilo de cada uno. Esta propuesta permitirá obtener de manera dinámica una lista de *TAGS* que permita distinguir estilísticamente a dos sujetos.

4.2.3 Selección dinámica de características

El método que se propone a continuación parte del estudio realizado por Hollingsworth (2012) acerca del uso de diagramas de cajas para la selección de características.

La selección de los *TAGS* se fundamenta en la base del problema: la obtención de aquellos indicadores de estilo que permitan diferenciar a los autores de los diferentes textos. En este caso, la búsqueda de estos elementos persigue maximizar la diferenciación de dos autores. De este modo, se obtendrá una lista de *TAGS* que permita la discriminación únicamente de los dos sujetos de estudio. La Tabla 4.2 muestra un ejemplo de una lista compuesta por los primeros 14 *TAGS* obtenida por el algoritmo del presente TFG. En cada fila se muestra un *TAG* diferente. Estos están compuestos por una cantidad variable de elementos gramaticales separados por un signo de sumar. El formato que se muestra no es el propio de la plataforma, sino que está decodificado al español para poder ser comprendido. Su forma natural aparece en la Tabla 4.3 que será explicada en secciones posteriores.

Para la selección del conjunto de *TAGS* finales, el primer paso es obtener las dos listas que recojan todas las combinaciones posibles de elementos gramaticales de cada

uno de los autores mediante el análisis de una serie de documentos de entrenamiento previamente clasificados. En cuanto a los problemas explicados en el apartado anterior, se buscarán todas las posibles combinaciones en base al número de componentes y de *TAGS*. De esta manera se elaboran dos listas, una diferente para cada autor. La Tabla 4.3 muestra un ejemplo del resultado de este proceso.

Adposición preposición simple+nombre propio lugar+puntuación coma
Nombre común femenino singular+adposición preposición simple+nombre propio lugar
Pronombre común impersonal e invariable+verbo principal indicativo presente tercera singular+adposición preposición simple
Verbo principal indicativo presente tercera singular+adposición preposición simple+determinante artículo masculino singular
Puntuación coma+adposición preposición simple+determinante artículo masculino singular+nombre común masculino singular
Determinante artículo masculino singular+nombre común masculino singular+pronombre relativo común impersonal e invariable
Adejetivo calificativo masculino singular+puntuación coma+adposición preposición simple
Verbo auxiliar indicativo imperfecto primera singular+verbo principal participio singular masculino+adposición preposición simple
Determinante artículo femenino singular+nombre común femenino singular+adposición preposición simple+nombre propio lugar
Pronombre relativo común impersonal e invariable+verbo principal indicativo presente tercera singular+adposición preposición simple
Adposición preposición simple+determinante posesivo primera común singular +nombre común masculino singular
Verbo principal indicativo presente tercera singular+determinante artículo femenino singular+nombre común femenino singular
Nombre común masculino plural+verbo principal participio plural masculino+adposicion preposición simple

Tabla 4.2: Lista de los 13 primeros *TAGS* en formato comprensible por una persona

TAG	Doc1	Doc2	Doc3	Doc4	Doc5
"aq0cs0 sps00 da0ms0"	0.000945	0.002835	0.000945	0.000945	0.000945
"aq0fs0 ncfs000 sps00"	0	0.001418	0	0	0.002836
"aq0fs0 sps00 da0ms0"	0.001112	0	0.001112	0.001112	0.002224
"aq0fs0 sps00 da0ms0 ncms000"	0.003325	0.003325	0.001108	0.001108	0.001108
"aq0ms0 ncms000 sps00"	0.001051	0	0.001051	0.001051	0

Tabla 4.3: Una lista con los 5 primeros TAGS (formato codificado) y la frecuencia de aparición normalizada en 5 documentos

Está compuesta por dos campos, el nombre del *TAG* y la frecuencia de aparición en el documento. En este caso, el rango de valores se encuentra entre el 0 y el 1 debido a que los datos han sufrido un proceso de normalización que se explicará más tarde en el Apartado 4.4. La gran diferencia que aparece en la Tabla 4.3 es que los *TAGS* están codificados. Esta codificación es la propia de la herramienta *Freeling (2014b)* y permite facilitar la tarea del programador a la hora de hacer que la máquina entienda los distintos elementos que integran el mismo. Para conocer su significado se ha desarrollado un diccionario que se encuentra en el Apéndice B.1.

La forma de utilizarlo es muy sencilla. La categoría gramatical de cada elemento es la primera letra y las siguientes sirven como modificadores que añaden significado. Para ello, en orden, se coge la letra siguiente y se le añaden dos más: la de su categoría y otra que indica la característica que representa, la cual dependerá de la posición de la letra que se está procesando. Por ejemplo el elemento *aq0fs0* se sabe que es un adjetivo porque la primera letra es la *a*. El siguiente paso es coger la letra *q* y añadir por delante la letra correspondiente a su categoría: *a,* y aquella que indica la propiedad que representa la segunda posición de un elemento adjetivo; en esta ocasión, el tipo de adjetivo, la *t*. Por tanto, quedaría de la siguiente manera: *atq*. De esta manera, mirando el diccionario sabemos que hace referencia a un adjetivo calificativo. Este procedimiento se haría con todas las letras hasta obtener el significado del indicador: *adjetivo calificativo común singular*.

Una vez obtenidos los dos catálogos de *TAGS* con las frecuencias asociadas para cada uno de los autores, el paso siguiente es la obtención de un valor de precisión ligado a cada uno, que permita ordenarlos de mayor a menor en una lista común. Para ello, se implementará un método cuya finalidad es exactamente la de obtener ese comparador denominado área de precisión (AP). Para su obtención se realizará el diagrama de cajas de los *TAGS* que sean comunes para ambas listas y se calculará el área de las zonas en las que no haya intersección de las cajas obtenidas. Sin embargo, para aquellos *TAGS* que sólo tengan aparición en una de ellas, no se realizará el procedimiento anterior, sino que directamente se calcula el área del diagrama de cajas de la lista en la que aparezca y se potenciará ese valor incrementando el AP en una unidad. La Figura 4.1 y la Figura 4.2 muestran un ejemplo ilustrativo del procedimiento para los dos casos. La franja de color rojo muestra la zona de conflicto entre los dos diagramas de cajas, indicando que esos rangos de valores son parecidos para ambos autores. Por ello, ese área no se tendrá en cuenta. El color verde indica los rangos en los que hay diferencia entre los sujetos. Por ello, cuanto más grandes sean estas franjas mejor será la propiedad diferenciadora del *TAG*. En el segundo caso donde sólo hay un diagrama de cajas, el AP se igualará al área de la caja dibujada más la unidad potenciadora.

Debido al diseño de la plataforma se genera un fichero que contiene los *TAGS* seleccionados mediante el proceso anterior, que permitirá diferenciar a los dos autores analizados. De este modo, es necesario crear una lista por cada par de autores; sin embargo, este método se puede extrapolar a la obtención de catálogos de *TAGS* para un número distinto de sujetos.

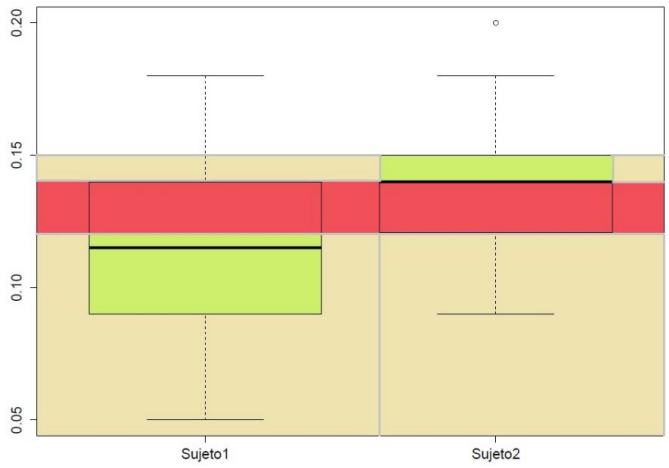


Figura 4.1: Diagrama de cajas para el cálculo del AP de un *TAG* que aparece en los dos autores.

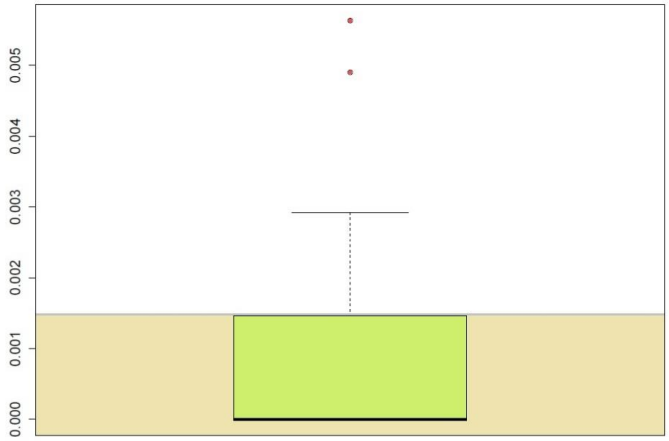


Figura 4.2: Diagrama de cajas para el cálculo del AP de un *TAG* que sólo está presente en un autor

4.3 Palabras Funcionales (FW)

4.3.1 Definición y obtención

El otro tipo de indicador estilístico que se va a utilizar en el algoritmo es el referente a las palabras funcionales. El Apartado 2.3.1 muestra la historia más relevante de este tipo de elementos que son utilizados para la resolución del problema de la atribución de autoría.

Estas características son independientes del contexto, propiedad imprescindible para el problema en cuestión. De esta manera, las FW serán aquellas palabras cuyo significado no dependa del tema que se expone. Así, términos como: *pájaro*, *avión*, *saxofón* etc. serán automáticamente descartados. En cambio, otros como *de*, *a*, *y*, etc., cuya esencia no aporta un significado completo por sí solos serán los escogidos como indicadores de estilo.

4.3.2 Selección supervisada

La única variable que existe en este escenario es el número de FW elegido que haga máxima la distinción estilística. Para resolver esta cuestión, se realizará un estudio

exhaustivo del problema empleando distintas cantidades de FW con el fin de obtener un número concreto. En el Capítulo 7 se realizará dicho análisis. Al igual que sucede con los *TAGS*, aparece el problema de la elección de las FW dentro del LN que permita maximizar el objetivo de estos indicadores. Para la obtención de estas palabras existen dos métodos: Uno de ellos es estático, en el que a partir de un estudio del LN se establece una lista fija y se buscará la frecuencia de aparición de cada FW que se encuentre en ella. Siguiente esta idea, Koppel *et ál*. (2007) usaron una lista de 250 FW mientras que Stamatatos (2006) extrajo una de 1000. El otro método, que es el utilizado en el TFG, no requiere de un estudio previo del LN y además es dinámico, ya que la lista se obtiene en base a un análisis inductivo supervisado a partir de textos conocidos de los autores. El siguiente apartado expondrá la manera de realizarlo.

4.3.3 Selección dinámica de características

Debido a la estructura del algoritmo, la lista de FW que se generará de manera dinámica diferenciará exclusivamente a dos autores. El proceso consiste en obtener una lista ordenada de todos los términos que aparecen en cada uno de los documentos. El número de apariciones será el valor que permite ordenar de mayor a menor cada uno de ellos. A partir de esa lista se analizará gramaticalmente cada palabra en orden y se seleccionarán aquellas cuya categoría gramatical no sea una de las siguientes: sustantivo, adjetivo, verbo o adverbio. Así, se descartan las palabras que dependen del contexto. La Tabla 4.4 muestra una lista de 20 FW obtenida por el algoritmo. El número de FW será un factor para analizar. Burrows (1987, 1992) consideró que un conjunto de 100 características era adecuado para representar el estilo individual de un autor. Para el presente TFG el número de características será una propiedad configurable desde el fichero de propiedades de la plataforma.

4.4 Normalización de las características

Uno de los efectos no deseables que aparece cuando se analizan documentos es el que produce la utilización de textos con longitudes diferentes. En este escenario se

Lista de FW	
1	de
2	el
3	la
4	que
5	y
6	en
7	a
8	un
9	los
10	con
11	se
12	una
13	las
14	del
15	por
16	al
17	no
18	su
19	lo
20	me

Tabla 4.4: Lista compuesta por 20 FW, obtenida por el algoritmo

utilizan dos indicadores que en ambos casos los datos que se obtienen son frecuencias de aparición. En consecuencia, si el texto es más largo, la probabilidad de que el valor calculado sea mayor aumenta. Por lo tanto, es necesario realizar un post-procesamiento de los datos obtenidos que permita anular o reducir en la medida de lo posible el impacto de este fenómeno. Esta transformación consiste en dividir al resultado la frecuencia de apariciones entre la longitud del documento, ya que al ser variables que escapan del

pensamiento humano, la tendencia de aparición evolucionará directamente proporcional a la longitud del texto (Zhao & Zobel 2005).

4.5 La firma del autor. Combinación de TAGS y FW

Un estudio realizado en la Universidad de Oxford contempla la precisión de algunas de las técnicas utilizadas en el problema de la atribución de autoría (Grieve 2007). En la Tabla 4.5 se muestran estas aproximaciones. Posteriormente, ejecutó otro análisis en el que combinaba varias técnicas cuyas precisiones obtenidas individualmente eran buenas. De esta manera obtuvo bondades que mejoraban o igualaban las precisiones ya obtenidas en cada una de las técnicas por separado. La Tabla 4.6 muestra los datos de precisión generados mediante algoritmos que combinan varios métodos estilísticos.

Textual measurement (Variant)	Test accuracy (%)						
	Possible authors						
	40	20	10	5	4	3	2
Word and punctuation mark profile (5-limit)	63	72	80	87	89	92	95
2-gram profile (10-limit)	65	72	79	86	88	91	94
3-gram profile (10-limit)	61	72	78	85	88	91	94
4-gram profile (10-limit)	55	64	73	83	85	89	93
Grapheme and punctuation mark profile	50	60	70	81	84	87	93
Multiposition graph profile (first and last six in word)	49	58	68	79	82	86	92
Word profile (5-limit)	48	57	67	77	80	85	88
5-gram profile (10-limit)	47	55	66	76	79	84	90
Multiposition grapheme profile (first six in word)	43	53	64	76	79	84	90
Multiposition grapheme profile (last six in word)	42	52	63	74	79	83	90
Punctuation mark profile (by character)	34	46	58	72	76	80	89
6-gram profile (10-limit)	35	45	56	68	72	78	86
Word-internal grapheme profile	28	39	51	65	70	76	85
Single-position grapheme profile (last in word)	27	36	49	63	68	73	84
Grapheme profile	25	35	47	62	67	74	83
7-gram profile (2-limit)	34	42	45	59	64	69	81
Single-position graph profile (2nd to last in word)	23	31	43	57	63	70	81
Single-position grapheme profile (1st in word)	20	30	41	56	62	69	80
Multiposition word profile (first four in sentence)	22	31	41	55	60	67	77
Word-length profile (fifteen intervals of one character)	18	26	39	54	60	68	79
Single-position word profile (1st word in sentence)	17	30	36	50	56	64	75
8-gram profile (2-limit)	18	24	36	50	55	62	74
2-word collocation profile	17	24	34	48	54	61	74
Tuldava's *LN*	11	18	31	49	55	64	77
Sentence-length profile (twelve intervals of twenty-five characters)	12	20	31	46	53	62	74
Sentence-length profile. (ten intervals of five words)	10	17	28	44	50	59	73
9-gram profile (2-limit)	12	18	28	41	46	55	68
Type–Token ratio	8	16	27	44	51	61	75
Herdan's *C*	7	14	25	42	49	59	73
Guiraud's *R*	7	13	24	41	48	58	73
Average word-length	7	12	22	39	46	55	70
Average sentence-length (in characters)	6	12	22	39	45	53	70
Average sentence-length (in words)	6	11	21	37	44	53	69
Yule's *K* and Simpson's *D*	6	10	18	33	38	49	65

Tabla 4.5: Tabla que muestra la bondad de las distintas técnicas de decisión de la autoría de textos por separado. Se varía el número de autores (Grieve 2007).

Textual measurement (Variant)	Test accuracy (%)						
	Possible authors						
	40	20	10	5	4	3	2
Weighted combination	69	78	85	91	93	95	97
Simple combination	58	72	82	90	92	94	96
Word and punctuation mark profile (5-limit)	63	72	80	87	89	92	95
2-gram profile (10-limit)	65	72	79	86	88	91	94

Tabla 4.6: Tabla que muestra la bondad de 4 algoritmos que combinan distintas técnicas de decisión de la autoría de textos (Grieve 2007).

En el presente TFG se utiliza el concepto de la *firma del autor* (FAu), que hace referencia al conjunto de características que permiten distinguir a dos autores. Al igual que sucede en el estudio de Grieve (2007), la FAu presenta tres configuraciones dependiendo de los indicadores de estilo que la compongan: (i) sólo *TAGS*, (ii) sólo FW y (iii) combinación de ambos indicadores.

- **Sólo *TAGS*.** La elección de este indicador fue en base al estudio propuesto por Hollingsworth (2012) y su método de optimización en cuanto a la selección dinámica de estas características. Asimismo, otros estudios han probado que esta técnica es efectiva para el contexto de la atribución de autoría (Halteren 2007; Peng *et ál*. 2004; Sanderson & Guenther 2006; Coyotl-Morales *et ál*. 2006).

- **Sólo Palabras funcionales.** La principal motivación para escoger este indicador de estilo es su proximidad con los *TAGS*. Ambos indicadores parten del análisis léxico y los métodos para su obtención son similares haciendo más sencilla la tarea de la fase de desarrollo. Además hay bastantes estudios que aportan buenos resultados (Wallace & Mosteller 1963; Abbasi & Chen 2005; Burrows 1987, 1992).

- **Combinación de sendos indicadores.** Partiendo de los resultados obtenidos por Grieve (2007) se estudiará el comportamiento del algoritmo en base a la combinación de ambos elementos estilísticos. Otra de las motivaciones para su elección fue precisamente la compatibilidad para su fusión. De esta manera, aunque aporten información distinta ambos representan frecuencias de aparición de características léxicas.

4.6 Reducción de la dimensionalidad del problema

Como ya se ha expuesto previamente, cuando intentamos resolver un problema real mediante técnicas asociadas al campo de *machine learning,* lo primero que tenemos que encontrar son aquellas características que mejor describan la cuestión que queremos resolver (Kaufmann 2005). En ciertas ocasiones, el número de variables que podemos encontrar puede llegar a ser tan grande que haga complicado el proceso de obtención de reglas o patrones que permitan alcanzar el resultado buscado. Para resolver este aspecto, se utilizan técnicas que permiten disminuir la dimensión del problema. En este caso, se empleará el, análisis por componentes principales (PCA, *principal component analysis)* (Dunteman 1989).

4.6.1 Análisis de componentes principales (PCA)

PCA se basa en encontrar combinaciones lineales entre el conjunto de variables que caracterizan el problema. De esta manera, tendremos *x* observaciones, siendo *x* el vector que incluye la frecuencia de aparición de los distintos indicadores de estilo. El objetivo de este método es encontrar esas combinaciones con la forma: a_1x, a_2x,..., a_nx, denominados componentes principales, que maximicen la varianza de los datos y además que estén incorreladas entre sí (Dunteman 1989). Resolviendo este problema de maximización, se obtienen los vectores: a_1, a_2,..., a_n. Estos elementos son los autovectores que representan un porcentaje de variabilidad de los datos (Dunteman 1989). Burrows (1987) utilizó este análisis en su estudio para resolver el problema de la atribución de autoría utilizando la frecuencia de aparición de palabras. La Figura 4.3 muestra una gráfica de barras obtenida en una de las pruebas del algoritmo que muestra los autovectores representando la variabilidad característica de cada uno de las componentes principales. El número de los mismos que se va a utilizar dependerá del problema de estudio. En consecuencia, la solución se calculará de manera inductiva, de tal modo que se fije un porcentaje de variabilidad mínimo y se seleccione un número de componentes que ofrezca un valor acumulado de varianza igual o mayor que el límite impuesto.

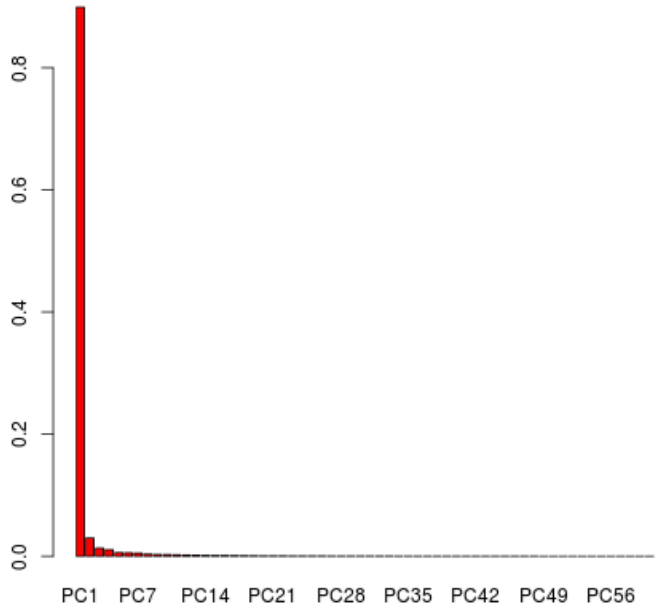


Figura 4.3: Gráfica de barras que indica el porcentaje de varianza que recogen cada una de las componentes principales

La Figura 4.3 muestra que la primera componente principal recoge la mayor varianza, mayor de un 80%. Por ello, en las pruebas que se basan en PCA, se seleccionarán las dos primeras componentes principales. De tal modo, cuando se utilice este análisis se configurará el fichero de propiedades del algoritmo presente en el Apéndice B.2, indicando que se va a realizar análisis PCA y el número de componentes principales que se desea obtener.

4.6.2 Obtención de las componentes principales

La propuesta del TFG para la obtención de estos vectores que permiten la reducción de la dimensión del problema sin aportar una pérdida considerable de información se realiza de manera dinámica. Para ello, lo primero es seleccionar un conjunto de documentos de entrenamiento. El siguiente paso es obtener las distintas características de cada documento en forma de matriz en la que las filas representan cada

uno de los textos y las columnas el valor asociado al elemento estilístico que representa. La Tabla 4.7 muestra la matriz que se obtendría en este paso, se muestra una parte de ella (se reduce el número de filas y de columnas). En este experimento se utilizaron un total de 80 indicadores, 40 FW y 40 TAGS por ello los vectores generados mediante PCA tendrán una longitud de 80. Una vez obtenida la matriz de datos compuesta por todos los documentos de entrenamiento se inicia la búsqueda de los autovectores, obteniendo las distintas componentes principales mostradas en la Tabla 4.8. Como se analizó anteriormente, se almacenarán las dos primeras, porque el nivel de información que recogen ambas es suficiente para caracterizar el problema. El resto de componentes aportan porcentajes muy bajos de varianza, por lo que serán descartados para las siguientes etapas.

	sps00.da0ms0.nccs000	ncfs000.sps00.np00sp0	da0ms0.nccs000.np00sp0	sps00.da0fs0.aq0fs0	sps00.da0ms0.w	da0ms0.ncms000.sps00.da0ms0.ncms000
1	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0000000000	0.0000000000
2	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0000000000	0.0000000000
3	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0027777778	0.0000000000
4	0.001960784	0.0000000000	0.0000000000	0.000000000	0.0000000000	0.0000000000
5	0.000000000	0.0021978023	0.0000000000	0.000000000	0.0000000000	0.0000000000
6	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0000000000	0.0000000000
7	0.000000000	0.0000000000	0.0000000000	0.005780347	0.0086705200	0.0000000000
8	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0044101435	0.0000000000
9	0.000000000	0.0000000000	0.0015197569	0.000000000	0.0000000000	0.0000000000
10	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0044395120	0.0000000000
11	0.000000000	0.0000000000	0.0000000000	0.003393665	0.0000000000	0.0000000000
12	0.000000000	0.0012936611	0.0000000000	0.001293661	0.0000000000	0.0000000000
13	0.000000000	0.0000000000	0.0000000000	0.004694836	0.0015649452	0.0031298904
14	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0032000000	0.0000000000
15	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0000000000	0.0000000000
16	0.000000000	0.0000000000	0.0000000000	0.002412545	0.0012062726	0.0024125453
17	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0000000000	0.0044247787
18	0.000000000	0.0000000000	0.0000000000	0.000000000	0.0000000000	0.0000000000
19	0.000000000	0.0000000000	0.0000000000	0.003454231	0.0000000000	0.0000000000
20	0.000000000	0.0000000000	0.0000000000	0.006600660	0.0000000000	0.0000000000

Tabla 4.7: Tabla de los 6 primeros elementos de la matriz de características asociada a 20 documentos.

	row.names	PC1	PC2	PC3
1	sps00.da0ms0.nccs000	-0.012531732	-0.0586368556	0.0601965078
2	ncfs000.sps00.np00sp0	-0.008675146	-0.0319545252	0.0111340616
3	da0ms0.nccs000.np00sp0	-0.009060822	-0.0345656267	0.0555825525
4	sps00.da0fs0.aq0fs0	-0.007322178	0.0120992183	-0.0056508621
5	sps00.da0ms0.w	-0.007753116	-0.0025353162	-0.0136232361
6	da0ms0.ncms000.sps00.da0ms0.ncms000	-0.008550884	0.0139442592	0.0268691575
7	sps00.da0mp0.nccp000	-0.006778419	-0.0073013274	-0.0026000385
8	sps00.np00sp0.sps00	-0.007875919	-0.0077709641	-0.0116633323
9	ncms000.sps00.np00o00	-0.009427240	0.0110701601	0.0083857644
10	ncfs000.sps00.da0fs0.ncfs000.sps00	-0.007938132	-0.0385223276	0.0209906325
11	sps00.da0fs0.ncfs000.sps00.da0fs0	-0.009331948	-0.0155935163	0.0202749092
12	sps00.da0fs0.ncfs000.sps00.da0fs0.ncfs000	-0.007666295	-0.0180522342	0.0181557351
13	ncfs000.sps00.da0ms0.np00o00	-0.007185582	-0.0012055652	-0.0033619235
14	sps00.ncmp000.aq0mp0	-0.007201036	0.0161201695	-0.0031611891
15	da0ms0.ncms000.sps00.ncms000	-0.006457675	-0.0006180487	-0.0025556821
16	da0ms0.ncms000.np00sp0	-0.007131697	0.0075760266	0.0459390835
17	ncms000.sps00.da0fs0.np00o00	-0.005278724	-0.0118977000	0.0313730546
18	sps00.da0ms0.nccs000.np00sp0	-0.006491708	-0.0342810678	0.0496186241
19	fc.ncms000.sps00	-0.006559490	-0.0133720205	0.0045165781
20	np00sp0.fc.ncms000	-0.005327662	-0.0283815374	-0.0120577234
21	ncmp000.sps00.da0mp0	-0.005661651	-0.0105115823	0.0077997182
22	sps00.da0fs0.np00v00	-0.005484314	-0.0087886449	0.0097139292
23	ncms000.sps00.da0ms0.ncms000.sps00	-0.009503769	-0.0029167837	0.0163129245
24	ncfs000.aq0cs0.fp	-0.005585253	-0.0057803112	0.0143431765
25	da0fp0.ncfp000.aq0fp0	-0.005519772	0.0224538932	0.0109956340
26	np00o00.fpa.np00o00	-0.005273052	-0.0055753336	-0.0040488511
27	np00o00.fpa.np00o00.fpt	-0.005273052	-0.0055753336	-0.0040488511
28	da0ms0.nccs000.sps00	-0.004840550	0.0010395079	0.0134678318
29	dp3cs0.ncms000.sps00	-0.007062922	0.0141612984	0.0137297271
30	aq0cs0.sps00.da0fs0	-0.006024725	-0.0254378728	-0.0141223074
31	da0fs0.np00o00.vaip3s0	-0.003820919	-0.0175754183	0.0002769744
32	da0fs0.np00o00.vaip3s0.vmp00sm	-0.003696052	-0.0170256027	-0.0025125406
33	sps00.da0ms0.np00g00	-0.005175515	0.0019919137	-0.0007047627
34	sps00.ncms000.sps00.da0ms0	-0.005692145	0.0167709815	-0.0042454051
35	da0fs0.ncfs000.sps00.da0mp0.ncmp000	-0.006140244	-0.0078403457	0.0207899009
36	fp.rg.fc	-0.006006920	0.0094381418	-0.0109191712
37	da0fs0.ncfs000.vmp00sf	-0.007503112	0.0171745350	-0.0207933817
38	ncfs000.sps00.np00o00	-0.005375673	0.0019480676	-0.0070621281
39	aq0cs0.sps00.da0ms0.ncms000	-0.006321113	0.0133948048	-0.0045065318
40	da0fs0.np00o00.sps00	-0.004917600	-0.0298106030	-0.0147407114

Tabla 4.8: Tabla que muestra las 3 primeras componentes principales. Parte de la componente principal asociada a los *TAGS*.

	row.names	PC1	PC2	PC3
41	de	-0.606862928	-0.4490505718	-0.4364467367
42	el	-0.259354372	0.1056371951	0.5076370604
43	en	-0.234768594	0.4145345027	0.0108787066
44	la	-0.325565772	-0.4837642349	0.3267780393
45	que	-0.205643563	0.0177447470	0.5093319908
46	a	-0.154544619	0.0875370258	-0.1320051647
47	y	-0.149540489	-0.0478799358	-0.0697805168
48	los	-0.126350450	0.0252094825	-0.0830710757
49	del	-0.116444405	0.0989233228	0.1673113489
50	las	-0.095590801	0.0514174356	0.0050219063
51	por	-0.096913324	0.0597397214	-0.0493421170
52	se	-0.088821745	0.1014363081	0.0670356555
53	un	-0.065291670	0.0274708467	-0.0111697598
54	al	-0.059376202	0.0172189280	0.0338406180
55	con	-0.054304621	0.0742893102	-0.0135964052
56	su	-0.047001938	0.0586075702	0.1113088287
57	para	-0.046314268	-0.0285655754	0.0505259890
58	una	-0.044714150	0.0618056551	-0.0469579914
59	no	-0.041271328	0.0193160944	0.1631486617
60	como	-0.024192650	-0.0211524855	0.0495841153
61	lo	-0.023631835	-0.0107161905	0.0583557762
62	más	-0.024315671	0.0816461023	-0.0119357293
63	o	-0.019593748	-0.0087915476	-0.0161142622
64	sobre	-0.019092713	0.0281417986	0.0211941287
65	también	-0.020720621	-0.0194410839	-0.0275043402
66	entre	-0.020479355	0.0212391807	-0.0104995869
67	X.	-0.403563987	0.3372576332	-0.0641259267
68	este	-0.016319329	0.0016911583	0.0453065827
69	sus	-0.014544487	0.0566731198	-0.0444257435
70	contra	-0.014772616	-0.0564191735	0.0164322836
71	según	-0.014697933	0.0338634815	-0.0252356780
72	sin	-0.012920862	0.0169654705	-0.0060557964
73	X..	-0.264830648	0.4244595964	-0.1962924200
74	esta	-0.012028862	-0.0356580778	0.0326571338
75	si	-0.011434081	0.0146387390	0.0124836343
76	pero	-0.010776295	0.0034770577	0.0139277576
77	desde	-0.009014508	0.0413383287	0.0069727742
78	hasta	-0.009351727	0.0111463503	-0.0347856267
79	ese	-0.009268414	0.0123097204	-0.0075774514
80	ya	-0.008915203	-0.0036383472	0.0253153932

Tabla 4.9: Tabla que muestra las 3 primeras componentes principales. Parte de la componente principal asociada a las FW.

Capítulo 5

Técnicas avanzadas de clasificación

5.1 Introducción

Las ciencias relacionadas con la IA son aquellas que buscan la integración del razonamiento humano dentro de una máquina, con el objetivo de que estas computadoras sean capaces de tomar decisiones. Para desarrollar estas acciones, en el caso del ser humano, es debido a unos estímulos externos que son percibidos (Russell & Norvig 1995). Su origen pueden provenir de cualquiera de nuestros cinco sentidos: vista, oído, olfato, tacto o gusto. Sin embargo, estos elementos no están presentes en las máquinas, que únicamente son capaces de procesar impulsos eléctricos en forma de unos y ceros. De esta manera, nació el área de *machine learning*, cuya misión es permitir a estos elementos inertes desarrollar la capacidad de decisión en base a unas características externas (Witten *et ál* 2011).

El algoritmo que se va a desarrollar en este TFG va a utilizar y analizar el uso de tres modelos basados en IA, que aportan a la máquina la capacidad de decidir la autoría de un documento. Para ello, generará unas reglas y patrones en base a las características establecidas en el Capítulo 4. Una de las formas en las que se pueden clasificar los modelos de IA es en cuanto a la manera que tienen de establecer los patrones y reglas: el proceso de aprendizaje. Pueden ser supervisados, no supervisados e híbridos (Hush & Horne 1992, 1993).

- **Supervisados.** Son algoritmos que utilizan muestras de entrenamiento ya clasificadas. De esta manera, las reglas y patrones se establecerán en base a la distribución de las características que hacen que esa observación tenga esa categoría. Algoritmos supervisados serían SVM, redes neuronales de tipo MLP y N*äive Bayes*.

- **No supervisados.** Estos modelos no conocen a priori lo que representa el conjunto de características. Por ello, la finalidad de estos algoritmos es la búsqueda de similitudes dentro de los datos para formar grupos que tengan algún tipo de parecido. Los algoritmos de agrupamiento, como *k-means*, sería un ejemplo claro.

- **Híbridos.** Son una mezcla de técnicas de aprendizaje supervisado y no supervisado con el fin de aprovechar las bondades de cada tipo. Es el caso de las redes neuronales con función de base radial (RBF).

A lo largo de este capítulo se estudiarán los conceptos básicos de los modelos utilizados para la detección de autoría: SVM, una red neuronal MLP y *Naive Bäyes*. Por último, se explicará la decisión de usar estos tres modelos combinados.

5.2 Support Vector Machine (SVM)

5.2.1 Definición

El modelo SVM es un algoritmo de IA supervisado, cuyo objetivo es encontrar una función que permita clasificar los vectores de entrada con las características de cada documento. En nuestro escenario el número de entradas dependerá de la cantidad de características seleccionadas. En cambio, es un problema con dos posibles salidas: uno de los dos posibles autores. El objetivo de SVM para el reconocimiento de patrones es encontrar la función que cumpla (Boser *et ál* 1992):

$$f: R^N \rightarrow \{\pm 1\}\,. \qquad (1)$$

La obtención de esta función se realiza a partir de ejemplos con la forma (*x*,*y*) siendo *x* un vector cuya longitud es igual al número de características e *y* la clase que representa. En este caso *y* valdrá ± 1, ya que es un problema con dos clases. De este modo, se obtendrá que $f(x) = y$ para cada uno de los datos de entrenamiento (Boser *et ál* 1992).

Los clasificadores SVM están basados en la clase de hiperplanos (Hearst 1998):

$$(w \cdot x) + b = 0 \in R^N, b \in R. \qquad (2)$$

Con la función de decisión:

$$f(x) = sign[(w \cdot x) + b]. \quad (3)$$

El objetivo será buscar el hiperplano con el mayor margen de separación entre las dos clases (Vapnik 1998). La Figura 5.1 muestra gráficamente el procedimiento de obtención del mismo. El hiperplano óptimo será aquel ortogonal a la línea que conecta los puntos más cercanos entre las dos clases. Estos puntos, que están dentro de un círculo en la Figura 5.1, se denominan vectores soporte, debido a que imponen las condiciones del problema. De este modo, si estos elementos cambiasen se obtendría otro hiperplano. Además, la distancia entre los vectores soporte de cada clase con respecto al hiperplano es la misma y la suma de éstos es el *margen*, parámetro que se pretende maximizar.

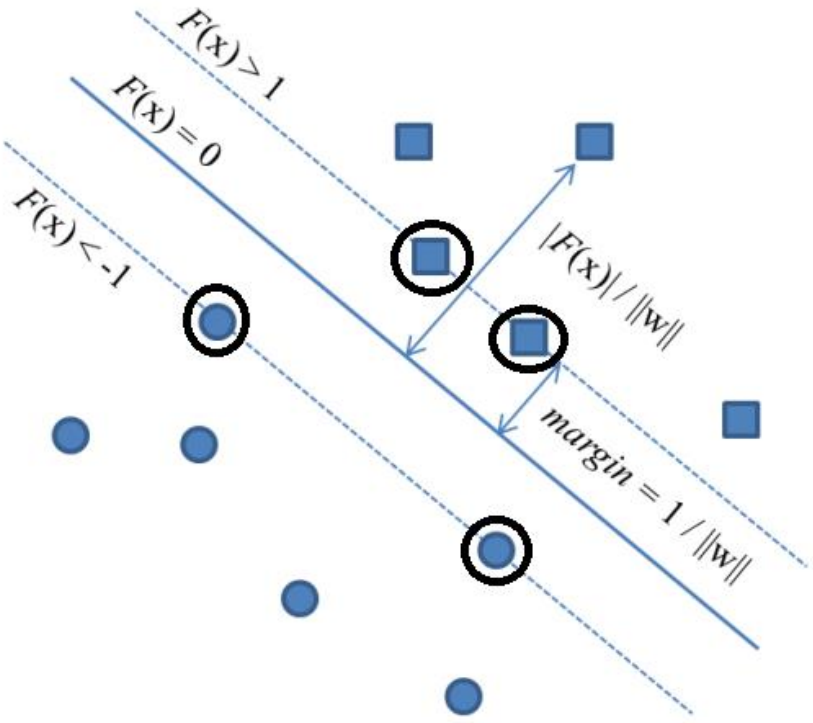


Figura 5.1: Gráfica que muestra el hiperplano óptimo que se obtiene en SVM para separar dos clases (Yu & Kim 2012).

Resolviendo el problema de optimización cuadrática (Hearst 1998) se obtiene que $w = \sum_i v_i \cdot x_i$, en términos del conjunto de características que se encuentran en el margen (Figura 5.1). De este modo, la función clasificadora quedaría de la siguiente manera:

$$f(x) = sign[(\sum_i v_i \cdot (x_i \cdot x)) + b]. \quad (4)$$

Uno de los problemas que aparece es la generalización de este algoritmo para casos no lineales. La solución es el mapeo del conjunto de datos no lineales a un espacio de mayor dimensión llamado *espacio de características* y construir en él el hiperplano de separación. Para realizar esta tarea se utiliza un núcleo que permita este proceso. La Figura 5.2 muestra una aproximación gráfica del proceso de mapeo. De esta manera, la definición del hiperplano buscado sería:

$$[w \cdot \varphi(x)] + b = 0 \in R^N, b \in R, \quad (5)$$

siendo $\varphi(x)$ una función no lineal que se encarga del mapeo de los datos al nuevo espacio de características. Entonces, la función clasificadora finalmente sería:

$$f(x) = sign\{ [\sum_i v_i \cdot K(x_i, x)] + b \}, \quad (6)$$

siendo $K(x_i, x) = \varphi(x_i) \cdot \varphi(x)$, que se corresponde con el núcleo utilizado.

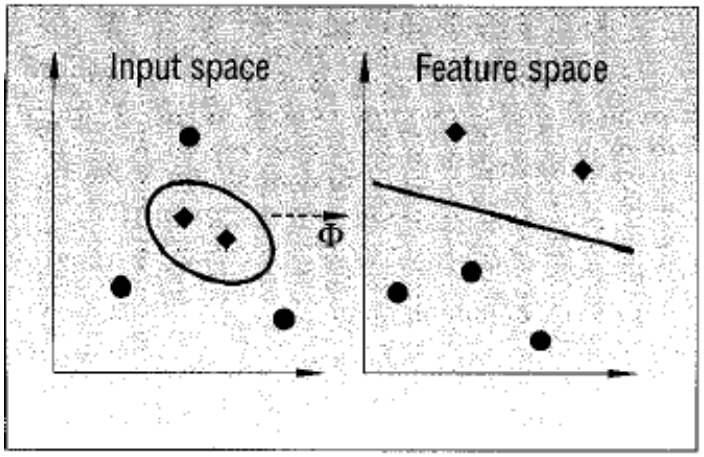


Figura 5.2: Mapeo del espacio inicial no lineal al espacio de características, donde se establecerá el hiperplano de separación (Hearst 1998).

5.2.3 Núcleos

La elección del núcleo determinará la forma en la que se proyectarán los datos de entrenamiento en el nuevo espacio de características, obteniendo diferentes hiperplanos que se corresponderá en el espacio inicial con fronteras de decisión más complejas. Hay que destacar aquellos que son más utilizados (Yu & Kim 2012):

- **Polinomial:** $k(a, b) = (a \cdot b + 1)^d$
- **Función de base radial (RBF):** $K(a, b) = \exp(-\gamma \|a - b\|^2)$
- **Sigmoidal:** $K(a, b) = \tanh(ka \cdot b + c)$

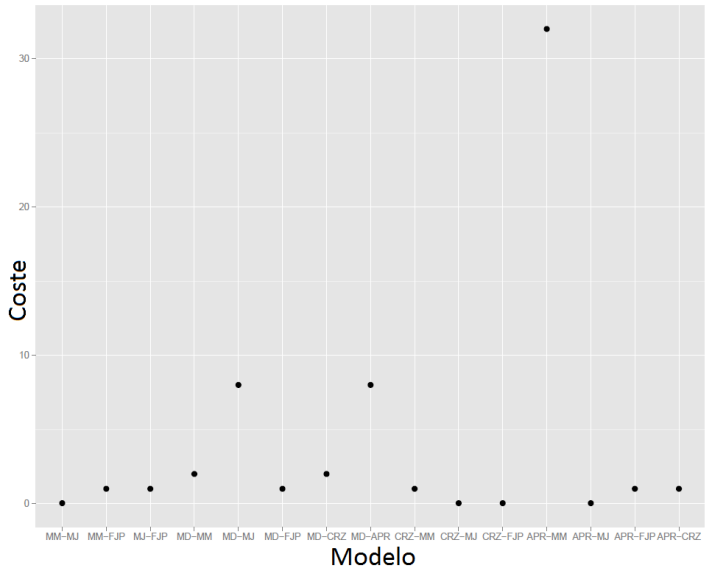
Como se puede observar cada núcleo lleva asociado algunos parámetros que posteriormente habrá que optimizar, para obtener la mayor precisión en el modelo de clasificación de SVM. A partir de un análisis de los 3 núcleos en la plataforma desarrollada en el TFG, se ha preferido utilizar el núcleo RBF debido a que en la mayoría de los casos la clasificación resultó ser más precisa.

5.2.3 Parámetros de SVM

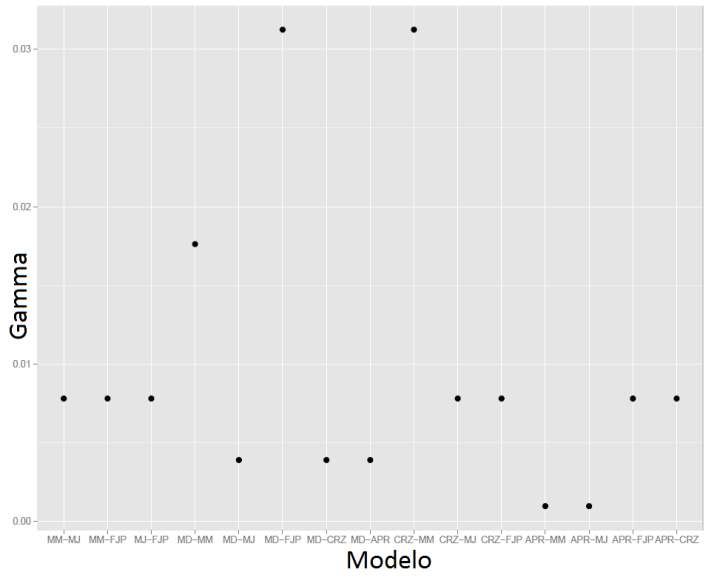
Para la generación del modelo SVM se han optimizado dos características: *gamma* y *coste* (Karatzoglou *et ál*. 2006; Hsu *et ál*. 2010): Para ello, se ha empleado la técnica de *k-fold-cross validation* que se explicará en el Apartado 5.5 y un banco de seis autores: Miguel Delibes Setién (MD), Carlos Ruiz Zafón (CRZ), Arturo Pérez Reverte (APR), Miguel Mora (MM), Miguel Jiménez (MJ) y Fernando J. Pérez (FJP), la información acerca del corpus de documentos utilizados se encuentra en el Apartado 7.2.1.

- ***Gamma*.** Parámetro propio del núcleo (RBF).
- **Coste.** Valor de la penalización de aquellas muestras que se encuentren en la clase que no les corresponde.

La elección de los rangos de las distintas características se ha realizado de manera inductiva. Para ello, se han realizado un conjunto de pruebas y se ha analizado los valores de estos elementos seleccionados. La Figura 5.3 muestra una gráfica de cada característica en la que se muestra el valor elegido de cada una con respecto a 15 modelos generados por el algoritmo. De esta manera, se selecciona un rango más acotado en relación a los resultados obtenidos.



(a)



(b)

Figura 5.3: Valores de las características de SVM obtenidos para 15 modelos distintos. (a) Valores del parámetro *coste*. (b) Valores obtenidos para el parámetro *gamma*.

Una vez analizados los resultados de la Figura 5.3 los rangos seleccionados para el proceso de optimización de características en SVM son:

- ✓ ***Coste*.** Los valores obtenidos en las 15 pruebas oscilan en la franja de 0 a 10 con un pico sobresaliente en 32. De esta manera, los límites escogidos tienen que contener todos los datos anteriores más una franja de error que permita cierto margen fuera del rango anterior. Así el límite inferior será 0.00001 y el superior será 100.

- ✓ ***Gamma*.** Para este parámetro los resultados recogidos se encuentran entre 0 y 0.04. El rango seleccionado tendrá presente también el margen de error y por ello los valores elegidos serán 0.0001 y 1.

5.3 Redes neuronales

5.3.1 Definición

Las redes neuronales artificiales (*artificial neural network*, ANN) surgieron con el objetivo de emular el comportamiento del cerebro humano, considerando que el fenómeno de la consciencia es causado por la compleja interacción de millones de neuronas (Hush & Horne 1992). Para representar el comportamiento de estas neuronas, se utilizan modelos matemáticos cuya arquitectura describe las formas de comunicación entre ellas. La Figura 5.4 muestra una primera clasificación de las neuronas artificiales cuya categorización dependerá de su relación con en el sistema (López & Fernández 2008):

- **Neuronas de entrada.** Reciben señales desde el entorno, provenientes de sensores o de otros sectores del sistema.

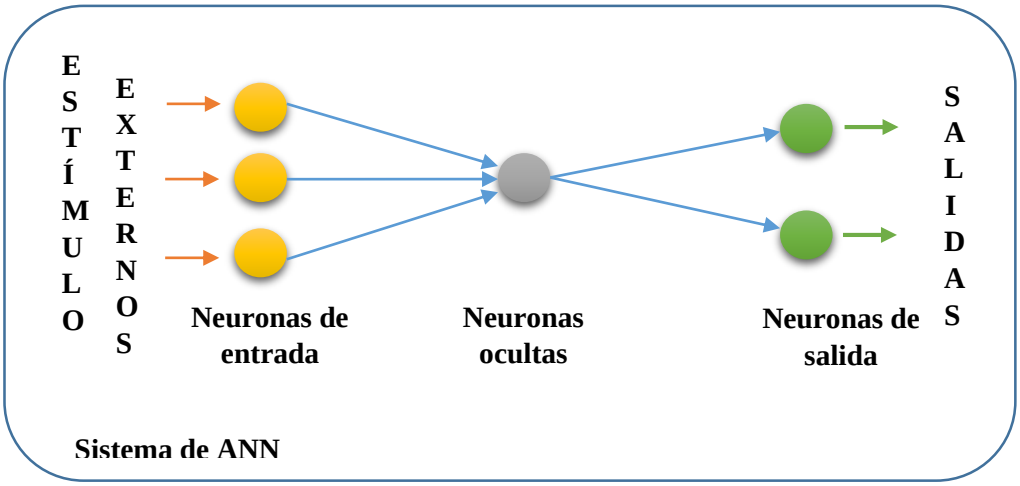


Figura 5.4: Esquema básico de los distintos tipos de neuronas y sus interacciones con el sistema

- **Neuronas ocultas.** Reciben estímulos y emiten salidas dentro del sistema, sin mantener contacto alguno con el exterior. En ellas se lleva a cabo el procesamiento básico de la información, estableciendo la representación interna de ésta.

- **Neuronas de salida.** Envían su señal directamente fuera del sistema una vez finalizado el tratamiento de la información y reciben estímulos del sistema.

El modelo matemático que caracteriza a las neuronas artificiales o perceptrones individuales contiene los siguientes elementos básicos que se plasman en la Figura 5.5 (López & Fernández 2008):

- ✓ Estado de activación inicial (EA), previo a la recepción de estímulos.

- ✓ Estímulos de entrada a la neurona con unos pesos asociados.

- ✓ Una función de activación (FA) o transferencia, cuyo objetivo es combinar el estado inicial de la neurona con los estímulos de entrada para establecer un nuevo valor de estado de activación.

- ✓ Una función de salida que transforma el estado final de activación en la señal de salida.

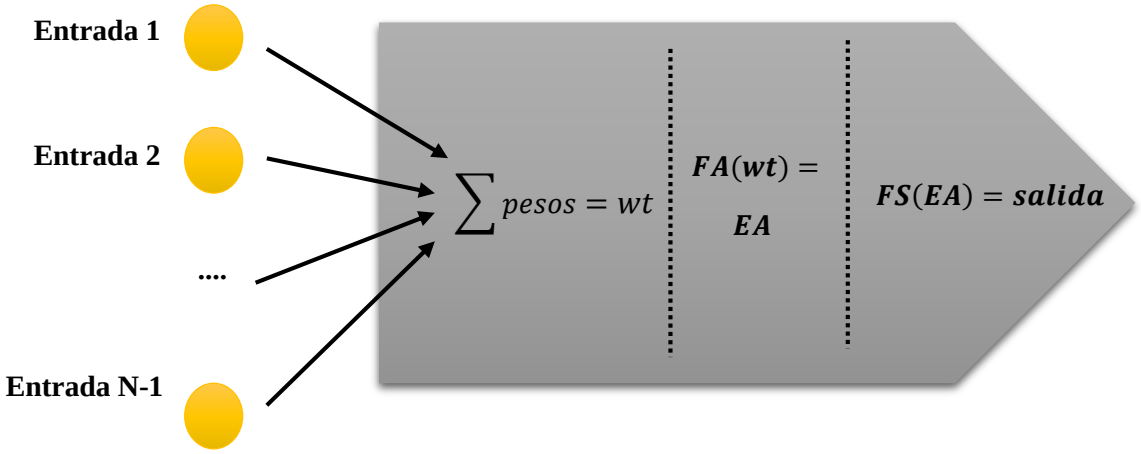


Figura 5.5: Esquema del funcionamiento de una neurona artificial o perceptrón individual

- ✓ Una regla de aprendizaje que determina la forma de actualización de los pesos de la red.

5.3.2 El perceptrón multicapa

El perceptrón multicapa o MLP es una arquitectura de ANN que se caracteriza por implementar en cascada capas de perceptrones, de forma que actuando conjuntamente pueden implementar complejas fronteras de decisión y expresiones booleanas arbitrarias (Hush & Horne 1992, 1993). La Figura 5.6 muestra un esquema de este tipo de redes neuronales compuesto por tres capas, dos de ellas ocultas.

Uno de los principales problemas de la red MLP es la definición de su arquitectura: la FA, la FS y sobre la topología de la red (López & Fernández 2008).

Referente a la topología de la red, la tarea de decisión tiene que ver con el número de capas que la integran y la cantidad de neuronas ocultas a incluir en cada una de ellas. Lippman (2007) demostró que con el uso de dos capas se podía implementar cualquier frontera de decisión convexa. Posteriormente añadió que mediante el aumento en la cantidad de neuronas en la capa oculta, se podía conseguir una aproximación arbitrariamente buena de cualquier frontera de decisión no lineal continua

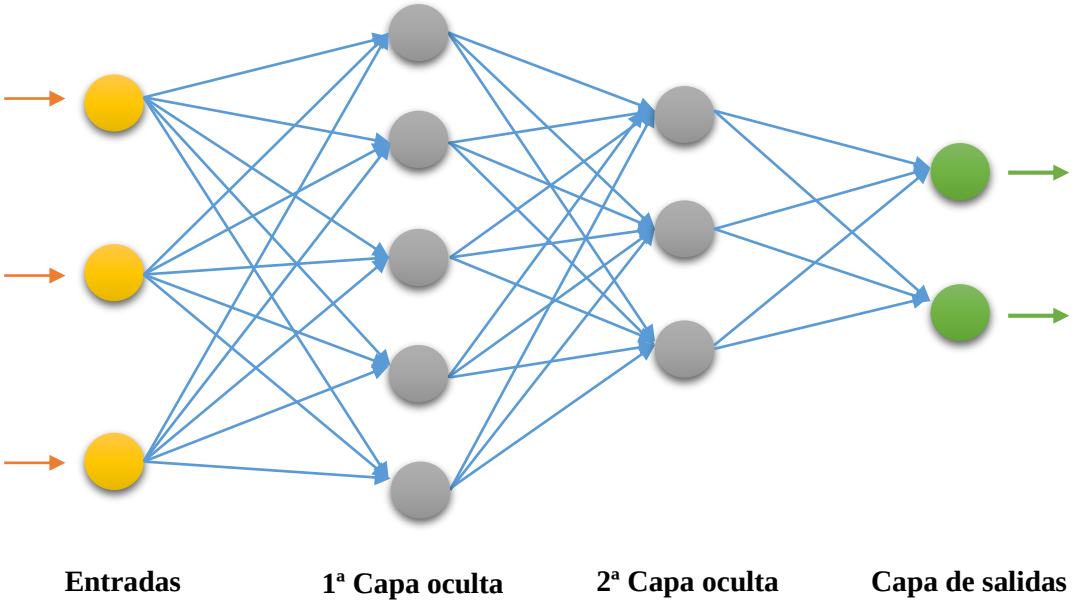


Figura 5.6: Esquema de una ANN MLP compuesta por 3 capas.

(Lippman 2007). De esta manera, por temas de optimización la ANN que se utilizará en el algoritmo de detección de autoría será una red neuronal MLP con dos capas, una oculta y otra de salida. El número de neuronas será un parámetro que se obtendrá de manera dinámica, según se explicará en el Apartado 5.3.3.

Por último queda mencionar el algoritmo de aprendizaje de la ANN, cuyo objetivo será el de configurar los distintos pesos de la red. En este trabajo se utilizará el método *backpropagation*, un algoritmo de búsqueda por gradiente, por su fácil uso dentro del entorno de desarrollo que utiliza la plataforma. Su uso implicará la optimización de la tasa de aprendizaje (Hush & Horne 1993).

5.3.3 Parámetros de una red neuronal

En la generación del modelo de red neuronal se ha tenido en cuenta la optimización de tres parámetros: la tasa de aprendizaje, el número de neuronas en la capa oculta y el parámetro *rang*.

- **Tasa de aprendizaje.** Este parámetro tiene que tener un compromiso entre un descenso rápido (valores elevados) en el que habrá casos que se ignoren ciertas características de la superficie de error y un descenso lento (valores pequeños) que podrían ocasionar el consumo de tiempos muy elevados durante la búsqueda del mínimo local (Hush & Horne 1993).

- **Número de neuronas en la capa oculta.** La elección del tamaño de la red es muy importante para formar aproximaciones arbitrariamente precisas. Si la red es demasiado pequeña, no será capaz de crear un modelo preciso del problema. En cambio, si es demasiado grande, podrían aparecer varias soluciones al problema creando modelos poco efectivos (Hush & Horne 1992,1993).

- **Rang.** Este parámetro es propio del paquete R que genera el modelo ANN (nnet package). Se utiliza para asociar aleatoriamente los pesos iniciales entre el rango ($-$ *rang* , *rang*). El valor que será elegido tenderá a la fórmula: $rang \cdot \max(|x|) \approx 1$, siendo x el vector que contiene el conjunto de datos. Esta fórmula es impuesta por el creador del paquete para R (nnet package).

Al igual que en la optimización del modelo SVM se utilizará la técnica *k-fold-cross validation* para obtener una mejor aproximación de la bondad de los valores obtenidos. De nuevo, para conocer los rangos seleccionados se ha realizado la misma prueba con 15 modelos distintos y se ha analizado los valores que toman estos parámetros. Las Figuras 5.7, 5.8 y 5.9 muestran los resultados obtenidos para el número de neuronas, la tasa de aprendizaje y el parámetro *Rang*, respectivamente.

Según los resultados obtenidos en esta prueba, los rangos seleccionados para la optimización son:

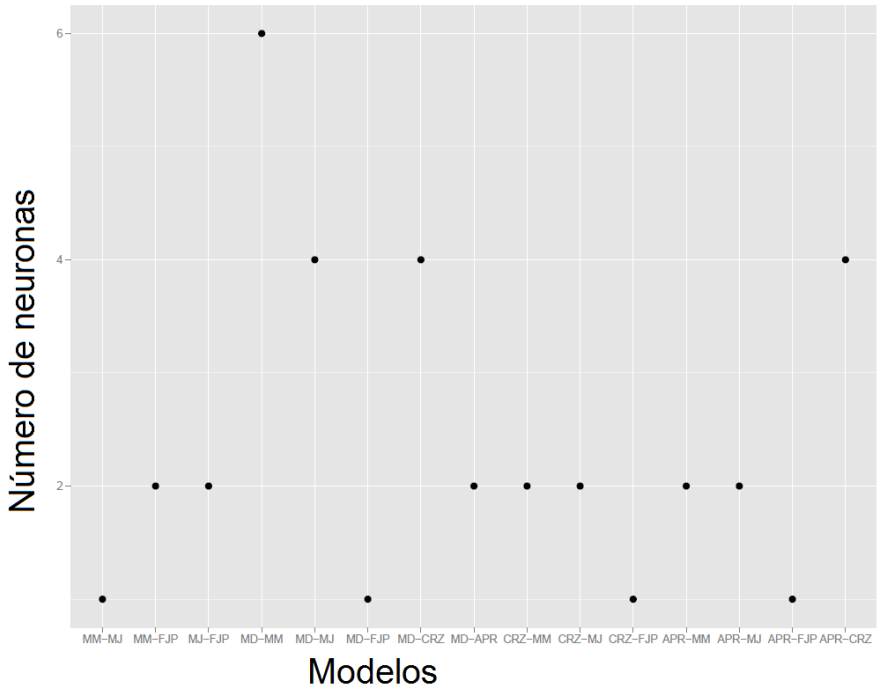


Figura 5.8: Valores de las características del modelo de ANN, obtenidos para 15 modelos distintos. Valores obtenidos para el número de neuronas

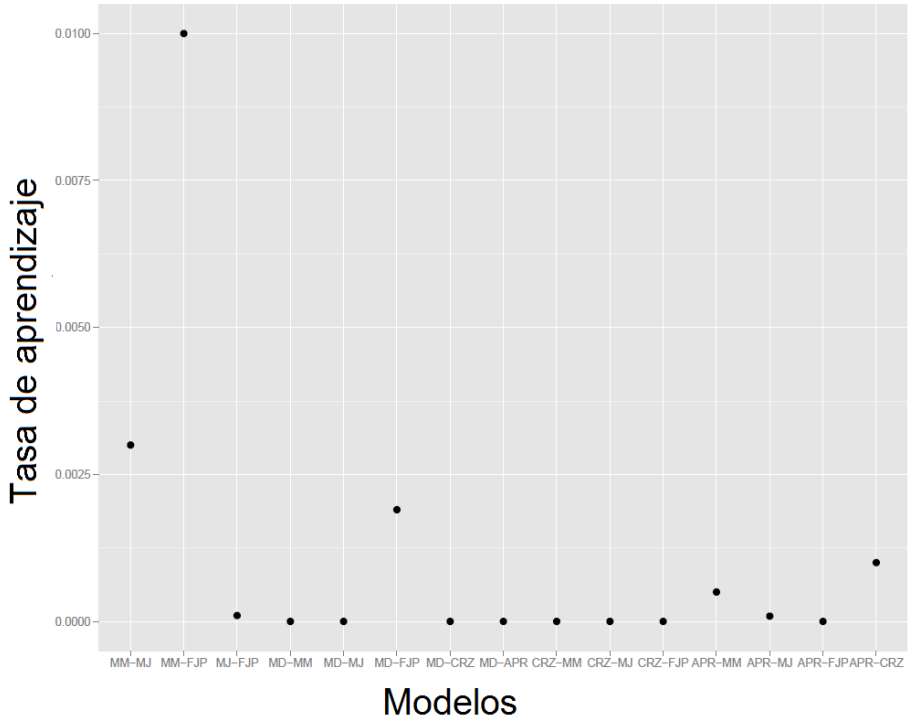


Figura 5.7: Valores de las características del modelo de ANN, obtenidos para 15 modelos distintos. Valores obtenidos para el parámetro *Rang*.

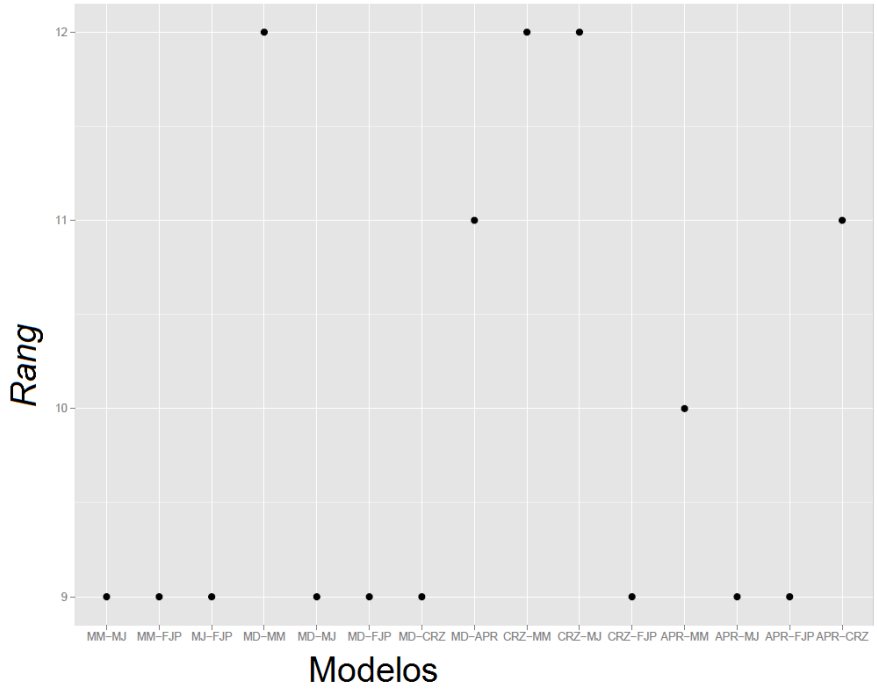


Figura 5.9: Valores de las características del modelo de ANN, obtenidos para 15 modelos distintos. Valores de la tasa de aprendizaje.

- ✓ **Tasa de aprendizaje.** Los valores obtenidos fluctúan entre el 0 y el 0.01 de esta manera el límite escogido será 0.0000000001 para el inferior y 0.1 para el superior.

- ✓ **Número de neuronas.** El número de neuronas que se ha obtenido se encuentra entre 1 y 6 por ello el rango elegido será: de 1 a 8 neuronas.

- ✓ ***Rang*.** En este caso el estudio realizado permite establecer un margen de error más preciso ya que según impone el paquete nnet (nnet 2014) el valor de este parámetro tiene que tender a $\frac{1}{\max(\|x\|)}$, de esta manera el rango de será: $[\,(\frac{1}{\max(\|x\|)} - 3\,)\,,(\frac{1}{\max(\|x\|)} + 3\,)\,]$. Siendo *max(x)* el máximo valor del conjunto de datos. El valor *3* se ha obtenido mediante la resta del máximo valor obtenido en la prueba, 12, con el mínimo, 9.

5.4 Näive Bayes

5.4.1 Definición

Estos clasificadores se basan en la presunción de que las características son condicionalmente independientes dentro de la clase etiquetada. De esta manera, la probabilidad de que una observación pertenezca a la clase *c* se rige bajo la siguiente fórmula (Murphy 2006):

$$p(x|y = c) = \prod_{i=1}^{D} p((x_i|y = c)\,, \qquad (1)$$

siendo *x* la observación que se quiere clasificar, la probabilidad de que pertenezca a la clase *c* será igual a la multiplicación de la probabilidad de cada una de las *D* características que integran x, lo que se denomina x_i, a pertenecer a la clase en cuestión. De este modo, el modelo elegirá aquella clase que tenga mayor probabilidad condicionada (Murphy 2006).

5.4.2 Parámetros de Näive Bayes

Para la creación del modelo de Näive *Bayes* se ha optimizado un parámetro: Laplace (Frank & Bouckaert 2006).

- **Laplace:** Este elemento sirve para prevenir que entradas desconocidas al modelo provoquen que la probabilidad de pertenecer a todas las clases sea nula.

Mediante el mismo análisis que en los casos anteriores (Apartado 5.2.3 y Apartado 2.3.3) el resultado obtenido en los 15 modelos para el parámetro *Laplace* ha sido constante, 0, esto es debido a que nunca va a existir una entrada desconocida que provoque que el modelo no funcione como es debido. Por ello, en la fase de creación de este modelo de clasificación el valor elegido para el parámetro *Laplace*, estará fijado en 0.

5.5 Proceso de optimización de características

5.5.1 K-fold-cross validation

El método k-fold-cross validation o estimación rotatoria permite evaluar un modelo a través de un conjunto de observaciones de entrenamiento (Kohavi 1995). Este procedimiento permite obtener unos resultados más precisos que se acercan más a la realidad. La Figura 5.10 muestra un esquema del proceso usando un valor para k igual a 5 que será el utilizado en el algoritmo desarrollado en el presente TFG. Esta decisión es debido a que las pruebas de entrenamiento de modelos se realizarán con un número reducido de muestras. De esta manera, con un k igual a 5 se consiguen grupos con una cantidad suficiente de observaciones en cada uno de ellos. La técnica consiste en dividir el conjunto de N muestras de entrenamiento en k subconjuntos de aproximadamente el mismo tamaño. Uno de ellos será excluido y el resto se utilizará para entrenar el modelo. El subconjunto que sobra se utilizará para probar el modelo. Este proceso se realizará k veces escogiendo en cada vuelta un subconjunto diferente para la fase de prueba. Con

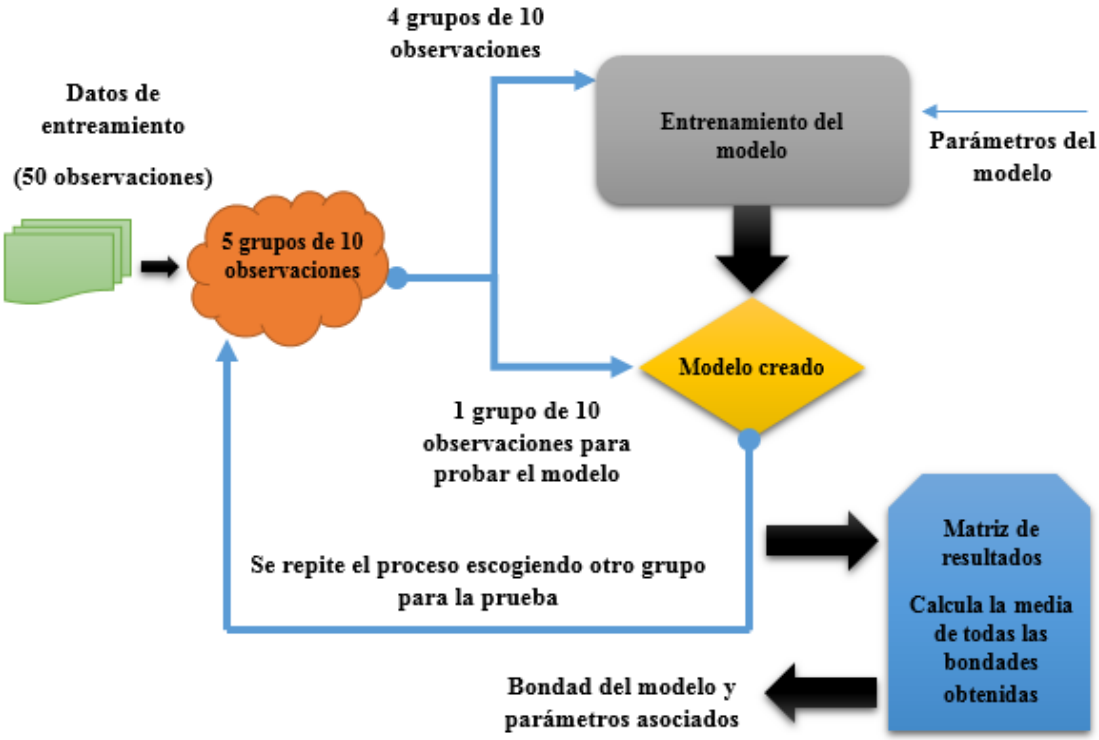


Figura 5.10: Esquema del método *k-fold-cross validation* que se ha tomado *k* = 5.

todos los resultados obtenidos se realiza la media, que se corresponderá con el resultado final de la prueba. De este modo, se generan resultados más precisos (Kohavi 1995).

5.5.2 Optimización granulada

Esta técnica tiene como objetivo la optimización granulada de los parámetros de cada uno de los modelos (Hsu *et ál*. 2010). La Figura 5.11 muestra un esquema del procedimiento. Basándose en la técnica de *k-fold-cross validation* propuesta en el Apartado 5.5.1 que permite conocer la bondad del algoritmo, la técnica de la "búsqueda en Grid" parte del establecimiento de un valor mínimo de precisión, denominado *Blim*, para poder finalizar la búsqueda. Se selecciona un rango de valores inicial para los distintos parámetros y mediante *cross-validation* se obtienen los valores dentro del rango

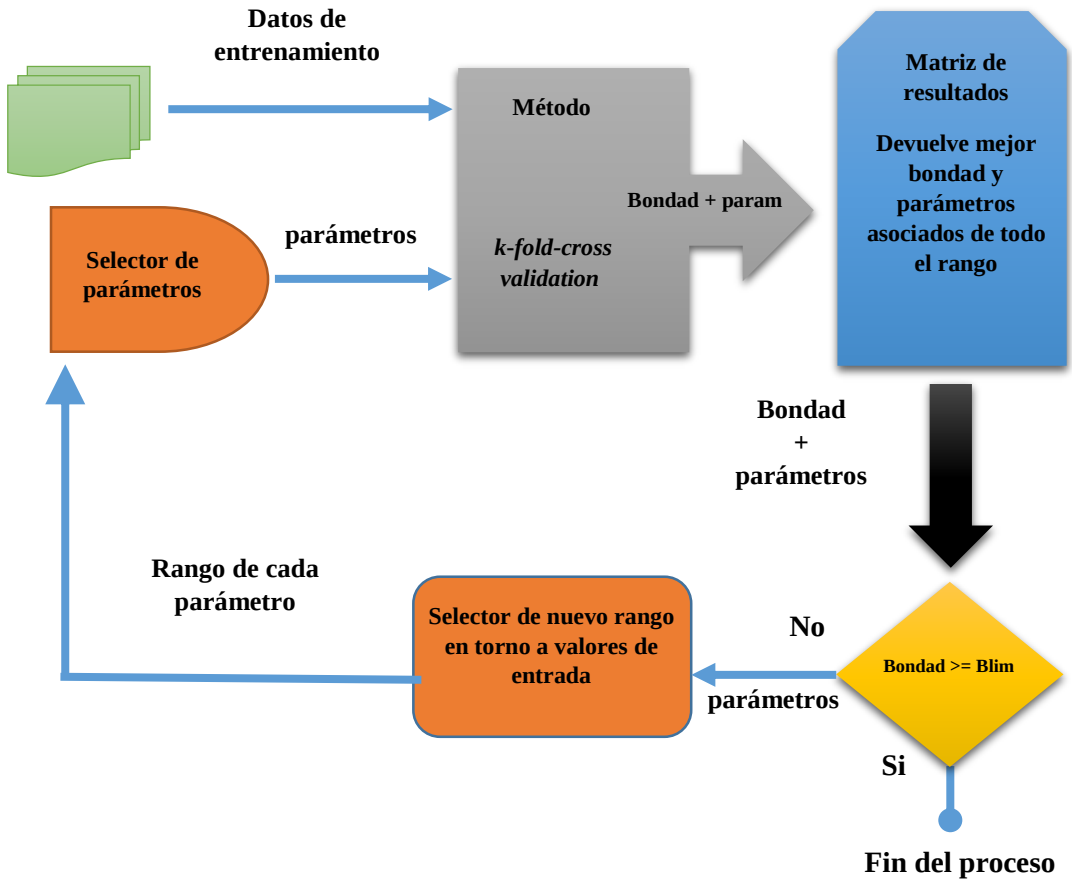


Figura 5.11: Esquema del proceso de optimización granulada

que hacen máxima la precisión. Si el resultado no es menor o igual que el límite impuesto se escogerá un nuevo rango para los distintos valores entorno a los generados por la técnica *cross-validation.* Así hasta tener una precisión que cumpla con las condiciones de *Blim*.

Este procedimiento comenzará recorriendo rangos muy generales (extensos) hasta alcanzar rangos más específicos que impliquen una mejora en la precisión de clasificación del modelo.

5.6 Combinación de modelos

La razón de utilizar tres modelos de IA es suavizar los fallos que puedan aportar cada uno por separado mediante un análisis del resultado en conjunto (Hansen & Salomon 1990; Dietterich 2000. El algoritmo desarrollado en el presente trabajo analizará dos mecanismos que se proponen en el TFG para la decisión de la autoría, cuyos esquemas se encuentran en la Figura 5.12:

- **Modelo de mejor esfuerzo (*best effort model*, BEM).** Esta técnica consiste en utilizar únicamente el modelo que mejor bondad generó en la fase de test.

- **Técnica de desempate.** Esta técnica utiliza los tres modelos para decidir el autor del documento analizado. Para este proceso de decisión, primeramente se escogerán los dos que mejor precisión hayan obtenido en la fase de test, los modelos primarios (MP), y el restante se le denominará el modelo secundario (MS) que será utilizado en la fase de desempate. El procedimiento es muy simple: cuando los MP elijan cada uno a un autor diferente, el documento se clasificará según lo indicado por el MS. Sin embargo, cuando ambos MP estén de acuerdo en la autoría, la decisión del MS no será tomada en cuenta.

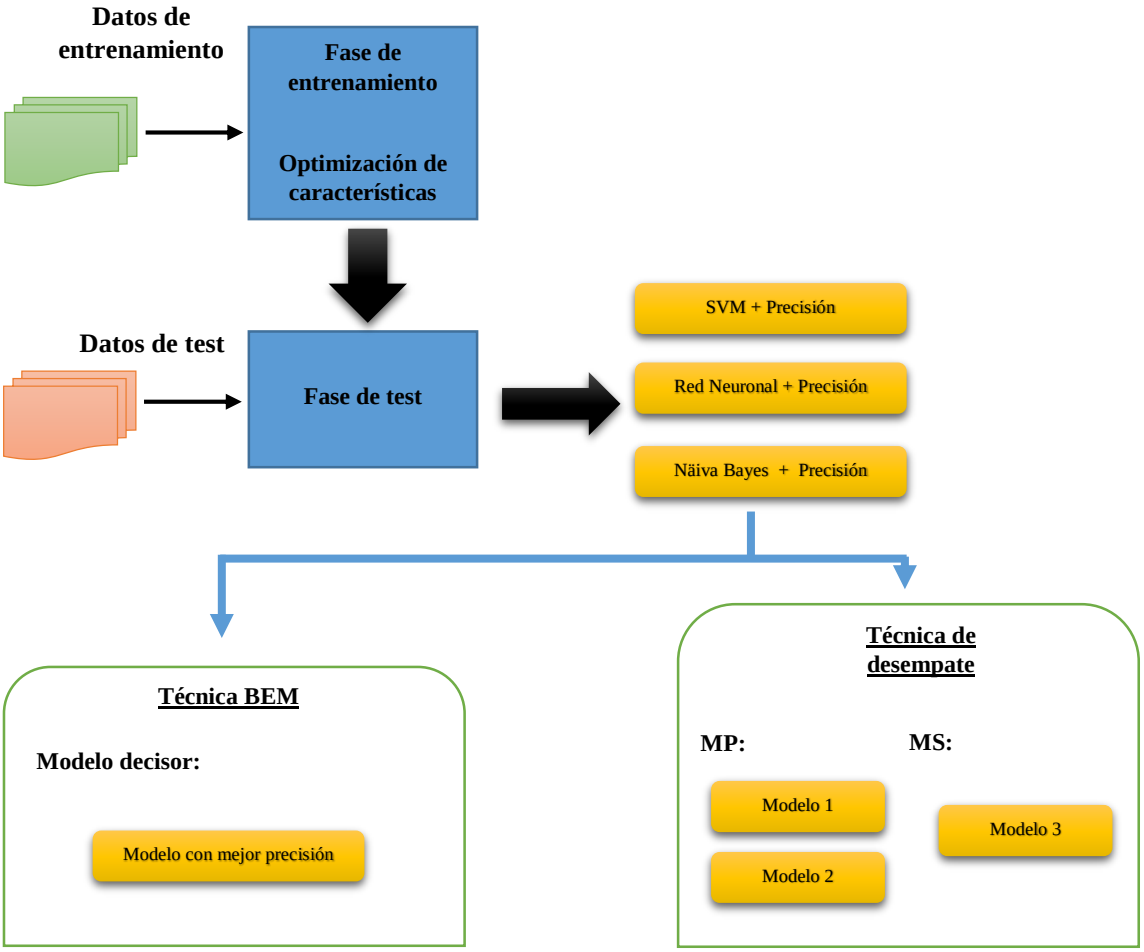


Figura 5.12: **Esquema del proceso de selección de modelos para cada una de las técnicas propuestas en el TFG: BEM y la técnica de desempate.**

Capítulo 6

Desarrollo software

6.1 Introducción

El contenido de este capítulo se divide en dos grandes bloques. El primero de ellos tiene que ver con las tecnologías que se han utilizado en el proyecto. En el segundo se explicará la arquitectura del algoritmo, profundizando en cada uno de los módulos que lo componen.

6.2 Tecnologías del proyecto

Una de las tareas más importantes para un programador cuando se plantea implementar una solución a un problema real, es la elección del lenguaje de programación que mejor se adapte a sus necesidades. Lo primero que hay que escoger es el paradigma de programación que se va a utilizar. Entre ellos se encuentran los paradigmas imperativos, funcionales, lógicos y los orientados a objetos (Louden & Lambert 2012).

El paradigma que se utilizará será el orientado a objetos y el lenguaje escogido es *Java* (*Java* API 2014) debido a su portabilidad para el conjunto de sistemas operativos. La razón del paradigma es por la sencillez y la afinidad que aporta en el proceso de implementación del algoritmo. Además, se utilizará una segunda tecnología, *R* (Proyecto *R* 2014), que funcionará conjuntamente con la primera y permitirá desarrollar los métodos basados en *text mining* y todas las funciones de creación y manipulación de los distintos modelos de IA. La elección del lenguaje R es por la facilidad para manipular modelos de IA.

6.2.1 Lenguaje R

R es un lenguaje de programación muy potente útil para resolver problemas que requieren un alto grado de computación estadística, es la razón por la que se ha escogido (Proyecto *R* 2014). Es un proyecto de código libre en el que investigadores desarrollan paquetes con funciones ya implementadas. En este trabajo se utilizarán una serie de paquetes con métodos que implementan técnicas de *text mining* y que permiten crear y manipular los distintos modelos de IA que se utilizarán: SVM, MLP y *Näive Bayes.*

Para el conjunto de técnicas de *text mining* se utilizará el paquete *tm* (Tm package 2014). En el desarrollo de los distintos modelos de IA se utilizará los siguientes paquetes: *nnet* (Nnet package 2014), para la gestión del modelo MLP, y *e1071* (*e1071 package* 2014), para los modelos SVM y *Näive Bayes.*

6.2.2 Java

Java es el lenguaje de programación elegido para desarrollar el proyecto (*Java* API 2014). La principal causa de su elección es porque es un lenguaje portable. De esta manera, con sólo tener instalada la máquina virtual de java, el programa puede funcionar en distintos sistemas operativos. Es orientado a objetos y su API ofrece multitud de posibilidades para el programador. Además, existe una librería que permite ejecutar *scripts* programados en R dentro del código ejecutable de programas basados en dicho lenguaje.

6.2.3 Combinación de tecnologías

Los lenguajes de programación están diseñados para satisfacer unas necesidades. Éstas son muy diferentes y es por ello que existen lenguajes que están mejor preparados para ciertos escenarios. Es el caso por ejemplo de ADA, un lenguaje orientado al desarrollo de sistemas basados en tiempo real (Burns & Wellings 2009). En cambio, R, como ya se ha dicho, está pensado para resolver problemas que requieran un cómputo estadístico muy complejo (Proyecto *R* 2014). La necesidad para el programador es que si quiere optimizar al máximo sus programas deberá adaptarse y utilizar el lenguaje que mejor convenga al contexto del problema.

Así, la solución de la atribución de autoría exigirá dos lenguajes de programación. El primero de ellos, Java, que será el que se utilice para el desarrollo de la plataforma que manejará los recursos y mostrará los resultados. Por otro lado, R, será utilizado para gestionar los textos y los distintos modelos de IA. De esta manera, se consigue la optimización máxima de las distintas tareas. La Figura 6.1 muestra un esquema del

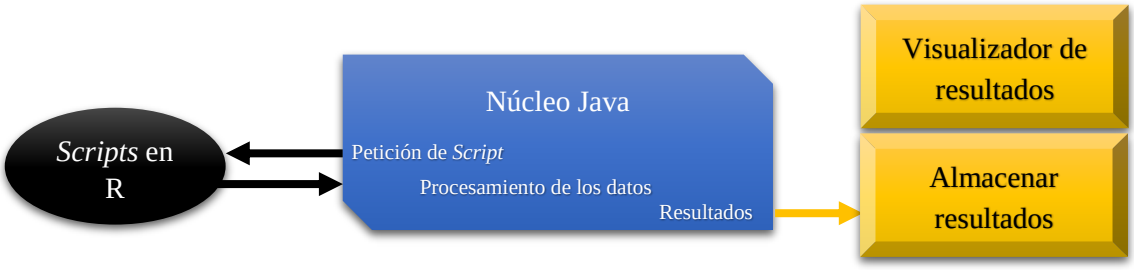


Figura 6.1: Esquema del funcionamiento de la plataforma

funcionamiento, donde el núcleo Java de la plataforma solicitará la ejecución de *scripts* en R y mostrará los resultados obtenidos al usuario.

6.3 Entornos utilizados

6.3.1 Ubuntu

El sistema operativo que se ha utilizado para el desarrollo del presente trabajo ha sido un Ubuntu 12.04 LTS con una arquitectura de 32 bits (Ubuntu 2014). Esta decisión es debido a requerimientos de compatibilidad con la herramienta NLP *Freeling* y también al conocimiento previo sobre la versión.

6.3.2 RStudio

Rstudio es un espacio de trabajo para el desarrollo de *scripts* en lenguaje. Se ha utilizado la versión 0.98.501 (Rstudio 2014). Su utilización hace más sencilla la tarea de generación de *scripts* mediante una interfaz de usuario muy sencilla e intuitiva. Es por ello que se ha utilizado para el desarrollo del algoritmo. De esta manera, para la generación de los *scripts* no es necesario interactuar directamente con la consola de comandos de R.

6.3.3 Eclipse

Como entorno de desarrollo para la implementación del núcleo Java de la plataforma se ha elegido la versión 3.7.2 *Indigo* (Eclipse 2014). Fue la versión que se instaló en la máquina virtual donde se comenzó el desarrollo del TFG. Es compatible con el entorno de desarrollo de Java (*Java development kit 7*, JDK 7). La elección de esta plataforma es por el conocimiento previo de uso adquirido durante la carrera.

6.4 Estructura del algoritmo

6.4.1 Sistema de directorios

El directorio raíz de la plataforma implementada en el TFG se dividirá en una serie de subdirectorios, que administrará el núcleo Java, para guardar y cargar información que intervenga en el flujo natural de operaciones de la plataforma. Los directorios son:

- ***Chooser*.** Se almacena toda la información que está relacionada con los modelos de IA. Por ello se encontrarán los ficheros que almacenan los TAGS, las FW, los autovectores (para PCA si procede) y los propios modelos. Asimismo, dentro de este directorio está la carpeta *pick* que se utiliza para cargar los textos de los que se analizará la autoría con respecto a un conjunto de autores.

- ***ScriptR*.** Esta carpeta contiene los *scripts* en R que ejecutará el núcleo Java.

- **Documentos.** En este lugar se guardan los documentos que se utilizarán para los procesos de test y entrenamiento. Estos documentos se encuentran almacenados en carpetas cuyo nombre es el asociado al autor que los creó.

- ***Logs.*** Carpeta que contiene las trazas que informan sobre el flujo de operaciones de la plataforma.

- ***Properties.*** Lugar para agrupar los distintos ficheros de propiedades que cargará el núcleo Java cuando sea necesario. En el caso del fichero *nucleo.properties*, se pueden configurar distintos parámetros del algoritmo que hará posible la realización de pruebas diferentes sin tener que volver a compilar el proyecto.

6.4.2 Arquitectura

La arquitectura de la plataforma se divide en cinco módulos que funcionan individualmente pero que se complementan. De esta manera, la salida de cada uno de ellos se utilizará como entrada para otro de los módulos. Así se distinguen: la etapa de preprocesado, el extractor de características, el selector de características, el extractor de autovectores, el creador de modelos y, por último, el decisor.

6.4.3 Etapa de preprocesado

Esta etapa tendrá como objetivo la adaptación de los documentos que analizará la plataforma en sus distintas fases: entrenamiento, *test* y decisión. Someterá cada texto a una serie de transformaciones que permitirán el correcto funcionamiento de las etapas siguientes. Para ello, primero buscará los DP de los autores especificados como argumento de entrada dentro del directorio *documentos*. En este lugar, el núcleo Java buscará aquellas carpetas que tengan como nombre uno de los introducidos. Allí se encontrarán los DP divididos en dos subdirectorios: *test* y *train*. De esta manera, la plataforma podrá diferenciar la finalidad de cada uno de ellos: fase de entrenamiento o fase de *test*. Posteriormente los cargará y ejecutará un *script* de R (*Transform.R)* que realizará las adaptaciones pertinentes. Una vez obtenidos los DA el núcleo realizará dos tareas. Primero guardará los DA de *test* y los de entrenamiento en dos subdirectorios diferentes: *results_test* y *resulst_train*, respectivamente. Asimismo, cada DA será analizado lexicalmente por la herramienta NLP *Freeling* que devolverá la lista de *TAGS* asociada. Posteriormente, el núcleo se encargará de crear la matriz de *TAGS* (MTG) que

agrupa los análisis de los DA de cada autor en ficheros separados. Por último, almacenará esta información dentro del directorio de cada autor.

La Figura 6.2 muestra un esquema del módulo de preprocesado en el que se han introducido los nombres *Eduardo* y *Ryohei* para adaptar sus documentos. Primero carga los DP a partir de los directorios correspondientes de cada autor. Luego, ejecuta el *script* que los transforma a los DA para ser almacenados en las carpetas *Results*_test y *Results_train* según su procedencia. A su vez, con ellos el núcleo obtiene la MTG de cada uno de los autores mediante la herramienta *Freeling*.

6.4.4 Extractor de características

Este módulo se encarga de la extracción de las características que mejor describan el estilo de escribir de los autores. En el Capítulo 4 se explica detalladamente aquellas que han sido seleccionadas como indicadores de estilo para el presente proyecto: las FW y los *TAGS*.

Para el correcto funcionamiento de este módulo, la etapa de preprocesado ha tenido que haber sido ejecutada previamente, ya que necesitará los ficheros generados en dicha fase (DA y MTG).

El flujo de operaciones se divide en dos bloques principales: la obtención del fichero de *TAGS* y la obtención del fichero de FW. Juntos constituirán la firma distintiva de los dos autores.

- **Parte *TAGS*.** Este bloque tendrá como misión la obtención del fichero *firma.dat*, donde están almacenados los *TAGS* que mejor diferencian a los dos autores. El primer paso consiste en cargar la configuración pertinente a este proceso, que se encuentra en el fichero de propiedades del núcleo (Apéndice B.2). De esta manera, la plataforma estará configurada correctamente para obtener el fichero. El siguiente paso es cargar y concatenar las matrices de *TAGS* obtenidas en la etapa de preprocesado.

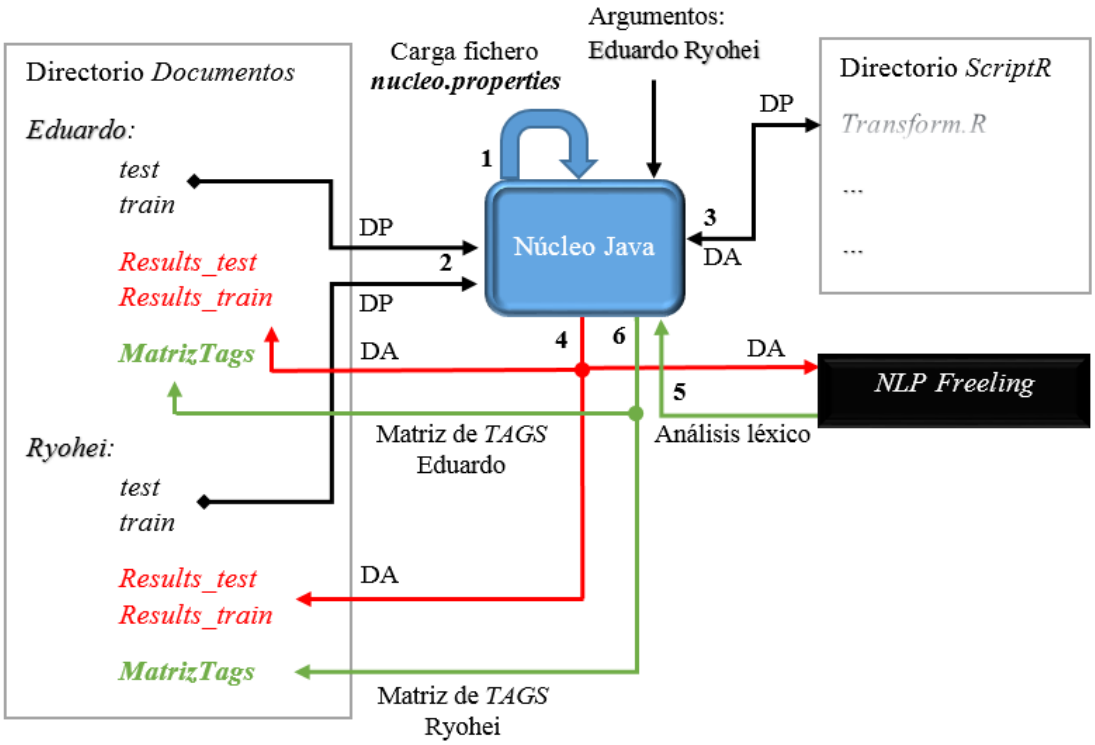


Figura 6.2: Esquema de la etapa de preprocesado

Esta información será procesada por un *script* en R (*Ngra6-outliers-new.R*) que escogerá los *TAGS* y almacenará el resultado en el directorio *Chooser/Firma/nombre_autor1-nombre_autor2/*. De este modo, cuando queramos obtener el fichero de TAGS de estos dos autores, sólo habrá que hacer referencia a este directorio y dentro de él a *firma.dat*.

- **Parte FW.** El objetivo de esta parte es generar el fichero *functionWords.dat*, en el que se encuentra la lista de los FW más representativos de los autores. Para ello, se cargarán y concatenarán en la plataforma los DA de cada autor. Con estos datos se ejecutará un *script* (*functionWords.R*), que construirá la lista de FW y almacenará el resultado en el directorio */Chooser/Function Words/nombre_autor1-nombre_autor2/*. De esta manera, */Chooser/Function Words/*, será el lugar donde se almacenen los ficheros *functionWords.dat*, correspondientes a un par de autores.

La Figura 6.3 muestra un esquema de esta etapa continuando con los autores propuestos en la Figura 6.2. Aparecen dos caminos distintos diferenciados por dos colores. El rojo identifica al camino que se encarga de la obtención de la MTG. En cambio, el verde tiene como objetivo obtener la lista de FW.

6.4.5 Selector de características

El selector de características es la parte encargada de cargar los ficheros de la firma de dos autores obtenidos en la etapa de extracción (FW + *TAGS*). Posteriormente, calculará las frecuencias asociadas a cada indicador a partir del DA analizado.

De nuevo, este módulo sólo funcionará si se han obtenido los ficheros en la etapa de extracción de características. La diferencia con el resto de procesos es que éste va a ser utilizado dentro de los módulos que necesiten obtener esta información y, por tanto, no se ejecutará por sí solo, sino cuando lo requiera otro proceso.

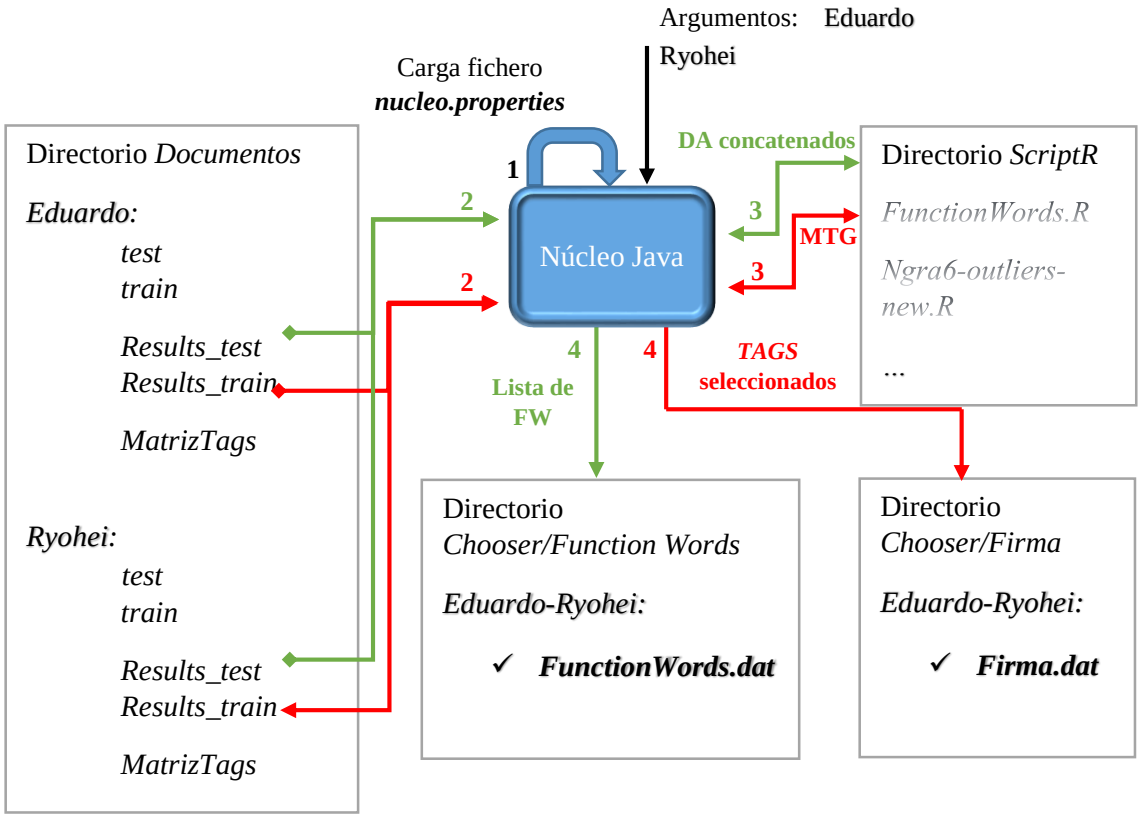


Figura 6.3: Esquema de la etapa de extracción de caracterísitcas. En rojo el bloque *TAGS*. En verde el bloque FW.

La Figura 6.4 muestra un esquema del flujo de operaciones. Primero cargará del fichero de propiedades del núcleo (Apartado B.2) la configuración necesaria. A continuación, buscará los ficheros de la firma (*FunctionWords.dat* y *firma.dat*), correspondientes a los dos autores que se introducen como argumento. Por último, calculará los pesos de los distintos elementos estilísticos y devolverá dos listas: una con los pesos de las FW y otra con los correspondientes a los *TAGS*.

6.4.6 Extractor de autovectores

El extractor de autovectores está integrado en la etapa de extracción de características. Sin embargo, debido a que se ejecuta una vez finalizada esta fase y, que además utiliza el módulo selector de características, se incluye como una sección aparte.

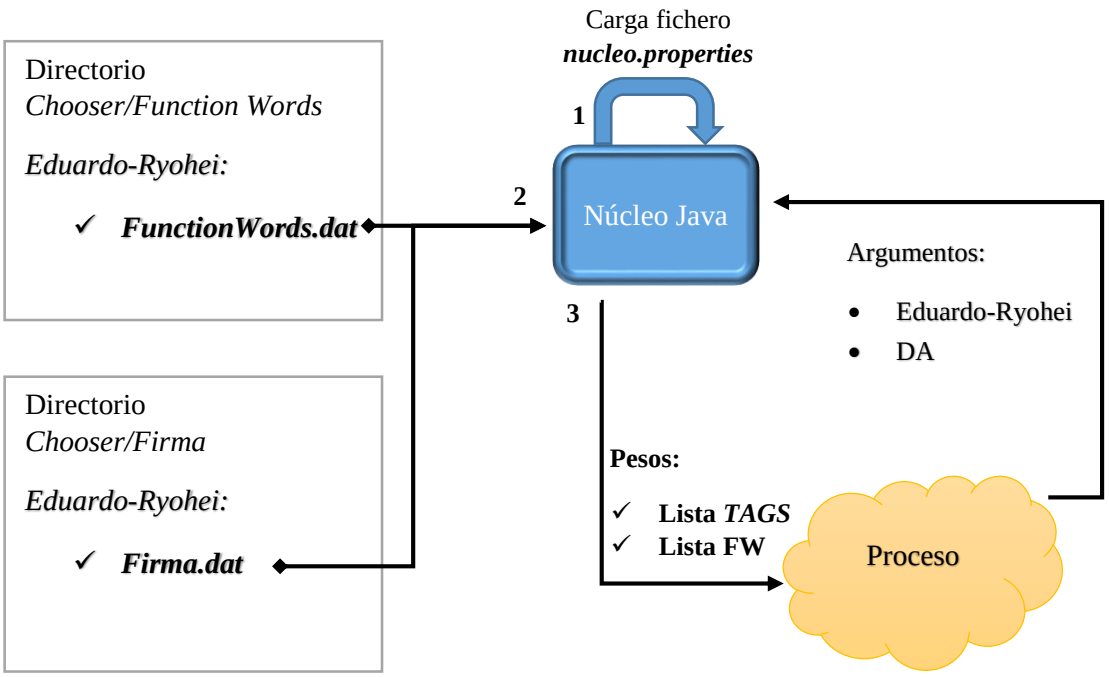


Figura 6.4: Esquema de la etapa de selección de características.

El objetivo es la obtención de los autovectores a partir de los pesos asociados a los DA, según la firma obtenida (FW + *TAGS*). Toda la información relativa al análisis PCA se encuentra en el Apartado 4.5.

Para la generación de los autovectores, primero se cargan los DA de entrenamiento correspondientes a cada autor. Con ellos se obtienen los pesos asociados, utilizando el módulo de selección de características. Luego, con esta información generará las componentes principales (PC) a través de la ejecución del *script PCA.R.* El número de PC vendrá determinado por el fichero de configuración de la plataforma (Apéndice B.2). Por último, se almacenará en el directorio */Chooser/PCA/nombre_autor1-nombre_autor2/* el fichero *PCAvector.Rdata*, en el que se encuentran los PC. La Figura 6.5 muestra un esquema para obtener los PC.

6.4.7 Creador de modelos

Este módulo se encargará de la creación de los tres modelos basados en IA: SVM, red neuronal MLP y *Näive Bayes.* Para el correcto funcionamiento de este proceso, las fases de preprocesado y de extracción de características han tenido que haber sido ejecutadas.

El creador de modelos divide su trabajo en dos etapas: (i) la primera es la creación y entrenamiento de los modelos y (ii) la segunda es la comprobación de la bondad de los mismos.

- **Creación y entrenamiento del modelo.** Primero leerá del fichero de propiedades del núcleo (Apéndice B.2) la configuración necesaria para esta etapa, en la que se incluye el número de características de cada tipo (FW y *TAGS*) y si se realizará análisis PCA. Posteriormente, cargará en el núcleo los DA tanto de *test* como de entrenamiento. Se obtendrán los pesos de cada *TAG* y FW de los autores obtenidos en la fase de extracción de características mediante el módulo selector de características.

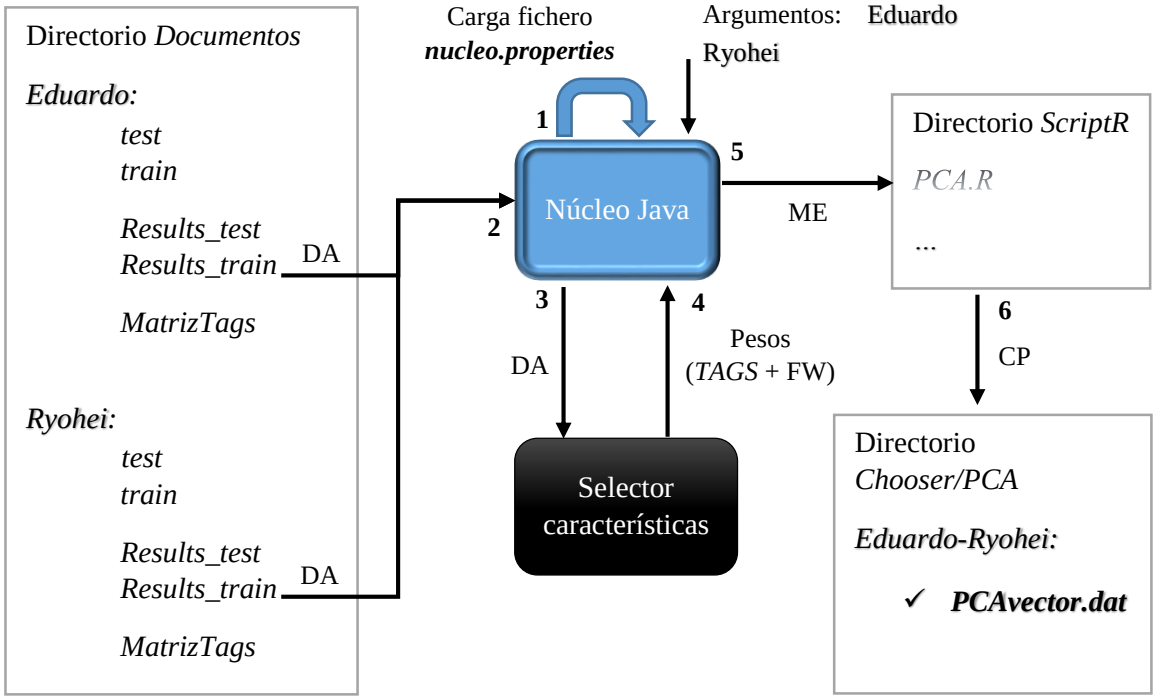


Figura 6.5: Esquema de la extracción de autovectores

Esta información se dividirá en dos ficheros: uno para guardar los pesos de los DA de entrenamiento y otro para los de *test*. El siguiente paso consiste en ejecutar el *script modelv4.R*. Este código primero guardará los ficheros con los pesos en dos matrices, una para la fase de *test* (MT) y otra para la de entrenamiento (ME). Para esta etapa únicamente se utilizará la ME. Una vez generadas las matrices ME y MT, lo primero es llevar a cabo la optimización de los parámetros de cada modelo. Este procedimiento está explicado en el Apartado 5.5. Para finalizar, se crearán los modelos en función de si se realiza análisis PCA. En el caso de que no se haga, se utilizan la ME y los parámetros obtenidos en el paso anterior. Si se realiza análisis PCA, se cargan los autovectores y se ejecutará el producto interno entre la ME y los autovectores, obteniendo la matriz de pesos de PCA de entrenamiento (MPPE). De esta manera, se crearían los modelos utilizando los parámetros optimizados y la MPPE.

- **Bondad del modelo.** Esta etapa comenzará una vez creados los modelos y su objetivo será el de evaluar cada uno de ellos utilizando los pesos

almacenados en la MT. En el caso de utilizar análisis PCA, habría que obtener la matriz de pesos de PCA de *test* (MPPT) de la misma manera que se obtuvo la MPPE y evaluar cada modelo con la MPPT. Finalmente, con los resultados de la prueba, se elabora un vector de errores que ordena de mayor a menor precisión los tres modelos generados. Por último, guardará los modelos y el vector de precisión en el directorio *Chooser/Modelos/nombre_autor1-nombre_autor2*

La Figura 6.6 muestra el flujo de operaciones del creador de modelos.

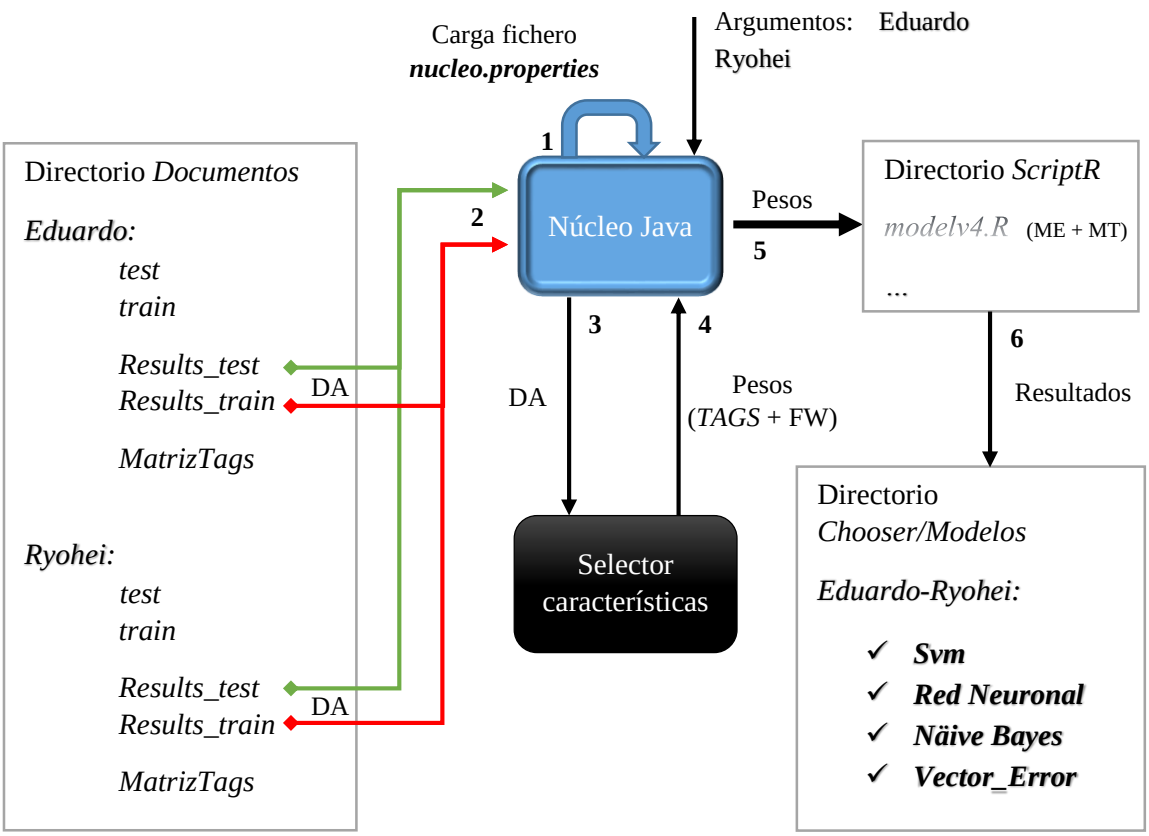


Figura 6.6: Esquema del módulo de creación de modelos

6.4.8 Decisor

El módulo de decisión es el encargado del proceso de elección de un autor entre un conjunto de ellos para un documento con la autoría disputada. Este proceso está implementado para funcionar con un número ilimitado de sujetos lo que se llamará el banco de autores (BA). Asimismo, los documentos que será procesada su autoría (DSA) serán aquellos que se encuentren en la carpeta */Chooser/pick*. El requisito necesario para el correcto funcionamiento es que se hayan obtenido los modelos de todas las posibles combinaciones de dos autores dentro del BA.

El algoritmo de decisión seguirá los siguientes pasos para la elección de la autoría asociada al DSA. Primero, la plataforma cargará en un vector los autores introducidos como argumento de entrada; serán los que conformen el BA. Con este vector, elaborará las posibles combinaciones de dos autores, para posteriormente realizar con cada una de ellas la siguiente operación: ejecutar el *script pickerv2.R* cuyo objetivo es devolver el nombre del autor que considera que es el creador del documento entre los dos posibles. Para esta elección utilizará una de las dos técnicas explicadas en el Apartado 5.6.

El núcleo Java guardará el resultado de cada combinación en una lista (LR) y, cuando se hayan obtenido todas, pasará a la fase de elección del autor mediante la siguiente técnica que se propone con el TFG. Para aplicar este método, la plataforma tendrá completada la LR para todas las posibles combinaciones de autores.

La LR tendrá la siguiente forma: *par-de-autores*|*autor-elegido*. Un ejemplo sería: *Edurado-Ryohei*|*Eduardo*. De esta manera, el núcleo sabe que para el DSA que se está analizando, entre *Eduardo* y *Ryohei* es más probable que sea *Eduardo* el autor del documento. Este proceso se realiza con todos los grupos de dos autores y se seleccionará el autor que haya vencido sobre el resto. En caso de que no exista un vencedor sobre el resto, la plataforma no asignará autoría al documento, el resultado será desconocido.

La Figura 6.7 muestra un esquema del método de decisión propuesto.

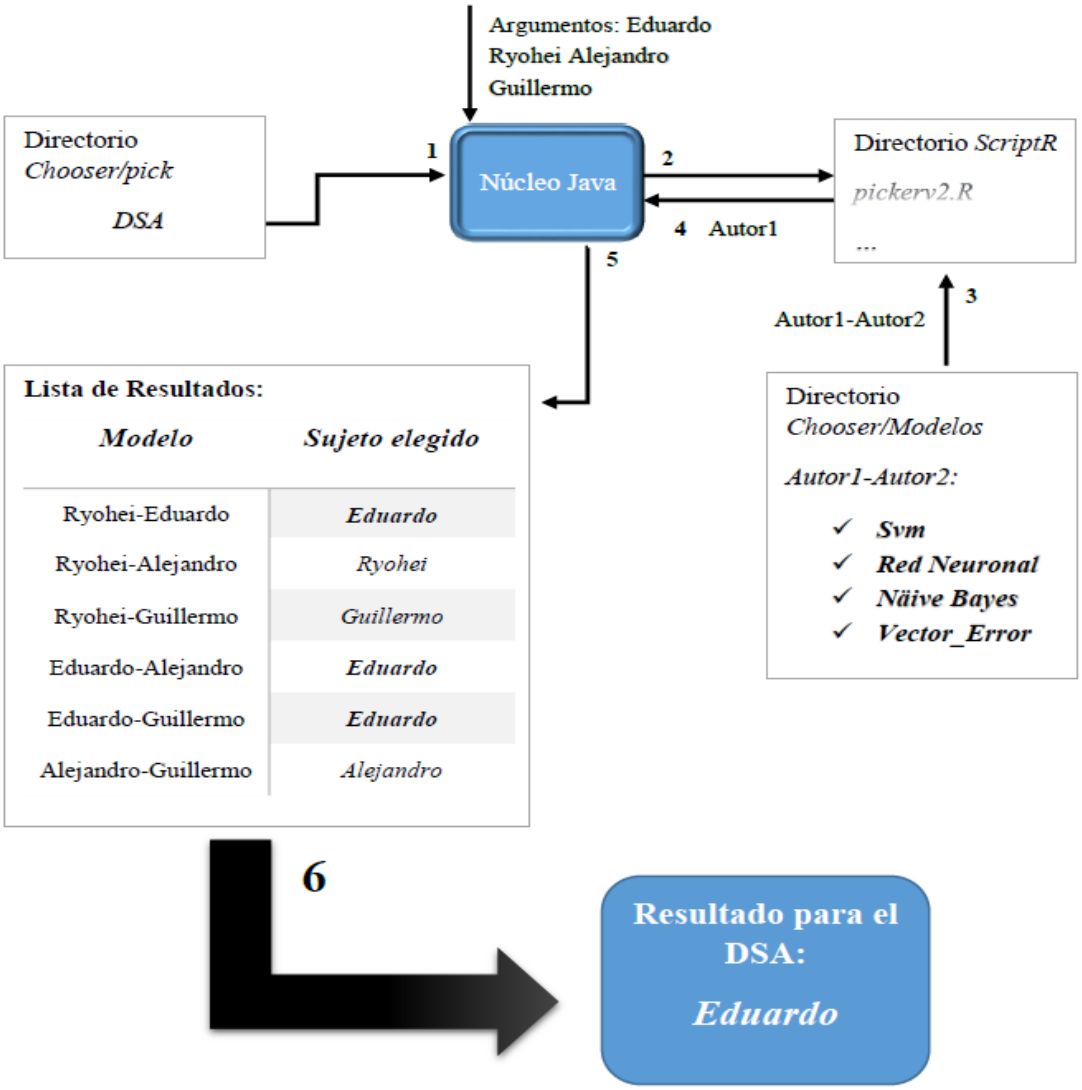


Figura 6.7: **Esquema del módulo de decisión**

Capítulo 7

Resultados obtenidos y discusión

7.1 Introducción

En este capítulo se van a exponer las distintas pruebas realizadas que fundamentan las conclusiones obtenidas en el trabajo. Para ello, se tendrán en cuenta los factores que modifican la bondad del algoritmo.

Esta sección se divide en tres apartados. El primero de ellos describirá el diseño de cada una de las pruebas. El siguiente, mostrará los resultados obtenidos. Finalmente, se realizará una discusión de lo generado en los distintos experimentos.

7.2 Descripción de las pruebas

7.2.1 El corpus

La primera decisión en la fase de diseño de las distintas pruebas es la elección de los autores que van a ser sujeto de estudio. En este trabajo se han seleccionado un conjunto de seis autores, todos contemporáneos entre sí. Se clasifican en dos grupos: escritores y periodistas.

- **Escritores.** Este grupo está compuesto por Miguel Delibes Setién (MDS), Carlos Ruiz Zafón (CRZ) y Arturo Pérez Reverte (APR). De ellos se han obtenido fragmentos de tres obras diferentes, que se utilizarán para las etapas de entrenamiento, *test* y decisión. La Tabla 7.1 muestra las obras seleccionadas de cada autor.

Miguel Delibes Setién	*Cinco horas con Mario*	*El camino*	*El Hereje*
Carlos Ruiz Zafón	*El juego del ángel*	*El prisionero del cielo*	*La sombra del viento*
Arturo Pérez Reverte	*El asedio*	*El pintor de batallas*	*El francotirador paciente*

Tabla 7.1: Tabla con las obras utilizadas de los escritores MD, CRZ y APR en las pruebas del algoritmo

- **Periodistas.** Tres periodistas del periódico *el País*, han sido escogidos para este grupo. Ellos son Miguel Mora (MM), Miguel Jiménez (MJ) y Fernando J. Pérez (FJP). Para la obtención de los documentos de cada autor se han guardado noticias publicadas dentro del periodo 2012-2014 de la página *web* del periódico en formato digital (Web el País 2014).

La elaboración de los diferentes documentos se ha realizado intencionadamente utilizando diferentes extensiones. De este modo, se comprueba la eficacia del algoritmo aplicando este condicionante en las fases iniciales de entrenamiento y *test*, así como posteriormente en la etapa de decisión. La Tabla 7.2 recoge la cantidad de documentos de cada autor que serán empleados en los distintos procedimientos del algoritmo.

7.2.2 Configuración de la plataforma

Para la realización de cada una de las pruebas hay que establecer la configuración pertinente de la plataforma para que pueda realizar el cometido que se le propone. De este modo, para realizar un experimento el primer paso es configurar el fichero *nucleo.properties***.** Fichero de propiedades de la plataforma (Apéndice B.2). En él se configura la cantidad de *TAGS* y de FW que se va utilizar en la generación de la firma. Asimismo, permite la utilización de PCA en el experimento. Una vez configurados los parámetros del fichero se comienza con la prueba:

Autor	***Número de documentos elaborados***
Miguel Delibes	240
Carlos Ruíz Zafón	178
Arturo Pérez Reverte	147
Miguel Mora	68
Miguel Jiménez	47
Fernando J. Pérez	47

Tabla 7.2: Número de documentos utilizados de cada autor

Primero hay que ejecutar la etapa de preprocesado (Apartado 6.4.3). Después se generan los 3 modelos que se utilizarán para la posterior etapa de decisión (Apartado 6.4.7). Para la generación se indica mediante los argumentos de entrada los nombres de los autores de los que se quiere obtener los modelos. Por último, la etapa de decisión. El BA de decisión que utilizará la plataforma será aquel que se introduzca como parámetros de entrada (Apartado 6.4.8). La Figura 7.1 muestra un esquema gráfico de una prueba en la que se utiliza PCA acompañado de la Tabla 7.3 en la que se encuentran los valores asociados a los parámetros de configuración de la plataforma.

Parámetro	***Valor***
num_firma	30
num_outliers_allow	2
range_valor	1.2
plot_box	FALSE
token_min	3
token_max	6
num_func_words	30
num_PCA	2
PCA	SI

Tabla 7.3: Valores de los parámetros de configuración del fichero *nucleo.properties* para un experimento

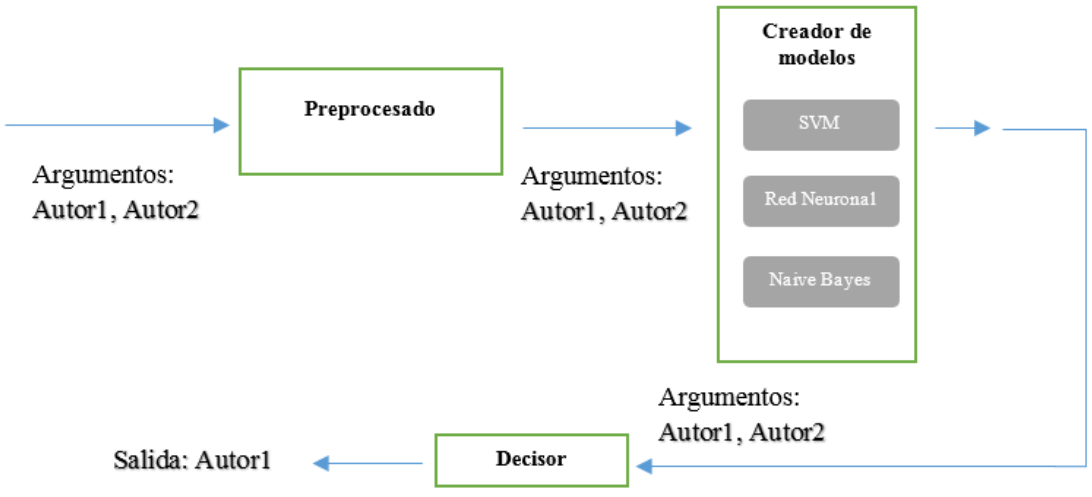


Figura 7.1: Esquema del flujo de operaciones para un experimento del algoritmo

7.2.3 Diseño de las pruebas

El conjunto de pruebas realizadas en este trabajo ha sido duplicado en dos escenarios distintos. El primero de ellos es aquel en el que no se realiza un análisis PCA (Apartado 4.5.1). En el segundo caso, se utilizó el análisis PCA.

El conjunto de pruebas que se han realizado a lo largo del presente trabajo son:

- **Comparativa de las técnicas de decisión.** El objetivo de esta prueba es evaluar y comparar las dos técnicas propuestas en el Apartado 5.6: BEM y el método de desempate. Para su realización se fija un número de 80 características, 40 de ellas TAGS y las otras 40 FW. Para este proceso se evalúan las 15 posibles combinaciones de dos autores a partir de los 6 escogidos. Para cada una de ellas se recoge por separado la precisión de cada modelo: SVM, MLP y *Näive Bayes*. Posteriormente, con los datos obtenidos se establece el vector de error para cada combinación de dos autores. Este vector ordenará de mayor a menor precisión los modelos de IA, que se encuentran entrenados y testeados. El último paso es obtener la bondad de cada una de las dos técnicas que se quieren comparar. Asimismo, para esta comparación se incluirá una estimación más: la media. Este valor se obtendrá haciendo la media de los dos modelos, cuya precisión calculada fue la mayor en la fase de *test.* Esta prueba no estará duplicada para el caso del escenario con análisis PCA, debido a que la naturaleza del análisis no se ve afectado por este proceso.

- **Número de características en los modelos.** Para el desarrollo de esta prueba se varía el número de características de entrada que utilizan los modelos basados en IA presentes en el algoritmo. Se establecen tres configuraciones diferentes en cuanto a la distribución del número de indicadores de estilo: (i) la primera de ellas está basada en el uso únicamente de 40 TAGS; (ii) la segunda configuración gestiona sólo 40 FW y (iii) la última de ellas utiliza 40 TAGS y 40 FW. La finalidad de esta prueba es comprobar si la combinación de estos dos elementos estilísticos

mejora la precisión del algoritmo implementado. En su realización se utilizaron 30 documentos de entrenamiento, la técnica de desempate y un BA de dimensión dos. Se utilizó la técnica de desempate porque fue la que se implementó en la plataforma.

- **Número de documentos de entrenamiento.** Otro de los factores determinantes en la obtención de modelos óptimos basados en IA es el número de documentos de entrenamiento. De esta manera, el objetivo de esta prueba es conocer la conducta de los distintos modelos, en base a la variación del número de documentos que se utilicen en la fase de entrenamiento. Para realizar este experimento se configurará la plataforma con las siguientes especificaciones. El número de características se fija en 80, divididas en los dos indicadores de estilo (40 TAGS + 40 FW). En cuanto a la técnica para decidir la autoría de los documentos será la de desempate ya que era la que se implementó en la plataforma. El número de autores del BA será dos. Por último, esta prueba se realizará dos veces, con el fin de analizar el comportamiento del algoritmo en presencia o ausencia del análisis PCA. El desarrollo de esta prueba consiste en calcular la precisión del algoritmo para 15 combinaciones de dos autores. El proceso se divide en 5 subtareas, en las que se obtiene el porcentaje de acierto individual para cada par de autores. Cada una de ellas está configurada en cuanto a la utilización de modelos entrenados con un número distinto de documentos: 15, 20, 25, 30 y 40. Se produce un salto entre 30 y 40 documentos, debido a la elevada carga computacional asociada.

- **Número de autores en el BA.** La última prueba realizada tiene por objetivo conocer la evolución de la precisión del algoritmo según aumenta el número de integrantes del BA. De esta manera, este estudio parte de un BA de dos sujetos hasta analizar uno de seis. La configuración de la plataforma para el desarrollo de este experimento utiliza 30 documentos de entrenamiento y 80 características de entrada a los modelos (40 TAGS + 40 FW). Además, la técnica de decisión de autoría elegida es la de

desempate ya que era la que utilizaba la plataforma De nuevo esta prueba no utilizará análisis PCA.

7.3 Resultados de las pruebas

Los resultados de las siguientes pruebas siguen el esquema propuesto en el Apartado 7.2, relativo al diseño de cada una de ellas. El fin que busca esta sección es analizar la bondad del algoritmo en el campo de la atribución de autoría para textos españoles. De este modo, se centrará en cada una de las pruebas propuestas anteriormente. La realización de cada una de ellas se duplicará, salvo en dos casos, en dos escenarios. El primero de ellos será aquel en el que no se realice análisis PCA. Por tanto, la dimensión del problema sea mucho mayor que en el segundo contexto, en el que se utiliza este procedimiento de reducción del número de variables.

7.3.1 Comparativa de las técnicas de decisión

La Figura 7.2 compara la precisión obtenida para cada modelo. El color rojo corresponde al modelo SVM, el verde al modelo de MLP y el azul al modelo de *Näive Bayes.* El eje de coordenadas muestra las distintas combinaciones de dos autores, mientras que el eje de ordenadas representa la bondad obtenida. Ésta última se calcula dividiendo el número de documentos correctamente clasificados entre el total de los utilizados para la prueba.

La Tabla 7.4 resume los datos recogidos por el vector de error para las 15 combinaciones de dos autores entre los 6 posibles. Esta información será procesada por la plataforma acorde a la técnica de decisión utilizada en la prueba (Apartado 5.6).

Por último, la Figura 7.3 compara la eficacia de cada técnica. En verde la técnica de desempate y en azul la de BEM. A estos valores se añade el estimador de la media de los dos mejores modelos en color rojo.

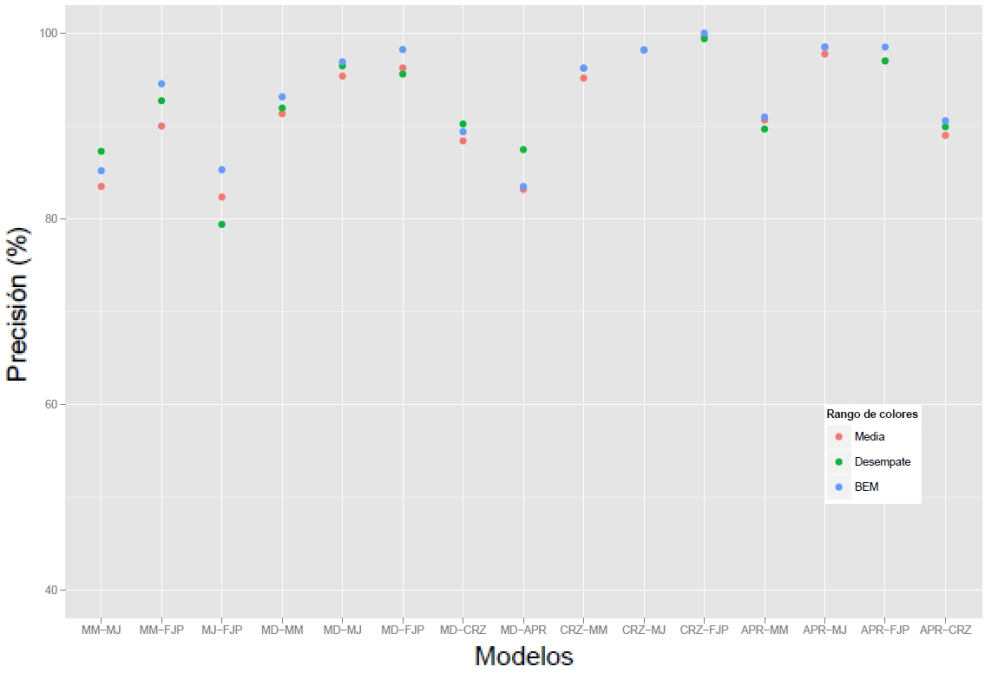


Figura 7.2: Gráfico comparativo entre los tres modelos basados en IA: SVM, MLP y *Näive Bayes*

Combinación de Autores	Modelo 1	Modelo 2	Modelo 3
MM-MJ	red neuronal	*svm*	*Näive Bayes*
MM-FJP	*svm*	red neuronal	*Näive Bayes*
MJ-FJP	red neuronal	*svm*	*Näive Bayes*
MD-MM	*svm*	red neuronal	*Näive Bayes*
MD-MJ	*svm*	red neuronal	*Näive Bayes*
MD-FJP	red neuronal	*svm*	*Näive Bayes*
MD-CRZ	red neuronal	*svm*	*Näive Bayes*
MD-APR	*svm*	red neuronal	*Näive Bayes*
CRZ-MM	*Näive Bayes*	red neuronal	*svm*
CRZ-MJ	*Näive Bayes*	*svm*	red neuronal
CRZ-FJP	*svm*	*Näive Bayes*	red neuronal
APR-MM	*Näive Bayes*	red neuronal	*svm*
APR-MJ	*svm*	red neuronal	*Näive Bayes*
APR-FJP	*svm*	red neuronal	*Näive Bayes*
APR-CRZ	*svm*	red neuronal	*Näive Bayes*

Tabla 7.4: Tabla con los vectores de errores obtenidos para cada par de autores

7.3.2 Número de características en los modelos

Los resultados se muestran en dos tablas (Tabla 7.5 y Tabla 7.6) con el fin de diferenciar los dos escenarios, con y sin análisis PCA.

Cada tabla está dividida en tres columnas, una para cada caso de estudio: sólo *TAGS*, sólo FW y uso de ambos indicadores de estilo. En cada una de ellas aparece la precisión recogida por el algoritmo mediante la técnica BEM. Las celdas que tienen un fondo verde serán las que mayor precisión se hayan obtenido para las tres configuraciones posibles. Se utilizarán los colores rojo y amarillo cuando alguno de los dos indicadores por solitario obtenga una mejor precisión que en el caso de utilizar ambos. El color rojo indicará una diferencia más pronunciada que el caso amarillo.

Por último la Tabla 7.7 recoge la media de los 15 modelos para cada distribución de características con y sin PCA, con el fin de comparar el comportamiento del algoritmo en los dos escenarios.

Combinación de Autores	**40 *TAGS* (Precisión %)**	**40 FW (Precisión %)**	**40 *TAGS* + 40 FW (Precisión %)**
MM-MJ	70.90	50.90	87.27
MM-FJP	63.63	76.36	92.72
MJ-FJP	79.41	94.11	79.41
MD-MM	83.46	90.32	91.93
MD-MJ	85.46	85.90	96.47
MD-FJP	98.23	96.47	95.59
MD-CRZ	75.41	75.41	90.22
MD-APR	81.34	80.42	87.46
CRZ-MM	93.01	94.62	96.23
CRZ-MJ	89.09	95.75	98.18
CRZ-FJP	95.75	93.93	99.39
APR-MM	87.09	88.38	89.67
APR-MJ	91.72	90.29	98.50
APR-FJP	95.59	94.77	97.01
APR-CRZ	*84.52*	83.77	89.81

Tabla 7.5: Tabla que recoge la precisión obtenida para cada distribución: 40 *TAGS*, 40 FW y 40 *TAGS* + 40 FW, sin utilizar análisis PCA

Combinación de Autores	40 *TAGS* (Precisión %)	40 FW (Precisión %)	40 *TAGS* + 40 FW (Precisión %)
MM-MJ	50.90	85.45	74.54
MM-FJP	76.36	81.81	83.63
MJ-FJP	94.11	58.82	52.94
MD-MM	90.32	69.35	60.48
MD-MJ	85.90	88.48	85.46
MD-FJP	96.47	89.42	91.18
MD-CRZ	75.41	70.11	72.62
MD-APR	80.42	65.13	62.69
CRZ-MM	94.62	80.64	80.64
CRZ-MJ	95.75	86.66	91.51
CRZ-FJP	93.93	94.54	94.54
APR-MM	88.38	78.70	68.38
APR-MJ	90.29	69.40	94.02
APR-FJP	94.77	94.02	93.28
APR-CRZ	83.77	63.01	60.75

Tabla 7.6: Tabla que recoge la precisión obtenida para cada distribución: 40 *TAGS*, 40 FW y 40 *TAGS* + 40 FW, utilizando análisis PCA

Escenario	40 *TAGS* (Precisión %)	40 FW (Precisión %)	40 *TAGS* + 40 FW (Precisión %)
Sin análisis PCA	84.97	86.09	92.66
Con análisis PCA	86.09	78.36	77.77

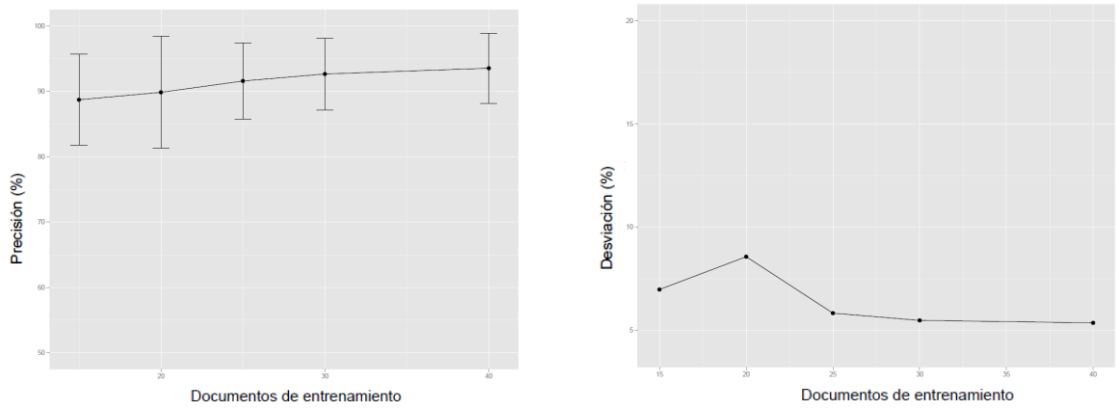
Tabla 7.7: Tabla que recoge la media de las precisiones de las 15 combinaciones de autores para cada una de las distribuciones de características y los dos escenarios: con y sin PCA

7.3.3 Número de documentos de entrenamiento

El conjunto de resultados obtenidos para cada uno de los escenarios ha sido resumido en las Figuras 7.4a y 7.4b para el análisis sin PCA, y en las Figuras 7.5a y 7.5b para el análisis con PCA.

Las Figuras 7.4a y 7.5a corresponden al análisis de la precisión media obtenida mediante el uso de la técnica de desempate. Se varía el número de documentos de entrenamiento utilizados para cada modelo: SVM, MLP y *Näive Bayes*. La barra de error

muestra la variabilidad obtenida en el conjunto de datos que componen la media de la precisión global. El objetivo es minimizar estos indicadores con el fin de obtener una efectividad (precisión) similar del algoritmo para cada grupo de dos autores que se quiera clasificar. Las Figuras 7.4b y 7.5b ilustran la evolución de la desviación estándar con respecto al número de documentos utilizados en la fase de entrenamiento. Esta información será útil para discriminar aquellos factores que hagan mínima esta desviación, ya que es uno de los objetivos de la prueba.



(a) (b)

Figura 7.3: Análisis del comportamiento del algoritmo con respecto a la variación del número de documentos de entrenamiento, sin utilizar PCA. (a) Gráfico de la precisión obtenida con barras de error para cada caso; (b) Evolución de la desviación estándar para cada cantidad de docuentos de entrenamiento.

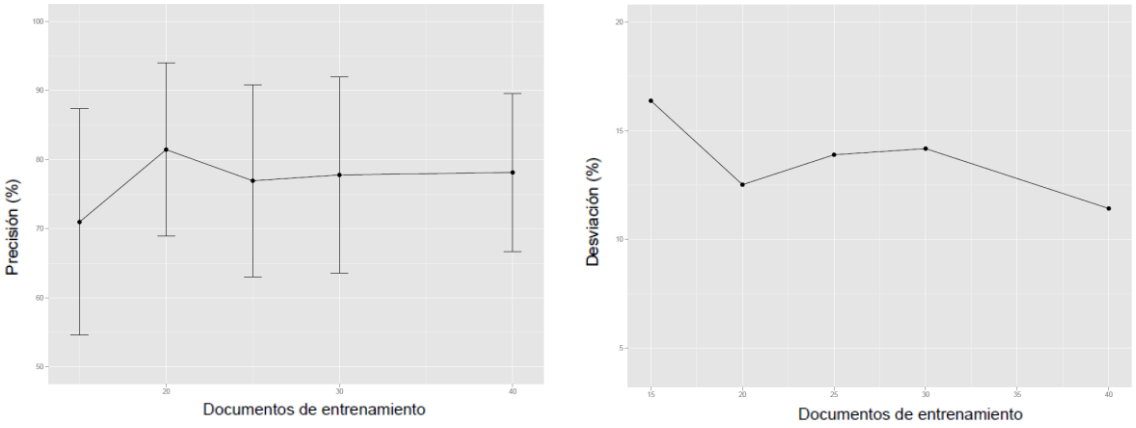


Figura 7.4: Análisis del comportamiento del algoritmo con respecto a la variación del número de documentos de entrenamiento, utilizando análisis PCA. (a) Gráfico de la precisión obtenida con barras de error para cada caso; (b) Evolución de la desviación estándar para cada cantidad de documentos de entrenamiento.

7.3.4 Número de autores en el banco de autores

Para mostrar los resultados obtenidos se ha creado un diagrama de cajas para cada caso: desde un BA de 2 hasta uno de 6 autores, incrementando en una unidad el número de sujetos para cada prueba. Este tipo de gráficos muestra una aproximación de la distribución de los resultados. De este modo, se analiza la precisión obtenida a medida que el número de autores presentes en el BA aumenta. El rombo que se encuentra dentro de la caja representa la media de las precisiones obtenidas para las combinaciones de autores y la línea que la atraviesa es la mediana. Dado que en el presente trabajo se han recogido muestras de 6 autores, éste será el límite impuesto para este experimento. Por ello, el número de combinaciones posibles dependerá de la cantidad de autores que se tomen en el BA para el proceso de decisión. La Figura 7.6 recoge los diagramas de cajas obtenidos.

7.4 Discusión de los resultados

7.4.1 Análisis PCA

La primera conclusión que se ha obtenido al analizar los resultados en los distintos experimentos realizados, es que la utilización de PCA en este contexto no supone una mejora en el rendimiento del algoritmo.

Para el estudio del número de características de cada tipo, *TAGS* o FW, los resultados son mucho peores que en el caso de no utilizar PCA, salvo en la opción de utilizar únicamente *TAGS* en el que el escenario basado en análisis PCA obtiene una precisión algo mejor (Tabla 7.6). En cuanto a la desviación de las precisiones obtenidas para cada par de autores es mucho mayor en PCA en dos de las tres pruebas. Este fenómeno es algo que se pretende evitar, ya que uno de los objetivos del algoritmo es la obtención de precisiones lo más parecidas posibles para el conjunto de autores del BA. La Tabla 7.8 compara la desviación estándar de las tres distribuciones probadas, en relación al uso o no de PCA. En ella se puede observar que la celda resaltada en color verde corresponde con el mínimo de este estimador y se encuentra en el escenario que no utiliza PCA.

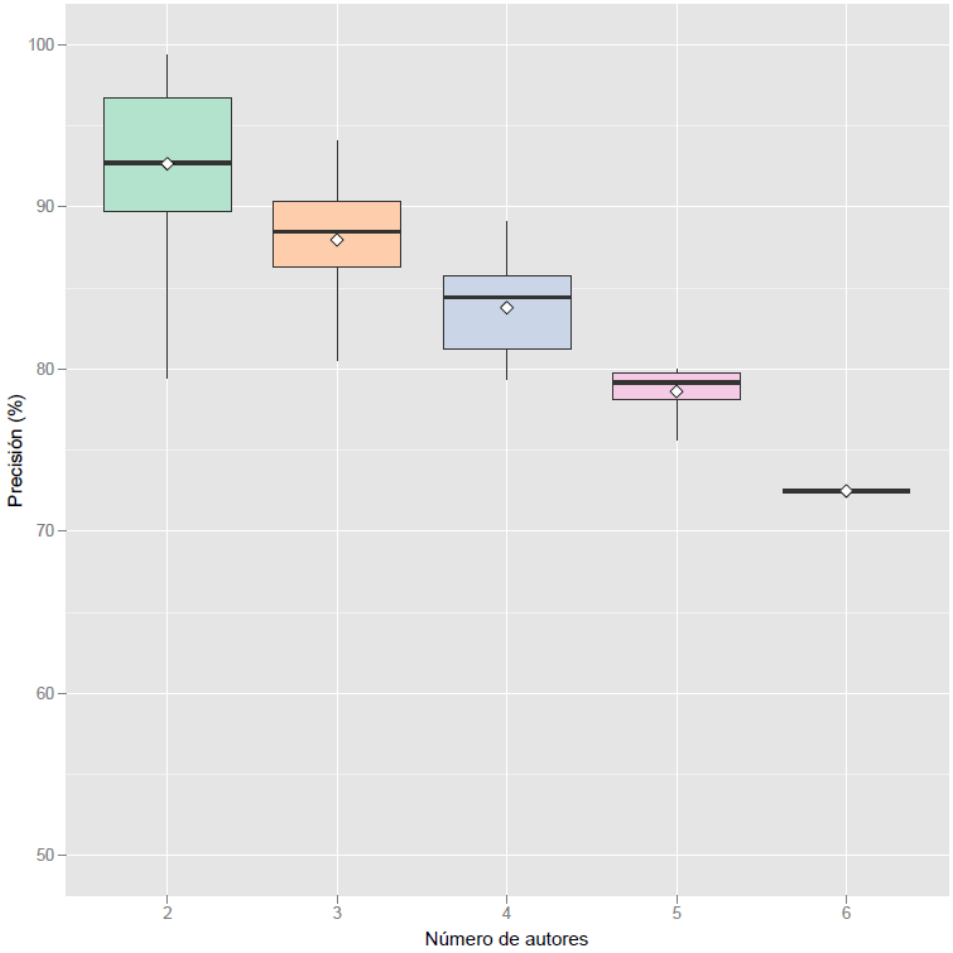


Figura 7.5: Diagramas de cajas que muestran la precisión del algoritmo para distintas cantidades de autores en el BA.

	40 TAGS *(Desviación Estándar)*	*40 FW* *(Desviación Estándar)*	*40 TAGS + 40 FW* *(Desviación Estándar)*
Con PCA	11.95	11.61	13.93
Sin PCA	9.74	11.95	5.47

Tabla 7.8: Desviación típica de los resultados obtenidos en la prueba de la distribución de características para los dos escenarios: con y sin PCA.

El comportamiento es similar en las pruebas realizadas con respecto al número de documentos de entrenamiento. Los resultados obtenidos con PCA son peores que cuando no se utiliza. Aun así, se ve claramente que la tendencia que siguen la precisión y la desviación estándar calculadas en ambos contextos es la misma.

En resumen, según lo analizado anteriormente, el análisis PCA, como se comentaba al inicio de esta sección, no es un método que permita obtener rendimientos mejores, sino que empeora la precisión de los modelos obtenidos. La explicación para este fenómeno es que el uso de dos componentes principales, como características de entrada para los modelos de IA, no es suficiente para determinar el estilo de un autor utilizando los indicadores propuestos en el algoritmo.

Baayen *et ál*. (2002) demostraron que el análisis PCA era adecuado en ciertos casos, como los estudios de Burrows (1987, 1989). Sin embargo, en algunos contextos resultaba insuficiente para distinguir el estilo entre autores. Esto se demostró en el estudio alemán, en el que el uso de palabras funcionales, junto al análisis PCA, no consiguió encontrar una estructura que pudiera caracterizar la autoría (Baayen *et ál*. 2002). Los resultados obtenidos en el TFG con respecto a PCA corroboran que para este escenario (*TAGS* + FW) este tipo de análisis no es adecuado.

Investigadores dentro del campo de la atribución de autoría consideran que los modelos basados en SVM para problemas de clasificación, generalmente mejoran el rendimiento del resto de algoritmos (Abassi & Chen 2005; Zhang *et ál*. 2006), lo que incluye al análisis PCA. La información contenida en la Tabla 7.4 muestra que efectivamente SVM es el modelo que aparece en el primer puesto del vector de errores en un mayor número de ocasiones.

7.4.2 Técnicas de decisión de autoría

A partir de los resultados obtenidos, se concluye que ambas técnicas (BEM y desempate) presentan un buen comportamiento en el ámbito del problema de la atribución de autoría. Aunque cabe decir que el comportamiento del método BEM es más preciso en la mayoría de los casos. Asimismo, ambas técnicas son capaces de suavizar el efecto negativo de un modelo mal entrenado y hacer que no influya en la fase de decisión. La bondad de un modelo está principalmente influida por los documentos de entrenamiento que se utilicen (Stamatatos 2009). Dado que se utilizan textos *reales* no homogéneos, una de las razones que explica este fenómeno de mal entrenamiento es la posible inclusión de

ruido a los modelos, provocando que el proceso de aprendizaje falle en alguno de los algoritmos. Es por ello que gracias a las técnicas propuestas (BEM y desempate) se puede suavizar este fenómeno.

En la Figura 7.2 se puede ver que la red neuronal MLP no ha sido correctamente configurada, en cambio en la clasificación obtenida en la Figura 7.3 se obtiene una precisión cercana al 100%. Esto es posible gracias al uso de varios modelos de IA. Ambas técnicas se basan en el descarte de modelos: uno en el caso de la técnica de desempate y dos para la técnica BEM. Como se observa en la Tabla 7.4, las precisiones generadas por cada algoritmo (SVM, MLP o *Näive Bayes*) no siguen siempre el mismo comportamiento para los distintos autores. De este modo, utilizando tres modelos diferentes se consigue solventar las limitaciones individuales suavizando los resultados generados con los obtenidos por el resto de algoritmos de clasificación utilizados en la plataforma.

La técnica que inicialmente se propuso para la plataforma fue la de desempate. Por este motivo es la que se incluyó en la plataforma y la utilizada en el resto de pruebas. Posteriormente se propuso la técnica BEM y la única prueba que se pudo realizar dentro del tiempo establecido fue la comparativa con la técnica anterior.

No se han encontrado estudios de atribución de autoría que permitan la comparación con estas dos técnicas, ya que se basa en la utilización de un único algoritmo de clasificación.

7.4.3 Combinación de TAGS y FW

La combinación de los dos indicadores de estilo (TAGS y FW) para el escenario que no utiliza análisis PCA, supone una mejora considerable en la fase de clasificación del algoritmo. Como se ha dicho en el punto anterior, el comportamiento de PCA no es el idóneo en este campo y por ello al combinar las características estamos acentuando aún más las deficiencias que aporta este método al algoritmo. La Tabla 7.5 muestra que las precisiones generadas cuando se combinan ambos indicadores tiene como resultado una mejora en la precisión de la clasificación. Este resultado era el de esperar, ya que la

información estilística contenida en la firma del autor es mayor, haciendo más sencilla la tarea de distinción del estilo de los autores.

En cuanto a la desviación estándar, la combinación de estos elementos de estilo hace que su valor se reduzca considerablemente, un 50 % con respecto al valor máximo obtenido mediante el uso de los indicadores de estilo por separado. Este comportamiento también era el esperado, ya que cuando uno de los indicadores no es capaz por sí solo de establecer una distinción clara de estilo, el algoritmo utilizará los elementos del indicador restante para establecer esa diferenciación. Un ejemplo sería la primera fila de la Tabla 7.5. Mientras que las precisiones obtenidas utilizando los indicadores de estilo por separado fueron: 70.90% y 50.90%, en el caso de la combinación de ambos se obtuvo un porcentaje de acierto del 87.27%. Esto supone un 36.37% más que en el peor de los casos. Además, la reducción de la desviación estándar es otro de los objetivos que persigue el algoritmo. El valor idóneo sería 0 lo que correspondería a una precisión común para todos los modelos generados para el BA.

Este resultado es similar al obtenido por Grieve (2007), quien realizó un estudio de la precisión de distintas técnicas para resolver el problema de la atribución de autoría. En él contempló la idea de combinar aquellos indicadores que mejor resultados ofrecían (precisiones mayores). Concluyó que en efecto las precisiones obtenidas mediante la combinación de elementos de estilo, eran en la mayoría de casos más altas, mientras que en el resto se mantenía (Grieve 2007).

Como conclusión al análisis expuesto, el efecto que produce la combinación de ambos indicadores se traduce en mejoras a nivel de precisión del algoritmo y de generalización de los modelos creados, dos propiedades muy beneficiosas para el problema de la atribución de autoría.

7.4.4 Implicación del número de documentos de entrenamiento

Como se había supuesto, el número de documentos de entrenamiento es un factor que condiciona la naturaleza de cada modelo. A partir de los resultados obtenidos (Figura 7.3) se observa que, a medida que se incrementa esta variable, la precisión obtenida crece

en la misma dirección. Sin embargo, la desviación estándar juega el papel contrario, su valor cada vez es menor. Este resultado es justamente el que se persigue, ya que minimizando la desviación estándar se consigue una mayor generalización del modelo para el conjunto de autores posibles.

El experimento se realizó hasta una cantidad de 40 documentos de entrenamiento, debido a que el número de documentos disponibles no permitía hacer pruebas más complejas y a que el número de documentos de entrenamiento para cada autor fuera similar. El análisis que se obtiene del comportamiento que sufre la desviación estándar y la precisión obtenida según aumenta el número de documentos es el esperado. Esto implica que los algoritmos de clasificación (SVM, MLP y *Näive Bayes*) presentan una mayor precisión. Por lo tanto, la etapa de aprendizaje es más efectiva. Sin embargo, aún no existen estudios que establezcan para el problema de la atribución de autoría una cantidad mínima requerida de textos de entrenamiento (Lyckx & Daelemans 2011). Es por ello que este valor se deberá obtener de forma analítica, a partir de experimentos. De esta manera, es necesario hacer pruebas con un número de documentos mucho mayor para conocer con mayor confianza el comportamiento de la metodología propuesta.

Stamatatos (2009) estableció unas condiciones sobre las cuales poder evaluar cualquier método basado en atribución de autoría:

- El tamaño del corpus de entrenamiento. Número de documentos y longitud.

- El tamaño del corpus de test en términos de longitud de los textos.

- El número del BA. Se estudiará en el Apartado 7.4.5.

- La distribución del corpus de entrenamiento con respecto al conjunto de autores del BA. Si el número de documentos de entrenamiento de cada autor es idéntico o no.

Tradicionalmente, los textos considerados como idóneos para las técnicas de atribución de autoría eran aquellos cuya longitud mínima era de 10.000 palabras (Burrows 2007). Sin embargo, cuando el problema exigía una longitud inferior, como por ejemplo en el estudio de Coytl *et ál* (2006) sobre la autoría de unos poemas, se utilizaba un corpus con una cantidad muy grande de documentos de entrenamiento. Asimismo, otros estudios consiguieron resultados prometedores utilizando también textos cortos (Sanderson & Guenter 2006; Hirst & Feiguina 2007). El presente TFG es otro de los casos en los que se ha usado textos relativamente cortos entre 150 y 1.000 palabras y los resultados obtenidos han sido óptimos en cuanto a la precisión obtenida (Apartado 7.3.4).

En cuanto a la distribución del corpus de entrenamiento, se ha utilizado el mismo número de documentos para cada autor. Sin embargo, se han combinado textos largos y cortos. La causa de esta decisión es proporcionar distintas observaciones para que el proceso de aprendizaje sea más completo. Sanderson y Guenter (2006) observaron que el conjunto de datos de entrenamiento tiene mayor influencia en el rendimiento que los datos de test.

En conclusión, son muchos los condicionantes que pueden modificar el resultado final del algoritmo en relación al corpus utilizado. Es por ello que la comparación entre los diferentes estudios resulta una tarea compleja y en ocasiones difícil de extrapolar. En este experimento, la idea fundamental que se obtiene es que cuanto mayor sea el corpus de entrenamiento utilizado, mejor será la capacidad de clasificación de los modelos.

7.4.5 Precisión del algoritmo en base al banco de autores

La finalidad del algoritmo es la clasificación de la autoría de un documento de texto, de la manera más precisa posible a partir de un BA establecido. No cabe duda de que este problema resulta más difícil en cuánto incrementamos el número de integrantes del BA. De esta manera, se ha probado con diferente número de autores, desde 2 hasta 6, y con el fin de conocer el comportamiento de la metodología propuesta.

Una vez analizados los resultados, como ya se había pensado, la precisión del algoritmo es inversamente proporcional al número de autores del BA. De esta manera, partimos de un porcentaje del 92.65 % de acierto en 2 autores hasta un 72.48 % para 6.

El estudio de Koppel *et ál.* (2010) se centró en el campo de la atribución de autoría para grandes BA. Descubrieron que el rendimiento desciende según aumenta el BA. El mismo comportamiento sucede en este escenario. Asimismo, Grieve (2007) también realizó un estudio en el que probaba distintas técnicas variando el número de autores (desde 2 hasta llegara a un BA de 40). Obtuvo el mismo resultado: una reducción de la precisión del modelo. No cabe duda que el número de autores del BA será un factor fundamental para el desarrollo de algoritmos que pretendan dar solución al campo de la atribución de la autoría. Por ejemplo el estudio de Abbasi & Chen (2008) se centraba en BA grandes lo que implicaba la utilización de miles de características.

La bondad del algoritmo del TFG está calculada utilizando textos españoles. Por ello, los resultados asociados a otras lenguas mediante el uso de la metodología propuesta pueden ser diferentes. Con la idea de comparar, la Tabla 7.9 hace una comparación de la clasificación de las técnicas que mejor resultados proporcionan con textos españoles y además lo comparan con los obtenidos usando textos ingleses (Blasco & Ruiz 2009).

Como se puede observar, la técnica que mejor rendimiento ofrece presenta una bondad del 90% para textos españoles y del 95 % para los ingleses. El algoritmo planteado

		LENGUA ESPAÑOLA		LENGUA INGLESA	
	Algoritmos de atribución	%		%	
		10	2	10	2
1	Marca de puntuación simple1	85	90	58	89
2	Palabra simple	70	90	67	88
3	Grafema multiposición (6 inic.)	65	70	64	90
4	Puntuación + palabra	60	70	80	95
5	2-gramas de palabra	55	80	34	74
6	5-gramas de grafema	50	60	66	90

Tabla 7.9: Comparativa de precisiones para BA de 2 y 10 autores utilizando diferentes técnicas para los idiomas español e inglés (Blasco & Ruiz, 2009)

en el presente trabajo presenta de media un porcentaje del 92.65%, llegando en algún experimento a conseguir el 100% de efectividad. De este modo, el trabajo realizado supera la precisión obtenida hasta el momento, para textos españoles, quedándose muy cerca de los resultados para los ingleses. No obstante, hay que tener en cuenta que los autores y documentos utilizados en estos estudios difieren. Por tanto, estas comparaciones hay que tomarlas con cuidado.

Las Tablas 7.10 y 7.11 muestran una comparación de las precisiones obtenidas por los estudios de otros investigadores con respecto al campo de la atribución de autora sobre textos ingleses, españoles y portugueses. Hay que tener en cuenta que los datos de entrenamiento y de test en muchos de los casos son diferentes y es por ello que las conclusiones que se obtengan de la tabla deben ser tomadas con mucha precaución.

Destacar dos de los estudios mostrados en las Tablas anteriores: Halvani et ál. (2013) y Seidman (2013). Corresponden a una competeción celebrada en Valecia en 2013. En ella los participantes tubieron que desarrollar un algoritmo que permitiera decidir si una serie de documentos pertenecían al mismo autor. Para ello les proporcionaron textos en tres idiomas diferentes: inglés,español y griego. De este modo, se podía comparar la efectividad de la propuesta en base a distintos idiomas. El ganador fue Seidman (2013) consiguiendo una media del 75.3% (Juola & Stamatatos 2013). Sin embargo, su propuesta consiguió un 60% de efectividad para el caso de los textos españoles. Halvani et ál (2013) obtuvo una precisión global del 71.8 % quedando en segunda posición (Juola & Stamatatos 2013). Pero en esta ocasión, su metodología consiguió un 84.0% de precisión para el caso de los textos españoles.

Con estos resultados se concluye que cada una de las propuestas que se realizan para el campo de la atribución de autoría dependen de muchos factores y uno de los principales es el idioma al que va destinado. De esta manera, todas las conclusiones que se puedan obtener de este Trabajo Fin de Grado realizado en la Universidad de Valladolid tienen que tener presente todos los factores que han influido en los resultados obtenidos.

Estudio	*Tipo de Textos*	*Características del algoritmo*	*Entrenamiento /Test*	*Número de autores*	*Precisión obtenida (%)*
Nirkhi et ál. 2014	Textos ingleses	Riqueza del vocabulario + Longitud de la frase + FW (SVM)	Conjunto de datos "Reuter_50_50"	5	92.0
Koppel et ál 2013	Textos ingleses	1000 palabras con mayor información + 10000 palabras más comunes (*Bayessian multi-class regression*)	Dos libros para cada autor de la literatura inglesa y americana	9	82.8
Seidman 2013	Textos españoles	Caracteres de 4-Gramas	Artículos de periódicos y de ficción	1 vs todos	60.0
	Textos ingleses	Unigramas	Artículos de periódicos		80.0
	Textos griegos	Unigramas	Extractos de libros sobre ciencia computacional y temas asociados		83.3
Halvani et ál. 2013	Textos españoles	Combinación de 12 características léxicas y de caractereres (K-Nearest Neighbor)	Artículos de periódicos y de ficción	1 vs todos	84.0
	Textos ingleses		Artículos de periódicos		70.0
	Textos griegos		Extractos de libros sobre ciencia computacional y temas asociados		63.3

Tabla 7.10: Comparativa de las precisiones obtenidas por algoritmos de atribución de la autoría

Estudio	***Tipo de Textos***	***Características del algoritmo***	***Entrenamiento /Test***	***Número de autores***	***Precisión obtenida (%)***
Varela et ál. 2011	Textos portugueses	FW (2 modelos SVM)	Artículos de periódicos	10	75.2
Blasco & Ruiz 2009	Textos españoles	Frecuencia de ocho de las palabras más frecuentes O Frecuencia de cuatro marcas de puntuación entre el número total de caracteres	Artículos periodísticos (2007-2009)	2	90.0
Grieve 2007	Textos ingleses	Combinación de 16 indicadores de estilo.	Artículos periodísticos (2000-2005)	2	97.0
Argamon ét al. 2007	Textos ingleses	FW+ combinación de frecuencias relativas de características del sistema de conjunción y de modalidad (SVM)	20 novelas del siglo XIX	8	90.0
Argamon & Levitan 2005	Textos ingleses	FW (SVM)	20 novelas del siglo XIX	8	99.0
Presente TFG	**Textos españoles**	**40 FW + 40 TAGS (SVM +MLP + *Näive Bayes)***	**Artículos periodísticos (2013-2014) y extractos de libros**	2	**92.7**

Tabla 7.11: Comparativa de las precisiones obtenidas por algoritmos de atribución de la autoría (continuación)

Capítulo 8

Conclusiones

8.1 Objetivos alcanzados

El presente TFG comenzó con el objetivo general de desarrollar un algoritmo que permitiera distinguir de manera precisa la autoría de un documento de texto a partir de un grupo de autores preestablecido. Para ello, se propusieron una serie de objetivos específicos, de los que a continuación se analiza su grado de cumplimiento:

- **Modificar el formato del texto para que la herramienta NLP sea capaz de analizar correctamente la información**. En este TFG, la herramienta que se utiliza es *Freeling* (Padró & Stanilovsky 2012) explicada en el Apartado 3.5.2. Para ello, se ha creado el módulo de preprocesado que tiene como objetivo la creación de los DA, textos que pueden ser procesados por la herramienta NLP (Apartado 6.4.3).

- **Identificar y obtener el conjunto de características (FW y *TAGS*) que permitan diferenciar el estilo de dos autores que dependerán de diversos factores.** Con respecto al objetivo de identificar los indicadores de estilo se ha desarrollado el módulo extractor de características (Apartado 6.4). Será el encargado de extraer la lista de *TAGS* y de FW de manera inductiva a partir de unos DA de entrenamiento. Estas dos listas conforman lo que se denomina la *firma del autor*, que permite la distinción estilística de los autores. La parte de extracción de los *TAGS* y los FW es llevada a cabo por el módulo extractor de características (Apartado 6.4.4) que a partir de la FAu, obtendrá la frecuencia de aparición de los elementos estilísticos propios de la FAu en cada DA.

- **Proponer una metodología basada en técnicas avanzadas de clasificación que permita maximizar el rendimiento de las características estilísticas seleccionadas.** La metodología propuesta se basa en la utilización de 3 algoritmos de IA (SVM, MLP y *Näive Bayes)*. Para la creación de los distintos modelos se ha implementado el módulo creador de modelos (Apartado 6.4.7). Asimismo, para la etapa decisión se

proponen dos técnicas: BEM y la desempate (Apartado 6.4.8) y será el módulo de decisión quien se encargue de la tarea de la atribución de la autoría para los distintos documentos.

- **Aplicar una serie de técnicas para ajustar lo más preciso posible los distintos modelos de IA para optimizar/maximizar los resultados de la fase de test.** Las técnicas que han permitido la optimización de los parámetros de clasificación de los distintos algoritmos basados en IA han sido*: cross-validation* y la optimización granulada (Apartado 5.5). Los resultados obtenidos en los distintos experimentos permiten establecer que el proceso de optimización dinámico de características es eficaz.

- **Comparar los resultados obtenidos con estudios previos dentro del campo de la atribución de autoría.** El Apartado 7.4 establece comparaciones con otros estudios del campo de la atribución de autoría. Sin embargo, los resultados de los distintos métodos que se comparan dependen de las observaciones utilizadas y éstas no son iguales para las distintas técnicas comparadas. Por ello, todos los resultados generados en este informe tienen en cuenta este factor.

8.2 Conclusiones

En el presente TFG se ha propuesto una solución para el problema de la atribución de autoría para textos españoles, utilizando técnicas de clasificación avanzadas: SVM, MLP y *Näive Bayes*.

Se ha demostrado en el Capítulo 7 que las características escogidas (FW y *TAGS*) son eficaces para el contexto del problema en textos españoles. Además, el uso combinado de ambas permite obtener precisiones del 92.65% (Apartado 7.4.3), mejorando las soluciones propuestas hasta el momento. Asimismo, la propuesta del TFG en cuanto al uso de múltiples algoritmos de IA para resolver el problema que se plantea, permite suavizar las deficiencias individuales de cada modelo y generar resultados

óptimos, en concepto de precisión obtenida (Apartados 7.4.2 y 7.4.5). Además, mediante los procesos de optimización (*cross-validation* y optimización granulada) utilizados en el presente TFG (Apartado 5.5), ha sido posible entrenar eficazmente los diferentes algoritmos de clasificación (SVM, MLP y *Näive Bayes)* de manera dinámica ofreciendo una configuración particular para cada modelo generado (Apartado 5.6). En cuanto al proceso de normalización que se ha tenido en cuenta en las distintas fases del algoritmo: entrenamiento, test y decisión, concluir que la técnica aplicada (Apartado 4.4) ha permitido procesar satisfactoriamente documentos de diferentes longitudes con resultados buenos. Haciendo referencia a PCA, con este estudio se ha probado que el uso de este tipo análisis en la metodología propuesta, no supone una mejora en términos de la eficacia del algoritmo de atribución de la autoría propuesto para textos españoles (Apartado 7.4.1). También se ha llegado a la conclusión de que el algoritmo que mejor se adapta al contexto del problema es: SVM. Esta deducción fortalece las conclusiones ya obtenidas por otros estudios previos (Apartado 7.4.2). Con respecto a la herramienta NLP *Freeling* utilizada en la plataforma (Apartado 3.5.2). Queda demostrado el buen rendimiento que ofrece con respecto al proceso de análisis gramatical y etiquetado (Apartados 4.2.3 y 4.3.3). A su vez, la facilidad, portabilidad y programabilidad que le caracteriza. Por último concluir que el paradigma de programación orientado a objetos y los distintos lenguajes d programación utilizados en el TFG (Apartado 6.2) han posibilitado la implementación del algoritmo de una manera sencilla y eficaz que se traduce en el desarrollo de un algoritmo óptimo tanto en uso de recursos como en términos de precisión obtenida.

Con todo ello se ha conseguido desarrollar un algoritmo que ha alcanzado en algunas pruebas porcentajes de acierto del 100%, con una media del 92.65 % de acierto, superando la precisión obtenida hasta el momento (Apartado 7.4.5). Cabe decir que en el proceso de comparación hay que tener en cuenta las limitaciones que impone el uso de un banco de pruebas distinto, lo que puede ocasionar cambios en las precisiones obtenidas.

La conclusión final de este TFG es que la metodología que se propone es eficaz y propia, que permite la clasificación de textos españoles con una precisión al nivel de los estudios actuales asociados al problema de la atribución de autoría de textos.

8.3 Limitaciones

La principal limitación se refiere a que el estudio se centra en textos españoles actuales. La mayoría de estudios en el campo de la atribución de la autoría utilizan textos ingleses, mientras que en el caso de documentos en español apenas existen trabajos al respecto. De esta manera, durante el TFG se ha tenido que extrapolar todo el conocimiento de las distintas técnicas existentes con respecto a la atribución de la autoría con textos no españoles al lenguaje español.

Otra de las limitaciones es la ausencia de un banco de pruebas general para documentos españoles que permita analizar y comparar la bondad del algoritmo en el ámbito de la lengua española. En el contexto inglés, en cambio, existen estas bases de datos, los *Federalist Papers* son un ejemplo de ello (Mosteller & Wallace 1963). Esto dificulta el proceso de comparación de la metodología propuesta con las técnicas ya existentes. Asimismo, este banco pruebas debe contemplar los factores que influyen en el rendimiento del proceso de clasificación. En este caso, serian: el tamaño del corpus, la longitud de los documentos y el número de autores de BA.

Los documentos utilizados para el desarrollo del algoritmo han sido obtenidos a partir de libros electrónicos y noticias publicadas en Internet. Esto implica la posibilidad de que el conjunto de textos seleccionado haya podido sufrir alguna modificación previa por parte de una tercera persona. La consecuencia es la posible inclusión de ruido en los documentos, ya que podría existir una aportación estilística de otra persona. Este hecho puede alterar la *firma del autor*. Por ello, la mejor manera de probar este tipo de algoritmos es mediante el uso de textos que únicamente hayan sido manipulados por el autor correspondiente. De este modo, se conseguirían resultados fiables y precisos.

La metodología propuesta en el TFG contempla el uso de múltiples algoritmos de clasificación avanzada (SVM, MLP y *Näive Bayes*). Los estudios analizados contemplan el uso de diferentes elementos estilísticos (Grieve 2007). Sin embargo, no ha sido posible encontrar soluciones en las que intervengan varios algoritmos de clasificación avanzada. De este modo, se complica aún más la tarea de comparación.

Por último indicar que el tiempo y los recursos disponibles no han permitido desarrollar todas las pruebas que se necesitaban realizar. Es el caso de la prueba que variaba el número de documentos de entrenamiento. Al no disponer de un banco de documentos muy extenso y que las pruebas tardaban en realizarse mucho tiempo (horas en algunos experimentos), las gráficas obtenidas no muestran todos los resultados posibles. Asimismo, con respecto a la técnica BEM no se ha podido comparar su efectividad con el resto de pruebas.

8.4 Líneas Futuras

Como ya se ha dicho en el punto anterior, es necesario seguir investigando en el campo de la atribución de autoría, sobre todo en el ámbito español (Blasco & Ruiz, 2009). De este modo, las principales líneas futuras que se proponen son:

- Explorar nuevas combinaciones de características, siempre y cuando no dependan del contexto. Por ejemplo, el caso de utilizar como indicador de estilo los *n*-gramas que se puede añadir fácilmente al algoritmo propuesto en este TFG.

- Analizar y profundizar más detalladamente en el proceso de normalización de textos con distintas longitudes.

- Estudiar el comportamiento de la metodología propuesta para distintos corpus de entrenamiento: balanceados, no balanceados.

- Creación de un banco de pruebas público, donde los documentos sean fiables y permitan la obtención de resultados fidedignos y precisos.

- Analizar el comportamiento del algoritmo con otros modelos de clasificación avanzada.

- Estudiar la naturaleza de los resultados con otros mecanismos de reducción de la dimensionalidad del problema.

- Medir la eficacia del algoritmo con textos escritos en otros idiomas. Por ejemplo medir su eficacia con los *Federalist Papers.*

- Analizar el comportamiento del algoritmo utilizando la técnica BEM mediante la realización de las pruebas propuestas en el Apartado 7.2 u otras que puedan surgir.

Bibliografía

Abbasi, A. & Chen, H. (2005) Applying authorship analysis to extremist-group web forum messages. *IEEE Intelligent Systems*, 20(5):67-75.

Aggarwal, C. (2006a) Chapter 4th, A survey of text clustering algorithms. *Mining Text Data.* Ed. Springer, Londres. UK.

Aggarwal, C. (2006b) Chapter 6th, A survey of text classification algorithms. *Mining Text Data.* Ed. Springer, Londres. UK.

Aggarwal, C.C. & Zhai, C. (2012) *Mining Text Data.* Ed. Springer. USA.

Antosch, F. (1969) The Diagnosis of Literary Stylewith the Verb-Adjective Ratio. *In Statistics and Style.* Eds. L. Doleztel and R.W. Bailey. USA.

Argamon, S. & Levitan, S. (2005) Measuring the usefulness of function words for authorship attribution. *(1-3).*

Argamon, S., Whitelay, C., Chase, S., Hota, R. S., Garg, N. & Levitan, S. (2007) Stylistic Text Classification Using Functional Lexical Features: Research Articles. *Journal of the American Society for Information Science and Technology*, 58(6):802-822.

Baker, L. & McCallum, A. (1998) Distributional Clustering of Words for Text Classification, *ACM SIGIR Conference.* (96-103).

Baayen, R., Halteren, H., & Tweedie, F. (1996) Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*. 11(3):121–131.

Boser, B.E., Guyon, I.M. & Vapnik, V.N (1992) A Training Algorithm for Optimal Margin Classifiers. *Proc. Fifth Ann. Workshop Computational Learning Theory*. ACM Press, New York. (144-152).

Blasco, J. & Ruiz, C. (2009) Evaluación y cuantificación de algunas técnicas de Atribución de autoría en textos españoles. *Revista Castilla. Estudios de Literatura*, 0:27-47.

Brainerd, B. (1973) On the Distinction Between a Novel and a Romance: A Discriminant Analysis. *Computer and the Humanities*, 7(5):259-270.

Brainerd, B. (1974) *Weighing Evidence in Language and Literature: A Statistical Approach.* Ed. University of Toronto Press. USA.

Burrows, J.F. (1987) Word patterns and story shapes: The statistical analysis of narrative style. *Literary and Linguistic Computing*. 2(2):61-70.

Burrows, J.F. (1989) An ocean where each kind. . . ': Statistical analysis and some major determinants of literary style. *Computers and the Humanities*, 23(4):309–21.

Burrows, J.F. (1992) Not unless you ask nicely: The interpretative nexus between analysis and information. *Literary and Linguistic Computing*, 7(2):91–109.

Burrows, J.F. (2007) All the way through: testing for authorship in different frequency strata. *Literary and Linguistic Computing*, 22(1):27–47.

Burns, A. & Wellings, A. (2009) *Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX 4th Edition*. Ed. Addison-Wesley, USA.

Chaski, C.E. (2005) Empirical evaluations of language-based author identification techniques. *Forensic Linguistics*, 8(1):1-65.

Chopra, A., Prashar A., & Sain, C. (2013) Natural Language Processing. *International Journal of Technology Enhancements and Emerging Engineering Research,* 1(4):131-134.

Cooley, R., Mobasher, B., & Srivastava, J. (1997). Web mining: information and pattern discovery on the World Wide Web. *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence.* (558-567).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcouglu, K., & Kuksa, P. (2011) Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493-2537.

Corazón, M. (2008) *Descartes: Un Nuevo modo de hacer filosofía*. Ed. Universidad de Navarra. España.

Coyotl-Morales, R.M., Villaseñor-Pineda, L., Montes-y-Gòmez, M., & Rosso, P. (2006) Authorship attribution using Word sequences. *In Proceedings of the 11th Iberoamerican Congress on Patterns Recognition.* (844-853).

Craven, M. & Kumlien, J. (1999) Constructing Biological Knowledge-Bases by Extracting Information from Text Sources. *In Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology.* (77–86).

Crescenzi, V., Mecca, G., & Merialdo, P. (2001) Road-Runner: Towards automatic data extraction from large Web sites. *In Proceedings ofthe27th International Conference on Very Large Data Bases.* (109-118).

Cutting, D.R., Karger, D. R., Pedersen, J.O., & Tukey, J.W. (1992). Scatter/Gather: A Cluster-Based Approach to Browsing Large Document Collections. *In Proceedings of the 15th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.* (318-329).

Dietterich, T.G. (2000) Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1-15

Dunteman, H. (1989) Principal Component Analysis. *Journal of Educational* Statistics, 16(2):141-144.

Ellegard, A. (1962) *A Statistical Method for Determining Authorship: The Junius Letters, 1769-1772.* Ed. University of Göteborg Press. Sweden.

Feldman, R. & Sanger, J. (2006) *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data.* Ed. Cambridge. UK.

Grant, T.D. (2007) Quantifying evidence for forensic authorship analysis. *International Journal of Speech Language and the Law,* 14(1):1-25.

Feldman, R. & Sanger, J. (2007) *The Text Mining Handbook. Advanced Approaches in Analyzing Unstrctured Data.* Cambridge University Press. USA.

Fellbaum, C. (1998) *WordNet: An electronic lexical database.* Ed. MIT Press. USA.

Fisher, R. A. (1963). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics,* 7:179-188.

Flórez, R. & Fernández, J.M. (2008) *Las redes Neuronales Artificiales.* Ed. Nebiblio. La Coruña, España.

Frank, E. & Buckaert, R. (2006) Naïve Bayes for Text Classification with Unbalanced Classes. *Lecture Notes in Computer Science*. 4213:503-510.

Frantzeskou, G., Stamatatos, E., Gritzalis, S., & Katsikas, S. (2006) Effective identification of source code authors using byte-level information. *In Proceedings of the 28th International Conference on Software Engineering*. (893-896).

Freeling (2014a) Manual de la herramienta *Freeling* versión 3.1. http://nlp.lsi.upc.edu/freeling/doc/userman/userman.pdf. Último acceso 20 de Junio de 2014.

Freeling (2014b) Página *web* con las etiquetas que utiliza la herramienta *freeling* para etiquetar la información morfológica de las palabras. http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html. Último acceso 20 de Junio de 2014.

Fucks, W. (1952) On the Mathematical Analysis of Style. *Biometrika*, 39(1-2):122-129.

Fucks,W. & Lauter, J. (1965) Mathematische Analyse des Literarischen Stils. *In Mathematik und Dichtung.* Ed. H. Kreuzer and R. Gunzenhausers. Munich: Nymphenburger Verlagsbuckhandlung.

Fukuda, K., Tamura, A., Tsunoda, T., & Takagi, T. (1998) Toward Information Extraction: Identifying protein names from biological papers. *In Proceedings of the Pacific Symposium on Biocomputing*. (707-718).

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

Grzybek, P. (2006) History and Methodology of Word Length Studies. *Contributions to the Science of Text and Language Text, Speech and Language Technology*, 31:15-90.

Grieve, J. (2007) Quantitative authorship attribution: An evaluation of techniques. *Literary and Linguistic Computing*, 22(3):251-270.

Halvani, O., Steinebach, M. & Zimmermann, R. (2013) Authorship Verification via k-Nearest Neighbor Estimation. *Notebook for PAN at Conference and Labs of the Evaluation Forum*. Valencia.

Hansen, L. K. & Salamon, P. (1990) Neural network ensembles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(10):993-1001.

Hirst, G. & Feiguina, O. (2007) Bigrams of syntactic labels for authorship discrimination of short texts. *Literary and Linguistic Computing,* 22(4):405-17.

Hsu, C., Chang, C., & Lin, C. (2010) *A Practical Guide to Support Vector Classification*. Ed. National Taiwan University. Taiwan.

Hush, R. & Horne, B.G. (1993) Progress in Supervised Neural Networks. *IEEE Signal Processing Magazine*, 10(1):8-39.

Hush, R. & Horne, B.G. (1992) An overview of neural networks. Part I: Static Networks. *Informática y Automática*. 25(1):19-36.

Hearst, M. (1998). Support vector machines. *Intelligent systems and their Applications, IEEE,* 13(4):18-28.

Holmes, D.I. (1994) Authorship Attribution. *Computers and the Humanities*, 28(2):87-106.

Holmes, D.I. (1998) The evolution of stylometry in humanities scholarship. *Literary and Linguistic Cooputing*, 13(3):111-117.

Honore, A. (1979) Some simple measures of richness of vocabulary. *Association for Literary and Linguistic Computing Bulletin*, 7(2): 172-177.

IntertekGroup (2002) *Leveraging Unstructured Data in Investment Management*, http://www.taborcommunications.com/dsstar/02/0604/104317.html. Último acceso 20 de Junio de 2014.

Jiang, J. (2006) Chapter 2nd, Information extraction from Text. *Mining Text Data.* Ed. Springer, Londres. UK.

Juola, P. & Stamatatos, E. (2013) Overview of the Author Identification Task at PAN 2013. *Proccedings of PAN* 2013. Valencia.

Karatzoglou, A., Meyer, D., & Hornik, K. (2006) Support Vector Machines in R. *Journal of Statistical Software,* 15(9):1-28.

Klopthenko, A., Eklund, T., Back, B., Karlsson, J., Vanharanta, H., & Visa, A. (2004) Combining Data and Text Mining Techniques for Analyzing Financial Reports. *International Journal of Intelligent Systems in Accounting and Finance Management.* 12(1):29-41.

Kohavi, R. (1995) A Study if Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence*. 2:1137-1143.

Koppel, M. & Schler, J. (2003). Exploting stylistic idiosyncrasies for authorsip attribution. *In Proceedings of IJCAI'03 Workshop on Computational approaches to Style Analysis and Synthesis.* (69-72).

Koppel, M., Schler, J., & Bonchek-Dokow, E. (2007) Measuring differentiability: Unmasking pseudonymous authors. *Journal of Machine Learning Research*. 8:1261-1276.

Koppel, M., Schler, J., & Argamon, S. (2013) Authorship Attribution: What's Easy and What's Hard? *Journal of Law & Policy*, 31:317-331.

Lippman, R. P. (1987) An introduction to computing with neural nets. *IEEE ASSP Magazine,* 4(2):4-22.

Liu, B. & Zhang, L. (2006) Chapter 13th. A survey of Opinion Mining and Sentiment Analysis. *Mining Text Data.* Ed. Springer, Londres.UK.

Louden, K. C. & Lambert, A. K. (2012) Chapter 1: Introduction. *Programming Languages: Principles and Practice, Third Edition.* Ed. Course Technology, USA.

Luyckx, K. & Daelemans, W. (2011) The effect of author set size and data size in authorship attribution. *Literary and Linguistic Computing*, 26(1):35-55.

Matsuura, T. & Kanada, Y. (2000) Extraction of authors' characteristics from Japanese modern sentences via n-gram distribution. *In Proceedings of the 3rd International Conference on Discovery Science.* (315-319).

McCarthy, P.M., Lewis, G.A., Dufty, D.F., & McNamara, D.S. (2006) Analyzing writing styles tih coh-metrix. *In Proceedings of the Florida Artificial Intelligence Research Society International Conference.* (764-769).

Mendenhall, T.C. (1887) The characteristic curves of composition. *Science*, 9(214):237-246.

Mendenhall, T.C. (1901) A mechanical solution of a literary problem. *Popular Science Monthly*, 60(2):97-105.

Morton, G. & Michaelson, S. (1990). *The qsum plot*. Technical Rport CSR-3-90, University of Edinburg, UK.

Mosteller, F. & Wallace, D.L. (1963) Inference in an Authorship problem. A comparative study of discrimination methods applied to the authorship of the disputed Federalist papers. *Journal of the American Statistical Association*, 58(302):275-309.

Mosteller, F., Fienberg S. E., Hoaglin C. D., & Tanur J.M. (2010) Chapter 4th Who wrote the Disputed Federalist Papers, Hamilton or Madison?. *The Pleasures of Statistics: The Autobiography of Frederick Mosteller*. Ed. Springer. Londres. UK.

Murphy, K.P. (2006) *Naive Bayes classifiers*. Ed. University of British Columbia. Canada.

Nirkhi, S., Dharaskar, R.V.& Thakare, V.M. (2014) Authorship Attribution of online messages using Stylometry: An Exploratory Study. *International Conference on Advances in Engineering and Technology (ICAET'*2014). (254-257).Padró, L. &

Stanilovsky, E. (2012) Freeling 3.0: Towards Wider Multilinguality. *Proceedings of the Language Resources and Evaluation Conference (LREC 2012). (2473-2479).*

Página *web* del grupo de procesado natural del lenguaje de la Universidad de Standford. http://www-nlp.standford.edu. Último acceso 20 de Junio de 2014.

Página web de TALP. http://www.talp.upc.edu. Último acceso 20 de Junio de 2014.

Página *web* del API de JAVA. http://docs.oracle.com/javase/7/docs/api/. Último acceso 20 de Junio de 2014.

Página *web* del proy*ecto* R. http://www.r-project.org/. Último acceso 20 Junio de 2014.

Página *web* del paquete TM. http://tm.r-forge.r-project.org/. Último acceso 20 de Junio de 2014.

Página *web* del manual del paquete e1071. http://cran.r-project.org/web/packages/e1071/index.html. Último acceso 20 de Junio de 2014.

Página *web* del manual del paquete nnet. http://cran.r-project.org/web/packages/nnet/index.html. Último acceso 20 de Junio de 2014.

Página *web* oficial de la plataforma Rstudio. http://www.rstudio.com/. Último acceso 20 de Junio de 2014.

Página *web* del periódico *El país.* http:// elpais.com. Último acceso 20 de Junio 2014.

Página *web* de la versión 12.04 LTS del sistema operativo Ubuntu. http://releases.ubuntu.com/precise/. Último acceso 20 de Junio de 2014.

Página *web* de la verisón *Indigo* de Eclipse. http://www.eclipse.org/indigo/. Último acceso 20 de Junio de 2014.

Peng, F., Shuurmans, D., & Wang, S. (2004) Augmenting naive Bayes classifiers with statistical language models. *Information Retrieval Journal,* 7(3-4):317-345.

Rindflesch, T. C., Hunter, L., & Aronson, A. R. (1999) Mining Molecular Binding Terminology from Biomedical Text. *In Proceedings of the '99 AMIA Symposium.* (127-131).

Russell, J. S. & Norvig, P. (1995) Chapter 1 & 2. Introduction. *Artificial Intelligence: A Modern Approach.* Ed. Prentice Hall. Nueva Jersey. USA.

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

Sebastiani, F. (2002) Machine learning in automated text categorization. *ACM Computing Surveys*. 34(1):1-47.

Sichel, H. S. (1975) On a Distribution Law for Words Frequencies. *Journal of the American Statistical Association,* 70(351):542-547.

Somers, H. H. (1966) Statistical Methods in Literary Analysis. *In The Computer and Literary Style.* Ed. Kent State University Press. USA.

Spenke, M. & Beilken, C. (1999) Visual, Interactive Data Mining with InfoZoom –The Financial Data Set. *In Proceedings of the 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases. (1-6).*

Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2000). Automatic text categorization in terms of genre and author. *Computational Linguistics,* 26(4):471–495.

Stamatatos, E., Fakotakis, N., & Kokkinakis, G. (2001). Computer-based authorship attribution without lexical measures. *Computers and the Humanities,* 35:193–214.

Stamatatos, E. (2006) Authorship attribution based on feature set subspacing ensembles. *International Journal on Artificial Intelligence Tools*, 15(5):823-838.

Stamatatos, E. (2009) A survey of moderns authorship attribution methods. *Journal of the American Society and Information Sciences*, 60(3): 538-556.

Tallentire, D.R. (1972) *An Appraisal of Methods and Models in Computational Stylistics, with Particular Refernce to Author Attribution.* PhD thesis. University of Cambridge.

Tambouratzis, G., Markantonatou, S., Hairetakis, N., Vassiliou, M., Carayannis, G., & Tambouratzis, D. (2004) Discriminating the registers and styles in Modern Greek language –Part2: Extending the feature vector to optimize author discrimination. *Literary and Linguistic Computing*, 19(2):221-242.

Varela, P.J. (2010) *O uso de atributos estilométricos na identificaçao da autoria de textos*. Ed. Universidad Pontificia de Paraná. Portugal.

Varela, P.J., Justino, E.J.R & Oliveira, L.E.S. (2011) Identificaçao de Autoria de Textos através o uso de Classes Linguísticas da Língua Portuguesa. *Proceedings of the 8th Brazilian Sumposium in Information and Human Language Technology*. (174-178).

Vapnik,V. (1998) *Statistical Learning Theory.* Ed. Wiley, Ney York. USA.

Wake, W.C. (1957) Sentence-Length Distributions of Greek Authors. *Journal of the Royal Statistical Society Series A,* 120(3):331-346.

Williams, C.B. (1940) A Note on the Statistical Analysis of Sentence-Length as a Criterion of Literary Style. *Biometrika,* 31(3-4):356.361.

Witten, I.H., Frank, E., & Hall, M.A. (2011) *Data Mining, Practical Machine Learning Tools and Techniques*. 3ª Edición. Ed. Elsevier, Madrid, España.

Yu, H. & Kim, S. (2012) Capítulo 15: SVM Tutorial – Classification, Regression and Ranking. *Handbook of Natural Computing.* Ed. Springer. Berlin. Alemania.

Yule, G.U. (1939) On sentence-length as a statistical characteristic of style in prose, with application to two cases of disputed authorship. *Biometrika,* 30(3-4):363-390.

Yule, G.U. (1944) *The statistical study of literary vocabulary*. Ed. Cambridge University Press. USA.

Zanasi, A. (2007) Chapter 8th. Open sources automatic analysis for corporate and government intelligence. *Text Mining and its Applications to Intelligence, CRM and Knowledge Management.* Ed. WIT Press. UK.

Zhao, Y. & Zobel, J. (2005) Effective and Scalable Authorship Attribution Using Function Words. *Information Retrieval Technology. Lecture Notes in Computer Science*, 3689:174-189.

Zheng, R., Li, J., Chen, H., & Huang, Z. (2006) A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the American Society for Information Science and Technology*, 57(3):378–393.

Zhou, M. & Cui, Y. (2004) GeneInfoViz: Constructing and Visualizing Gene Relation Networks. *Silico Biology,* 4:323–333.

Zipf, G.K. (1932) *Selected studies of the principle of relative frequency in language*. Ed. Harvard University Press. USA.

Apéndice A: Siglas

ANN: Red neuronal artificial (*artificial neural network*)

AP: Área de precisión

APR: Arturo Pérez Reverte

as: Asturiano

cy: Galés

BA: Banco de autores

BEM: Modelo de mejor esfuerzo (*best effort model*)

ca: Catalán

CC: Curvas características

CRZ: Carlos Ruiz Zafón

CUSUM: Suma acumulativa (*cumulative sum)*

DA: Documento adaptado

DP: Documento plano

DSA: Documento sin autoría

en: Inglés

es: Español

FA: Función de activación

FAu: Firma del autor

FJP: Fernando J. Pérez

fr: Francés

FS: Función de salida

FW: Palabras funcionales (*function words*)

gl: Gallego

IA: Inteligencia artificial

it: Italiano

LN: Lenguajes naturales

LR: Lista de resultados

MD: Miguel Delibes Setién

ME: Matriz de entrenamiento

MJ: Miguel Jiménez

MLP: Perceptrón multicapa

MM: Miguel Mora

MP: Modelo primario

MS: Modelo secundario

MT: Matriz de test

MTG: Matriz de *TAGS*

NLP: Natural language processor

PC: Componente principal (*principal component*)

PCA: Análisis de componentes principales (*principal component analysis*)

POS: *Part-of-Speech*

RBF: Función de base radial (*radial basis function*)

ru: Ruso

sl: Esloveno

TFG: Trabajo de Fin de Grado

Apéndice B: Ficheros de propiedades

B.1 Fichero "tags.properties"

```
#Base de datos con los TAGS del Programa

#ADJETIVOS: TAGS que empiecen con A
A=Adjetivo

#Modificadores del adjetivo

#Tipo
ATQ=Calificativo
ATO=Ordinal
AT0=No

#Grado
AGR0=No
AGRA=Aumentativo
AGRC=Disminutivo
AGRS=Superlativo

#Genero
AGM=Masculino
AGF=Femenino
AGC=Comun

#Numero
ANS=Singular
ANP=Plural
ANN=Invariable

#Funcion
AF0=No
```

```
AFP=Participio
################################################################

############
#Adverbios
############
R=Adverbio

#Valor
RVG=General
#Solo para el adevervbio no
RVN=Negativo
##############################

##########################################
#Determinantes
##########################################
D=Determinante
#Tipo
DTD=Demostrativo
DTP=Posesivo
DTT=Interrogativo
DTE=Exclamativo
DTI=Indefinido
DTA=Articulo
#Persona
DP1=Primera
DP2=Segunda
DP3=Tercera
#Genero
DGM=Masculino
DGF=Femenino
```

DGC=Comun

DGN=Neutro

#Numero

DNS=Singular

DNP=Plural

DNN=Invariable

#Poseedor

DPS=Singular

DPP=Plural

DP0=No

##

####################

#Nombres

####################

N=Nombre

#Tipo

NTC=Comun

NTP=Propio

#Genero

NGM=Masculino

NGF=Femenino

NGC=Comun

NG0=No

#Numero

NNS=Singular

NNP=Plural

NNN=Invariable

NN0=No

#Clasificacion Semantica

NCSP=Persona

NCG0=Lugar

NCO0=Organizacion

NCV0=Otros

NC00=No

#Grado

NGA=Aumentativo

NGD=Diminutivo

NG0=No

##

#################

#Verbos

#################

V=Verbo

#Tipo

VTM=Principal

VTA=Auxiliar

VTS=Semiauxiliar

VT0=No

#Modo

VMI=Indicativo

VMS=Subjuntivo

VMM=Imperativo

VMN=Infinitivo

VMG=Gerundio

VMP=Participio

VM0=No

#Tiempo

VTP=Presente

VTI=Imperfecto

VTF=Futuro

VTS=Pasado

VTC=Condicional

VT0=No

#Persona

VP1=Primera

VP2=Segunda

VP3=Tercera

VP0=No

#Numero

VNS=Singular

VNP=Plural

VN0=No

#Genero

VGM=Masculino

VGF=Femenino

VG0=No

##

####################

#Pronombres

####################

P=Pronombre

#Tipo

PTP=Personal

PTD=Demostrativo

PTX=Posesivo

PTI=Indefinido

PTT=Interrogativo

PTR=Relativo

PTE=Exclamativo

PT0=No

#Persona

PP1=Primera

PP2=Segunda

PP3=Tercera

PP0=No

#Genero

PGM=Masculino

PGF=Femenino

PGC=Comun

PGN=Neutro

PG0=No

#Numero

PNS=Singular

PNP=Plural

PNN=Impersonal e Invariable

PN0=No

#Caso

PCN=Nominativo

PCA=Acusativo

PCD=Dativo

PCO=Oblicuo

PC0=No

#Poseedor

PPS=Singular

PPP=Plural

#Educacion(Usted, ustedes y vos)

PEP=Cortesia

PE0=No

##

###################

#Conjunciones

##

C=Conjuncion

#Tipo

CTC=Coordinada

CTS=Subordinada

##

######################

```
#Interjecciones  (Ah eh ejem ele...)
######################
I=Interjeccion
#####################################################
######################
#Preposiciones
######################
S=Adposicion
#Tipo
SPP=Preposicion
SP0=No
#Forma
SFS=Simple
SFC=Contraida
SF0=No
#Genero
SGM=Masculino
SG0=No
#Numero
SNS=Singular
SN0=No
############################################################
############################
#Signos de puntuacion
############################
F=Puntuacion
FAA=Exclamacion inicio
FAT=Exclamacion final
FC=coma
FCA=Corchete inicio
FCT=Corchete final
FD=Dos punto
```

FE=Comillas

FG=Guion

FH=Contrabarra

FIA=Interrogacion inicio

FIT=Interrogacion final

FLA=Corchete inicio

FLT=Corchete cierre

FP=Punto

FPA=Parentesis inicio

FPT=Parentesis cierre

FRA=Angulos inicio

FRC=Angulos cierre

FS=Puntos suspensivos

FT=Porcentaje

FX=Punto y coma

FZ=Signo matematico

##

##################

#Cifra y numerales

##################

Z=Cifra

#Tipo

#Partitivo: Docena, un millon, centenar...

ZTD=Partitivo

ZTM=Moneda

ZTP=Porcentaje

ZT0=No

#Unidad: 80 km por hora

ZTU=Unidad

###

##############################

#Fecha y Hora

```
##############################
W=Fecha/Hora
#################################################################
```

B.2 Fichero "nucleo.properties"

```
#Este fichero contiene los parametros configurables
#Que utilizará la herramienta de detección de autoria


#Firma de Tags#
################################################

#Numero de tags que tendrá la firma
num_firma=30
#Numero máximo de outliers permitidos
num_outliers_allow=2
#Tamaño de los whiskers (bigotes) del boxplot valor mayor que 1
range_valor=1.2
#Mostrar resultado del boxplot (TRUE o FALSE) FALSE por defecto
plot_box=FALSE
#Nº minimo de componentes dentro de un mismo tag de la firma
token_min=3
#Nº máximo de componenetes dentro de un mismo tag de la firma
token_max=6
################################################

#Function words#
################################################

#Numero de palabras funcionales que tendrá la firma
num_func_words=30
```

```
################################################

#PCA#
################################################

#Numero de componenetes principales
num_PCA=2

#Variable PCA: Permite realizar analisis PCA
#Valores posibles:
#SI : Se realiza anlisis PCA
#NO : No se realiza analisis PCA (Valor por defecto)
PCA = NO
```

Apéndice C: Pliego de condiciones

El presente documento contiene los requisitos legales que se han de cumplir para la realización del TFG *Desarrollo e implementación de un algoritmo basado en text mining aplicado a la atribución de autoría*. Para ello se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa de desarrollo de software, con la finalidad de utilizar las herramientas desarrolladas como medio para el análisis de la autoría de un documento.

La empresa desarrolladora ha de seguir una línea de investigación con objeto de desarrollar el proyecto. Esta línea de investigación, junto con el posterior desarrollo del programa, está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulara por las siguientes condiciones:

- **Condición 1:**

La modalidad de contratación será el concurso. El proyecto se adjudicará a la propuesta más favorable, sin atender exclusivamente al presupuesto del proyecto, sino atendiendo también a las garantías ofertadas. La empresa que somete el proyecto a concurso se reserva el derecho de declararlo desierto.

- **Condición 2:**

El montaje y mecanización completa de los equipos necesarios para el desarrollo del proyecto será realizado en su totalidad por la empresa licitadora.

- **Condición 3:**

En la oferta se hará constar el precio total por el que se compromete a realizar la obra y el porcentaje de baja que se supone este precio en relación con un importe límite, si este se hubiera fijado.

- **Condición 4:**

La obra o proyecto se realizara bajo la dirección técnica de un graduado en Telecomunicación, auxiliado por el número de Ingenieros Técnicos Programadores que sea preciso para el desarrollo de la misma.

- **Condición 5:**

Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

- **Condición 6:**

El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

- **Condición 7:**

Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que, con arreglo a sus facultades, le haya comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ejecutado de acuerdo a los preceptos de los pliegos de condiciones. Las modificaciones y la valoración de las diversas unidades se harán de acuerdo con dichos preceptos, sin que el importe total pueda exceder de los presupuestos aprobados.

Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrán servir de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

- **Condición 8:**

Tanto en las certificaciones de obra como en la liquidación final, se abonaran los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

- **Condición 9:**

Si, excepcionalmente, se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que, sin embargo, es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero considere justa y, si la Dirección resolviera aceptar la obra, quedara el contratista obligado a conformarse con la rebaja acordada.

- **Condición 10:**

Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluara su importe en función de los precios asignados a otras obras o materiales análogos si los hubiere y, cuando no los hubiere, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento se sujetaran siempre al establecido en el punto anterior.

- **Condición 11:**

Cuando el contratista, con la autorización del Ingeniero Director de obras, emplee material de calidad más elevada o de mayores prestaciones de lo estipulado en el proyecto, sustituya una clase de desarrollo por otra que tenga asignado mayor precio, ejecute con mayores dimensiones cualquier otra parte de las obras o, en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sino a lo que le correspondería si se hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

- **Condición 12:**

Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final, no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen o, en su defecto, por lo que resulte de su medición final.

- **Condición 13:**

El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras, así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

- **Condición 14:**

Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

- **Condición 15:**

La garantía definitiva será del 4% del presupuesto y, la provisional, del 2%.

- **Condición 16:**

La forma de pago será por certificaciones mensuales de la obra ejecutada de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

- **Condición 17:**

La fecha de comienzo de la obra será a partir de los quince días naturales del replanteo oficial de las mismas, y la definitiva, al año de haber ejecutado la provisional, procediéndose, si no existe reclamación alguna, a la reclamación de la fianza.

- **Condición 18:**

Si el contratista, al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras pues, transcurrido ese plazo, será responsable de la exactitud del proyecto.

- **Condición 19:**

El contratista está obligado a designar una persona responsable que se reunirá con el Ingeniero Director o con el delegado que este designe, para todo lo relacionado con ella. Al ser el Ingeniero Director el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

- **Condición 20:**

Durante la realización de la obra, habrá visitas de seguimiento del desarrollo por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro total o parcial de ella, aunque sea por agentes externos a su empresa, deberá ser reparado o construido por su cuenta.

- **Condición 21:**

El contratista deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa por retraso en la ejecución siempre que este no sea debido a causas de fuerza mayor. A la terminación de la obra se hará una recepción

provisional, previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, con la conformidad del contratista.

- **Condición 22:**

Hecha la recepción provisional, se certificara al contratista el resto de la obra, reservándose a la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas establecidas.

- **Condición 23:**

Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de octubre de 1961, se aplicaran sobre el denominado en la actualidad "Presupuesto de Ejecución de contrata", y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

La empresa de desarrollo de software que ha desarrollado este proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

- **Condición particular 1:**

La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenecen por entero a la empresa desarrolladora, representada por el Ingeniero Director del Proyecto.

- **Condición particular 2:**

La empresa desarrolladora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada bien para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos posteriores, para la misma empresa cliente o para otra.

- **Condición particular 3:**

Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contara con la autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuara en representación de la empresa de desarrollo de software.

- **Condición particular 4:**

En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

- **Condición particular 5:**

En todas las reproducciones se indicara su procedencia, explicando el nombre del proyecto, nombre del Ingeniero Director y empresa desarrolladora.

- **Condición particular 6:**

Si el proyecto pasa a la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y, a criterio de éste, la empresa desarrolladora decidirá aceptar o no la modificación propuesta.

- **Condición particular 7:**

Si la modificación se acepta, la empresa desarrolladora se hará responsable, al mismo nivel que el proyecto inicial, del proyecto que resulta al añadirla.

- **Condición particular 8:**

Si la modificación no es aceptada, por el contrario, la empresa desarrolladora declinara toda responsabilidad que se derive de la aplicación o influencia de la misma.

- **Condición particular 9:**

Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa desarrolladora.

- **Condición particular 10:**

La empresa desarrolladora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

- **Condición particular 11:**

La empresa desarrolladora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario realizar para dicha aplicación industrial, siempre que no haga explicita una renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

- **Condición particular 12:**

El Ingeniero Director del proyecto será responsable de la dirección de la aplicación industrial siempre que la empresa desarrolladora lo estime oportuno. En caso contrario, la persona asignada deberá contar con la autorización del mismo, quien delegara en el las responsabilidades que ostente.

DOCUMENTACIÓN

- Servicio de la Biblioteca de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid.
- Artículos de las revistas académicas en su versión *online*.
- Servicio de acceso online a revistas científicas de la Universidad de Valladolid
- Acceso a Internet.

HARDWARE

- Ordenador Intel Core i5-4670K 3.4 GHz, con 16GB de RAM.
- Impresora
- Disco duro extraíble Seagate® 1TB
- Medios de almacenamiento portátil.
- Consumibles (material papelería, CD-ROMs, DVDs,etc).

SOFTWARE

- VMware® Workstation 8.0.0 build-471780
- Eclipse versión Indigo
- Rstudio® 0.98.501
- Java® SE Development Kit 7u65
- R
- Microsoft® Windows 7 Profesional, Microsoft Corporation.
- Ubuntu® 12.04 LTS Precise 32 bits
- Microsoft® Office 2013, Microsoft Corporation

Apéndice D: Presupuesto

EJECUCIÓN MATERIAL

Ordenador Intel Core i5-4670K 3.4 GHz, con 16GB de RAM……..….....	1100,00 €
Impresora.....……...….	300,00 €
Software……....	1.200,00 €

TOTAL EJECUCCIÓN MATERIAL…….....	2.600,00 €

GASTOS GENERALES

18% sobre la ejecución material.……	468,00 €

BENEFICIO INDUSTRIAL

6% sobre la ejecución material……..	156,00 €

MATERIAL FUNGIBLE

Gastos de impresión...…….....	132,00 €
Gastos de encuadernación.……..…..	80,00 €
Medios de almacenamiento masivo.…….....	30,00 €
CD-ROM…….....	10,00€

TOTAL MATERIAL FUNGIBLE	240,00 €

HONORARIOS

360 horas a 30 €/hora…….....	10.800,00 €
SUBTOTAL……	**13.640,00 €**

IVA APLICABLE

21% sobre el subtotal del proyecto …….....	2.864,40 €
TOTAL DEL PROYECTO …….....	16.504.40 €

El total del presente presupuesto, asciende a:

DIECISEIS MIL QUINIENTOS CUATRO EUROS CON CUARENTA CÉNTIMOS

Valladolid, 2014-09-9 Fdo: Alejandro Pedrosa Antón