INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

Detección automática de plagio usando información sintáctica

TESIS

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

PRESENTA
M. en C. Juan Pablo Francisco Posadas Durán

DIRECTORES DE TESIS
Dr. Grigori Sidorov
Dr. Ildar Batyrshin

México, Ciudad de México marzo 2016

SIP-14 BIS

INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 14:15 horas del día 29 del mes de febrero de 2016 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Detección automática de plagio usando información sintáctica"

Presentada por el alumno:

Posadas	Durán	Juan Pablo Francisco
Apellido paterno	Apellido materno	Nombre(s)

Con registro: B 1 2 1 0 0 6

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

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

LA COMISIÓN REVISORA

Directores de tesis

Dr. Grigori Sidorov

Dr. Ildar Batyrshin

Dr. Sergio Suárez Guerra

Dr. Alexander Gelbukh

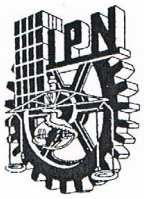
Dra. Sofía Natalia Galicia Haro

Dra. Nareli Cruz Cortés

PRESIDENTE DEL COLEGIO DE PROFESORES

Dr. Luis Alfonso Villa Vargas

INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION
EN COMPUTACION
DIRECCION



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 9 del mes marzo del año 2016, el que suscribe M. en C. Juan Pablo Francisco Posadas Durán alumno del Programa de Doctorado en Ciencias de la Computación con número de registro B121006, adscrito al Centro de Investigación en Computación, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del Dr. Grigori Sidorov y Dr. Ildar Batyrshin, cede los derechos del trabajo intitulado *"Detección automática de plagio usando información sintáctica"*, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

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

Juan Pablo Francisco Posadas Durán

Resumen

La acción de plagiar consiste en utilizar, de manera parcial o total, el trabajo creativo de alguien más sin el debido reconocimiento a los autores de dicho trabajo. Se considera al plagio como una acción delictiva porque quién lo realiza busca obtener beneficios mediante la falsa atribución de la autoría del trabajo creativo.

La detección automática de plagio de textos se puede realizar desde un enfoque intrínseco o un enfoque extrínseco. En el enfoque extrínseco se realiza una comparación directa entre un conjunto de textos para determinar cuáles han cometido plagio, mientras que en el enfoque intrínseco se realiza un análisis sobre el estilo de escritura presente en las diferentes secciones del texto para detectar aquellas que presenten un estilo marcadamente diferente y que se asume corresponden a un autor diferente.

La atribución de autoría es un caso particular de detección de plagio intrínseco donde se busca determinar si un texto fue escrito por un autor o si fue escrito por alguien más. Para resolver el problema de atribución de autoría se cuenta con un conjunto de autores candidatos y de algunas muestras de textos escritos por ellos, mediante un análisis de las muestras se construye un modelo sobre el estilo de escritura de los autores candidatos así como la obtención del estilo presente en un texto de autoría desconocida, se busca asociar el texto de autoría desconocida necesariamente con alguno de los autores candidatos.

En el problema de atribución de autoría se requiere de la construcción de un modelo sobre el estilo de escritura de un autor, es decir, la forma en que un autor utiliza el lenguaje escrito para expresar sus ideas. El estilo de escritura de un autor o simplemente el estilo de un autor se construye utilizando marcadores de estilo, los cuales hacen referencia a característica sobre el uso del lenguaje escrito que permite identificar y cuantificar hábitos del autor.

En esta tesis se propone un método para la atribución de autoría cerrada (detección de plagio intrínseca) utilizando un enfoque de aprendizaje automático supervisado. Se proponen dos estrategias para modelar el estilo de un autor: la primer estrategia utiliza una representación distribuida a nivel de documentos (*doc2vec*) en la que se analiza el contexto de ocurrencia de las palabras y bigramas de palabras para obtener un modelo sobre el estilo de un autor, la segunda estrategia se basa en el uso de n-gramas sintácticos analizando las variante de n-gramas mixtos.

El método propuesto se evaluó utilizando diferentes corpus reportados en la literatura para la atribución de autoría en los que se abarcan escenarios con distintos tipos de texto y con diferentes temáticas. Los resultados obtenidos utilizando una representación distribuida igualan o superan los resultados reportados en el estado del arte, mientras que el uso de n-gramas sintácticos demostró igualar la eficiencia obtenida en algunos corpus del estado del arte.

Abstract

Plagiarism can be defined as the action of using, partially or totally, the creative work of someone else without the corresponding recognition to the authors of that work. The plagiarism is considered as a criminal action because who performs it seeks benefits through the false authorship attribution of the creative work.

The plagiarism is committed in different areas, for example, in the cultural field (paintings, songs, photos, movies, plays), commercial field (software, registered patents, trademarks), scientific field (articles, methods, research, presentations), school filed (homeworks, reports, examinations, teaching materials), among others. The incidence of the plagiarism of texts has grown due to the lack of security mechanisms in the *Web*, so it is necessary to development methods for the automatic detection of plagiarism in texts.

The automatic plagiarism detection can be performed from an intrinsic or extrinsic approach. In the extrinsic approach a direct comparison between a set of texts is done to determine which one have committed plagiarism, on the other hand, in the intrinsic approach an analysis of the writing style present in the different sections of a text is performed to detect those sections that show a different style, which correspond to a different author.

The task of authorship attribution can be considered as a particular case of the intrinsic plagiarism detection that seeks to determine whether a text was written by an author or if it was written by someone else. In the problem of attribution of authorship there are a set of candidate authors and some samples of texts written by them, a model of the writing style of the authors is obtained by analyzing their samples and a model of an anonymous text is obtained. The goal is to assign the anonymous text to one of the candidate authors. If the author of the anonymous text is a member of the set of candidate authors then the task is known as closed authorship attribution, if the author is not necessarily in the set of candidate authors then the task is known as open authorship attribution.

The problem of authorship attribution requires to obtain a model for the writing style of an author, that is, the way an author uses written language to express his ideas. The writing style of an author or simply the style of an author can be obtained by using markers style, which refer to features about the use of written language that can identify and quantify habits of the author.

The written language offers different features (morphological, lexical, syntactic and semantic) that can be used as style markers. The character and word n-grams are the style markers that best performance have shown in the state of the art, however, the n-grams generate considerable noise plus they are not robust in scenarios with several authors (more 5 authors).

In this thesis a method for closed authorship attribution (intrinsic plagiarism detection) using a supervised machine learning approach is presented. Two strategies are proposed to model the style of an author: the first strategy uses a distributed document-level representation (*doc2vec*) where the context of words and bigrams of words are analyzed to obtain a model for the style of an author, the second strategy is based on the use of a variant of syntactic n-grams named mixed n-grams.

The proposed method was evaluated using different corpora reported in the literature that comprises scenarios with different types of text and different topics. The results obtained using a distributed exceed the results reported in the state of the art, while the use of syntactic n-grams demonstrated match the efficiency obtained for some corpora.

Dedicatoria

Dedico esta tesis a mi madre Irene Durán Orozco q.e.p.d y a mi padre Salomón Posadas Calderón

Agradecimientos

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

Agradezco a mis sinodales: **Dr. Alexander Gelbukh**, **Dr. Sergio Suárez Guerra**, **Dra. Nareli Cruz Cortés** y **Dra. Sofía Galicia Haro** por sus comentarios, sugerencias y observaciones.

Agradezco a mi padre **Salomón Posadas Calderón** y hermana **Gabriela Posadas Durán** por el apoyo y la comprensión a lo largo de este trabajo.

Agradezco a **Elibeth Mirasol Meléndez** por toda la paciencia y por todo su apoyo incondicional.

Agradezco a mis **compañeros de laboratorio** por su apoyo.

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

Índice general

Índice de figuras

Índice de tablas

Índice de algoritmos

Capítulo 1

Introducción

La acción de plagiar consiste en utilizar, de manera parcial o total, el trabajo creativo de alguien más sin el debido reconocimiento a los autores de dicho trabajo. El plagio es considerado un delito porque quién lo comete busca obtener beneficios al infringir la propiedad intelectual que los autores tienen sobre su trabajo creativo mediante la falsa atribución de la autoría de dicho trabajo.

El fenómeno del plagio se presenta en diferentes ámbitos, por ejemplo, en el ámbito cultural (pinturas, canciones, fotografías, películas, obras de teatro), comercial (uso no autorizado de patentes, software, logos), científico (artículos, métodos, investigaciones, presentaciones), escolar (tareas, reportes, exámenes, material didáctico), por mencionar algunos.

El plagio de textos consiste en copiar texto compuesto por alguien más de manera idéntica o distinta y hacerlo pasar como un texto original del plagiario al no incluir ninguna referencia del autor original (cita bibliográfica o nota al pie de página). Con el desarrollo de la *Web* surgió una diversidad de alternativas para la publicación de textos con exposición a nivel, sin embargo, la falta de mecanismos de seguridad en contra del plagio ha provocado la proliferación del plagio de textos dentro de la *Web*, por lo que es necesario el desarrollo de métodos para la detección automática del plagio de textos.

La detección de plagio de textos no es una tarea trivial porque el plagiador utiliza diferentes técnicas que le permitan engañar sobre la legitimidad de la autoría y ocultar su delito, por lo que se pueden considerar diferentes niveles de plagio de textos diferenciando cada uno de ellos por la dificultad para detectar el plagio (existen casos en los que es difícil inclusive para los especialistas llegar a un consenso). Por otra parte la detección de plagio de textos debe lidiar con la complejidad propia del lenguaje escrito, por ejemplo, diferentes géneros de texto (novela, noticia, reseña, crítica, entre otros), diferentes idiomas, por mencionar algunos.

La detección automática de plagio de textos se puede realizar desde un enfoque intrínseco o un enfoque extrínseco. En el enfoque extrínseco se realiza una comparación directa entre los textos de un conjunto para determinar cuales han cometido plagio, mientras que en el enfoque intrínseco se realiza un análisis sobre el estilo de escritura presente en las diferentes secciones de un texto para detectar aquellas que presentan un estilo marcadamente diferente y asumir que fueron plagiadas porque corresponden a un autor diferente (detección de anomalías, en inglés *outlier detection*). Una diferencia importante entre los dos enfoques es que en el enfoque extrínseco se dispone de los textos originales que han sido plagiados mientras que en el enfoque intrínseco no se cuenta con ellos.

La atribución de autoría se puede considerar como un caso particular de detección de plagio intrínseco en el que se busca determinar si un texto fue escrito por un autor o si fue escrito por alguien más. En el problema de atribución de autoría se cuenta con un conjunto de autores candidatos y con algunas muestras de textos escritos por ellos, mediante un análisis de las muestras se construye un modelo sobre el estilo de escritura de cada uno de los autores candidatos y el objetivo es identificar al autor de un texto anónimo mediante la comparación del estilo de escritura presente en el texto y los estilos de los autores candidatos. Si el autor del texto anónimo se encuentra dentro del conjunto de autores candidatos entonces la tarea se conoce como atribución de autoría cerrada, por otro lado, en el caso de que el autor no necesariamente se encuentre dentro del conjunto de autores candidatos entonces la tarea se conoce como atribución de autoría abierta.

En el problema de atribución de autoría, al igual que en el problema de detección de plagio intrínseca, se requiere de la construcción de un modelo sobre el estilo de escritura de un autor, es decir, identificar la forma en que un autor utiliza el lenguaje escrito para expresar sus ideas. Un modelo sobre el estilo de un autor se construye utilizando marcadores de estilo, los cuales hacen referencia a una característica sobre el uso del lenguaje escrito (morfológicas, léxicas, sintácticas, semánticas) que permite identificar y cuantificar hábitos que tienen los autores.

La eficiencia de un marcador de estilo depende de su capacidad para modelar el estilo de un autor independientemente de cambios en el tipo de texto y en la temática en que pueda incurrir un autor al componer un texto. Debido a la diversidad que presenta el lenguaje escrito difícilmente un solo marcador de estilo es suficiente para construir un modelo confiable sobre el estilo de un autor, por lo que típicamente se utilizan varios marcadores de estilo para la conformación de un modelo.

Los n-gramas de caracteres y los n-gramas de palabras son los marcadores de estilo que mejor eficiencia han reportado en investigaciones previas, sin embargo, esto marcadores de estilo son ruidosos y no son robustos en escenarios heterogéneos (corpus no balanceados, corpus que abarquen distintas temáticas, corpus que abarquen distintos géneros, entre otros). Existe la necesidad de métodos para la atribución automática de autoría que sean robustos y que permitan obtener modelos sobre el estilo de escritura para escenarios como los descritos anteriormente, por esta razón el análisis del estilo de un autor y la definición de marcadores de estilo siguen siendo temas de investigación dentro de la comunidad científica.

En esta tesis se propone un método para la atribución de autoría cerrada (detección de plagio intrínseco) utilizando un enfoque de aprendizaje automático supervisado. Se proponen dos estrategias para modelar el estilo de un autor: la primer estrategia se basa en un representación distribuida a nivel de documentos (*doc2vec*) y la segunda estrategia basada en el uso de n-gramas sintácticos considerando la variante de n-gramas mixtos.

El método propuesto fue evaluado utilizando diferentes corpus reportados en el estado del arte para la tarea de atribución de autoría cerrada en los que se abarcan escenarios con distintos números de autores candidatos, distintos géneros de texto (multi-género) y con distintas temáticas (multi-tema). El uso de una representación distribuida a nivel de

documentos demostró ser robusta al mejorar los resultados reportados en trabajos previos, especialmente al ser probado en uno de los corpus más desafiantes reportado en el estado del arte, un corpus desbalanceado, multi-tema y multi-genero, por otro lado, el uso de n-gramas sintácticos demostró igualar la eficiencia obtenida en algunos corpus del estado del arte.

1.1. Planteamiento del problema

Aunque todo ser humano es capaz de expresar contenido mediante el lenguaje escrito el estilo de un autor se considera único, de tal forma que no existen dos autores que expresen sus ideas de manera idéntica. El estilo de un autor hace referencia a los hábitos que éste tiene sobre el uso de los elementos y reglas del lenguaje escrito para componer un texto, estos hábitos se realizan de manera inconsciente y se ven reflejados a lo largo de diversos textos del autor [Uzuner et al. 2005].

Desde el punto de vista de la lingüística es posible construir un modelo sobre el estilo de un autor analizando la forma en que éste compone textos y describiendo el estilo en términos de los patrones encontrados, por ejemplo, combinaciones de palabras más utilizadas, forma en que generalmente comienza las oraciones, longitud promedio de las oraciones, entre otros. El análisis del estilo de un autor es utilizado para resolver diversas tareas dentro del Procesamiento del Lenguaje Natural (PLN) como son atribución de autoría, detección de plagio, obtención de perfil de usuario, validación de autoría, entre otras.

Los marcadores de estilo se pueden clasificar en función del tipo de características que utilizan para representar el estilo de un autor. Algunos marcadores de estilo utilizan la información que aparece de manera explícita en el texto sin necesidad de realizar algún procesamiento complejo, éstos generalmente trabajan a un nivel caracteres o palabras y algunos ejemplos son los n-gramas de caracteres, n-gramas de palabras, diversidad del vocabulario y frecuencia de uso de palabras auxiliares.

Por otro lado, existen marcadores de estilo que requieren del uso de herramientas lingüísticas (por ejemplo analizadores sintácticos, diccionarios especializados o etiquetadores de categorías gramaticales) para procesar un texto y permitan detectar hábitos que un autor tiene a un nivel sintáctico o semántico. Algunos ejemplos de este tipo de marcadores son los n-gramas de etiquetas POS, frecuencia de uso de los distintos tipos de palabra, uso de voz pasiva y de voz activa, entre otros.

En el estado del arte, el uso de marcadores de estilo basados en el análisis a nivel de palabras o a nivel de caracteres han sido predominantes debido a su sencilla implementación (no requiere de herramientas lingüísticas que realicen algún procesamiento complejo) y a la eficiencia que han obtenido en algunos corpus, sin embargo, un desventaja que presenta este tipo de marcadores de estilo es que requieren de un corpus de entrenamiento con una extensión grande (alrededor de miles de palabras), además los marcadores de estilo basados en palabras han mostrado dependencia temática, es decir, obtienen una baja eficiencia en escenarios donde los autores candidatos hablan de temas similares [Stamatatos 2009*b*].

En aplicaciones reales, como por ejemplo en el ámbito forense, comúnmente se trabaja con escenarios desafiantes en los que no existe una distribución homogénea del corpus (número desigual de muestras de los autores candidatos), los textos son de extensión corta (cientos de palabras), el número de autores candidatos es grande (mayor a 10), entre otras características [Juola 2007]. Debido a las características que presentan los escenarios de aplicación, se requiere de métodos para la atribución automática de autoría que permitan obtener un modelo confiable sobre el estilo de un autor en escenarios como los descritos previamente.

1.2. Propuesta de solución

Desde un enfoque de aprendizaje automático, la tarea de atribución de autoría se considera como un problema de clasificación multiclase en el que se cuenta con un conjunto de autores candidatos y se requiere asociar un texto de autoría desconocida con alguno de ellos. Utilizando técnicas de aprendizaje automático supervisado se implementan clasificadores que son entrenados con la representación vectorial del conjunto de entrenamiento y posteriormente se les pide que clasifiquen un conjunto de textos anónimos con alguno de los autores candidatos.

En la presente tesis se propone un método para la atribución de autoría de textos cerrada (detección de plagio intrínseco) basado en un enfoque de aprendizaje automático en el que se considera a la tarea de atribución de autoría como un problema de clasificación multiclase: dónde se cuenta con un conjunto de entrenamiento conformado por muestras de textos de un

conjunto de autores (autores candidatos) y se busca atribuir la autoría un texto con alguno de los autores candidatos. Se proponen dos estrategias para el modelado de la información: representación distribuida a nivel de documento (doc2vec) y n-gramas sintácticos obtenidos del árbol de dependencias.

La representación distribuida a nivel de documentos (doc2vec) obtiene una representación analizando el contexto de las palabras, es decir, identifica las características de las palabras vecinas a una palabra y las codifica en un vector de características; se utilizará una implementación disponible libremente llamada GENSIM [1]. El método de representación distribuida explota la información sintáctica contenida en el texto busca caracterizar las palabras que lo componen en términos de las palabras del contexto, es decir, las palabras que se encuentran antes y después de cierta palabra en el texto.

Los algoritmos que obtienen una representación distribuida de un texto trabajan a nivel de palabras respetando el orden en que aparecen. En esta tesis se propone utilizar una representación en términos de n-gramas de palabras tradicionales con la finalidad de identificar patrones en el contexto de ciertas combinaciones de palabras y se espera que la información capturada por los n-gramas de palabras ayude a mejorar la eficiencia obtenida solamente con palabras.

Por su parte, los n-gramas sintácticos codifican subárboles del árbol de dependencias de las oraciones y capturan las relaciones sintácticas existentes entre los elementos del subárbol, permitiendo realizar comparaciones sobre su uso entre los diferentes autores. Se propone este marcador de estilo porque la información sintáctica provee un conjunto de características robustas que no dependen del tema sobre el que se hable en el texto.

Además, en esta tesis se describe un algoritmo para la extracción de los n-gramas sintácticos que consideren la variante de n-gramas homogéneos (todos los elementos constituyentes del n-grama son de la misma naturaleza) y n-gramas sintácticos heterogéneos (los elementos constituyentes del n-grama son de distinta naturaleza).

En términos generales, el método que se propone comienza realizando un preprocesamiento estándar (eliminación de elementos innecesarios, segmentación del texto en párrafos y oraciones) de todos los textos que conforman el corpus. Para obtener una representación distribuida se crean archivos que reescriban la información contenida en los textos del corpus que se analiza en términos de n-gramas tradicionales de palabras, específicamente se analizan

[1] Véase `https://radimrehurek.com/gensim/`

los tamaños de 2 hasta 4 que son los que mejores resultados han reportado dentro del estado del arte. El método doc2vec es alimentado con las representaciones en términos de palabras y de n-gramas de palabras, el método obtiene un modelo sobre el estilo de los autores por cada tipo de representación identificando las palabras o n-gramas de palabras y sus respectivos contextos. Una vez que se tienen los modelos, se obtienen una representación vectorial de cada texto del corpus obtenido un vector usando cada modelo y finalmente concatenando los vectores.

En caso de la representación en términos de n-gramas sintácticos, los textos son procesados por un analizador sintáctico para obtener los árboles de dependencia de cada oración. Los árboles de dependencia alimentan un algoritmo que extrae subárboles codificados en forma de n-gramas así como su frecuencia de ocurrencia en cada texto; el algoritmo obtiene dos versiones de n-gramas sintácticos: n-gramas con un solo tipo de información sintáctica (n-gramas homogéneos) y n-gramas que combinan distinta información sintáctica (n-gramas mixtos). Se construye un modelo sobre el estilo de los autores utilizando una representación vectorial en el que las dimensiones corresponden a los n-gramas sintácticos

De manera general, se utiliza una representación vectorial para cada documento en la que cada entrada representa una característica cuyo sentido depende de la estrategia utilizada: frecuencia de ocurrencia de un n-grama sintáctico o probabilidad de ocurrencia de una palabra dado un contexto. Después de la obtención de las representaciones de los textos del corpus, un clasificador es entrenado con la información de entrenamiento y posteriormente se le solicita que asigne un autor a cada uno de los textos anónimos.

Se evaluará el desempeño de las dos estrategias propuestas para el modelado de la información utilizando los siguientes corpus compilados para la tarea de atribución de autoría: PAN 2012 [Juola 2012], *The Reuters Corpus Volume 1* (RCV1) [Lewis et al. 2004] y *The Guardia* [Sapkota et al. 2015]. Cada uno de los corpus propuestos ofrece diferentes escenarios en los que existe diversidad en el género de los textos utilizados (noticias, novelas, reseñas de libros), diversidad temática y distintas configuraciones sobre el número autores candidatos.

1.3. Objetivo general

Desarrollar un método para la detección de plagio intrínseco orientado a la atribución de autoría cerrada utilizando un enfoque de aprendizaje automático supervisado, en el que se exploran dos posibles estrategias para la representación: representación distribuida de textos (doc2vec) y n-gramas sintácticos.

1.4. Objetivos específicos

Las metas que se proponen en el trabajo son:

- Obtener una representación distribuida a nivel de documentos el corpus de prueba utilizando como entrada palabras y n-gramas de palabras.
- Buscar parámetros óptimos para la construcción de un modelo distribuido a nivel de documentos en los distintos corpus de prueba para la atribución de autoría.
- Diseñar e implementar un algoritmo para obtener n-gramas sintácticos (n-gramas homogéneos y n-gramas mixtos).
- Cuantificar la presencia de todas las variantes de n-gramas sintácticos en un texto.
- Obtener un modelo sobre el estilo en términos de los n-gramas sintácticos.
- Obtener una representación vectorial de los corpus de prueba para usando el modelo generado para cada una de las posibles estrategias de representación (n-gramas sintácticos y representación distribuida).
- Entrenar clasificadores usando la representación vectorial de los textos de entrenamiento de los diferentes corpus de prueba.
- Evaluar la eficiencia de las representaciones (n-gramas sintácticos y representación distribuida) para resolver la tarea de tarea de atribución de autoría.

1.5. Aportaciones científicas esperadas

Las aportaciones científicas esperadas de este trabajo son:

- Proponer un método para la atribución de autoría que utilice una representación distribuida a nivel documentos.
- Mostrar la eficiencia que tiene una representación distribuida a nivel de documentos para modelar el estilo de un autor.
- Mostrar el impacto que tienen los formatos de entrada de palabras y n-gramas de palabras en una representación distribuida para modelar el estilo de un autor.
- Diseñar e implementar de un algoritmo para la obtención de n-gramas sintácticos (homogéneos y mixtos).
- Proponer un método para la atribución de autoría que utilice como marcadores de estilo n-gramas sintácticos (homogéneos y mixtos).
- Mostrar la eficiencia que el uso de n-gramas sintácticos tiene para modelar el estilo de un autor.
- Compilar un corpus sobre la atribución de autoría con información sintáctica.

Capítulo 2

Marco teórico

2.1. Detección de plagio de textos

Una de las áreas de investigación que ha llamado la atención de la comunidad científica recientemente ha sido la detección automática de plagio en textos o simplemente detección de plagio. El plagio en textos no es una problemática nueva, sin embargo, su incidencia en textos ha incrementado considerablemente en los últimos años en la Web.

La gran cantidad de información disponible y la falta de mecanismos de control para el manejo de la información, son algunas de las causas que han favorecido su crecimiento y han motivado el desarrollo de técnicas para su detección [Barrón-Cedeno et al. 2010].

Las técnicas desarrolladas para la detección de plagio han sido utilizadas en ámbitos escolares y científicos, por ejemplo, algunas universidades utilizan software especializado para detectar plagio entre los trabajos realizados por los alumnos, de igual manera, revistas científicas utilizan este tipo de software para detectar plagio entre los artículos enviados por los investigadores [Collberg & Kobourov 2005]. La lingüística forense es otra área, que aplica técnicas para la detección de plagio en la resolución casos de propiedad intelectual.

De manera general se define la acción de plagiar como [1]: "*Copiar en lo sustancial obras ajenas, dándolas como propias*". Dentro del PLN la detección de plagio de textos se define como [Alzahrani et al. 2012]:

[1] Véase *Diccionario de la Real Academia de la Lengua Española, Febrero 2016,* `http://dle.rae.es/?id=TIZy4Xb`.

> "*el uso de ideas o trabajo creativo, parcial o total, de alguien más expresado en textos de distintas autorías, sin el reconocimiento al autor original*".

El trabajo creativo de un texto siempre se atribuye a sus autores, sin embargo, un texto puede contener el trabajo de otros autores solo si se expresa adecuadamente el reconocimiento a éstos mediante el uso de citas bibliográficas o referencias. El plagio de textos consiste en la falsa atribución de autoría del trabajo creativo expresado en texto ante terceros.

El trabajo creativo de un texto se evalúa en términos de los siguientes elementos [Uzuner et al. 2005]:

- el contenido que hace referencia a las ideas o la información que el autor quiere compartir.
- la expresión que hace referencia a la forma en que el autor expresa el contenido utilizando el lenguaje, por ejemplo, la estructura del texto, el léxico utilizado, el uso de voz pasiva o activa, el uso de oraciones complejas o simples, entre otros.

En la literatura se han propuesto varias taxonomías sobre el plagio, cada una de ellas le asigna una categoría al plagio, considerando la forma en la que éste se lleva a cabo. La copia literal de bloques de texto, la copia de la estructura de un texto y el parafraseo de la información son algunos ejemplos de plagio, siendo este último él más difícil de detectar.

En el año 2010 la IEEE propuso una taxonomía para el plagio que abarca cinco niveles que ordena el plagio de texto comenzando con el nivel más ofensivo y terminando con el nivel menos ofensivo [HaCohen-Kerner et al. 2010]:

1. Nivel uno: Copia literal del texto completo o de la mayor parte de él (más de la mitad del texto original) sin el reconocimiento adecuado al autor original.
2. Nivel dos: Copia literal de una parte considerable del texto (menos de la mitad del texto original) sin el reconocimiento adecuado al autor original.
3. Nivel tres: Copia literal de elementos individuales (párrafos, oraciones, figuras, entre otros) sin el reconocimiento adecuado al autor original.
4. Nivel cuatro: Parafraseo de páginas o párrafos sin el reconocimiento adecuado al autor original.

5. Nivel cinco: Copia literal de una porción del texto con reconocimiento al autor original pero sin delineadores bien establecidos (sangría, comillas o cita bibliográfica).

En el trabajo presentado por Alzahrani et al. [2012] se propone una taxonomía del plagio en la que se incluye una mayor cantidad casos. La taxonomía divide el plagio de textos en dos tipos tomando en cuenta la conducta del plagiador:

- Plagio literal: en este caso el contenido y la forma tienden a mantenerse. Dentro de esta categoría tenemos los siguientes casos:
  - Copia exacta: copia literal, parcial o completa, del documento.
  - Copia casi exacta: copia casi literal de un fragmento del texto que se obtiene realizando pequeñas modificaciones a nivel de palabras (inserción, eliminación y sustitución)
  - Copia modificada: copia con cambios en la sintaxis respecto al original.
- Plagio inteligente: en este caso el plagiador intenta ocultar el delito manteniendo la información y modificando la forma en que se expresa. Algunas técnicas utilizadas incluyen:
  - Manipulación de texto: comprende acciones como resumir y parafrasear texto.
  - Traducción: consiste en plagiar texto mediante la traducción del texto de su idioma original hacia algún otro.
  - Adopción de idea: consiste en el plagio de la forma en que se presenta la información y de otros elementos como figuras, tablas, diagramas, etc.

La detección automática de plagio de textos consiste en desarrollar métodos computacionales para identificar las secciones de un texto que presenten algún tipo de plagio, basándose en un análisis del trabajo creativo del texto (contenido y forma) en el que se utilizan técnicas de áreas como el Procesamiento de Lenguaje Natural (PLN), Minería de Datos (MD), Recuperación de Información (RI), entre otras.

Las metodologías desarrolladas para la detección automática de plagio se pueden clasificar, de manera general, en dos enfoques: extrínseco e intrínseco. El enfoque extrínseco detecta las posibles secciones que presenten algún nivel de plagio mediante una comparación directa entre las características de los elementos del texto sospechoso y las de un conjunto de textos fuente [Alzahrani et al. 2012]. El resultado que se obtiene es un conjunto de relaciones entre los elementos del texto en duda y los elementos de los textos fuente que indican la existencia de cierto nivel de plagio.

El enfoque extrínseco ha sido ampliamente utilizado para detectar plagio literal, algunos ejemplos de trabajos relacionados los podemos encontrar en [Kang et al. 2006, Bhavani et al. 2009, Sánchez-Vega et al. 2010]. Una variante del enfoque extrínseco es la detección de plagio a través de diferentes idiomas (en inglés, *cross-lingual plagiarism*), en la que el plagio se comete al usar indebidamente el trabajo creativo de alguien más, a pesar de que éste se exprese en un idioma diferente al utilizado originalmente, en esta variante de plagio se realiza un proceso de traducción. Para la detección de plagio a través de diferentes idiomas se realiza un análisis similar al que se lleva a cabo en el enfoque externo pero en este enfoque se requiere de una etapa de traducción o alineación de textos. Algunos ejemplos de trabajos que se han desarrollado para la detección de plagio en diferentes idiomas se presentan en [Barrón-Cedeño et al. 2010, Pereira et al. 2010, Potthast, Barrón-Cedeño, Stein & Rosso 2011].

El enfoque intrínseco identifica las posibles secciones de un texto que presenten algún nivel de plagio, mediante un análisis del texto se busca identificar cambios no justificados en el estilo del autor. A diferencia del enfoque extrínseco, en el enfoque intrínseco no se requiere de manera explícita el conjunto de documentos fuente. En la actualidad, la búsqueda de información relacionada con un tema en particular se vuelve cada vez más complicado debido al volumen de información disponible a través de Web, por lo que el enfoque intrínseco se vuelve una alternativa más favorable.

Se han presentado diversos trabajos relacionados con la detección de plagio aplicando un enfoque intrínseco, de manera general un trabajo que utiliza este enfoque se compone de las siguientes etapas [Alzahrani et al. 2012, Potthast, Eiselt, Barrón-Cedeño, Stein & Rosso 2011]:

- **Segmentación**. Esta etapa consiste en dividir el texto a analizar en unidades de interés. Algunas unidades propuestas coinciden con la división natural de un texto, por ejemplo, párrafos u oraciones, en otros casos se divide al texto en bloques con cierta longitud medida en términos de palabras o caracteres.

- **Modelado de la información**. En esta etapa se busca una representación para el *estilo de escritura* contenido en las unidades que componen al texto; la representación del estilo, generalmente, se realiza usando la información lingüística contenida en el texto. Además, se debe proponer una función que permita establecer la similitud entre dos representaciones de distintas unidades.

- **Detección de plagio**. En esta etapa se identifican aquellas unidades del texto que sean considerablemente diferentes del resto, las cuales se consideraran como posibles fragmentos atribuibles a un diferente autor.

La etapa de modelado de la información es la más importante en el enfoque intrínseco, porque a partir de la representación que se proponga, se desarrollarán el resto de las etapas del método. Aunque parece algo natural pensar que cada persona posee un estilo de escritura único, el cómo modelar el estilo de escritura aún sigue siendo objeto de estudio, inclusive la idea misma sobre la existencia de un estilo de escritura único en cada persona ha sido causa de polémica entre la comunidad científica. En este escenario, los avances realizados en el área llamada Estilometría (en inglés, *Stylometry*), enfocada al estudio del estilo de escritura de las personas a partir de un análisis estadístico sobre el uso de elementos lingüísticos, han ayudado a consolidar la idea de que existe un estilo de escritura único para cada persona y que es posible cuantificarlo.

El problema de la detección de plagio intrínseca se puede considerar como una generalización del problema de atribución de autoría, con la diferencia de que no se tiene un conjunto de textos para construir el perfil del estilo del autor y que trabaja con secciones de un texto en lugar de textos completos [Stein et al. 2011].

2.2. Análisis del estilo de un autor

Estudios recientes en el campo de la genética y de la lingüística han aportado resultados que apoyan la teoría de que la capacidad para utilizar el lenguaje es una característica propia de los seres humanos que se encuentra codificada en la genética de éstos [Holmes 1998].

Todo ser humano es capaz de utilizar el lenguaje y la forma en que éste es utilizado por un individuo depende de varios factores como la idea que desea expresar, la emoción que desea transmitir, los conocimientos previos que tiene sobre el tema, el tipo de público a quién se dirige, el medio que utiliza para expresarse, entre otros. En general la forma en que una persona utiliza el lenguaje depende tanto de rasgos propios de su personalidad como de características del medio que lo rodea y de la situación en específico en la que se encuentre [Uzuner et al. 2005].

Los rasgos de la personalidad de un autor influyen de manera inconsciente en las elecciones que éste hace sobre el lenguaje para componer un texto y se espera que estas elecciones aparezcan de manera repetitiva a lo largo de diferentes textos del autor dado que no se realizan de manera consciente. A este patrón de elecciones, sobre el uso del lenguaje que aparecen de manera frecuente en los textos de un autor, se le conoce como estilo de escritura del autor o estilo del autor.

Investigadores consideran que todo ser humano tiene un estilo de escritura que se considera único, el cuál tiende a mantenerse constante a través del tiempo y es independiente del tema sobre el que se habla en el texto. Debido a las características que presupone el estilo de escritura de un autor, los avances realizados en el análisis del estilo son utilizados con frecuencia para resolver diferentes áreas dentro del Procesamiento del Lenguaje Natural (PNL) como son atribución de autoría [Stamatatos 2009*b*, Sidorov et al. 2014], detección de plagio [Stein et al. 2011, Sanchez-Perez et al. 2014], obtención de perfiles de usuario [Estival et al. 2007, Rangel et al. 2015], investigaciones forenses [Solan 2013, Koppel et al. 2013], entre otras .

Un texto se puede considerar como una secuencia de elecciones hechas por su autor sobre los elementos y las reglas del lenguaje escrito. Los rasgos de la personalidad de un autor influyen de manera inconsciente en las elecciones que éste hace al componer un texto y se espera que estas elecciones aparezcan repetidamente a lo largo de diferentes textos del autor, a este patrón de elecciones sobre el uso del lenguaje escrito se le conoce como estilo de escritura del autor o simplemente estilo del autor [Uzuner et al. 2005].

El análisis del estilo de escritura no es una tarea fácil debido a la complejidad implícita del lenguaje natural. Una línea de investigación enfocada en este tipo de análisis es la conocida como Estilometría (en inglés, *Stylometry*), en la que se busca definir características sobre el lenguaje escrito que permitan cuantificar el estilo de un autor.

Algunas de las características que se han propuesto en esta línea de investigación son [Stamatatos 2009*b*]: tamaño de las oraciones, tamaño de las palabras, frecuencia de las palabras, frecuencia de caracteres, riqueza del vocabulario utilizado, entre otros.

El lenguaje escrito se compone de un conjunto de elementos y de reglas que rigen la forma en que se asocian estos elementos para expresar ideas. Una peculiaridad que presenta el lenguaje escrito es la de ofrecer más de una alternativa para expresar una idea en particular, utilizando diferentes elementos y aplicando diferentes reglas.

Los elementos y las reglas del lenguaje escrito, utilizadas en la composición de un texto son elecciones realizadas por el autor. Desde este enfoque, podemos considerar a una sección de un texto como una secuencia de elecciones realizadas por el autor, sobre los elementos del lenguaje escrito para expresar una idea, que pueden tener o no, una presencia repetitiva en el resto del texto [Alzahrani et al. 2012].

El análisis del estilo de un autor consiste en proponer un conjunto de rasgos sobre el lenguaje escrito, llamados marcadores de estilo (en inglés, *style markers* o *fingerprints*), que sean representativos y robustos, para que al ser cuantificados permitan identificar un texto escrito por el autor.

Un marcador de estilo busca hacer referencia a un rasgo de la forma en que compone un autor, tomando como eje el conocimiento sobre el lenguaje escrito. El estudio del lenguaje escrito implica el análisis de los elementos que lo componen y de las reglas de su uso, desde un nivel básico (estudio de los caracteres) hasta un nivel más complejo (estudio del impacto comunicativo del texto en el lector); típicamente su estudio se divide en los siguientes niveles: lexicográfico, morfológico, sintáctico, semántico, pragmático y discursivo [Stamatatos 2009*b*].

El lenguaje escrito ofrece un conjunto de características (morfológicas, léxicas, sintácticas, semánticas) que son mutuamente independientes y por lo tanto es posible definir diferentes marcadores de estilo, cada uno de ellos capaz de capturar información que otros niveles no pueden [Uzuner et al. 2005].

Debido a la gama de posibilidades para definir marcadores de estilo, el reto es encontrar aquellos marcadores de estilo para la caracterización del estilo de un autor que presenten el mejor desempeño en la tarea que se desea resolver.

Se han propuesto diferentes taxonomías que agrupan a los marcadores de estilo considerando el tipo de característica que analizan y los recursos lingüísticos que necesitan para su aplicación. En los trabajos [Stamatatos 2009*b*, Juola 2007] se propone la siguiente clasificación para marcadores de estilo: nivel de caracteres, nivel léxico, nivel sintáctico, nivel semántico y nivel de formato.

En el nivel de caracteres los marcadores de estilo analizan el uso de los distinto caracteres que conforman el texto como son letras, números y signos de puntuación, también incluye el análisis de subunidades léxicas como los morfemas. Algunos ejemplos de marcadores de estilo propuestos son el número de caracteres en mayúscula y minúscula, frecuencia de uso de las letras, número de signos de puntuación, frecuencia de uso de n-gramas a nivel de caracteres, entre otros [Stamatatos 2009*b*, Juola 2007].

Los marcadores de estilo en esta categoría tienen la ventaja de que son fáciles de representar desde el punto de vista computacional, no requieren recursos lingüísticos especiales y son independientes del idioma; sin embargo, su principal desventaja es que el tamaño de su representación tiende a ser grande y no son lo suficientemente robustos.

En el nivel léxico los marcadores de estilo analizan las palabras y su uso en el texto. Algunos ejemplos de marcadores de estilo propuestos son el tamaño de las palabras, el tamaño de las oraciones medido en palabras, razón palabra-lema (*type-token ration*), número de *hapax legonema*, frecuencia de uso de palabras específicas, entre otros [Stamatatos 2009*b*].

A diferencia del análisis a nivel de caracteres, el análisis a nivel de las palabras permite identificar rasgos más afines con el estilo del autor, por ejemplo, la diversidad de su vocabulario o las palabras que más utiliza al componer un texto y su representación es sencilla desde el punto de vista computacional. De los inconvenientes que presentan este tipo de marcadores de estilo son que requieren de herramientas lingüísticas (analizador morfológico) para ser cuantificados, dependen del idioma utilizado en el texto y en algunos casos existe dependencia con el tema sobre el que se habla en el texto.

En el nivel sintáctico se analizan patrones sobre el uso de las reglas que rigen sobre los elementos del lenguaje escrito para la composición de un texto. Algunos ejemplos de marcadores de estilo propuestos son la frecuencia de uso de reglas sintácticas, análisis de árboles sintácticos generados por el texto, frecuencia de uso de n-gramas a nivel de palabras, frecuencia de uso de los tipos de palabras (*Part of Speech*, POS), entre otros [Stamatatos 2009*b*]. Los n-gramas sintácticos propuestos en [Sidorov 2013*b*, Sidorov et al. 2012*a*] son de especial interés porque explotan las relaciones de dependencia entre las palabras que componen el texto y ofrecen una manera de representar las relaciones no lineales de dependencia contenidas en el árbol sintáctico bajo el concepto de n-grama.

Los marcadores propuestos en este nivel se consideran más confiables para construir un estilo de escritura de un autor, debido a que el uso de las reglas sintácticas se tiende a realizar de manera inconsciente y no depende del tema que se aborde en el texto. Sin embargo, el uso de este tipo de marcadores de estilo requiere de herramientas lingüísticas robustas que permitan obtener información sintáctica sobre un texto y en algunos casos, se requiere de una representación compleja para poder manipular la información sintáctica.

En el nivel semántico los marcadores de estilo se enfocan en el estudio de las relaciones semánticas entre las palabras utilizadas en el texto e identifican patrones que permitan caracterizar el estilo del autor, por ejemplo, se analizan el uso de sinónimos, la cantidad de temas que se abordan en el texto, por mencionar algunos. El principal inconveniente que presenta este tipo de marcadores es la falta de herramientas que realicen un análisis semántico

eficiente, en consecuencia el número de trabajos que utilizan este tipo de marcadores es limitado y los resultados reportados no muestran una mejoría sustancial. Sin embargo, este tipo de marcadores ha sido utilizado con éxito en la tarea de detección de plagio extrínseco al permitir detectar cambios en el tema entre párrafos de un textočitepotthast09.

En el nivel de formato los marcadores de estilo se enfocan en identificar patrones sobre aquellos elementos relacionados con la presentación del texto. Algunos ejemplos de marcadores de estilo propuestos son: tipo de atributos utilizados en el formato del texto (por ejemplo, palabras en negrita, palabras subrayadas y palabras en cursiva), tipo de encabezados utilizados (por ejemplo, para iniciar o terminar una carta o un discurso), tipo de estructura utilizada en el texto (por ejemplo, número de secciones y subsecciones en la que se organiza el documento o nivel de profundidad utilizado para organizar el documento), entre otras [Stamatatos 2009*b*].

Los marcadores de estilo que se basan en el análisis del formato dependen completamente del tipo de texto que se compone, por lo que no requieren de herramienta lingüística alguna, sin embargo, su aplicación se encuentra limitada a casos específicos. Un caso en el que este tipo de marcadores ha mostrado un buen desempeño es en la atribución de la autoría de correos electrónicos [Koppel & Schler 2003], blogs [Koppel et al. 2006] y mensajes vía redes sociales [Layton et al. 2010].

Las categorías de marcadores de estilo descritas previamente analizan varias de las características del lenguaje escrito, no obstante existen propiedades a nivel pragmático y discursivo que no han sido exploradas como lo han sido otros niveles (morfológico, léxico y sintáctico). La razón por la que no han sido exploradas se debe a que las herramientas necesarias para su análisis son complejas desde un punto de vista computacional lo que implica una limitante para su aplicación, no obstante existen propuestas de marcadores de estilo que se enfocan en estos niveles [Stamatatos 2009*b*].

Otro aspecto importante además del tipo de marcadores de estilo que se pueden utilizar para modelar el estilo de un autor es la cantidad de información necesaria para poder obtener un modelo confiable. Sería deseable poder obtener modelos confiables al analizar unidades pequeñas de texto como las oraciones lo que permitiría resolver tareas en escenarios en los que se cuenta con pocas cantidades de texto (por ejemplo tuits), sin embargo, trabajar con poca información no es confiable desde un punto de vista estadístico. Investigaciones

realizadas para determinar el tamaño mínimo sobre el que permita obtener un modelo confiable sobre el estilo del autor, se reporta que secciones con un tamaño entre 200 y 500 caracteres proveen la suficiente información para caracterizar el estilo de quién escribió dichas secciones [Alzahrani et al. 2012, Holmes & Kardos 2003].

2.3. Atribución de autoría

La atribución de autoría consiste en identificar, a partir de un conjunto de posibles autores, al autor o autores de un texto en particular mediante el análisis de ejemplares de textos escritos por los posibles autores.

Los primeros trabajos se enfocaban en identificar rasgos o características significativas presentes en el escrito que pudieran vincularse de manera inequívoca a alguno de los posibles autores. La mayoría de las características utilizadas estaban relacionadas con aspectos sobre la presentación del escrito, como por ejemplo, tipo de papel utilizado, tipo de tinta utilizado, algún rasgo distintivo de la caligrafía en el caso de textos escritos a mano o algún error tipográfico en el casos de textos impresos, errores ortográficos repetitivos, tipo de formato utilizado, entre otros.

Con la llegada de las nuevas tecnologías, cambió la forma en que se escribían los textos, originando nuevos retos a los que se tenía que enfrentar la *atribución de la autoría de textos*. En esencia, estos nuevos retos van dirigidos en dos sentidos:

1. El uso de editores de texto ha favorecido la composición de textos de manera digital, sin embargo, ha provocado la desaparición de algunos rasgos del texto utilizados para realizar la atribución de la autoría (por ejemplo el tipo de papel utilizado, características de la caligrafía, entre otros) o que dejarán de ser representativos (por ejemplo, el uso de correctores ortográficos disminuye los posibles errores cometidos por el escritor o el uso de plantillas prediseñadas que homogeneizan el formato de los textos).

2. La creciente cantidad de documentos publicados alrededor del mundo en la Web, así como las facilidades para acceder a éstos mediante las tecnologías de la comunicación, ha dificultado la selección de posibles autores para un texto cuya autoría se encuentra en duda.

En un problema de atribución de autoría se maneja gran cantidad de información, por otro lado, el crecimiento de la Web y la falta de mecanismos que regulen su contenido ha provocado que el número de casos de atribución de autoría por resolver crezca rápidamente. Para poder atacar de manera eficiente los casos de atribución de autoría conviene no solamente pensar en herramientas computacionales que asistan en la solución del problema sino utilizar un enfoque computacional que permita automatizar la solución de este problema.

Ante este escenario, se ha propuesto el uso de un nuevo enfoque para la atribución de la autoría de textos, basado en el uso de modelos estadísticos y técnicas computacionales, conocido como atribución de autoría de textos asistida por computadora o atribución automática de textos. En este enfoque se busca desarrollar métodos que permitan identificar a un autor mediante un análisis estadístico sobre la forma en que utiliza el lenguaje y que realicen la atribución de la autoría de un texto, a partir de la comparación entre el análisis estadísticos del texto de interés y el de los ejemplares de los posibles autores.

El primer trabajo en el área de atribución automática de autoría fue realizado por [Mosteller & Wallace 1964]; en este trabajo se buscó determinar la autoría de un conjunto de textos históricos (*The Federalist Papers*) cuya autoría se encontraba en disputa entre tres posibles autores; este primer trabajo tenía características muy peculiares, por ejemplo, se contaba con un conjunto pequeño y bien definido de posibles autores, con un corpus considerable de ejemplos de textos escritos por los posibles autores, la extensión del texto de interés también era considerable y todos los textos eran homogéneos (del mismo tipo y hablaban del mismo tema) [Stamatatos 2009*b*]. El método propuesto demostró un buen desempeño, a decir de especialistas en lingüística, sin embargo, por las características del corpus utilizado se dejó entre ver que existían toda una serie de posibilidades por analizar.

La atribución de la autoría de textos tiene aplicación en diversos ámbitos, especialmente en aquellos relacionados con cuestiones legales, por ejemplo, en disputas por la autoría de textos publicados con anónimos o seudónimos, atribución de notas suicidas o intimidatorias, detección de plagio de textos, falsificación de identidad, entre otros. La atribución de autoría también ha sido utilizada para resolver problemas en ámbitos como el escolar, el científico y en las humanidades.

Resolver un problema de atribución de autoría, típicamente requiere cuantificar el estilo de cada uno de los posibles autores y obtener el estilo presente en el texto en disputa; para cuantificar el estilo de un autor requiere realizar diferentes cálculos sobre los ejemplares de textos escritos por éste. Además se requiere realizar la atribución de la autoría del texto en disputa comparando la información obtenida de los estilos cuantificados de los posibles autores.

El proceso de atribución de autoría, en un sentido amplio, consiste en [Bozkurt et al. 2007]:

1. Selección de posibles autores del texto en disputa y recolección de ejemplares de textos de su autoría.

2. Cuantificación del estilo de escritura de todos los posibles autores y del estilo plasmado en el texto en disputa.

3. Asociación del texto en disputa con alguno de los autores candidatos.

El primer paso en el proceso de atribución de autoría implica una ardua labor de búsqueda, en la que se debe lidiar con las siguientes cuestiones: se requiere definir criterios confiables sobre los cuáles se realiza la selección de los posibles autores; se requiere recopilar ejemplos de textos escritos por cada uno de los autores, previendo los casos en los que no se encuentren ejemplos disponibles en la *Web*; se deben organizar los ejemplos encontrados tomando en cuenta información como la fecha, el tipo de texto, el tema sobre el que se habla, el idioma utilizado, entre otros. Se han realizado algunos trabajos que buscan automatizar esta tarea, sin embargo, aún no se ha logrado automatizar completamente, en su lugar este se suele trabajar en casos específicos sobre los que se aplican heurísticas para realizar la búsqueda.

En vista de la complejidad del primer paso, típicamente en un problema de atribución de autoría se asume que este paso ya se ha realizado, entonces, en un problema de atribución de autoría se cuenta con un conjunto de ejemplares de textos de cada uno de los posibles autores del texto en disputa llamado corpus de entrenamiento y con el texto cuya autoría se encuentra en duda llamado corpus de prueba.

El segundo paso hace referencia al estilo de escritura de un autor y la forma de cuantificar éste estilo utilizando principalmente las propiedades del lenguaje escrito; en la sección anterior se ha comentado sobre los marcadores de estilo utilizados, el tipo de herramientas lingüísticas que requieren para ser cuantificados, las ventajas y desventajas generales que ofrecen según su tipo. La experiencia en este tema, indica que difícilmente un solo marcador de estilo podrá representar de manera eficiente el estilo de escritura un autor, lo que conlleva a utilizar más de un marcador de estilo en la práctica.

En el tercer paso se busca realizar la asociación de la autoría del texto en duda, comparando la información recabado del paso anterior sobre el estilo de escritura de los posibles autores y el presente en el texto en duda. Para ello se utilizan técnicas computacionales de áreas como Reconocimiento de Patrones, Minería de Datos y Recuperación de Información, que realizan de manera automática éste paso; el éxito de una técnica depende de las características del corpus de entrenamiento y de las características de los marcadores de estilo que serán utilizados.

Distintas áreas de la computación participan en el desarrollo de métodos destinados a la atribución de autoría como: Recuperación de Información (*Information Retrieval, IR*), Reconocimiento de Patrones (*Machine Learning, ML*) y Procesamiento de Lenguaje Natural (*Natural Language Processing, NLP*).

Desde un enfoque computacional la atribución de autoría se puede plantear como un problema de clasificación multiclase en el que cada instancia pertenece a una sola clase. En este caso cada uno de los estilos de escritura de los posibles autores representa una clase y el objetivo es asociar el estilo de escritura presente en el texto de interés con el estilo de uno de los posibles autores, que sea el más similar de acuerdo con alguna estrategia de clasificación.

Las diferentes estrategias de clasificación que se han empleado en problemas de atribución de autoría pueden clasificarse a partir de las diferentes características que presenta su metodología, por ejemplo, la forma en la que representa la información, la forma en la que asigna la clase, su tolerancia al ruido, entre otros. Una forma de clasificarlos es a partir de la manera en cómo se trata el corpus de entrenamiento [Stamatatos 2009*b*]: tratar cada texto del corpus de prueba de manera individual o de manera acumulativa por autor.

El enfoque basado en el perfil se unen todos los ejemplos de textos escritos por un autor en un solo texto acumulativo y se obtiene el estilo del autor o perfil del autor, posteriormente se realiza la asociación de la autoría utilizando un función de distancia que calcula la diferencia entre los estilos de escritura del perfil de un autor y del texto de interés, de tal forma que el texto se asocie con el autor que tenga la menor diferencia.

Se han propuesto diversas funciones de distancia para el enfoque basado en perfil, por ejemplo, algunas se basan en el uso de modelos de probabilidad [Mosteller & Wallace 1964], otras se basan en el uso técnicas de compresión de información [Kukushkina et al. 2001] y algunas más se basan en el uso de n-gramas de palabras o caracteres [Kešelj et al. 2003*a*].

Por otro lado, el enfoque basado en ejemplos obtiene por cada muestra de texto de un autor un estilo de escritura que son utilizados de manera independiente en la construcción de un modelo del estilo del autor, el cuál será utilizado para realizar la asociación de la autoría del texto de interés; las metodologías basadas en este enfoque constan de dos etapas, la etapa de entrenamiento en donde se construye un modelo del estilo de escritura de cada autor, requiere de la mayor cantidad de ejemplos de textos para obtener un modelo confiable y la etapa de asociación donde se utiliza una estrategia para asignar un texto de autoría en duda con alguno de los posibles autores.

Se han realizado varios trabajos bajo este enfoque que utilizan diferentes modelos para realizar la tarea de clasificación, uno de los más utilizados es el de Espacios Vectoriales en el que se usa un vector con diferentes marcadores de estilo para representar el estilo de un autor y realiza la asociación del texto utilizando algoritmos como Máquinas de Soporte Vectorial [Diederich et al. 2003, de Vel et al. 2001], árboles de decisión [Zhao et al. 2006], redes neuronales [Matthews & Merriam 1993*a*, Tweedie et al. 1996] y algoritmos genéticos [Stamatatos 2006, Oakes 2004].

Para evaluar el desempeño de un método de atribución de autoría, el corpus que se utiliza para realizar las pruebas (corpus de evaluación), juega un papel importante. Se sugiere tomar en cuenta las siguientes características cuando se evalúa un método de atribución de autoría [Stamatatos 2009*b*]:

- Tamaño del corpus de entrenamiento en términos del número de ejemplares y de la extensión de cada ejemplar.
- Tamaño del corpus de prueba.

- Número de posibles autores.
- Distribución del corpus de entrenamiento entre los autores (balanceado o no balanceado).
- Número de tipos de documentos o géneros utilizados.

Varios tipos de corpus han sido utilizados en los trabajos del área, por ejemplo, los textos literarios han sido ampliamente utilizados [Uzuner et al. 2005] debido a que este tipo de corpus ofrece textos con una extensión amplia y con un formato sencillo (básicamente compuesto por texto plano); otros corpus utilizados han sido utilizados textos de dominio específico como son artículos periodísticos [Stamatatos et al. 2001, Diederich et al. 2003], mensajes publicados en foros [Abbasi & Chen 2005, Zheng et al. 2005], blogs [Koppel et al. 2006] y correos electrónicos [de Vel et al. 2001].

Se han realizado esfuerzos a nivel internacional por compilar un corpus que pueda utilizarse como un estándar para evaluar y comparar el desempeño de los diversos trabajos propuestos; en este caso, se busca que el estilo de escritura del autor sea el factor discriminante más importante entre los textos [Stamatatos 2009*b*], además de ofrecer diferentes escenarios para la atribución de autoría. Un ejemplo en este sentido, lo podemos encontrar en la Competencia de Atribución de Autoría organizada en el 2004, en donde, se recopiló un corpus de evaluación que incluía textos en diferentes idiomas y se prepararon escenarios con diferentes niveles de dificultad [Juola 2004]; otro ejemplo que se reporta en Chaski [2001], que comprende un corpus de cartas realizados por 92 personas sobre 10 temas específicos (por ejemplo, carta para la aseguradora, carta de perdón para su mejor amigo, entre otros); otro ejemplo se reporta en Baayen et al. [2002], que comprende un corpus de 72 textos realizados por 8 personas sobre temas específicos en tres géneros diferentes (por ejemplo, una historia ficticia sobre un asesinato, un ensayo sobre la unificación de Europa, entre otros).

2.4. Representación distribuida de texto

El método de representación distribuida presentado en los trabajos [Mikolov, Chen, Corrado & Dean 2013, Mikolov, Sutskever, Chen, Corrado & Dean 2013] sirve para modelar distintas estructuras de un texto, desde estructuras sencillas como palabras hasta estructuras más complejas como documentos completos. La representación distribuida se basa en la semántica de las estructuras de un texto analizando su contexto.

La representación distribuida de un documento (doc2vec) es una extensión de la representación distribuida a nivel de palabras. El objetivo de la representación distribuida a nivel de palabras es predecir la ocurrencia de una de ellas conociendo el contexto que la rodea dentro de un texto.

Para la construcción de la representación distribuida de palabras, cada una de ellas es asignada a un vector único representado por una columna en una matriz W y dada una secuencia de palabras w_1, w_2, w_3, ..., w_T del conjunto de entrenamiento se busca maximizar el promedio de la probabilidad logarítmica dada por:

$$\frac{1}{T}\sum_{t=k}^{T-k} log\ p(w_t|w_{t-k}, ..., w_{t+k})$$

La tarea de predecir el vector de una palabra dado el contexto se realiza utilizando el clasificador multiclase *softmax*, es decir,

$$p(w_t|w_{t-k}, ..., w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}},$$

donde cada y_i se obtiene para cada palabra i de acuerdo con la ecuación 2.1:

$$y\ =\ b + Uh(w_{t-k}, ..., w_{t+k}; W) \tag{2.1}$$

En la ecuación 2.1 U y b son parámetros del método *softmax*, mientras que h se obtiene de la concatenación de los vectores de palabras extraídos de W. Una vez que el algoritmo de entrenamiento termina, las palabras con significado similar (contextos similares) se asignan a una posición similar en el espacio vectorial [Mikolov, tau Yih & Zweig 2013].

La representación distribuida a nivel de documentos (doc2vec) efectúa los mismos pasos que la distribución a nivel de palabras, pero utilizando una representación intermedia a nivel de párrafos que se obtienen concatenando las representaciones vectoriales de las palabras. La información contenida en los vectores de los párrafos es utilizada en la tarea de predecir la ocurrencia de una palabra generando diferentes contextos con las palabras contenidas en el párrafo. Los vectores de palabras o de párrafos son inicializados aleatoriamente pero al final de la tarea de predicción de palabras, las representaciones vectoriales capturan la semántica de las palabras.

En el modelo de vectores distribuidos de párrafos, cada párrafo es descrito como un vector único que representa una columna en la matriz W. La ecuación 2.1 cambia para construir a h a partir de W y D.

Un párrafo puede ser considerado como otra palabra que actúa como una memoria que permite recordar lo que hace falta del contexto, a este modelo se le conoce como memoria distribuida (en inglés Distributed Memory DM). Una variante del modelo anterior se llama bolsa de palabras distribuida (Distributed Bag of Words DBOW) es aquella en la que se ignoran las palabras del contexto de la entrada y se predicen aleatoriamente palabras extraídas del párrafo.

2.5. Representación basada en el uso de n-gramas sintácticos

El concepto de n-grama hace referencia de una secuencia de n elementos que componen el texto manteniendo el orden en el cuál aparecen. El procedimiento para obtener n-gramas utiliza una ventana imaginaria tamaño n que se desplaza que se desplaza un elemento en cada iteración hasta alcanzar el final del texto; los elementos que se encuentran enmarcado en la ventana imaginaria durante una iteración corresponden a un n-grama. El concepto de n-grama ha sido utilizado en distintas tareas de diferentes áreas como el Procesamiento de Lenguaje Natural, la Visión por Computadora la Minería de Datos y el Reconocimiento de Patrones por mencionar algunos.

La frecuencia de ocurrencia de los n-gramas se puede utilizar como características para modelar un fenómeno en el texto. Un aspecto importante sobre los n-gramas es su capacidad para capturar el contexto que contienen un elemento, permitiendo modelar no solamente la ocurrencia de un elemento sino también la ocurrencia de sus posibles contextos.

Los n-gramas a nivel de palabras y a nivel de caracteres han sido utilizados en trabajos previos para resolver el problema de atribución de autoría [Stamatatos 2009*b*]. Una variante de n-gramas propuesto en [Sidorov 2013*b*, Sidorov 2013*a*, Posadas-Duran et al. 2014] son los n-gramas sintácticos (sn-gramas), los cuales se obtienen recorriendo los árboles de dependencia de las oraciones de un texto. Los n-gramas sintácticos codifican subárboles del árbol de dependencias de las oraciones y capturan las relaciones sintácticas existentes entre los distintos elementos de un subárbol.

Una diferencia importante con los n-gramas tradicionales es que los n-gramas sintácticos representan relaciones fundamentadas en la sintaxis de un idioma, mientras que los n-gramas tradicionales incluyen combinaciones sin fundamentos que pueden generar ruido en la representación.

En el capítulo 4 se explica a detalle el concepto de de n-gramas sintácticos, se presentan dos tipos de n-gramas sintácticos (homogéneos y mixtos) y se presenta un algoritmo para su obtención.

2.6. Evaluación de algoritmos de clasificación

Los algoritmos de aprendizaje automático que realizan clasificación se encuentran estrechamente vinculados con las características de la información utilizada para su entrenamiento; debido a que no en todos los casos existen bancos de datos estándar, además se requiere de estrategias que permitan evaluar el desempeño de los algoritmos, con la finalidad de conocer la tasa de error esperada que genera el algoritmo y poder realizar comparaciones entre distintos algoritmos.

Para conocer la tasa de error de un algoritmo de clasificación, se usa éste para generar diferentes clasificadores, cada uno generado a partir de modificaciones aleatorias sobre elementos iniciales (en los datos de entrenamiento, en los parámetros) y se prueban en conjuntos de datos diferentes a los de entrenamientos (datos de validación), llevando el registro del desempeño de cada clasificador (error de validación). Finalmente, se obtiene la tasa de error esperada a partir de la distribución de los errores de validación.

En la práctica no se cuenta con bancos de datos de gran extensión que permitan realizar una división de datos de entrenamiento y datos de validación, asegurando un tamaño significativo para cada uno de ellos; en general se trabaja con conjuntos de datos pequeños y se utiliza la técnica conocida como validación cruzada (en inglés, *k-fold cross-validation*) que trabaja de manera repetida con los mismos datos pero haciendo una partición distinta en cada ocasión.

La estrategia de validación cruzada divide de manera aleatoria un conjunto de datos inicial X en k partes de igual tamaño, $X_i, i = 1, \ldots, k$. Se generan pares de datos, datos de entrenamiento T y datos de evaluación V, manteniendo una de las k partes como datos de evaluación y las $k - 1$ partes restantes como datos de entrenamiento; repitiendo este procedimiento k veces, en cada vez, seleccionando un conjunto de entrenamiento diferente, se obtienen k pares de datos:

$$V_1 = X_1 \qquad T_1 = X_2 \cup X_3 \cup \ldots \cup X_k$$

$$V_2 = X_2 \qquad T_2 = X_1 \cup X_3 \cup \ldots \cup X_k$$

$$\vdots$$

$$V_k = X_k \qquad T_k = X_1 \cup X_2 \cup \ldots \cup X_{k-1}$$

Con esta técnica se debe lidiar con dos problemas: el tamaño de los datos de evaluación no debe ser muy pequeño y el traslape de los datos de entrenamiento. Para lidiar con estos problemas se propuso realizar divisiones de igual tamaño para los datos de entrenamiento y datos de validación, se repite este procedimiento en 5 ocasiones (5×2 *cross-validation*), este valor disminuye el traslape de los datos de entrenamiento [Wettschereck 1994].

Para analizar los errores cometidos por un algoritmo de clasificación se utiliza la llamada matriz de confusión. Si se tienen $k > 2$ clases, se tiene una matriz $k \times k$ en la que cada entrada (i, j) contiene el número de instancias que corresponden a la clase C_i pero fueron asignadas a la clase C_j; para el caso en que no ocurre ningún error de clasificación todas las entradas distintas a las que se encuentran en la diagonal deben de tener un valor de cero.

Para una clase en particular, se puede obtener su tasa de error definida como:

$$tasa\ error = \frac{FN + FP}{N}, \tag{2.2}$$

donde:

- FN representa los falsos negativos
- FP representa los falsos positivos
- N representa el número total de instancias en los datos de validación

La tasa de error de un clasificador es el promedio de las tasas de error de cada una de las clases existentes en los datos de validación.

La eficiencia (en inglés, *accuracy*) representa el porcentaje de instancias que son correctamente clasificadas y ésta se define como:

$$eficiencia = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2.3}$$

donde:

- TP representa los verdaderos positivos
- TN representa los verdaderos negativos
- FN representa los falsos negativos
- FP representa los falsos positivos
- N representa el número total de instancias en los datos de validación

En el capítulo 3 se describen brevemente algunos de los trabajos relacionados con las dos estrategias de represetación propuestas en esta tesis.

Capítulo 3

Estado del arte

Los n-gramas de caracteres y de palabras son los marcadores de estilo que mejor eficiencia han reportado para la tarea de atribución de autoría [Stamatatos et al. 2001, Keselj et al. 2003*b*] también para otra tareas relacionadas [Stamatatos 2013, Plakias & Stamatatos 2008, Escalante et al. 2011]. Sin embargo, su desventaja es que generan vectores muy dispersos con una alta dimensionalidad, además de que generan representaciones de texto sin ningún tipo de interpretación semántica clara, por ejemplo, en el caso de n-gramas de caracteres, éstos pueden capturar inicios o terminaciones de palabra pero también considera elementos que no corresponden a ninguno de estos casos.

En [Stamatatos 2009*b*], se menciona que el uso de características semánticas para la tarea de atribución de autoría suele mejorar los resultados obtenidos, sin embargo, se han hecho muy pocos intentos de explotar las características de alto nivel para la creación de marcadores de estilo. En este trabajo, se considera que el uso de una representación distribuida la palabra representación distribuida para la tarea autoría.

El modelo típico bolsa de palabras consiste en representar el texto como un conjunto de palabras con sus frecuencias respectivas, es decir, las palabras son características en el modelo de espacio vectorial correspondiente. Para el problema de la atribución de autoría, las palabras más frecuentes (palabras auxiliares) demostraron tener una alta eficiencia. Por otro lado, se proponen las características basadas en n-gramas de palabras con el fin de tomar ventaja de la información contextual y los resultados que se obtienen son ligeramente mejor que el modelo de bolsa de palabras.

En [Segarra et al. 2013] los autores proponen redes de adyacencia de palabras de función (en inglés *Function Word Adjacencay Networks WANs*), donde los nodos son las palabras de función y los bordes dirigidos representan la probabilidad de encontrar una palabra función objetivo en la proximidad ordenada de una palabra función fuente. En ese trabajo los autores informan que la precisión alcanzada por la WAN es más alta que la obtenida por las metodologías tradicionales que se basan solamente en las frecuencias de palabras función. A pesar de que las redes WAN consiguen una alta precisión en textos muy largos, siempre y cuando un acto de reproducción o una novela, sólo obtienen tasas razonables de textos cortos, tales como artículos de opinión periódico si el número de autores candidatos es pequeño.

Recientemente, el uso de la representación distribuida ha mostrado un gran poder en la captura de la semántica de las palabras, frases y oraciones, lo que beneficia a las aplicaciones de procesamiento de lenguaje natural.

En [Le & Mikolov 2014*a*], los autores presentan una representación distribuida de párrafos, un algoritmo no supervisado que aprende representaciones de entidades de longitud fija a partir de documentos de longitud variable. La idea es combinar la semántica de las palabras para la construcción de la semántica de documentos utilizando el modelo de memoria distribuida. El uso de representaciones distribuidas supera a los modelos de bolsa de palabras y-n-gramas que establecen nuevos resultados para el estado del arte en varias tareas de clasificación de textos y análisis de emociones.

En [Li & Shindo 2015], los autores presentan un enfoque supervisado en el que se utilizan redes neuronales recursivas y recurrentes con el fin de obtener la representación distribuida de un documento. Este enfoque especifico de la tarea que se desea resolver porque no se aprende ninguna representación general de las oraciones de un párrafo. Sin embargo, este enfoque supera a las líneas de base existentes en tareas como la clasificación binaria, clasificación con varias categorías y regresión.

Otro trabajo interesante utiliza representación distribuida a nivel de párrafos el vector de párrafo y un modelo de n-gramas con salto (en inglés, *skip-grams*) para la tarea de discriminar lenguajes similares presentado en [Franco-Salvador et al. 2015]. Los autores utilizan el modelo de *skip-grama* continuos [Mikolov, Sutskever, Chen, Corrado & Dean 2013] para generar

representaciones distribuidas de palabras y estimar el promedio de sus dimensiones con el fin de generar la representación documento. También evalúan su modelo de clasificación utilizando la representación vectorial a nivel de párrafos para aprender la representación del documento.

La combinación de vectores de palabras (*skip-gramas*) obtuvo mejores resultados en promedio, en comparación con el uso de vectores generados directamente de los párrafos. Sin embargo, esta conclusión es válida para la tarea de identificación de la variación lingüística y no para la atribución de autoría.

Existen pocos intentos por utilizar las representaciones distribuidas para la atribución de autoría. En [Kiros et al. 2014], los autores proponen un método para el aprendizaje de las representaciones distribuidas de atributos. Los atributos corresponden a una amplia variedad de conceptos, tales como indicadores de documentos (para aprender vectores de oraciones), indicadores de idiomas (para aprender las representaciones lingüísticas distribuidas), metadatos e información adicional acerca de los autores (para aprender perfil de autor, tales como la edad, el género y ocupación). El método se evalúa lo largo de varias tareas: clasificación sentimiento, la clasificación de documentos en varios idiomas, y el blog atribución de autoría. Para la tarea de atribución de autoría el método se evalúa sobre un corpus de los blogs en el que los atributos se basan sólo en los metadatos del autor y no en los propios textos. Por el contrario, para nuestro trabajo usamos sólo la información textual con el fin de extraer automáticamente representaciones vectoriales significativas de documentos.

Una nueva arquitectura de red neuronal llamada red neuronal convolucional (en inglés, *Convolutional Neural Network CNN*) ha sido utilizada para obtener representaciones distribuidas de textos [Rhodes 2015]. Se evaluó más de dos conjuntos de datos, una línea de base desarrollado por el autor y el conjunto de datos PAN 2012 [Juola 2012]. Para la representación de los documentos el autor utiliza el conjunto vectores de palabras obtenidos de Google Noticias [1], los vectores de palabras fueron formados a través del método de skip-grama y muestreo presentado en [Mikolov, Chen, Corrado & Dean 2013, Mikolov, Sutskever, Chen, Corrado & Dean 2013].

[1] *Veáse* `https://code.google.com/archive/p/word2vec/`

El autor utiliza el enfoque estándar para los modelos convolucionales (operación de concatenación sencilla) para codificar secuencias en lugar de palabras y sin promediar las dimensiones como se hizo en [Franco-Salvador et al. 2015]. La clasificación se realiza mediante regresión logística y los resultados muestran una alta precisión sobre el conjunto de datos de línea de base, pero la arquitectura CNN no superan el mejor método presentado en el PAN de 2012, mientras que nuestro método que se presenta a continuación obtiene mejores resultados para este conjunto de datos.

Capítulo 4

N-gramas sintácticos como marcadores de estilo

4.1. Información sintáctica

El conjunto de elementos que se encuentran presentes en el lenguaje escrito es demasiado extenso, por lo que su estudio típicamente se divide en cuatro niveles: nivel fonológico, nivel morfológico, nivel sintáctico y nivel semántico. En cada nivel se analiza un determinado aspecto del lenguaje y se definen los elementos del lenguaje que serán sus objetos de estudio [Galicia-Haro & Gelbukh 2007].

En el nivel sintáctico se busca analizar la sintaxis de un texto, es decir, la forma en la que las palabras se relacionan entre sí para expresar una idea y la función que tienen éstas dentro de un texto. El estudio de la sintaxis se centra básicamente en conocer las reglas que gobiernan la asociación de palabras y asignar una clasificación a las palabras según la función que desempeñan dentro de un texto.

Al conjunto de palabras que se utilizan en un idioma en particular se le conoce como léxico. Se puede observar que no todas las palabras se utilizan de la misma forma para la generación de texto, esto se debe a que cada una de ellas representa un concepto diferente; en el caso del idioma inglés, a las palabras se le asigna alguna de las siguientes clases, conocidas como clases gramaticales (en inglés, *Part of Speach POS*) [Galicia-Haro & Gelbukh 2007]: nombres propios, pronombres, sustantivos, verbos, adverbios, adjetivos, artículos, preposiciones y conjunciones.

Al conjunto de reglas que gobiernan la forma en que las palabras se pueden asociar para generar texto se le conoce como gramática. La gramática de un idioma en particular puede llegar a ser compleja, debido a la gran cantidad de reglas que la conforman y excepciones a las reglas.

Desde el punto de vista sintáctico, la unidad de estudio se conoce como sintagma, que es una estructura lingüística formada con por lo menos dos morfemas y debe ejercer una función en la estructura lingüística. Las funciones sintácticas se clasifican en:

- **Primaria:** se denominan núcleos y se caracterizan por ser independientes.
- **Secundaria:** se denominan modificadores o complementos y se caracterizan por ser dependientes, es decir, sólo tienen valor gramatical cuando se unen a un núcleo.
- **Terciaria:** se denominan nexos o relacionantes y se caracterizan por relacionar sintagmas o palabras.

Una oración representa la estructura sintáctica mínima y se define como una palabra o conjunto de palabras con que se expresa un sentido gramatical completo [Beristáin 2007]. Para representar las relaciones que existen entre los sintagmas que conforman una oración se utilizan los siguientes formalismos gramaticales: **gramática de dependencias** y **gramática de constituyentes**.

La gramática de dependencias muestra la relación entre pares de palabras, donde una de ellas es la palabra rectora y la otra palabra es dependiente de la primera. Utiliza una estructura de árbol que inicia con una palabra raíz, que es la palabra rectora de orden más general, posteriormente se crean derivaciones con sus palabras dependientes, en donde cada arco representa una dependencia entre palabras; las palabras dependientes a su vez se vuelven a dividir en palabras rectoras y palabras dependientes con lo que se genera un nuevo nivel en el árbol, hasta que se obtienen solamente palabras rectoras.

Como ejemplo tomemos la siguiente oración: *John smoked with a little more dignity and surveyed them in silence*. Utilizando el analizador sintáctico *Stanford Parser*[1], se obtiene la siguiente información: la categoría gramatical de las palabras, los lemas de las palabras y la relación de dependencia entre palabras, como se muestra en la tabla 4.1. Los signos de puntuación son omitidos en la tabla.

Tabla 4.1: Ejemplo de información sintáctica de dependientes

Número palabra	**Palabra**	**Etiqueta POS**	**Relación**	**Dependencia**
1	John	NNP	nsubj	2
2	smoked	VBD	root	0
3	with	IN	prep	2
4	a	DT	det	7
5	little	RB	advmod	6
6	more	JJR	amod	7
7	dignity	NN	pobj	3
8	and	CC	cc	2
9	surveyed	VBD	conj	2
10	them	PRP	dobj	9
11	in	IN	prep	9
12	silence	NN	pobj	11

[1] Veáse http://nlp.stanford.edu:8080/parser/

La información sobre las relaciones de dependencia entre las palabras se puede representar de manera gráfica mediante un árbol como se muestra en la figura 4.1. A ésta representación de la información de dependencias entre las palabras se conoce como árbol de dependencias.

En la tabla A.2 se muestran las etiquetas que utiliza el analizador sintáctico *Stanford Parser* para etiquetar las relaciones de dependencias entre palabras, la información completa se puede consultar en [De Marneffe & Manning 2008].

Por otro lado, la gramática de constituyentes supone que una oración se puede descomponer en bloques más pequeños, los cuales a su vez se pueden descomponer en bloques aún más pequeños. Los bloques de construcción se componen de un conjunto de palabras que forman agrupamiento sintácticos denominados: grupo nominal (GN), grupo verbal (GV), grupo proposicional (GP), sustantivo, adjetivo, verbo, etc.; también utiliza una estructura de árbol en la cual se muestra la forma en que cada bloque es conformado por otros bloques en las derivaciones de un nodo, hasta llegar a bloque más pequeños que son las palabras.

Para la misma oración utilizada anteriormente, la información sobre las reglas de constitución de la oración se puede representar de manera gráfica mediante un árbol como se muestra en la figura 4.2.

4.2. Definición de n-gramas sintácticos

Dentro del Procesamiento de Lenguaje Natural los n-gramas han sido utilizados para representar textos en tareas como la atribución de autoría, creación de perfiles, detección de plagio, entre otros. Es posible obtener diferentes tipos de n-gramas dependiendo del tipo de elementos utilizados para su construcción siendo algunos de los más utilizados palabras, caracteres, lemas y etiquetas POS.

Una variante de n-gramas propuesto en [Sidorov 2013*b*, Sidorov 2013*a*, Posadas-Duran et al. 2014] son los llamados n-gramas sintácticos (sn-gramas), los cuales se obtienen recorriendo los árboles de dependencia de las oraciones de un texto. Los sn-gramas son capaces de representar las relaciones entre palabras desde un punto de vista sintáctico, a diferencia de los n-gramas tradicionales de palabras o caracteres que pueden incluir n-gramas en el que sus elementos no tienen más relación que el orden de aparición en el texto.

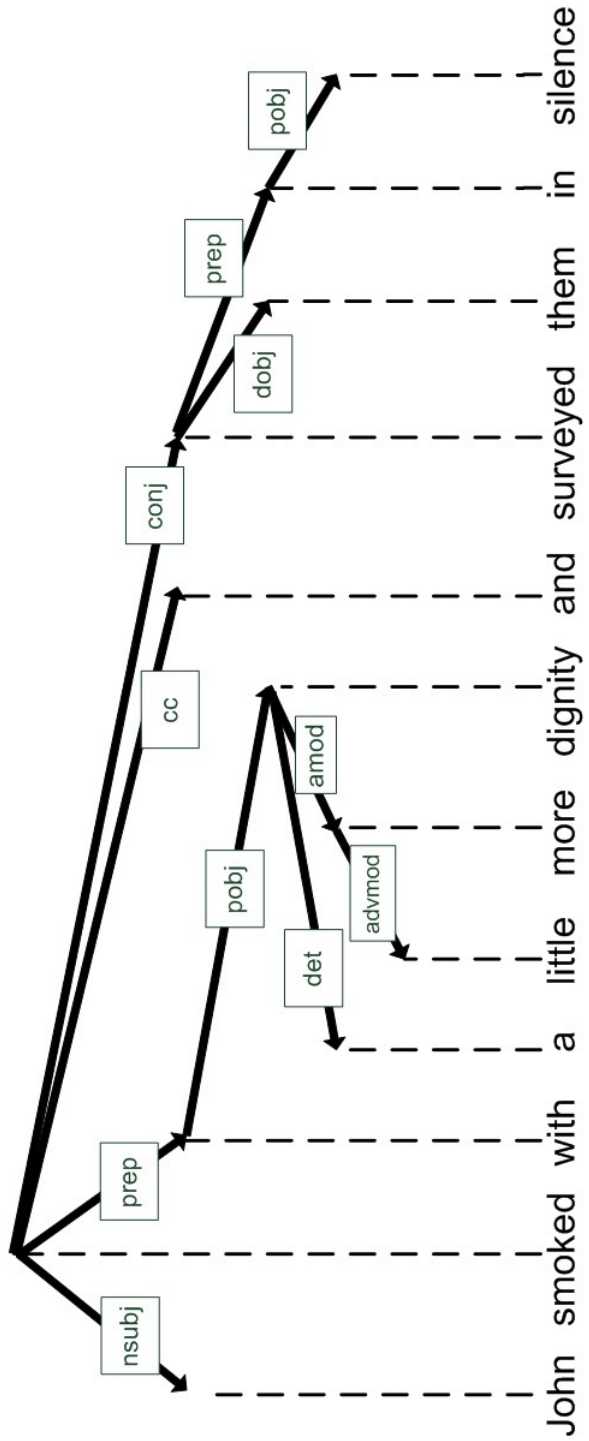


Figura 4.1: *Ejemplo de árbol de dependencias*

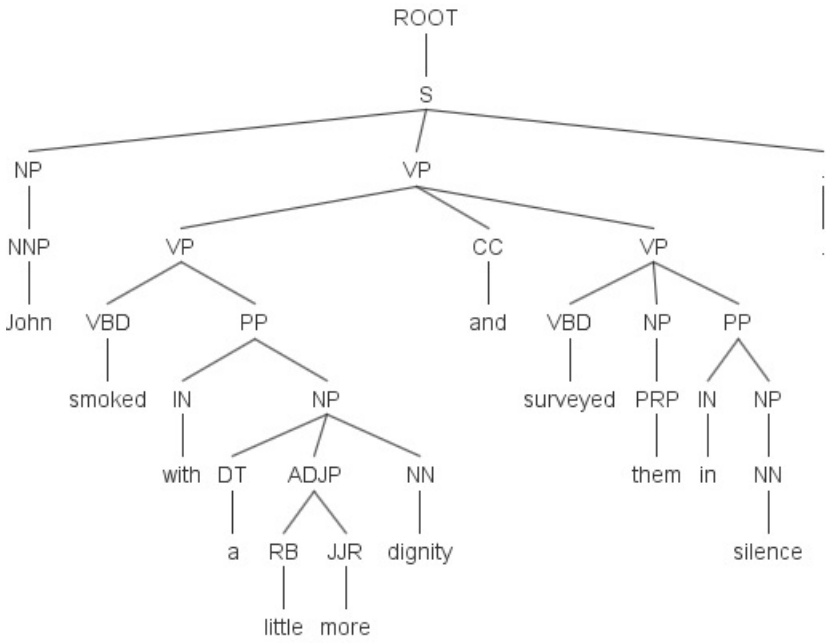


Figura 4.2: *Ejemplo de árbol de constituyentes*

Consideremos como ejemplo la oración: "*Victor sat at the counter on a plush red stool.*", en la figura 4.2 se muestran todos los n-gramas de tamaño 3 que se obtienen de la oración de ejemplo junto con su frecuencia.

Tabla 4.2: N-gramas de palabras de la oración "*Victor sat at the counter on a plush red stool*"

N-grama	**Frecuencia**
Victor-sat-at	1
sat-at-the	1
at-the-counter	1
the-counter-on	1
counter-on-a	1
on-a-plush	1
a-plush-red	1
plush-red-stool	1

En la figura 4.3 se muestra el árbol de dependencias de la oración de ejemplo mencionada anteriormente y en la tabla 4.3 se muestran los sn-gramas de palabras de tamaño 3 que se obtienen. Para representar los sn-gramas que se muestran en la tabla 4.3 se utiliza el metalenguaje presentado en [Sidorov 2013*b*] en el que se plantean convenciones para representar los diferentes subárboles que se obtienen recorriendo el árbol de dependencias; de acuerdo con el metalenguaje el elemento que se encuentra en el extremo izquierdo representa el nodo padre del subárbol, los nodos hermanos se encuentran separados por comas respetando el orden de aparición en el árbol de izquierda a derecha y los elementos encerrados entre paréntesis *[]* representan nodos hijo del nodo que se encuentra fuera de los paréntesis en el extremo izquierdo.

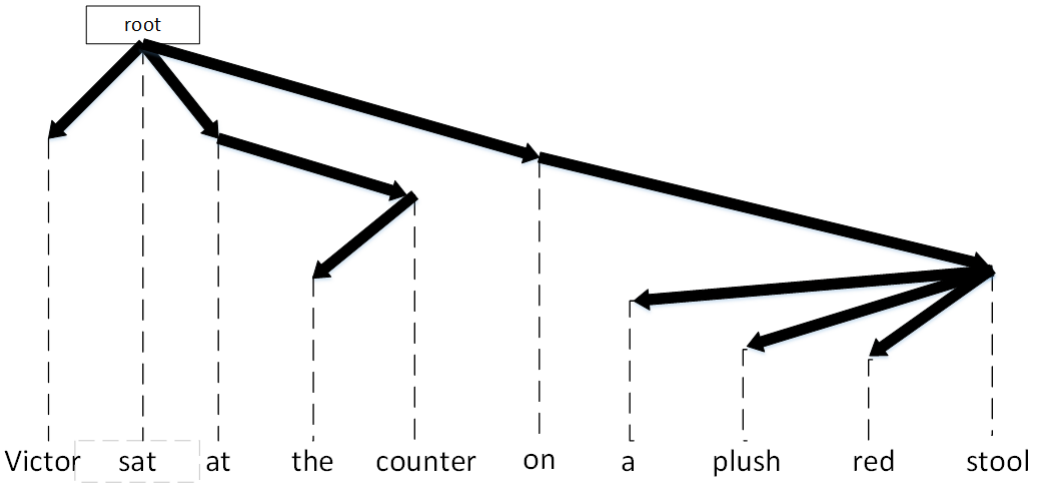


Figura 4.3: Árbol de dependencias de la oración "*Victor sat at the counter on a plush red stool*"

Los n-gramas tradicionales de palabras capturan secuencia de palabras respetando el orden en que aparecen las palabras en el texto (análisis superficial del texto), en contraste, los sn-gramas capturan las relaciones sintácticas entre las palabras que corresponden a las reglas gramaticales del lenguaje (análisis de profundo del texto), por ejemplo, al comparar el n-grama *Victor-sat-at* y el sn-grama *sat[Victor,at]* se observa que aunque ambos n-gramas se conforman por las mismas palabras, en los sn-gramas se establece una relación entre las palabras *Victor* y *at*, así como una relación entre éstas y la palabra *sat*.

Tabla 4.3: N-gramas de palabras de la oración "*Victor sat at the counter on a plush red stool*"

N-grama	**Frecuencia**
sat[Victor,at]	1
sat[Victor,on]	1
sat[at,on]	1
sat[on[stool]]	1
sat[at[counter]]	1
on[stool[plush]]	1
on[stool[a]]	1
on[stool[red]]	1
stool[a,plush]	1
stool[a,red]	1
stool[plush,red]	1
sat[at[counter]]	1
at[counter[the]]	1

En la tabla 4.3 se observa que los sn-gramas capturan una mayor cantidad de información lo que permite obtener una representación más completa del texto, además al capturar las reglas utilizadas por el autor para la composición del texto se espera que los sn-gramas modelen de una mejor manera las preferencias del autor sobre el uso del lenguaje a diferencia de los n-gramas tradicionales.

Los elementos que constituyen los sn-gramas pueden ser de distinta naturaleza, en general, se puede utilizar cualquier información contenida en el árbol de dependencias, es decir, elementos léxicos, etiquetas POS y etiquetas de dependencias. Una variante de los sn-gramas es combinar representaciones de distinta naturaleza para su composición, denominados sn-gramas mixtos (en inglés, *mixed sn-grams*), que se obtienen al fijar la información sobre el elemento padre del sn-grama de un tipo y utilizar un tipo diferente para el resto de los elementos, por ejemplo, una combinación que se puede hacer utilizando palabras para los elementos padre y etiquetas POS para el resto de los nodos, a esta combinación la denominaremos como *word+POS sn-gram* [Sidorov 2014]. Otras posibles combinaciones son *SR+POS sn-grams*, *SR+words sn-grams*, *POS+SR sn-grams*, *POS+words sn-grams* and *words+SR sn-grams* [Sidorov 2013*b*, Sidorov et al. 2012*b*].

En la tabla 4.5 se muestra parte de la información sintáctica que se obtiene de la oración "*Victor sat at the counter on a plush red stool*" al ser procesada por un analizador sintáctico y en la tabla 4.4 se muestran las diferencias de de los sn-gramas que utilizan el mismo tipo información en cada elemento y los sn-gramas mixtos para la combinación *word+POS sn-gram*.

Tabla 4.4: Etiquetas POS de la oración "*Victor sat at the counter on a plush red stool*"

Id palabra	**Palabra**	**Etiqueta POS**
1	Victor	NNP
2	sat	VBD
3	at	IN
4	the	DT
5	counter	NN
6	on	IN
7	a	DT
8	plush	JJ
9	red	JJ
10	stool	NN

Tabla 4.5: Comparativo de sn-gramas de palabras vs Word+POS sn-gramas

Palabras	**Word+POS**
sat[Victor,at]	sat[NNP,IN]
sat[Victor,on]	sat[NNP,IN]
sat[at,on]	sat[IN,IN]
sat[on[stool]]	sat[IN[NN]]
sat[at[counter]]	sat[IN[NN]]
on[stool[plush]]	on[NN[JJ]]
on[stool[a]]	on[NN[DT]]
on[stool[red]]	on[NN[JJ]]
stool[a,plush]	stool[DT,JJ]
stool[a,red]	stool[DT,JJ]
stool[plush,red]	stool[JJ,JJ]
sat[at[counter]]	sat[IN[NN]]
at[counter[the]]	at[NN[DT]]

En la tabla 4.4 se observa que, a diferencia de los sn-grams de palabras, los sn-gramas mixtos son capaces de identificar nuevos patrones debido al hecho de que combinan información de contextos diferentes, por ejemplo, los sn-gramas *sat[NNP,IN]*, *sat[IN[NN]]*, *on[NN[JJ]]* son patrones que ocurren con una mayor frecuencia en comparación con los sn-gramas de un solo tipo.

Capítulo 5

Método propuesto

En este trabajo doctoral se propone un método para la atribución de autoría utilizando un enfoque de aprendizaje automático y dos estrategias para el modelado del estilo de un autor: uso de n-gramas sintácticos y representación distribuida de textos. De las dos variantes sobre el problema de atribución de autoría solamente (abierta o cerrada) este trabajo se enfoca en la atribución cerrada.

La tarea de atribución de autoría (atribución de autoría) cerrada puede verse como un problema de clasificación multiclase que se define como sigue: dado un conjunto de autores conocidos $\mathbf{A} = \{A_1, A_2, \ldots, A_i\}$ y un conjunto de textos de muestra de cada autor $\mathbf{T} = \left\{t_1^1, t_2^1, \ldots, t_n^1, \ldots, t_j^i\right\}$, donde el elemento t_j^i corresponde al j-ésimo ejemplo del autor A_i, el problema consiste en construir un clasificador F que asigne a cada elemento del conjunto de textos de autoría desconocida $\mathbf{X} = \{x_1, x_2, \ldots, x_m\}$ a sólo un autor conocido, es decir, $F : \mathbf{X} \rightarrow \mathbf{A}$.

Se propone utilizar un enfoque de aprendizaje automático supervisado en el que el conjunto de textos muestra T corresponden al conjunto de entrenamiento que serán la entrada de un clasificador y los textos de autoría desconocida X corresponden al conjunto de prueba. La información del conjunto de entrenamiento será procesada utilizando una estrategia basada en ejemplos, es decir, cada muestra de texto de un autor se considera como un ejemplo sobre el estilo de un autor llamado instancia y cada instancia es analizada de manera independiente para la construcción de un modelo sobre el estilo de un autor.

La información sobre el estilo de un autor contenido en cada instancia es codificada utilizando un espacio vectorial de tal forma que v_j^i corresponde a la representación vectorial de texto t_j^i. Las dimensiones que conforman el espacio vectorial corresponden a una selección de marcadores de estilo y se asume que las características son independientes entre sí (modelo de bolsa de palabras). Los textos del conjunto de prueba también son codificados en el espacio vectorial seleccionado donde r_m corresponde a la representación vectorial del texto x_m.

Dependiendo de la naturaleza de los marcadores de estilo seleccionados para la conformación del espacio vectorial puede existir ruido que afecte la eficiencia de la atribución de autoría, por ejemplo, marcadores poco discriminantes o marcadores redundantes. Para disminuir la cantidad de ruido se pueden aplicar estrategias para reducir la cantidad de características utilizadas para la conformación del espacio vectorial a un conjunto más pequeño y más representativo.

Para realizar la asignación de un texto de autoría desconocida con un algún autor del conjunto de autores candidatos A se utiliza un clasificador, el cual se entrena con las instancias del conjunto de entrenamiento en su representación vectorial. En esta propuesta se utilizan cuatro métodos de clasificación utilizados ampliamente en el estado del arte: máquinas de soporte vectorial (SVM), clasificador bayesiano con distribución multinomial (NBM), vecinos más cercanos (kNN) y regresión logística (LR). Los clasificadores utilizados en esta propuesta fueron seleccionados porque las representaciones propuestas generan espacios vectoriales con una alta dimensionalidad (más de 100 dimensiones) y los clasificadores antes mencionados han reportado un buen desempeño en este tipo de escenarios.

En la figura 5.1 se muestra el diagrama a bloques de la propuesta y en las secciones siguientes se describe a detalle las características del método propuesto.

5.1. Representación distribuida a nivel de documentos

Para resolver la tarea atribución de autoría usamos un enfoque de aprendizaje de máquina, compuesto por dos etapas: la etapa de entrenamiento y la fase de pruebas. En la etapa de entrenamiento se obtiene una representación vectorial de los textos del conjunto T usando un modelo distribuido, es decir, $V = \left\{v_1^1, v_2^1, \ldots, v_j^i\right\}$ donde $v_j^i = \{f_1, f_2, \ldots, f_n\}$ es la representación vectorial del texto t_j^i. En la figura 5.2 se muestra un diagrama a bloques para la solución del problema de atribución de autoría utilizando una representación distribuida.

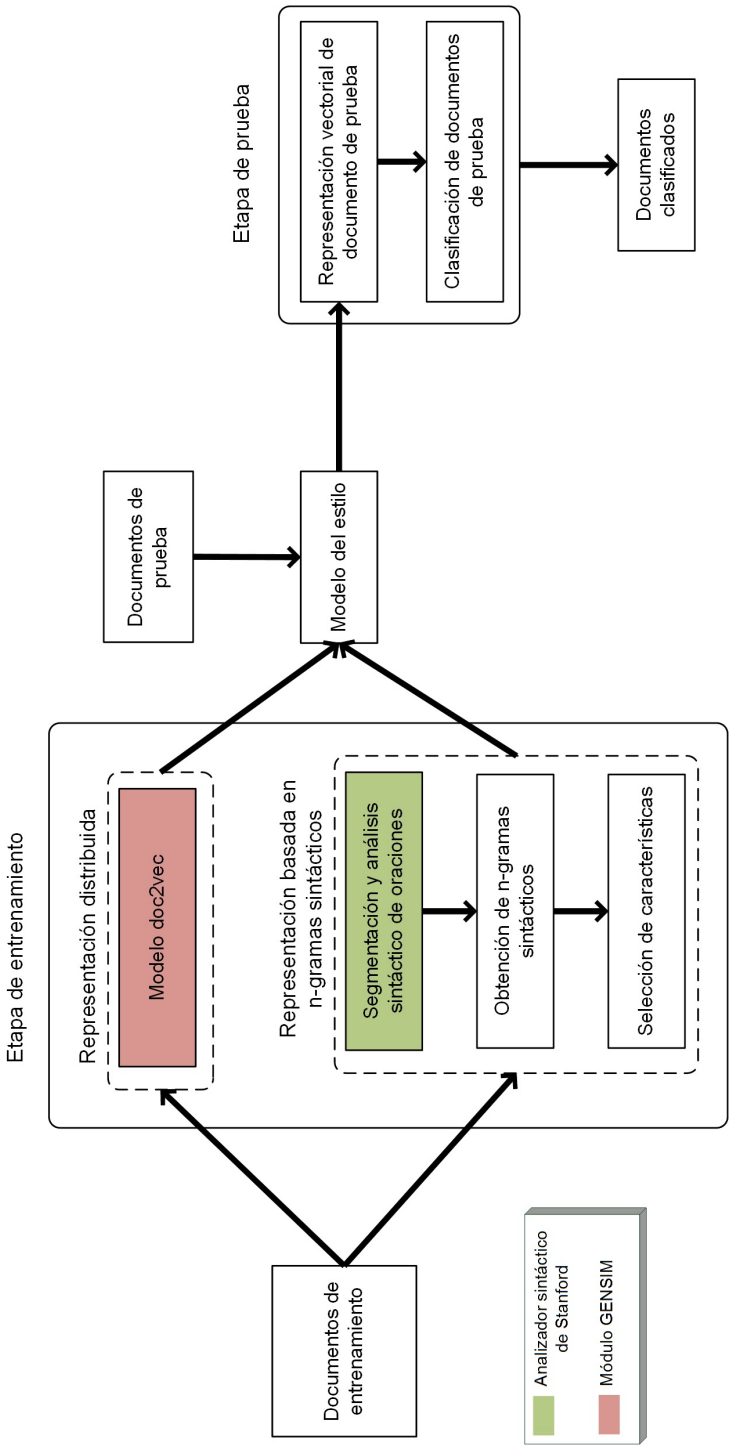


Figura 5.1: *Diagrama a bloques de la propuesta para la atribución de autoría*

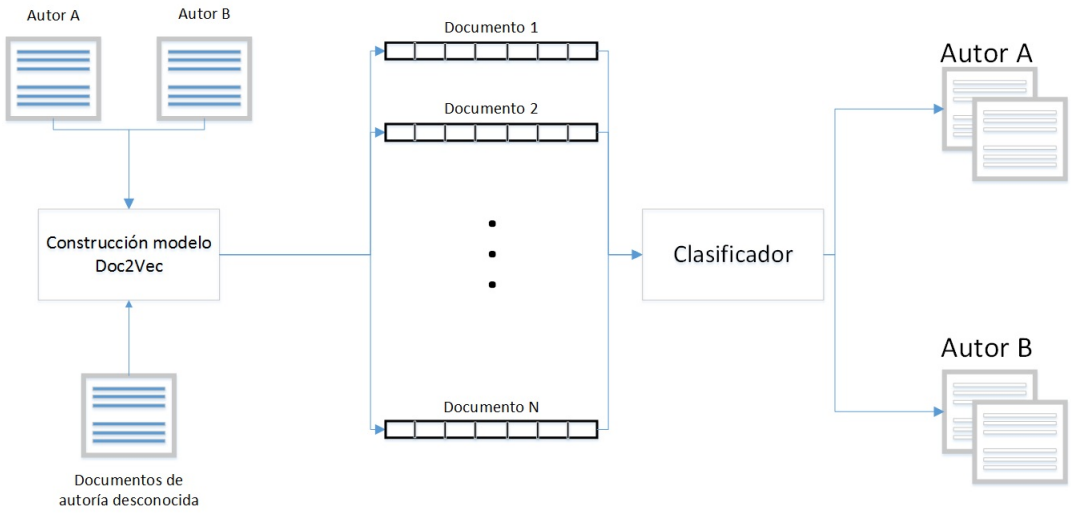


Figura 5.2: *Diagrama a bloques del método para resolver la atribución de autoría utilizando un representación distribuida a nivel de documentos*

La representación vectorial de las instancias se obtiene aplicando el método de representación distribuida a nivel de documentos llamado doc2vec [Le & Mikolov 2014*b*] que se encuentra implementado en una librería que se encuentra disponible libremente en la *Web* llamada GENSIM [1].

El método doc2vec construye un modelo para obtener las representaciones distribuidas a nivel de documentos, el cual ofrece dos enfoques posibles para construir el modelo: memoria distribuida (DM) y bolsa de palabras distribuida (DBOW). En el estado del arte relacionado con el uso del modelo doc2vec se reporta que el concatenar las representaciones que se obtienen usando ambos enfoques por separado mejora la eficiencia respecto a usar solamente una representación de alguno de los dos enfoques.

La implementación del modelo doc2vec requiere de tres parámetros: el número de características que serán devueltas, el tamaño de la ventana que capturan el contexto y la frecuencia mínima de las palabras que se consideran en el modelo. Los valores de los parámetros dependen de la naturaleza de la información que se desea modelar, en el caso de la tarea de atribución de autoría no se han reportado trabajos previos que utilizan doc2vec para representar la información, sin embargo, para el caso de la tarea de clasificación de opiniones

[1] Véase `https://radimrehurek.com/gensim/`

(en inglés *sentiment analysis*) se reporta utilizar representaciones con 300 características, tamaño de ventana igual a 10 y frecuencia mínima de 5 [Mikolov, Sutskever, Chen, Corrado & Dean 2013, Liang et al. 2015]. Tomando como referencia la información anterior, en esta tesis se propone realizar una búsqueda sobre los siguientes rangos para los diferentes parámetro que requiere el modelo: en el caso del número de características se propone el rango de $[50, 500]$, en el caso del tamaño de ventana se propone el rango de $[2, 40]$ y para la frecuencia mínima se propone el rango de $[2, 40]$.

Para construir un modelo sobre un conjunto de datos, el método doc2vec recibe como entrada el texto plano de los documentos que conforman el conjunto de entrenamiento sin etiquetar (sin especificar la clase a la que pertenecen) y el texto de los documentos del conjunto de prueba sin etiquetar. Se ha reportado que el método doc2vec obtiene un buen desempeño en conjuntos de datos de alrededor de miles de ejemplos e incluso millones de ejemplos [Mikolov, Sutskever, Chen, Corrado & Dean 2013, Liang et al. 2015]. Los corpus relacionados con la atribución de autoría que se utilizan en la presente tesis son de alrededor de cientos de ejemplos por autor, no se considera el uso de recursos adicionales porque se dese analizar la eficiencia del método en escenarios con poca información.

Los textos utilizados no requieren de algún preprocesamiento especial, solamente son segmentados a nivel de elementos (palabras y signos de puntuación). Diferentes representaciones de los texto son utilizadas con la finalidad de evaluar el comportamiento de conjuntos de elementos, particularmente se utilizan la representación en términos de bigramas, trigramas y cuatrigramas siguiendo el procedimiento estándar para su obtención. En la figura 5.3 se muestra el ejemplo de un texto utilizando el formato de división a nivel de palabras presentado como una lista y en la figura 5.4 se muestra el mismo documento pero utilizando un formato de entrada de bigramas en el que las palabras que conforman cada bigramas se encuentran separadas por un guion bajo (_).

```
[['it', 'is', 'curious', 'that', 'the', 'one', 'appealing', 'thing', 'about',
'ken', 'clarke', '-', 'his', 'enormously', 'advanced', 'age', '-', 'should',
'be', 'perceived', 'as', 'a', 'weakness', '.', 'there', 'are', ',', 'of',
'course', ',', 'a', 'hundred', 'anti-', 'age-discrimination', 'reasons', ',',
'too', 'worthy', 'to', 'list', 'here', ',', 'to', 'applaud', 'the',
'promotion', ',', 'at', 'his', 'third', 'attempt', ',', 'of', 'a', 'man', 'who',
'has', 'reached', 'the', 'official', 'age', 'of', 'retirement', '.', 'but', 'it',
'is', 'the', 'return', 'from', 'holiday', 'of', 'tony', 'blair', ',', 'still', 'in',
'the', 'grip', 'of', 'what', 'seems', 'to', 'be', 'one', 'of', 'the', 'longest',
'and', 'most', 'florid', 'mid-life', 'crises', 'in', 'modern', 'history', ',',
'that', 'now', 'shows', "clarke's", 'seniority', 'in', 'such', 'an',
'appealing', 'light', '.', 'on', 'a', 'superficial', 'level', ',', 'there', 'is',
```

Figura 5.3: *Representación de un texto utilizando el formato de palabras*

```
[['it_is', 'is_curious', 'curious_that', 'that_the', 'the_one',
'one_appealing', 'appealing_thing', 'thing_about', 'about_ken',
'ken_clarke', 'clarke_-', '-_his', 'his_enormously',
'enormously_advanced', 'advanced_age', 'age_-', '-_should',
'should_be', 'be_perceived', 'perceived_as', 'as_a', 'a_weakness',
'weakness_.', '._there', 'there_are', 'are_,', ',_of', 'of_course',
'course_,', ',_a', 'a_hundred', 'hundred_anti-', 'anti-_age-
discrimination', 'age-discrimination_reasons', 'reasons_,', ',_too',
'too_worthy', 'worthy_to', 'to_list', 'list_here', 'here_,', ',_to',
'to_applaud', 'applaud_the', 'the_promotion', 'promotion_,', ',_at',
'at_his', 'his_third', 'third_attempt', 'attempt_,', ',_of', 'of_a',
```

Figura 5.4: *Representación de un textos utilizando el formato de bigramas de palabras*

5.2. Representación basada en n-gramas sintácticos

Un estilo de escritura se define a partir de una serie de preferencias sobre el uso del lenguaje natural, sin embargo, debido a la complejidad inherente del lenguaje natural se utiliza una representación enfocada a ciertos aspectos que se consideran relevantes para la detección de plagio. En este método se propone el uso de la información sintáctica para caracterizar el estilo de escritura de una sección de texto, porque la información sintáctica es robusta a cambios de tema y a diferentes tipos de plagio, además de que en la actualidad existen herramientas para efectuar análisis sintáctico de un texto con una eficiencia aceptable.

El estilo de escritura de un texto se representa mediante un vector de rasgos que contiene información sintáctica del texto, el valor de cada una de las entradas del vector es determinado por un conjunto propuesto de marcadores de estilo a nivel sintáctico. Los marcadores de estilo propuestos se basan en la frecuencia de ocurrencia de los n-gramas sintácticos mencionados en la sección 4.2.

Para representar el estilo contenido el texto t_j^i se utiliza un vector v_j^i en el que cada entrada contiene la información relativa a un marcador de estilo sintáctico (característica), es decir,

$$v_j^i = \{sm_1, sm_2, \ldots\},$$

donde sm_1 representa la frecuencia de aparición del marcador de estilo 1 en el texto t_j^i del autor A_i, sm_2 representa la frecuencia de ocurrencia del marcador de estilo 2 en el texto t_j^i y así sucesivamente.

Se proponen los siguientes marcadores de estilo para construir la representación sobre el estilo de un autor de cada una de las instancias descritas en la sección 4.2:

- SR sn-grama,
- POS sn-grama,
- Word sn-grama,
- Lema sn-grama,
- SR + POS sn-grama,
- SR + Word sn-grama,

- SR + Lema sn-grama,
- POS + SR sn-grama,
- POS + Word sn-grama,
- POS + Lema sn-grama,
- Word + POS sn-grama,
- Word + SR sn-grama,
- Word + Lema sn-grama,
- Lema + SR sn-grama,
- Lema + POS sn-grama,
- Lema + Word sn-grama.

EL SR-sngrama, por ejemplo, es un marcador que hace referencia a la frecuencia de ocurrencia de subárboles decodificados con la etiquetas de relaciones de dependencia que existe entre las palabras que conforman una oración; el marcador de estilo indica la frecuencia con que las estructuras de los subárboles se repiten a lo largo de las oraciones que conforman un texto. Por ejemplo, para la oración *John smoked with a little more dignity and surveyed them in silence*, la información de dependencia entre palabras se muestran en la figura 5.5 y la frecuencia de todos los n-gramas de tamaño 4 se muestran en la tabla 5.1.

Dependiendo de la longitud las oraciones que conforman un texto, pueden existir n-gramas de dependencia de diferentes tamaños, en especial para oraciones muy largas y estructuras de árboles en la que los nodos tienen muchos hijos (más de tres hijos) como por ejemplo en oraciones en las que se listan objetos, el número de n-gramas puede ser demasiado grande. En esta propuesta se propone limitar el tamaño de los n-gramas con los que se trabaja a un tamaño entre 3 y 7.

Los n-gramas de dependencia con tamaño menor a 3 son descartados debido a que este tipo de n-gramas no capturan las relaciones no lineales del árbol de dependencia en su lugar capturan las relaciones lineales como lo hacen los n-gramas tradicionales, por otro lado, los n-gramas de tamaño mayor a 7 son demasiado raros y su ocurrencia eventual no es descriptiva para el estilo de un autor.

Tabla 5.1: Ejemplo de n-gramas de relaciones de dependencia

n-grama	**frecuencia**
root[conj[prep[pobj]]]	1
root[prep,conj[dobj]]	1
root[nsubj,conj[dobj]]	1
root[conj[dobj,prep]]	1
pobj[det,amod[advmod]]	1
root[prep[pobj],conj]	1
root[cc,conj[prep]]	1
root[nsubj,prep,conj]	1
root[nsubj,cc,conj]	1
prep[pobj[det,amod]]	1
root[prep[pobj[det]]]	1
root[nsubj,prep[pobj]]	1
prep[pobj[amod[advmod]]]	1
root[cc,conj[dobj]]	1
root[prep[pobj],cc]	1
root[nsubj,prep,cc]	1
root[prep,conj[prep]]	1
root[nsubj,conj[prep]]	1
root[prep,cc,conj]	1
conj[dobj,prep[pobj]]	1
root[prep[pobj[amod]]]	1

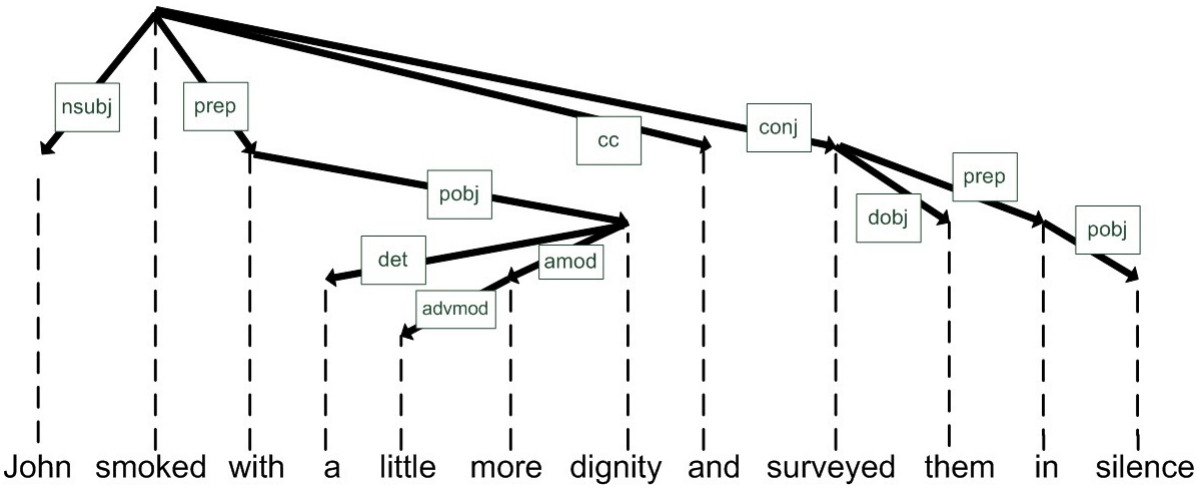


Figura 5.5: *Ejemplo de obtención n-gramas de relación de dependencia*

Otro aspecto importante sobre los n-gramas de dependencia es su frecuencia de ocurrencia, en general si un n-grama de dependencia ocurre de manera esporádica en una instancia se considera que éste no es representativo del estilo por lo que el uso de los n-gramas de dependencia se limita a una frecuencia de ocurrencia mayor a 1 por instancia.

El concepto de los n-gramas de dependencia fue descrito por primera vez en los trabajos [Sidorov et al. 2012*b*, Sidorov 2013*b*], donde se explican los diferentes tipos de información sintáctica que se puede utilizar para su conformación, se establece un metalenguaje para su representación y muestran algunos ejemplos sobre su obtención, sin embargo, no existía una implementación para su obtención.

Un n-grama sintáctico corresponde a un subárbol del árbol de dependencias y la obtención de todos los n-gramas sintácticos corresponde de manera general al problema de enumeración de subárboles de un árbol. Algunos trabajos relacionados con la enumeración de subárboles se presentan en [Wasa et al. 2012, Zaki 2002]. En esta trabajo doctoral se propone un algoritmo y una implementación para la obtención de los n-gramas de dependencia, el algoritmo toma como entrada un objeto que contenga la información sintáctica sobre el árbol de dependencia y regresa como salida un listado con los distintos n-gramas de dependencia que existen y su ocurrencia a lo largo del texto.

Considerando como ejemplo la oración "*Victor sat at the counter on a plush red stool*", en la figura 5.6 se muestra una representación gráfica del árbol de dependencias de la oración y en la tabla 5.2 se muestra la información sintáctica que se requiere como entrada al algoritmo de obtención de n-gramas de dependencia.

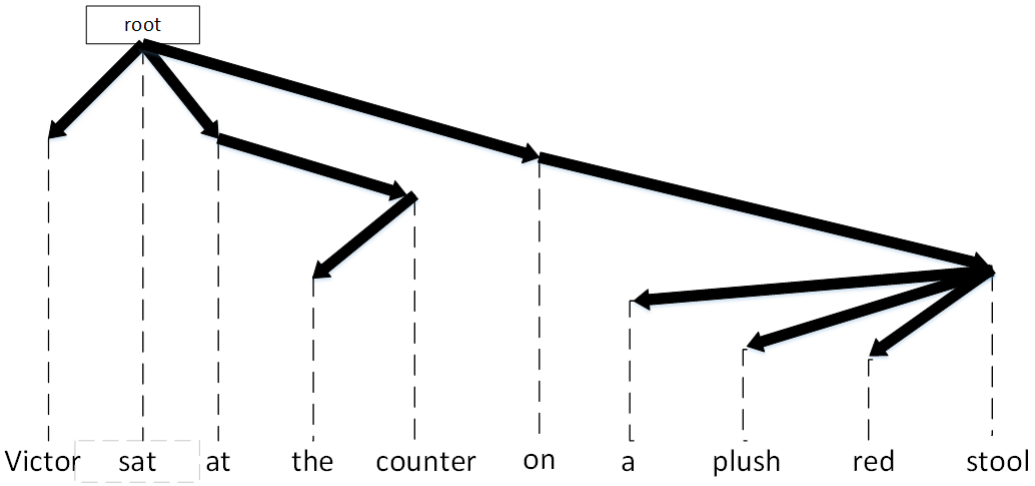


Figura 5.6: Árbol de dependencias de la oración "*Victor sat at the counter on a plush red stool*"

Tabla 5.2: Información sintáctica de la oración "*Victor sat at the counter on a plush red stool*"

Id	Palabra	Lema	POS	Dependiente	SR	Hijos
1	Victor	Victor	NNP	2	nsubj	
2	sat	sit	VBD	0	root	$[1, 3, 6]$
3	at	at	IN	2	prep	$[5]$
4	the	the	DT	5	det	
5	counter	counter	NN	3	pobj	$[4]$
6	on	on	IN	2	prep	$[10]$
7	a	a	DT	10	det	
8	plush	plush	JJ	10	amod	
9	red	red	JJ	10	amod	
10	stool	stool	NN	6	pobj	$[7, 8, 9]$
Hojas (nodos que no tienen hijos): $[1, 4, 7, 8, 9]$						

En el algoritmo 7 se muestra el pseudocódigo de la función *ObtenDepngrams* que es la función principal que obtiene el listado de los n-gramas de dependencia de un texto. Los algoritmo del 1 al 6 muestran el pseudocódigo de funciones auxiliares que requiere la función principal.

En el algoritmo 1 se muestra el pseudocódigo de la función *Combinaciones* que calcula el número de combinaciones de un tamaño especifico que se pueden hacer con un listado de elementos. En el algoritmo 2 se presenta la función *subárboles* que obtiene todos los índices de los nodos que pueden pertenecer a un subárbol conociendo el nodo raíz de ese subárbol.

En los algoritmo del 3 al 6 se presentan las funciones *Prepare_SNgram* y *Obtensubárboles*. La función *Obtensubárboles* construye los subárboles utilizando los índices de los elementos y el metalenguaje propuesto en [Sidorov 2013*a*] y la función *Prepare_SNgram* traduce los índices de los elementos por su correspondiente etiqueta de información sintáctica y dependiendo del tipo de n-grama de dependencia que se haya solicitado.

Algoritmo 1 Función *Combinacion*

```
1:  función COMBINACIONES(número de elementos(sz), tamaño de las combinaciones(r))
2:      Variables: numerador, divisor, aux
3:      si sz == r entonces
4:          numerador ← 1
5:      si no
6:          numerador ← sz
7:      fin si
8:      para todo i en [1, sz] hacer
9:          numerador ← numerador * (sz − i)
10:     fin para
11:     aux ← r
12:     para todo i en [1, r] hacer
13:         aux ← aux * (r − i)
14:     fin para
15:     divisor ← sz − r
16:     para todo i en [1, sz − r] hacer
17:         divisor ← divisor * (sz − r − i)
18:     fin para
19:     numerator ← numerator/ (aux * divisor)
20:     regresa numerator
21: fin función
```

Algoritmo 2 Función *Subarboles*

```
1: función SUBÁRBOLES(índice del nodo (idx), nodos hijo (hijos), nodos hojas (hojas))
2:     Variables: ngram[ ], opciones[ ], combinaciones[ ], lista[ ], val_max, m
3:     para todo r en [1, longitud(hijos)] hacer
4:         para todo j en [1, r + 1] hacer
5:             combinaciones[j − 1] ← j − 1
6:         fin para
7:         opciones ← [ ], ngram ← [ ], ngram.agregar(idx, −delizq−)
8:         para todo z en [0, r] hacer
9:             ngram.agregar(hijos[combinaciones[z]])
10:            si hijos[combinaciones[z]] ∉ hojas entonces
11:                opciones.agregar(hijos[combinaciones[z]])
12:            fin si
13:            ngram.agregar(−delsep−)
14:        fin para
15:        ngram.agregar(−delder−), lista.agregar(ngram, opciones)
16:        top ← Combinacion(longitud(hijos), r)
17:        para todo j en [2, top + 1] hacer
18:            m ← r, val_max ← longitud(hiijos)
19:            mientras combinaciones[m − 1] + 1 == val_max hacer
20:                m ← m − 1, val_max ← val_max − 1
21:            fin mientras
22:            combinaciones[m − 1] ← combinaciones[m − 1] + 1
23:            para todo k en [m + 1, r + 1] hacer
24:                combinaciones[k − 1] ← combinaciones[k − 2] + 1
25:                options ← [ ], ngram ← [ ]
26:                ngram.agregar(value, −delizq−)
27:                para todo z en [0, r] hacer
28:                    ngram.agregar(children[combinations[z]])
29:                    si hijos[combinations[z]] ∉ hojas entonces
30:                        options.agregar(children[combinations[z]])
31:                    fin si
32:                    ngram.agregar(−delsep−)
33:                fin para
34:                ngram.agregar(−delder−), lista.agregar(ngram, options)
35:            fin para
36:        fin para
37:    fin para
38:    regresa lista
39: fin función
```

Algoritmo 3 Función *Prepare_SNgram*

1: **función** PREPARE_SNGRAM(índices n-grama (line), información sintáctica de la oración (sentence), tipo de n-grama (op))
2: $ngram \leftarrow$ “”
3: **para todo** $item \in line$ **hacer**
4: **si** $tipo\_dato\,(item)$ es str **entonces**
5: $ngram \leftarrow ngram + item$
6: **si no si** $tipo\_dato\,(item)$ es int **entonces**
7: **si** $op == -1$ **entonces**
8: $ngram \leftarrow ngram + convierte\_cadena\,(item)$
9: **si no si** $op == 0$ **entonces**
10: $ngram \leftarrow ngram + sentence.word[item]$
11: **si no si** $op == 1$ **entonces**
12: $ngram \leftarrow ngram + sentence.rel[item]$
13: **si no si** $op == 2$ **entonces**
14: $ngram \leftarrow ngram + sentence.pos[item]$
15: **si no si** $op == 3$ **entonces**
16: $ngram \leftarrow ngram + sentence.lemma[item]$
17: **si no si** $op == 4$ **entonces**
18: $ngram \leftarrow ngram + sentence.word[item]$
19: $op \leftarrow 1$
20: **si no si** $op == 5$ **entonces**
21: $ngram \leftarrow ngram + sentence.word[item]$
22: $op \leftarrow 2$
23: **si no si** $op == 6$ **entonces**
24: $ngram \leftarrow ngram + sentence.rel[item]$
25: $op \leftarrow 0$
26: **si no si** $op == 7$ **entonces**
27: $ngram \leftarrow ngram + sentence.rel[item]$
28: $op \leftarrow 2$
29: **si no si** $op == 8$ **entonces**
30: $ngram \leftarrow ngram + sentence.pos[item]$
31: $op \leftarrow 0$
32: **si no si** $op == 9$ **entonces**
33: $ngram \leftarrow ngram + sentence.pos[item]$
34: $op \leftarrow 1$
35: **fin si**
36: **si no**
37: $ngram \leftarrow ngram + prepare\_SNgram\,(item, sentence, op)$
38: **fin si**

Algoritmo 5 Función *GeneraDepngrams*

```
    fin para
    regresa ngram
fin función

función OBTENSUBARBOLES(subárbol (original), árbol de dependencias (sentence), tamaño mínimo (min_size), tamaño máximo (min_size))
    Variables: combinations[ ], candidates[ ], size,
    para todo combination ∈ original hacer
        si longitud(combination[1]) > 0 entonces
            size ← longitud(prepare_SNgram(combination[0], sentence, −1))
            combinations.agregar([combination[0], combination[1], size])
        fin si
        si combination[0][0] <> root_idx entonces
            size = longitud(prepare_SNgram(combination[0], sentence, −1))
            candidates.agregar([combination[0], combination[1], size])
        fin si
    fin para
    mientras longitud(candidates) > 0 hacer
        candidate ← candidates.sacar[0], value ← candidate[0][0]
        para todo combination ∈ combinations hacer
            si value ∈ combination[1] entonces
                position ← combination[0].posicion(value)
                sngram ← combination
                sngram[0].sacar(position)
                sngram[0].agregar([position, candidate[0]])
                sngram[1].sacar(value)
                sngram[2] ← longitud(prepare_SNgram(sngram[0], sentence, −1))
                si (min_size > 0) y (max_size > 0) entonces
                    si sngram[2] ∈ [min_size, self.max_size + 1] entonces
                        DepNgrams.agregar(sngram[0])
                    fin si
                    si sngram[2] < max_size entonces
                        si sngram[0][0] == root_idx entonces
                            si longitud(sngram[1]) > 0 entonces
                                combinations.agregar((sngram))
                            fin si
                        si no
                            si longitud(sngram[1]) > 0 entonces
```

Algoritmo 6 Función *Obtensubarboles (continuación)*

```
                                    combinations.agregar((sngram))
                                fin si
                                candidates.agregar((sngram))
                            fin si
                        fin si
                    si no
                        DepNgrams.agregar((sngram[0]))
                        si sngram[0][0] == root_idx entonces
                            si longitud(sngram[1]) > 0 entonces
                                combinations.agregar((sngram))
                            fin si
                        si no
                            si longitud(sngram[1]) > 0 entonces
                                combinations.agregar((sngram))
                            fin si
                            candidates.agregar(sngram)
                        fin si
                    fin si
                fin si
            fin para
        fin mientras
    fin función
```

Algoritmo 7 Función ObtenDepngrams

```
1: función OBTENDEPNGRAMS(algo)
2:     Variables: unigrams [ ], combinations [ ], aux [ ], log
3:     si sentence.root_idx > 0 entonces
4:         unigrams, combinations, log ← get_subtrees (sentence)
5:         si len (unigrams) > 0 entonces
6:             DepNgrams.agregar ([sentence.root_idx])
7:             DepNgrams.agregar (unigrams)
8:         fin si
9:         para item en combinations hacer
10:            si self.min_size! = 0 o self.max_size! = 0 entonces
11:                size ← longitud (self.prepare_SNgram (item [0], sentence, −1))
12:                si size >= self.min_size y size <= self.max_size entonces
13:                    DepNgrams.agregar (item [0])
14:                fin si
15:                si size < self.max_size entonces
16:                    aux.agregar (item)
17:                fin si
18:            si no:
19:                DepNgrams.agregar (copy.deepcopy (item [0]))
20:            fin si
21:        fin para
22:        si min_size! = 0 o max_size! = 0 entonces
23:            compound_sngrams (aux, sentence)
24:        si no
25:            compound_sngrams (combinations, sentence)
26:        fin si
27:    si no
28:        imprimir: raíz no encontrada
29:    fin si
30:    regresar DepNgrams
31: fin función
```

Se construye una representación vectorial de cada texto, en la que cada dimensión corresponde a un n-grama sintáctico y cada entrada corresponde a la frecuencia de ocurrencia de ese n-grama en un texto en particular, como se muestra en la figura. 5.7

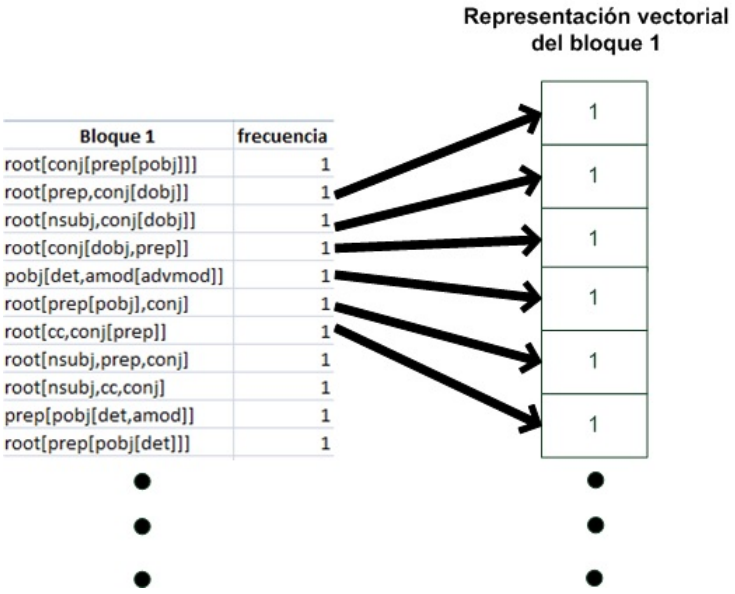


Figura 5.7: *Representación vectorial de un texto*

Capítulo 6

Pruebas y resultados

6.1. Descripción de los corpus de prueba

Aunado a los adelantos en las investigaciones referentes a la tarea de atribución de autoría, diferentes conjuntos de datos se han compilado para probar el rendimiento de las propuestas. Los primeros trabajos se centraron en problemas en el dominio de las humanidades, comúnmente relacionados con la asignación de los documentos controvertidos [Mosteller & Wallace 1963, Matthews & Merriam 1993*b*] o revelando autores anónimos [Holmes 1998]. Los conjuntos de datos utilizados consistieron en documentos históricos o documentos literarios con gran longitud (más de 1000 palabras) y con un número reducido de autores candidatos (no más de tres autores).

En los últimos años surgió un interés en el conjunto de datos más controlado, el control está en torno a tres aspectos: el género de los documentos, los temas contenidos en los documentos y la longitud de los documentos. Los corpus más recientes compilados especialmente para la tarea atribución de autoría ahora abarcan diferentes géneros de texto y diferentes temas, también los documento utilizado se recolectan de periódico o de los medios sociales como blogs o Twitter, lo que implica una reducción significativa en la longitud de los documentos y la inclusión del lenguaje social de los medios de comunicación (hashtags, emoticonos, argot). Los conjuntos de datos controlados representan escenarios más

adecuados para el desarrollo de métodos dentro de la atribución de autoría en contraste con los conjuntos de datos anteriores y han surgido diferentes intentos de construir un punto de referencia unificado para probar y comparar dichos métodos [Juola 2004, Argamon & Juola 2011, Juola 2012].

Se recogieron cinco diferentes corpus relacionados con el problema de atribución de autoría cerrada para evaluar nuestra propuesta. Los datos recogidos se centra en el idioma Inglés y abarca diferentes posibilidades de los siguientes aspectos: el número de autores conocidos, el tamaño de los datos de entrenamiento, diferentes géneros (novelas, opiniones, noticias) y diferentes temas.

La iniciativa de evaluación PAN [1] es un foro importante para los avances en la detección de plagio, la atribución de autoría y el mal uso de software social, donde se compilan corpus desafiantes sobre estos temas y se les pide a los participantes que presenten nuevas metodologías para resolverlos. El PAN 2012 fue la última edición que incluía la tarea atribución de autoría cerrado como parte de su sección de atribución de autoría, en esta edición se presentan tres corpus balanceados para la tarea atribución de autoría con el nombre de Problema A, Problema C y Problema I; los corpus se conformaron por fragmentos de novelas escritos por autores ingleses. Los tres corpus fueron utilizados para evaluar la propuesta.

Para el problema A, se conocen tres autores y los datos de entrenamiento se componen de dos ejemplos para cada uno de ellos, mientras que los datos de prueba constan de dos ejemplos de cada autor para ser clasificado (seis ejemplos en total), la longitud de los ejemplos fueron entre 1800 y 6060 palabras; para el problema C, el número de autores conocidos aumentan a ocho con dos ejemplos por autor, mientras que los datos de prueba se reducen a un ejemplo de cada autor para ser clasificado, la longitud de los ejemplos aumentan de tamaño aproximadamente a 13000 palabras; finalmente para el problema I consisten en catorce autores conocidos con dos ejemplos de cada uno como datos de entrenamiento, mientras que los datos de prueba incluyen un ejemplo para cada autor, pero las muestras corresponden a completar novelas o novela corta con una longitud que varía de 40000 a 170000 palabras [Juola 2012]. En otras secciones vamos a hacer referencia al índice de referencia del Problema A como PAN A, al Problema C como PAN C y al Problema I como PAN I. En la tabla 6.1 se muestran las características de los corpus utilizados en el PAN 2012.

[1] Véase `http://www.uni-weimar.de/medien/webis/events/pan-15/pan15-web/`

Tabla 6.1: PAN 2012 corpus para la tarea de atribución de autoría cerrada

	PAN A	**PAN C**	**PAN I**
Número de autores conocidos	3	8	14
Número de ejemplos por autor	2	2	2
Número de textos de prueba por autor	2	1	1
Tamaño de los ejemplos (en palabras)	$1,800$ a $6,060$	a lo más 13,000	40,000 a 170,000

El conjunto de corpus del PAN 2012 permiten explorar tres cuestiones discutidas en el marco teórico: en primer lugar, se sabe que con un mayor número de autores de la tarea se hace más difícil, en este sentido, los corpus permiten explorar con tres tamaños diferentes de autores conocidos a partir de un caso simple con sólo tres autores a un escenario más desafiante con catorce autores; en segundo lugar, los corpus proponen diferentes longitudes para los textos utilizados lo que permite evaluar diferentes posibilidades acerca de la cantidad necesaria de datos para crear un modelo fiable para el estilo del autor; en tercer lugar, los corpus ofrecen un escenario con diferentes temas ya que los textos corresponden a novelas.

Otro corpus utilizado para probar la propuesta fue el corpus *The Reuters Corpus Volume 1* (RCV1) [Lewis et al. 2004], que consiste en una colección de noticias escritas en inglés que cubren cuatro tópicos principales: corporativo/industrial (CCAT), economía (ECAT), gobierno/social (GCAT) y mercados (MCAT). Aunque el corpus no fue compilado inicialmente para la tarea de atribución de autoría, éste ha sido adaptado para la tarea por trabajos previos en atribución de autoría, por ejemplo, en [Stamatatos 2008, Plakias & Stamatatos 2008] los 10 autores con mayor cantidad de artículos escritos fueron seleccionados de la categoría CCAT y 50 ejemplos de cada autor se escogieron para conformar el conjunto de entrenamiento, de igual manera 50 ejemplos de cada autor se seleccionaron para formar el conjunto de prueba, procurando que no existiera solapamiento entre los conjuntos de entrenamiento y prueba. En las siguientes secciones se hará referencia a este corpus como RCV1- 10.

Otra adaptación del corpus RCV1 para la tarea atribución de autoría fue presentado en [Houvardas & Stamatatos 2006], a diferencia de la última actualización, en esta ocasión se eligieron los 50 autores más prolíficos, manteniendo 50 ejemplos por el autor para la formación y 50 ejemplos por autor para las pruebas sin solapamiento entre ellos. Nosotros haremos referencia a este corpus como RCV1- 50.

El corpus RCV1-10 y el corpus RCV1-50 son ambos balanceados entre los diferentes autores y han fijado su género a las noticias, la temática principal de las noticias en ambos casos se fija a lo corporativo/industrial (CCAT) sin embargo, se abarcan diferentes subtemas en las noticias y la longitud de los textos es corto (de 2 KB hasta 8 Kb). Estos corpus asemejan a un escenario más realista donde la cantidad de textos es limitada y el número de autores candidatos es grande.

Finalmente, se presenta otro corpus para la tarea de atribución de autoría que fue presentado en el trabajo [Stamatatos 2013], que consiste en una colección de artículos publicados en el periódico *The Guardian* en un periodo desde 1999 hasta 2009. Los artículos fueron escritos por 13 autores y se agrupan en cinco categorías temáticas (política, sociedad, mundo, del Reino Unido y reseñas de libros) descartando aquellos artículos cuyo contenido abarca más de una categoría, es decir, cada artículo sólo pertenece a una categoría.

Los artículos fueron descargados desde el repositorio en línea del periódico y preprocesados para obtener versiones de texto sin formato de los artículos. El número de ejemplos de cada autor sobre las diferentes categorías no es homogéneo, este número corresponde a la producción de cada autor a lo largo del periodo antes mencionado. La distribución completa del corpus se presenta en el trabajo [Stamatatos 2013].

Los experimentos utilizando el corpus de *The Guardia* se realizaron siguiendo el procedimiento presentado en el trabajo [Sapkota et al. 2015]: elegimos máximo diez documentos por autor para cada una de las cinco categorías, la nueva distribución se muestra en la tabla 6.2.

El corpus *The Guardian* ofrece la oportunidad de explorar un escenario con diferentes temas en el mismo género, con la excepción de la categoría de reseñas de libros que se considera como otro género y se asume que cada categoría representa un tema que es lo suficientemente diferente de las otras categorías. En contraste con los corpus descritos anteriormente, *El Guardian* ofrece el escenario más desafiante al ser un corpus desbalanceado, que incluye dos diferentes géneros de textos y diferentes temáticas.

Tabla 6.2: Distribución del corpus *The Guardian*

Autor	Política	Sociedad	Mundo	Reino Unido	reseñas de libros
CB	10	4	10	10	10
GM	6	3	10	3	0
HY	8	6	10	5	3
JF	9	1	10	10	2
MK	7	0	10	3	2
MR	8	10	10	10	4
NC	10	2	9	7	5
PP	10	1	10	10	10
PT	10	10	10	5	4
RH	10	4	3	10	10
SH	10	5	5	6	2
WH	10	6	10	5	7
ZW	4	10	10	6	4
Totals	112	62	117	90	63

6.2. Descripción de experimentos

En esta sección se describen los experimentos realizados para resolver la tarea de atribución de autoría cerrada sobre los diferentes corpus compilados para dicha tarea. Se analizó por separado la eficiencia que obtienen los dos diferentes métodos propuestos para representar el estilo de un autor (uso de n-gramas sintácticos y representación distribuidas) con la finalidad de identificar el método que modela mejor el estilo de un autor.

En los experimentos se utilizó el procedimiento descrito en el capítulo 5 basado en un enfoque de aprendizaje automático. La librería *Scikit-learn*[2][Pedregosa et al. 2011] que contiene distinto módulo de aprendizaje automático supervisado fue utilizada para implementar los clasificadores y evaluar la eficiencia que obtienen las representaciones en los distintos corpus.

[2]Descargar en `http://scikit-learn.org/stable/`

6.2.1. Representación distribuida de documentos

Se llevaron a cabo experimentos en los diferentes corpus descritos en la sección anterior. Debido a que se utiliza un enfoque de aprendizaje automático, los experimentos consisten en el entrenamiento de un clasificador con el conjunto de datos correspondiente utilizando una representación distribuida de los documentos, posteriormente se prueba el clasificador con el conjunto de prueba y se reporta la eficiencia obtenida.

El modelo doc2vec [Le & Mikolov 2014*b*] está implementado en el módulo de GENSIM [3] que se puede descargar de manera gratuita libremente. La implementación del modelo doc2vec requiere de tres parámetros: el número de características que serán devueltas, el tamaño de la ventana que capturan el contexto y la frecuencia mínima de las palabras que se consideran en el modelo. El valor del parámetro puede depender del conjunto de datos utilizados y debido a que el estado del arte no arrojó ningún trabajo similar al momento de desarrollar la propuesta, se realizó una búsqueda dentro de intervalos fijos propuestos.

Los experimentos se realizaron con diferentes clasificadores tradicionales como son las máquinas de soporte vectorial (SVM), regresión lineal (LR) y vecinos más cercanos (NN). Los mejores resultados se lograron utilizando la librería LibSVM [Chang & Lin 2011] y sólo el resultado obtenido por este clasificador se reporta a continuación.

Se llevaron a cabo experimentos para determinar la eficiencia de los formatos de entrada propuestos (palabras y n-gramas de palabras) utilizando un representación distribuida. Se probaron los diferentes tamaños propuestos para los n-gramas (2, 3, 4) y combinaciones con el formato de palabras, para cada una de las representaciones propuestas se realizó una búsqueda para los parámetros del método doc2vec para los siguientes rangos: número de características $[50, 500]$, tamaño de ventana $[2, 40]$ y frecuencia mínima $[2, 40]$. Los experimentos se realizaron solamente en el corpus de *The Guardian* porque ofrece un escenario con diversidad tanto en temático como en género, en la tabla 6.3 se muestra la mejor eficiencia obtenida cuando se seleccionan los textos referentes política como conjunto de entrenamiento y los textos referente al mundo como conjunto de prueba utilizando como clasificado una máquina de soporte vectorial (SVM), así como el valor de los parámetros utilizados en el mismo orden en que fueron mencionados previamente.

[3] Veáse `https://radimrehurek.com/gensim/`

Tabla 6.3: Eficiencia de los distinto formatos de entrada para el corpus *The Guardian* considerando *Politics* como el conjunto de entrenamiento y *World* como conjunto de prueba

Formato	**Eficiencia**	**Parámetros**
word	86.32	$[100, 7, 5]$
bigram	83.76	$[50, 13, 10]$
trigram	41.12	$[250, 8, 2]$
fourgram	20.56	$[50, 4, 2]$
word + bigram	**90.59**	$[50, 40, 5]$
word + trigram	89.71	$[350, 10, 4]$
word + fourgram	88.78	$[50, 18, 5]$

Se observa de la tabla 6.3 que la combinación de las representaciones que utilizan el formato de palabras y bigramas de palabras obtiene la mejor eficiencia para los textos Politics-World, al realizar experimentos con las demás combinaciones de temas del corpus *The Guardian* se observa que el uso de las representaciones de palabras y bigramas de palabras obtienen los mejores resultados, seguidos de la combinación de palabras y trigramas de palabras que solamente en una combinación (Politics-Society) igualó los resultados obtenidos por la combinación que incluye bigramas de palabras. En las siguientes secciones se muestran experimentos realizados solamente con palabras y bigramas de palabras, otras combinaciones de formatos como por ejemplo la combinación de palabras con bigramas y trigramas se plantean como trabajo futuro.

En la tabla 6.4 se muestran los resultados de los experimentos realizados con el corpus PAN. Las tres primeras filas corresponden a las diferentes representaciones propuesto para el modelo doc2vec y las filas restantes corresponden a los cinco mejores resultados obtenidos en el evento. En la tabla aparece el nombre del equipo utilizado en la evaluación, los detalles se pueden obtener a partir de la información general del evento [Juola 2012]. Los parámetros utilizados en las representaciones distribuidas fueron: D2V words $[50, 3, 6]$, D2V bigrams $[50, 5, 5]$ y D2V words+bigrams $[50, 17, 10]$.

Experimento con el corpus RCV1-10 y RCV1-50 se realizaron utilizando las tres representaciones propuestas para el modelo doc2vec, al igual que los corpus del PAN 2012 los corpus RCV1-10 y RCV-50 fueron divididos en entrenamiento y prueba. Para todos los autores del corpus se utilizaron 50 textos por autor en fase de entrenamiento y 50 textos por autor en fase de pruebas.

Tabla 6.4: Eficiencia al usar representación distribuida para los corpus del PAN

Nombre	**PAN A**	**PAN C**	**PAN I**
D2V words	**100**	**100**	**92.85**
D2V bigrams	**100**	**100**	**92.85**
D2V words+bigrams	**100**	**100**	**92.85**
Brainsignals	**100**	**100**	**92.85**
Sapkota	**100**	**100**	**92.85**
Lip6 1	100	100	85.71
EVL Lab	100	87.50	85.71
de Graaff 1	100	87.50	71.42
Bar I U	100	75.00	85.71
Lip 6 2	100	75.00	**85.71**
Lip 6 3	100	62.50	78.57
CLLE-ERSS 3	100	37.50	92.85
CLLE-ERSS 4	100	37.50	92.85
CLLE-ERSS 1	100	35.00	85.71
Zech terms	83.33	62.50	64.28
Vilarino 2	83.33	62.5	57.14
Ruseti	66.66	75.00	85.71
CLLE-ERSS 2	66.66	25.00	85.71
Vilarino 1	66.66	25.00	71.42
Zech stats	66.66	25.00	35.71
Surrey	66.66	12.5	50.00
de Graaff 2	50.00	50.00	50
Zech stylo	33.33	25.00	35.71

Un clasificador LibSVM fue entrenado para realizar la atribución de los textos de prueba. La eficiencia obtenida con la representación distribuida doc2vec y la obtenida por trabajos anteriores (histogramas locales de carácter n-gramas [Escalante et al. 2011], modelos tensoriales [Plakias & Stamatatos 2008], n-gramas de caracter y palabra [Stamatatos 2008], n-gramas de caracter filtrados [Sapkota et al. 2015] y la función de selección de n-gramas [Houvardas & Stamatatos 2006]) se muestran en la tabla 6.5. Los parámetros utilizados en las representaciones distribuidas para el corpus RCV1-10 fueron: D2V words $[50, 10, 4]$, D2V bigrams $[300, 25, 30]$ y D2V words+bigrams $[100, 21, 3]$; para el corpus RCV1-50 fueron: D2V words $[250, 21, 3]$, D2V bigrams $[150, 28, 3]$ y D2V words+bigrams $[400, 25, 3]$

Tabla 6.5: Results for RCV1 corpora

Name	**RCV1-10**	**RCV1-50**
D2V words	81.92	74.60
D2V bigrams	84.60	75.40
D2V words+bigrams	86.00	**77.80**
Local histograms of character n-grams	**86.40**	–
Tensor space models	80.80	–
Character and word n-grams	79.40	–
Filtered character n-grams	78.80	69.30
N-gram feature selection	–	74.04

Para el corpus *The Guardian* se utiliza el mismo esquema de la prueba establecida en el trabajo [Stamatatos 2013]:

- **Fase de entrenamiento:** a partir de las cinco categorías diferentes en *The Guardian* corpus (política, sociedad, Reino Unido, mundial y reseñas de libros), la categoría Política se selecciona como el conjunto de entrenamiento usando como máximo diez textos por autor.
- **Fase de prueba:** el clasificador es entrenado sobre el resto de las diferentes categorías, teniendo un total de cuatro pares (política-sociedad, política-Reino Unido, política-mundo, política-reseña de libros) y a lo más 10 textos por autor son seleccionados.

En la tabla 6.6 se muestran los resultados obtenidos para cada par de categorías junto con el número total de textos utilizados en cada categoría y los resultados obtenidos en trabajos anteriores. Los parámetros utilizados en las representaciones distribuidas para cada conjunto de prueba fueron: Society (D2V words $[100, 3, 30]$, D2V bigrams $[100, 18, 10]$ y D2V words+bigrams $[100, 33, 5]$), UK(D2V words $[200, 2, 30]$, D2V bigrams $[250, 5, 10]$ y D2V words+bigrams $[50, 16, 3]$), World(D2V words $[100, 7, 5]$, D2V bigrams $[50, 13, 10]$ y D2V words+bigrams $[50, 40, 5]$) y Books reviews (D2V words $[150, 2, 30]$, D2V bigrams $[150, 2, 10]$ y D2V words+bigrams $[50, 26, 2]$).

Tabla 6.6: Eficiencia para el corpus *The Guardian* considerando política como el conjunto de entrenamiento

Name	**Society**	**UK**	**World**	**Books reviews**
D2V words	**96.77**	90.00	86.32	80.95
D2V bigrams	93.54	86.66	83.76	80.95
D2V words+bigrams	**96.77**	**92.22**	**90.59**	**84.12**
Char 3-grams	$\simeq$ 91.00	$\simeq$ 88.00	$\simeq$ 82.50	$\simeq$ 79.50

De las tablas 6.4,6.5 y 6.6 se observa que la representación distribuida entrega mejores resultados que los reportados en el estado del arte inclusive en escenarios no homogéneos en los que se cuenta con textos de distintos géneros, se abarcan distintos temas y con un tamaño reducido (cientos de palabras).

6.2.2. N-gramas sintácticos

Se realizaron experimentos utilizando los n-gramas sintácticos como marcadores de estilo solamente utilizando los corpus propuestos en el PAN 2012 (PAN A, PAN C y PAN I). El procedimiento descrito en la sección 5.2 fue implementado para la obtención de las distintas variantes de n-gramas sintácticos, considerando los siguientes parámetros: el tamaño de los n-gramas se fijó a un valor entre 3 y 7 porque son los tamaños que mejores resultados han reportado para el caso de n-gramas de caracteres y palabras considerados como línea base; los clasificadores Naive Bayes con distribución multinomial (MNB), máquinas de soporte vectorial (SVM) y k-vecinos más cercanos (NN) fueron utilizados para realizar la clasificación debido a su capacidad para trabajar con información dispersa y vectores con un alto

número de dimensiones; para cada tipo de n-grama sintáctico se construyó el espacio vectorial utilizando todos los n-gramas de tamaño en el rango de 3 a 7 y se aplicó una estrategia de selección de características basada en la ganancia de información (en inglés, *information gain*).

Se realizaron experimentos utilizando los n-gramas sintácticos incluyendo los mixtos pero sin tomar en cuenta los n-gramas con filtrado. A continuación se muestran los tipos que obtuvieron el mejor desempeño en los diferentes corpus de prueba.

En la tabla 6.7 se muestra la eficiencia obtenida por los n-gramas de etiquetas de relaciones de dependencia para el corpus PAN A.

Tabla 6.7: Eficiencia para el corpus PAN A utilizando SR n-grams

Umbral	**# características**	**NBM (%)**	**SVM (%)**	**1-NN (%)**
>= 0.00	5315	33.33	33.33	33.33
>= 0.38	2745	66.66	66.66	83.33
>= 0.43	2423	66.66	33.33	**100**
>= 0.63	604	66.66	33.33	83.33
>= 0.91	257	83.33	66.66	33.33

En la tabla 6.8 se muestra la eficiencia obtenida por los n-gramas de Word+POS sn-gramas para el corpus PAN A.

Tabla 6.8: Eficiencia para el corpus PAN A utilizando Word + POS sn-gramas

Umbral	**# características**	**NBM (%)**	**SVM (%)**	**1-NN (%)**
>= 0.00	67115	33.33	50.00	33.33
>= 0.3899	25927	83.33	**100**	66.66
>= 0.4354	25768	83.33	**100**	66.66
>= 0.6383	798	**100**	**100**	**100**
>= 0.9182	469	**100**	**100**	83.33

En la tabla 6.9 se muestran la eficiencia obtenida por todas las variantes de n-gramas sintácticosd y los diferentes clasificadores.

En la tabla 6.10 se muestra la eficiencia obtenida por los n-gramas de etiquetas de relaciones de dependencia para el corpus PAN C y en la tabla 6.11 se muestra la eficiencia obtenida utilizando Word + POS sn-gramas.

Tabla 6.9: Eficiencia para el corpus PAN A utilizando n-gramas sintácticos

Tipo n-grama	**NBM (%)**	**SVM (%)**	**1-NN (%)**
SR sn-grama	66.66	33.33	**100**
POS sn-grama	**100**	**100**	**100**
Word sn-grama	33.33	50.00	33.33
Lema sn-grama	33.33	33.33	33.33
SR + POS sn-grama	**100**	**100**	**100**
SR + Word sn-grama	83.33	83.33	83.33
SR + Lema sn-grama	66.66	66.66	50.00
POS + SR sn-grama	**100**	**100**	83.33
POS + Word sn-grama	66.66	83.33	83.33
POS + Lema sn-grama	66.66	66.66	50.00
Word + POS sn-grama	83.33	83.33	66.66
Word + SR sn-grama	66.66	83.33	66.66
Word + Lema sn-grama	66.66	83.33	83.33
Lema + SR sn-grama	66.66	66.66	66.66
Lema + POS sn-grama	33.33	50.00	33.33
Lema + Word sn-grama	33.33	33.33	33.33

Tabla 6.10: Eficiencia para el corpus PAN C utilizando SR sn-grams

Umbral	**# características**	**NBM (%)**	**SVM (%)**	**1-NN (%)**
>= 0.00	4366	37.50	50.00	37.50
>= 0.41	1947	**75.00**	50.00	37.50
>= 0.43	1345	37.50	37.50	37.50
>= 0.47	810	37.50	50.00	50.00
>= 0.54	331	25.00	37.50	37.50

Tabla 6.11: Eficiencia para el corpus PAN C utilizando Word + POS sn-grams

Umbral	**# características**	**NBM (%)**	**SVM (%)**	**1-NN (%)**
>= 0.00	5512	37.50	37.50	37.50
>= 0.41	2512	**87.50**	50.00	62.50
>= 0.43	2081	62.50	37.50	50.00
>= 0.47	1530	37.50	37.50	50.00
>= 0.54	860	62.50	62.50	37.50

En la tabla 6.12 se muestran la eficiencia obtenida por todas las variantes de n-gramas sintácticos y los diferentes clasificadores.

Tabla 6.12: Eficiencia para el corpus PAN A utilizando n-gramas sintácticos

Tipo n-grama	**NBM (%)**	**SVM (%)**	**1-NN (%)**
SR sn-grama	75.00	50.00	37.50
POS sn-grama	37.50	37.50	37.50
Word sn-grama	37.53	50.00	37.53
Lema sn-grama	12.5	25.00	12.50
SR + POS sn-grama	62.50	75.00	50.00
SR + Word sn-grama	62.50	50.00	25.00
SR + Lema sn-grama	37.50	37.50	12.50
POS + SR sn-grama	62.50	75.00	50.00
POS + Word sn-grama	62.50	62.50	25.00
POS + Lema sn-grama	37.50	37.50	25.00
Word + POS sn-grama	**87.50**	50.00	62.50
Word + SR sn-grama	62.50	37.50	37.50
Word + Lema sn-grama	12.50	25.00	12.50
Lema + SR sn-grama	62.50	50.00	50.00
Lema + POS sn-grama	37.50	50.00	37.50
Lema + Word sn-grama	25.00	25.00	25.00

De las tabla 6.7, 6.8, 6.10 y 6.11 se observa que el uso de estrategias para la selección de características basadas en la informatividad son útiles para reducir el ruido generado por los n-gramas sintácticos.

Se observa de la tabla 6.9 y de la tabla 6.12 que los n-gramas sintácticos que utilizan los lemas o las palabras en su conformación tienen una eficiencia menor en comparación con el uso de la etiquetas de dependencia (SR) y las etiquetas (POS).

Para los experimentos en el corpus PAN I se consideraron solamente los n-grmas SR + POS, Word + POS, POS + SR, SR y POS debido a que mostraron una mejor eficiencia respecto al resto de los n-gramas sintácticos. En la tabla 6.13 se presenta un comparativo de los mejores resultados obtenidos utilizando n-gramas sintácticos y los equipos con las más altas puntuaciones de la edición PAN 2012.

Los métodos presentados en la tabla 6.13 que participaron en la evaluación del PAN 2012 emplearon además de n-gramas de caracteres y palabras, marcadores de estilo relacionados con el léxico utilizado (análisis de los tiempos de conjugación de verbos y tamaño del léxico utilizado); como métodos de clasificación utilizaron en máquinas de soporte vectorial (SVM) y redes neuronales. Al comparar la eficiencia obtenida al usar n-gramas sintácticos contra

Tabla 6.13: Comparativo eficiencias PAN A, PAN C y PAN I

Nombre	**PAN A (%)**	**PAN C (%)**	**PAN I (%)**
Brainsignals	**100**	**100**	**98.85**
Sapkota	100	100	92.85
Lip6 1	100	100	85.71
EVL Lab	100	87.50	85.71
Método propuesto	100	87.50	92.85

marcadores a nivel de caracteres o léxicos se observa que los n-gramas sintácticos no superan a los métodos propuesto basados en marcadores a nivel de caracteres o palabras, sin embargo, el método reporta resultados aceptables, especialmente en escenarios donde se cuenta mayor cantidad de información como el corpus PAN I.

Capítulo 7

Conclusiones

La estrategia que se implementó basada en representaciones distribuidas permite obtener modelos que explotan la información que se puede obtener del contexto de una palabra y generar marcadores de estilo basados en la probabilidad de ocurrencia de una palabra a partir de su contexto. Los experimentos realizados sobre diferentes corpus muestran que el uso de una representación distribuida a nivel de documentos es una opción robusta para modelar el estilo de un autor, la eficiencia que se obtuvo mejora a la reportada en el estado del arte o iguala los resultados obtenidos. Dos formatos de entrada diferentes se utilizaron para construir el modelo distribuido: palabras y bigramas de palabras. Los experimentos muestran que para la construcción de modelos distribuidos el combinar los formatos de entrada propuestos (palabras y bigramas) usando los dos enfoques planteados en el método doc2vec (DM y DBOW) es mejor que usar solamente uno de ellos.

En los trabajos previos no se reporta una metodología para encontrar los valores óptimos de los parámetros que requiere el método doc2vec para construir un modelo, de los experimentos realizados se observa que un tamaño de ventana menor a 20 y una frecuencia mínima de ocurrencia menor a 10 permiten obtener una buena eficiencia.

El algoritmo para la obtención de n-gramas sintácticos de árboles de dependencia (homogéneos y mixtos) ofrece marcadores de estilo basados en el uso de información sintáctica desde un punto de vista detallado ofreciendo una alternativa robusta al problema de atribución de autoría en comparación con el uso de n-gramas de caracteres debido a que muestra un buen desempeño con un espacio de características menor.

Al obtener su frecuencia de uso en los diferentes corpus de prueba se construyó una representación vectorial sobre el estilo de autores. Los mejores resultados se obtienen con los n-gramas que utilizan etiquetas de relaciones de dependencia y etiquetas POS en sus versiones de homogéneas y mixtas.

Los resultados obtenidos para el resto de conjuntos de datos también buenos resultados alcanzados, en el PAN / CLEF conjuntos de datos de nuestra propuesta es igual a los resultados obtenidos por los mejores equipos en el taller (ver tabla 6.13), en el RCV1-50 nuestra propuesta supera a los resultados obtenidos por trabajos anteriores (véase la tabla 6.5) pero en el RCV1-10 del resultado obtenido en el mejor resultado, al utilizar el corpus completo nuestra propuesta era $0,4\,\%$ abajo (ver tabla 6.5) y al usar la versión desequilibrada de RCV1-10 conjunto de datos los resultados fueron inferiores a los reportados por trabajos anteriores (véase la tabla 6.5), esto puede ser explicado por la pequeña cantidad de datos de entrenamiento que muestran que la representación doc2vec es sensible a la cantidad de datos utilizados para la construcción del modelo.

En general, el uso de la representación doc2vec ofrece una manera robusta para representar el estilo de un autor inclusive en escenarios considerados difíciles (más de 10 autores, textos con cientos de palabras conjuntos de datos no homogéneos, con textos cortos y con varios pos no homogéneos) y mejora los resultados que obtienen con los métodos de n-gramas tradicionales y n-gramas sintácticos.

7.1. Aportaciones científicas

Las aportaciones científicas de la presente tesis son:

- Diseño de un método para la atribución de autoría que utilice una representación distribuida a nivel documentos en la que se combinan los modelos obtenidos al usar distintos formatos de entrada.
- Evaluación de la eficiencia de formatos de entrada basados en n-gramas de palabras de tamaño 2, 3 y 4.
- Diseño de un método para la atribución de autoría que utilice como marcadores de estilo n-gramas sintácticos.

- Diseño de una representación del estilo de un autor basado en el uso de las frecuencias de ocurrencia de n-gramas sintácticos.
- Demostración de la eficiencia que el uso de n-gramas sintácticos tienen para modelar el estilo de un autor.
- Diseño e implementación de un algoritmo para la obtención de n-gramas sintácticos (homogéneos y mixtos) en lenguaje Python versión 2.7 disponible en la dirección `http://www.cic.ipn.mx/~sidorov/MultiSNGrams_2.py`
- Compilación de un corpus sobre la atribución de autoría con información sintáctica.

7.2. Trabajo futuro

Como trabajo futuro se propone lo siguiente:

- Evaluar la eficiencia de modelos que combinen más de tres formatos de entrada basados en n-gramas de palabras.
- Proponer un método para la selección de parámetros para la obtención de representaciones distribuidas.
- Diseñar y evaluar un método que combine la estrategia basada en n-gramas sintácticos con la estrategia de representación distribuida para modelar el estilo de escritura.
- Evaluar la eficiencia de las representaciones basadas en n-gramas sintácticos y basadas en representaciones distribuidas para la atribución de autoría abierta (ninguno de los autores candidatos escribió el texto).
- Evaluar la eficiencia de la combinación de las diferentes variantes de n-gramas sintácticos y n-gramas de caracteres o palabras.
- Evaluar el desempeño de una segunda etapa de selección de características basada en la frecuencia de ocurrencia para descartar n-gramas aleatorios.

Bibliografía

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

Akiva, N. [2011], Using clustering to identify outlier chunks of text - notebook for PAN at CLEF 2011, *in* 'CLEF (Notebook Papers/Labs/Workshop)'.

Akiva, N. [2012], Authorship and plagiarism detection using binary bow features., *in* 'CLEF (Online Working Notes/Labs/Workshop)'.

Alzahrani, S., Salim, N. & Abraham, A. [2012], 'Understanding plagiarism linguistic patterns, textual features, and detection methods', *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **42**(2), 133 –149.

Argamon, S. & Juola, P. [2011], Overview of the international authorship identification competition at PAN 2011, *in* 'CLEF (Notebook Papers/Labs/Workshop)'.

Baayen, H., van Halteren, H., Neijt, A. & Tweedie, F. [2002], An experiment in authorship attribution, *in* '6th JADT', Citeseer, pp. 29–37.

Barrón-Cedeño, A., Rosso, P., Agirre, E. & Labaka, G. [2010], Plagiarism detection across distant language pairs, *in* 'Proceedings of the 23rd International Conference on Computational Linguistics', COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 37–45.

Barrón-Cedeno, A., Vila, M. & Rosso, P. [2010], 'Detección automática de plagio: de la copia exacta a la paráfrasis', *Garayzábal Heinze et al.(2010)* pp. 76–96.

Beristáin, H. [2007], Gramática estructural de la lengua española, Technical report, Universidad Nacional de Mexico, Mexico (Mexico).

Bhavani, M., Thammi Reddy, K. & Shashi, M. [2009], A rough set based approach to detect plagiarism, *in* 'TENCON 2009 - 2009 IEEE Region 10 Conference', pp. 1 –8.

Bozkurt, I., Baghoglu, O. & Uyar, E. [2007], Authorship attribution, *in* 'Computer and information sciences, 2007. iscis 2007. 22nd international symposium on', IEEE, pp. 1–5.

Brin, S., Davis, J. & García-Molina, H. [1995], 'Copy detection mechanisms for digital documents', *SIGMOD Rec.* **24**(2), 398–409.

Ceska, Z., Toman, M. & Jezek, K. [2008], Multilingual plagiarism detection, *in* 'Proceedings of the 13th international conference on Artificial Intelligence: Methodology, Systems, and Applications', AIMSA '08, Springer-Verlag, Berlin, Heidelberg, pp. 83–92.

Chang, C.-C. & Lin, C.-J. [2011], 'LIBSVM: A library for support vector machines', *ACM Transactions on Intelligent Systems and Technology* **2**, 27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chaski, C. [2001], 'Empirical evaluations of language-based author identification techniques', *Forensic Linguistics* **8**, 1–65.

Clement, R. & Sharp, D. [2003], 'Ngram and bayesian classification of documents for topic and authorship', *Literary and linguistic computing* **18**(4), 423–447.

Collberg, C. & Kobourov, S. [2005], 'Self-plagiarism in computer science', *Commun. ACM* **48**(4), 88–94.

Curran, D. [2010], An evolutionary neural network approach to intrinsic plagiarism detection, *in* 'Proceedings of the 20th Irish conference on Artificial intelligence and cognitive science', AICS'09, Springer-Verlag, Berlin, Heidelberg, pp. 33–40.

De Marneffe, M.-C. & Manning, C. D. [2008], Stanford typed dependencies manual, Technical report, Technical report, Stanford University.

de Vel, O., Anderson, A., Corney, M. & Mohay, G. [2001], 'Mining e-mail content for author identification forensics', *SIGMOD Rec.* **30**(4), 55–64.

Diederich, J., Kindermann, J., Leopold, E. & Paass, G. [2003], 'Authorship attribution with support vector machines', *Applied intelligence* **19**(1), 109–123.

Escalante, H. J., Solorio, T. & Montes-y Gómez, M. [2011], Local histograms of character n-grams for authorship attribution, *in* 'Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1', HLT '11, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 288–298.

Estival, D., Gaustad, T., Pham, S. B., Radford, W. & Hutchinson, B. [2007], Author profiling for english emails, *in* 'Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING'07)', pp. 263–272.

Franco-Salvador, M., Rosso, P. & Rangel, F. [2015], Distributed representations of words and documents for discriminating similar languages, *in* 'Proceeding of the RANLP Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects (LT4VarDial)'.

Galicia-Haro, S. & Gelbukh, A. [2007], *Investigaciones en Análisis Sintáctico para el español*, Instituto Politécnico Nacional.

HaCohen-Kerner, Y., Tayeb, A. & Ben-Dror, N. [2010], Detection of simple plagiarism in computer science papers, *in* 'Proceedings of the 23rd International Conference on Computational Linguistics', COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 421–429.

Holmes, D. & Forsyth, R. [1995], 'The federalist revisited: New directions in authorship attribution', *Literary and Linguistic Computing* **10**(2), 111–127.

Holmes, D. I. [1998], 'The evolution of stylometry in humanities scholarship', *Literary and linguistic computing* **13**(3), 111–117.

Holmes, D. I. & Kardos, J. [2003], 'Who was the author? An introduction to stylometry', *Chance* **16**(2), 5–8.

Houvardas, J. & Stamatatos, E. [2006], Stamatatos e.: N-gram feature selection for authorship identification, *in* 'In: 12th International Conference on Artificial Intelligence: Methodology, Systems, Applications', Springer, pp. 77–86.

Juola, P. [2004], Ad-hoc authorship attribution competition, *in* 'Proceedings of the Joint Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing', pp. 175–176.

Juola, P. [2007], Future trends in authorship attribution, *in* P. Craiger & S. Shenoi, eds, 'Advances in Digital Forensics III', Vol. 242 of *IFIP International Federation for Information Processing*, Springer Boston, pp. 119–132.

Juola, P. [2012], An overview of the traditional authorship attribution subtask., *in* 'CLEF (Online Working Notes/Labs/Workshop)'.

Kang, N., Gelbukh, A. F. & Han, S.-Y. [2006], Ppchecker: Plagiarism pattern checker in document copy detection, *in* 'TSD', pp. 661–667.

Kern, R., Klampfl, S. & Zechner, M. [2012], Vote/veto classification, ensemble clustering and sequence classification for author identification., *in* 'CLEF (Online Working Notes/Labs/Workshop)'.

Kešelj, V., Peng, F., Cercone, N. & Thomas, C. [2003*a*], N-gram-based author profiles for authorship attribution, *in* 'Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING', Vol. 3, pp. 255–264.

Keselj, V., Peng, F., Cercone, N. & Thomas, C. [2003*b*], 'N-gram-based author profiles for authorship attribution'.

Kiros, R., Zemel, R. S. & Salakhutdinov, R. R. [2014], A multiplicative model for learning distributed text-based attribute representations, *in* 'Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada', pp. 2348–2356.

Koppel, M. & Schler, J. [2003], Exploiting stylistic idiosyncrasies for authorship attribution, *in* 'Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis', Vol. 69, p. 72.

Koppel, M., Schler, J. & Argamon, S. [2013], 'Authorship attribution: What's easy and what's hard?', *Available at SSRN 2274891* .

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

Kukushkina, O., Polikarpov, A. & Khmelev, D. [2001], 'Using literal and grammatical statistics for authorship attribution', *Problems of Information Transmission* **37**(2), 172–184.

Layton, R., Watters, P. & Dazeley, R. [2010], Authorship attribution for twitter in 140 characters or less, *in* 'Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second', IEEE, pp. 1–8.

Le, Q. V. & Mikolov, T. [2014*a*], Distributed representations of sentences and documents, *in* 'Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014', pp. 1188–1196.

Le, Q. V. & Mikolov, T. [2014*b*], 'Distributed representations of sentences and documents', *arXiv preprint arXiv:1405.4053* .

Lewis, D. D., Yang, Y., Rose, T. G. & Li, F. [2004], 'Rcv1: A new benchmark collection for text categorization research', *J. Mach. Learn. Res.* **5**, 361–397.

Li, R. & Shindo, H. [2015], Distributed document representation for document classification, *in* T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, D. Cheung & H. Motoda, eds, 'Advances in Knowledge Discovery and Data Mining', Vol. 9077 of *Lecture Notes in Computer Science*, Springer International Publishing, pp. 212–225.

Liang, H., Fothergill, R. & Baldwin, T. [2015], 'Rosemerry: A baseline message-level sentiment classification system', *SemEval-2015* p. 551.

Matthews, R. & Merriam, T. [1993*a*], 'Neural computation in stylometry i: An application to the works of shakespeare and fletcher', *Literary and Linguistic Computing* **8**(4), 203–209.

Matthews, R. & Merriam, T. [1993*b*], 'Neural computation in stylometry i: An application to the works of shakespeare and fletcher', *Literary and Linguistic Computing* **8**(4), 203–209.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. [2013], 'Efficient estimation of word representations in vector space', *CoRR* **abs/1301.3781**.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. [2013], Distributed representations of words and phrases and their compositionality, *in* 'Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.', pp. 3111–3119.

Mikolov, T., tau Yih, W. & Zweig, G. [2013], Linguistic regularities in continuous space word representations, *in* 'NAACL HLT, Atlanta, Georgia, June 9, 14', pp. 746–751.

Mnih, A. & Hinton, G. E. [2009], A scalable hierarchical distributed language model, *in* D. Koller, D. Schuurmans, Y. Bengio & L. Bottou, eds, 'Advances in Neural Information Processing Systems 21', Curran Associates, Inc., pp. 1081–1088.

Mosteller, F. & Wallace, D. [1964], 'Inference and disputed authorship: The federalist'.

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.

Oakes, M. [2004], Ant colony optimisation for stylometry: The federalist papers, *in* 'Proceedings of the 5th International Conference on Recent Advances in Soft Computing', pp. 86–91.

Oberreuter, G., L'Huillier, G., Ríos, S. A. & Velásquez, J. D. [2011], Outlier-based approaches for intrinsic and external plagiarism detection, *in* 'Proceedings of the 15th international conference on Knowledge-based and intelligent information and engineering systems - Volume Part II', KES'11, Springer-Verlag, Berlin, Heidelberg, pp. 11–20.

Paredes, R. G., Sanchez, J. A. & Razo, A. [2007], Drawing the line between fair use and plagiarism for digital documents, *in* 'Proceedings of the Eighth Mexican International Conference on Current Trends in Computer Science', ENC '07, IEEE Computer Society, Washington, DC, USA, pp. 113–122.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. [2011], 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830.

Pereira, R. C., Moreira, V. P. & Galante, R. [2010], A new approach for cross-language plagiarism analysis, *in* 'Proceedings of the 2010 international conference on Multilingual and multimodal information access evaluation: cross-language evaluation forum', CLEF'10, Springer-Verlag, Berlin, Heidelberg, pp. 15–26.

Petras, V., Forner, P. & Clough, P., eds [2011], *Intrinsic Plagiarism Detection Using Character Trigram Distance Scores*, Amsterdam, The Netherlands.

Plakias, S. & Stamatatos, E. [2008], Tensor space models for authorship identification, *in* J. Darzentas, G. Vouros, S. Vosinakis & A. Arnellos, eds, 'Artificial Intelligence: Theories, Models and Applications', Vol. 5138 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 239–249.

Posadas-Duran, J.-P., Sidorov, G. & Batyrshin, I. [2014], Complete syntactic n-grams as style markers for authorship attribution, *in* 'Human-Inspired Computing and Its Applications', Springer, pp. 9–17.

Potthast, M., Barrón-Cedeño, A., Stein, B. & Rosso, P. [2011], 'Cross-language plagiarism detection', *Lang. Resour. Eval.* **45**(1), 45–62.

Potthast, M., Eiselt, A., Barrón-Cedeño, A., Stein, B. & Rosso, P. [2011], Overview of the 3rd International Competition on Plagiarism Detection, *in* M. Braschler & D. Harman, eds, 'Notebook Papers of CLEF 2011 LABs and Workshops, 19-22 September, Amsterdam, The Netherlands'.

Potthast, M., Gollub, T., Hagen, M., Graßegger, J., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeño, A., Gupta, P. et al. [2009], 'Overview of the 4th international competition on plagiarism detection'.

Rangel, F., Rosso, P., Potthast, M., Stein, B. & Daelemans, W. [2015], Overview of the 3rd author profiling task at PAN 2015, *in* 'CLEF'.

Rao, S., Gupta, P., Singhal, K. & Majumder, P. [2011], External & intrinsic plagiarism detection: Vsm & discourse markers based approach - notebook for PAN at CLEF 2011, *in* 'CLEF (Notebook Papers/Labs/Workshop)'.

Rhodes, D. [2015], Author attribution with cnns, Technical report, CS224, Stanford University.

Sanchez-Perez, M. A., Sidorov, G. & Gelbukh, A. F. [2014], A winning approach to text alignment for text reuse detection at PAN 2014, *in* 'CLEF (Working Notes)', pp. 1004–1011.

Sánchez-Vega, F., Villaseñor Pineda, L., Montes-Y-Gómez, M. & Rosso, P. [2010], Towards document plagiarism detection based on the relevance and fragmentation of the reused text, *in* 'Proceedings of the 9th Mexican international conference on Advances in artificial intelligence: Part I', MICAI'10, Springer-Verlag, Berlin, Heidelberg, pp. 24–31.

Santorini, B. [1990], 'Part-of-speech tagging guidelines for the penn treebank project (3rd revision)'.

Sapkota, U., Bethard, S., Montes-y Gómez, M. & Solorio, T. [2015], Not all character n-grams are created equal: A study in authorship attribution, *in* 'Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL', pp. 93–102.

Segarra, S., Eisen, M. & Ribeiro, A. [2013], Authorship attribution using function words adjacency networks, *in* 'IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013', pp. 5563–5567.

Sidorov, G. [2013*a*], *Construcción no lineal de n-gramas en la lingüistica computacional*, Sociedad Mexicana de Inteligencia Artificial.

Sidorov, G. [2013*b*], 'Non-continuous syntactic n-grams', *Polibits* pp. 67–75.

Sidorov, G. [2014], 'Should syntactic n-grams contain names of syntactic relations', *International Journal of Computational Linguistics and Applications* **5**(1), 139–158.

Sidorov, G., Gelbukh, A., Gómez-Adorno, H. & Pinto, D. [2014], 'Soft similarity and soft cosine measure: Similarity of features in vector space model', *Computación y Sistemas* **18**(3), 491–504.

Sidorov, G., Velazquez, F., Stamatatos, E., Gelbukh, A. & Chanona-Hernández, L. [2012*a*], 'Syntactic dependency-based n-grams as classification features', *LNAI* pp. 1–11.

Sidorov, G., Velazquez, F., Stamatatos, E., Gelbukh, A. & Chanona-Hernández, L. [2012*b*], 'Syntactic n-grams as machine learning features for natural language processing', *Expert Systems with Applications* pp. 853–860.

Solan, L. M. [2013], 'Intuition versus algorithm: The case of forensic authorship attribution', *Brooklyn Journal of Law and Policy* **21**(551).

Stamatatos, E. [2006], 'Authorship attribution based on feature set subspacing ensembles', *International Journal on Artificial Intelligence Tools* **15**(05), 823–838.

Stamatatos, E. [2008], 'Author identification: Using text sampling to handle the class imbalance problem', *Inf. Process. Manage.* **44**(2), 790–799.

Stamatatos, E. [2009*a*], 'Intrinsic plagiarism detection using character n-gram profiles', *threshold* **2**, 1–500.

Stamatatos, E. [2009*b*], 'A survey of modern authorship attribution methods', *Journal of the American Society for Information Science and Technology* **60**(3), 538–556.

Stamatatos, E. [2011], Plagiarism detection based on structural information, *in* 'Proceedings of the 20th ACM international conference on Information and knowledge management', CIKM '11, ACM, New York, NY, USA, pp. 1221–1230.

Stamatatos, E. [2013], 'On the robustness of authorship attribution based on character n-gram features', *Journal of Law and Policy* **21**(2).

Stamatatos, E., Fakotakis, N. & Kokkinakis, G. [2001], 'Computer-based authorship attribution without lexical measures', *Computers and the Humanities* **35**(2), 193–214.

Stein, B., Lipka, N. & Prettenhofer, P. [2011], 'Intrinsic plagiarism analysis', *Lang. Resour. Eval.* **45**(1), 63–82.

Tanguy, L., Sajous, F., Calderone, B., Hathout, N. et al. [2012], 'Authorship attribution: Using rich linguistic features when training data is scarce.', *Notebook for PAN at CLEF 2012* .

Tweedie, F., Singh, S. & Holmes, D. [1996], 'Neural network applications in stylometry: The federalist papers', *Computers and the Humanities* **30**(1), 1–10.

Uzuner, O., Katz, B. & Nahnsen, T. [2005], Using syntactic information to identify plagiarism, *in* 'Proceedings of the second workshop on Building Educational Applications Using NLP', EdAppsNLP 05, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 37–44.

Wasa, K., Kaneta, Y., Uno, T. & Arimura, H. [2012], Constant time enumeration of bounded-size subtrees in trees and its application, *in* 'Computing and Combinatorics', Springer, pp. 347–359.

Wettschereck, D. [1994], A study of distance-based machine learning algorithms, PhD thesis.

White, D. R. & Joy, M. S. [2004], 'Sentence-based natural language plagiarism detection', *J. Educ. Resour. Comput.* **4**(4).

Zaki, M. J. [2002], Efficiently mining frequent trees in a forest, *in* 'Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 71–80.

Zhao, Y., Zobel, J. & Vines, P. [2006], 'Using relative entropy for authorship attribution', *Information Retrieval Technology* pp. 92–105.

Zheng, R., Li, J., Chen, H. & Huang, Z. [2005], '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.

Anexo A

Etiquetado de información sintáctica

En la tabla A.1 se muestran el conjunto de etiquetas estándar que utiliza el analizador sintáctico *Stanford Parser* para etiquetar las categorías gramaticales de las palabras (en inglés, *POS tags*) [Santorini 1990].

Tabla A.1: Etiquetas de categorías gramaticales

Etiqueta	**Descripción**
CC	Conjunción coordinada
CD	Numeral cadinal
DT	Determinante
EX	Existencial
FW	Palabra extranjera
IN	Preposicion o conjuncion subordinada
JJ	Adjetivo o numeral ordinal
JJR	atribución de autoríadjetivo comparativo
LS	Identificador de un elemento de una lista
MD	Modal auxiliar
NN	Sustantivo común singular
NNP	Sustantivo propio singular
NNPS	Sustantivo propio plural

Tabla A.1 – *Etiquetas de categorías gramaticales (continuación)*

Etiqueta	**Descripción**
NNS	Sustantivo común plural
PDT	Pre-determinante
POS	Marca genitiva
PRP	Pronombre personal
PRP$	Pronombre posesivo
RB	Adverbio
RBR	Adverbio comparativo
RBS	Advebio superlativo
RP	Particula
SYM	Simbolo
TO	Perposicion *to* o un marcador infinitivo
UH	Interjección
VB	Verbo en infinitivo
VBD	Verbo en pasado simple
VBG	Verbo en presente participativo o gerundio
VBN	Verbo en pasado participio
VBP	Verbo en presente simple singular
VBZ	Verbo en presente simple tercera persona
WDT	Determinante *Wh*
WP	Pronombre *Wh*
WP$	Pronombre *Wh* posesivo
WRB	Advebio *WH*

En la tabla A.2 se muestran las etiquetas que utiliza el analizador sintáctico *Stanford Parser* para etiquetar las relaciones de dependencias entre palabras [De Marneffe & Manning 2008].

Tabla A.2: Etiquetas de relaciones de dependencia

Etiqueta	**Descripción**
acomp	Complemento de adjetivo
advcl	Modificador de clausula adverbial
advmod	Modificador de adverbo
agent	Agente
amod	Modificador Adjetival
appos	Modificador aposicional
aux	Auxiliar
auxpass	Auxiliar pasivo
cc	Coordinación
ccomp	Complemento clausal
conj	Conjunción
cop	Cópula
csubj	Sujeto clausal
csubjpass	Sujeto clausal pasivo
dep	Dependiente
det	Determinante
discourse	Elemento discursivo
dobj	Objeto directo
expl	Expletivo
goeswith	Va con
iobj	Objeto indirecto
mark	Marcador
mwe	Expresión multi-palabra
neg	Modificador de negación
nn	Modificador de sustantivo compuesto
npadvmod	Frase nominal como modificador adverbial
nsubj	Sujeto nominal
nsubjpass	Sujeto nominal pasivo

Tabla A.2 – *Etiquetas de relaciones de dependencia (continuación)*

Etiqueta	**Descripción**
num	Modificador numérico
number	Elemento de un número compuesto
parataxis	Parataxis
pcomp	Complemento preposicional
pobj	Objeto de una preposición
poss	Modificador de posesión
possessive	Modificador posesivo
preconj	Preconjunción
predet	Predeterminante
prep	Modificador preposicional
prepc	Modificador preposicional clausular
prt	Frase verbal
punct	Signo de puntuación
quantmod	Modificador de frase cuantificada
rcmod	Modificador clausula relativa
ref	Referente
root	Raíz del árbol de dependencias
tmod	Modificador temporal
vmod	Modificador reducido verbal
xcomp	Complemento de cláusula abierta
xsubj	Sujeto de control