PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO

FACULTAD DE INGENIERIA

ESCUELA DE INGENIERIA INFORMATICA

DETERMINACIÓN DE AUTORÍA POR MEDIO DE REDES DE PALABRAS

Fernando Javier Püschel Araya

Nicolás Ignacio Zárate Guzmán

Profesor Guía: **Dr**. **Rodrigo Alfaro Arancibia**

Profesor Co-referente: **Héctor Allende Cid**

Carrera: **Ingeniería Civil Informática**

Septiembre 2016.

Índice

Lista de Tablas

Lista de Figuras

Resumen

El objetivo de este proyecto es desarrollar un prototipo de algoritmo capaz de clasificar de manera automática la Autoría de textos basado en redes de palabra. Cada texto se representará mediante un grafo dirigido, y será analizado mediante características basadas en la Estilometría para determinar si el perfil de un autor es parecido a otro y así determinar su Autoría.

El algoritmo es implementado por medio de redes de palabras, al cual se evaluará su desempeño en la clasificación con algoritmos alternativos, tales como SimRank Adaptado y Costo de construcción, el primero de estos es un algoritmo propuesto en este informe.

Palabras Claves: Red de Palabras, Clasificación Automática de Texto, Estilometría, Atribución de Autoría.

1. Introducción.

En la actualidad existe una gran cantidad de información disponible en internet, donde gran parte de esta existe en formato de texto, tanto en redes sociales, foros, blogs, correos electrónicos, revistas, artículos, entre otros. También con la masificación de computadoras y dispositivos móviles el acceso, edición y creación de tal información se ha facilitado, lo que ha demandado mayores y mejores herramientas para realizar análisis de textos; despertando un particular interés en la determinación de autoría de estos textos.

La Atribución de Autoría es la capacidad de determinar a un autor, por sus características o perfil, de una obra específica; para este trabajo en particular se considerará como obra los textos escritos en formato digital. Las problemáticas que puede ayudar a resolver la Atribución de Autoría son: identificar acoso sexual, atribución de mensajes terroristas, verificación de autenticidad de notas suicidas, disputas por derechos de autor, detección de plagios, entre otros. Para abordad el problema de la determinación de autoría se emplearán redes de palabras, las cuales consisten en descomponer un texto por sus palabras, para luego almacenarlas como vértices en un grafo dirigido, y las palabras co-ocurrentes tienen un enlace que los une. Luego de establecer las redes de palabras para cada autor, se realiza la clasificación de un texto sin autor determinado, utilizando 2 tipos de medidas, una de costo y otra de similitud.

Este documento se ha organizado en las siguientes secciones:

En la sección 2 se definen los objetivos que se desean conseguir con este trabajo.

En la sección 3 se muestra la definición de la problemática a tratar, la evolución de la técnica y las aplicaciones posibles que puede tener.

En la sección 4 se presenta el marco teórico necesario para entender el trabajo, las técnicas utilizadas y algunos resultados esperados.

En la sección 5 se muestra la experimentación con el marco teórico propuesto y los métodos realizados.

En la sección 6 se presenta el plan de trabajo para proyecto 2.

En la sección 7 se presentan las conclusiones del trabajo.

Finalmente en la sección 8 se muestran todas las referencias que fueron necesarias para la elaboración del trabajo.

.

2. Objetivos

A continuación se presenta el objetivo general y los objetivos específicos para este proyecto.

2.1. Objetivo general

Análisis de atribución de autoría utilizando algoritmo de ranking basado en redes de palabras.

2.2. Objetivos específicos

- Establecer marco teórico de la clasificación automática de autoría de textos y marco teórico para la utilización de redes de palabras.
- Proponer método de comparación de redes de palabras.
- Implementar algoritmo de clasificación automática de autoría de textos basada en redes de palabras.
- Establecer pruebas de comparación de algoritmo de clasificación.
- Realizar pruebas del algoritmo y compararlas con otros algoritmos de clasificación.

3. Problema

3.1. Definición

La atribución de autoría, permite establecer la pertenencia de un texto determinado mediante diferentes métodos de comparación, los cuales establecen un perfil de reconocimiento de un autor, por ejemplo, se pueden reconocer autores por su forma de escribir, por la utilización de palabras clave, por la frecuencia de utilización de algunas palabras o por la omisión de estas, entre otras.

La técnica a utilizar en este proyecto son las redes de palabras, la cual consiste en tomar el texto analizar, obtener todas las distintas palabras que lo componen y establecer uniones entre ellas, las cuales representan su relación, transformando el texto en un grafo dirigido. Esta técnica permite establecer un patrón del autor, en base a la representación que se obtenga del texto.

Este proyecto busca encontrar la similitud de un texto a analizar con un conjunto de textos ya procesados y categorizados, comparando la información que entregan las redes de palabras obtenidas de estos, es decir ve la similitud en la construcción de la nueva red de palabras con la ya existente, donde si su valor de similitud es cercano a 1, este es más parecido y si su valor es cercano a 0 este es más distinto. De acuerdo al valor de similitud se podría determinar si el texto pertenece a cierto autor con algún grado de exactitud.

3.2. Avances

La idea principal detrás de la atribución de autoría en informática es la medición de características textuales que distinguen un autor de otro, la primera aparición de esto es en el siglo 19 con un estudio de Mendenhall [1] que jugaba con la autoría de los textos de Shakespeare, seguido por estudios estadísticos en la primera mitad del siglo 20 con Yule [2] [3] y Zipf [4] Luego de estos inicios se obtiene un detallado estudio de autoría para “The Federalist Papers” realizado por Mosteller y Wallace [5] cual fue el trabajo con más influencia en la atribución de autoría. Los métodos utilizados por ellos fueron basados en el análisis de estadística Bayesiana el cual reconocía la frecuencia de apariciones de un set pequeño de palabras comunes, obteniendo a través de los resultados una discriminación significativa entre los candidatos a autor del texto, este método inicia los estudios no tradicionales de atribución de autoría, opuestos al método tradicional de expertos lingüistas. A finales de los 90s la investigación en atribución de autoría se basó en definir formas para cuantificar la forma de escritura, aquí nace lo que se conoce como “Estilometria” [6] [7]. En ella nace una gran variedad de mediciones, incluyendo el largo de frases, largo de palabras, la frecuencia de palabras, la frecuencia de caracteres y funciones para identificar la riqueza del vocabulario. Rudman [8] estima que existen alrededor de 1000 diferentes formas de medición para un texto.

Las metodologías propuestas para la atribución de autoría son computer-assited más que computer-based, esto quiere decir que este software raramente realiza la clasificación de manera totalmente automatizada, en ciertos casos este método tiene resultados preliminares increíbles haciendo pensar que la solución a este problema es muy cercana.

Luego del fin de los años 90s, las cosas habían cambiado en el estudio de la atribución de autoría. El aumento de los textos electrónicos y los recursos obtenidos de internet, incrementaron la necesidad del manejo de la información de manera eficiente, esto genero un gran impacto en el área de la información,

máquinas de aprendizaje y procesamiento de lenguaje natural (NLP). El desarrollo de estas áreas influencio en la tecnología de atribución de autoría de estas maneras:

- Formas eficientes de representar y clasificar grandes volúmenes de textos.
- Algoritmos poderosos de máquinas de aprendizaje se hacen disponibles para manejar datos de manera multidimensional, además de estandarizar métodos de evaluación para comparar distintos tipos de aproximación de datos.
- La investigación en NLP desarrolla herramientas para analizar textos de manera eficiente y provee nuevas formas para medir y representar los estilos de escritura.

En la última década podemos ver una nueva era de tecnología para el análisis de autoría, ahora dominada por la búsqueda y desarrollo de aplicaciones prácticas que buscan relacionarse con los textos del mundo real (por ejemplo: correos, blogs, mensajes de foros, etc.). El énfasis esta ahora dado por la evaluación de los métodos utilizados para la comparación. Además los factores esenciales que afectan en la atribución son la cantidad de textos utilizados para el entrenamiento, el número de autores candidatos y la distribución de los textos de cada autor.

3.3. Aplicaciones

Las aplicaciones de atribución de autoría existentes, son relacionadas principalmente al ámbito académico, donde se emplean para reconocer la existencia de plagios entre distintos autores, determinar si una persona posee más de un usuario en un sitio web o foro, detectar identidad de acosadores que se ocultan en redes sociales en base a su forma de escribir, entre otros. A continuación se hará referencia a algunos trabajos que mencionen estas aplicaciones:

- Detección automática de plagios en textos. [9]: Hace uso de diversas técnicas de recuperación y extracción de características, las cuales son empleadas para la comparación de los textos. De acuerdo a esa comparación se puede establecer si dos textos o fragmentos de textos son procedentes del mismo autor. Los posible plagios que pueden ser detectados con esta forma son los se hace una copia, incluso con algunas palabras modificadas, de fragmentos de textos sin incluir su fuente, y los que un autor se atribuye la autoridades un trabajo que fue realizado por otra persona; ya que estos pueden verse reflejados en la lectura del texto.

- Reconocimiento de autoría mediante patrones de estilos de escritura. [10]: Determina la atribución de la autoría por medio del estilo del autor, estilometría. El texto es tratado como una red co-ocurrente, ya que es mejor para trabajar los atributos relacionados al estilo escrito de los textos. Cada palabra del texto es representada como un nodo y como medida de medición se necesita la información de los nodos vecinos hasta en segundo grado (los vecinos de los vecinos) del nodo que se está analizando.

- Reconocimiento de acuerdo a los post en foros web. [11]: Se determina la atribución de autoría de los distintos autores que escriben post en un foro web, la idea detrás del método es generar meta características que capturen la modalidad similar especifica de las relaciones entre los textos de los diferentes autores, empleando como medida de clasificación medidas sintácticas, léxicas y de estilo. En el experimento se tenía un solo documento por autor, el cual fue divido en 10 fragmentos, de los cuales 9 fueron empleados para entrenar el modelo y el restante se emplea para probar.

- Reconocimiento de autor de acuerdo a distintos tipos de atributos. [12]: Se hace uso de distintos tipos de atributos proporcionan distintas perspectivas del estilo de cada documento. Dentro del trabajo se emplean utilizar conjuntos de atributos que pueden retener el estilo de los autores, obtener características que tengan una relación entre el documento y el autor, y que puedan ser utilizadas como un modelo de clasificación.

4. Marco Teórico

4.1. Planteamiento Inicial

4.1.1. Estilometría

Es la aplicación del estudio de estilo lingüístico que analiza ciertos rasgos del estilo del autor y los utiliza para comparas dos o más textos, para determinar la autoría de documentos anónimos o en disputa. El punto base de la estilometría es que el estilo es algo propio de cada autor, ya que se encuentra dentro de su subconsciente, y por esta razón, cada quien tiene un estilo propio. Dentro de las aplicaciones de la estilometría se encuentran la determinación de autoría de una obra, la autenticidad, la clasificación de textos, la medición de frecuencias de palabras, la identificación de lenguas, entre otras.

Cada texto tiene marcadores lexicales de estilo, los cuales determinan las características propias del estilo de cada autor, permitiendo una comparación para lograr determinar la autoría. Los marcadores lexicales de estilo se dividen en dos: la riqueza del vocabulario y la frecuencia de las palabras de función. Para la realización automática de los textos se emplean los n –gramas; que permiten obtener una secuencia de ítems cualquiera dentro de una frase o palabra.

La estilometría moderna emplea en gran medida la ayuda de computadores para el análisis estadístico, la inteligencia artificial y el acceso a todos los textos disponibles en internet. Con el uso de computadores es más fácil la detección de patrones dentro de los textos, así como el almacenamiento de las secuencias para su posterior análisis cuantitativo e identificación de secuencias de palabras. Algunos parámetros a considerar para poder realizar el análisis son el número de textos, la cantidad de autores existentes, la extensión de la lista de palabras, la frecuencia de utilización de las palabras, entre otras.

- **Técnicas**

Las técnicas del análisis de estilometría pueden categorizarse en supervisadas y no supervisadas. Las técnicas supervisadas son aquellas que requieren la clasificación y características de los autores, mientras que las no supervisadas hacen la categorización sin conocimiento previo del autor.

Dentro de las técnicas supervisadas usadas para el análisis de autoría se encuentran las Máquinas de Soporte Vectoriales (SVMs), Redes Neuronales, Arboles de Decisión y Análisis Lineal del Discriminante.

Las técnicas no supervisadas incluyen Análisis del Componente Principal (PCA) y Análisis de Clúster. PCA tiene la habilidad de capturar la esencia variante a lo largo de numerosas características dentro de una reducida dimensionalidad, lo que hace que sea atractiva para el análisis de problemas de textos.

Tabla 4.1 Tipos de características estilo métricas con su herramienta computacional y recurso requerido para la medición ("[ ]" indica herramienta opcional)

	Características	Herramientas requeridas y recursos
Léxico	Basado en tokens(largo de palabras, largo de oraciones, etc)	Tokenizer,[Sentence splitter]
	Riqueza de vocabulario	Tokenizer
	Frecuencia de palabras	Tokenizer, [Stemmer, Lemmatizer]
	N-gramas de palabras	Tokenizer
	Errores	Tokenizer, Orthographic spell checker
Caracteres	Tipo de caracteres (letras, dígitos, etc)	Diccionario de caracteres
	N-gramas de caracteres (tamaño fijo)	-
	N-gramas de caracteres (tamaño variable)	Feature selector
	Métodos de compresión	Text compression tool
Sintáctico	Parte del discurso	Tokenizer, Sentence splitter, POS tagger
	Pedazos	Tokenizer, Sentence splitter, [POS tagger], Text chunker
	Estructura de frase y sentencia	Tokenizer, Sentence splitter, POS tagger, Text chunker, Partial parser
	Reglas de frecuencia de re-escritura	Tokenizer, Sentence splitter, POS tagger, TEXT chunker, Full parser
	Errores	Tokenizer, Sentence splitter, Syntactic spell cheker
Semántico	Sinónimos	Tokenizer,[POS tagger],Thesaurus
	Dependencias semánticas	Tokenizer, Sentence splitter, POS tagger, Text Chunker, Partial parser, Semantic parser
	funcionales	Tokenizer,Sentence Splitter, POS tagger,Diccionarios especializados
Específicas de aplicación	Estructuras	HTML parser, parsers especializados
	Contenido especifico	Tokenizerr,[Stemmer,lemmatizer],diccionarios especializados
	Lenguaje especifico	Tokenizerr,[Stemmer,lemmatizer],diccionarios especializados

4.1.2. Atribución de Autoría

Es la ciencia de inferir características de un autor de las características que poseen los documentos escritos por este. La Atribución de Autoría es considerada como uno de los problemas más viejos y nuevo a la vez dentro de la recuperación de la información.

Los principales focos de la Atribución de Autoría están relacionados con los textos escritos, aunque también es posible determinar la Autoría de obras de artes como de música. Dentro de los textos escritos se consideran las estructuras de las sentencias y las elecciones léxicas; que a su vez se descomponen en más de mil opciones para seleccionar características posibles para determinar a un autor.

En términos amplios existen tres principales problemas con la Atribución de Autoría. El primero de ellos es dado una muestra en particular de texto y un conjunto conocido de autores, poder determinar a qué autor pertenece. El segundo, dado una muestra en particular de texto que se cree que pertenece a un grupo de autores, determinar cuál de ellos escribió que parte del texto. Y el tercer problema es poder determinar cualquiera de las propiedades relacionadas al autor de una muestra de texto, ya que puede existir mucha diversidad en la forma de escribir de un autor de acuerdo a un género literario frente a otro; como también en el cambio del público objetivo, entre otros.

La Atribución de Autoría está íntimamente relacionada con la Estilometría, ya que permite la generación de características distintivas entre diferentes autores, para así poder hacer la comparación. Algunas de las principales características están relacionadas con la forma de escribir, la riqueza léxica, la frecuencia de utilización de algunas palabras, el género literario preferido.

Método

Se empleó el siguiente método para llevar a cabo la elaboración del proyecto

Aprendizaje Computacional

Se emplean métodos para categorizar, principal es Método de Aprendizaje de Máquina de Soporte Vectorial, para marcar límites entre las clases asignadas a los autores para facilitar la clasificación, disminuyendo de esta manera las malas clasificaciones.

Características para la Atribución de Autoría

La característica de la atribución de autoría en la cual se trabajó es la siguiente:

Características Léxicas

Este tipo de características toman en cuenta al texto como una secuencia de tokens. Los tokens podrían ser palabras, números, signos de puntuación o abreviaturas.

En este contexto, es posible definir distintas características léxicas basadas en esta secuencia de tokens.

Las características léxicas tienen la ventaja de que muchas de ellas pueden ser extraídas de igual forma para distintos idiomas, con un nivel de análisis relativamente sencillo utilizando herramientas existentes, como los Tokenizers, salvo algunas excepciones como en el Chino, donde esta tarea es un tanto más complicada. [13]

4.1.3. Clasificación automática de textos

El proceso de crear una clasificación automática de textos consiste en descubrir que variables son útiles en la discriminación de los textos que pertenecen a clases prexistentes distintas. En particular, los clasificadores (programas que ejecutan algoritmos de clasificación) son entrenados con un grupo de documentos, previamente clasificados y etiquetados por un humano, acorde a algún criterio particular, conformando una clase. De esta manera, el objetivo de estos clasificadores es decidir en qué categoría debe ir cada nuevo texto, partiendo de un esquema de clasificación previo [14]. También se dice que la clasificación automática de documentos puede ser entendida como una tarea en la cual, en base a la identificación por medios matemático-estadístico, un nuevo documento es asignado a una clase particular de documentos pre-existentes [15]..

En términos prácticos, la utilidad de la clasificación automática de documentos se basa en la posibilidad de poder efectuar una adecuada recuperación de documentos no conocidos, asumiendo que aquellos textos que tratan, por ejemplo, de la misma materia están clasificados juntos, o en sectores cercanos.

En síntesis, se puede decir que la clasificación automática de textos es un proceso de aprendizaje matemático-estadístico, durante el cual un algoritmo implementado capta las características que distinguen cada categoría o clase de documentos de las demás, es decir, aquellas que deben poseer los documentos para pertenecer a esa categoría.

Estas características no tienen por qué indicar de forma absoluta la pertenencia a una clase o categoría, sino que lo haces más bien en función de una escala o graduación. De este modo, por ejemplo, documentos que posean una cierta característica tendrán un factor de posibilidades de pertenecer a determinada clase, de modo que la acumulación de dichas características arrojara un resultado que consiste en un coeficiente asociado a cada una de las clases ya conocidas. Este coeficiente lo que expresa en realidad es el grado de confianza de que el documento en cuestión pertenezca a la clase asociada al coeficiente resultante.

Figura 4.1 Clasificación de textos.

En la figura se puede apreciar un esquema de clasificación automática de textos, donde cada documento del conjunto de prueba, se representa de tal manera que sirve de entrada al clasificador, para ser clasificados según su contenido, para luego poder ser evaluados los resultados del clasificador. En este esquema el clasificador es una función binaria del tipo $\varphi: DxC \rightarrow \{0,1\}$. En esta función D, corresponde el conjunto de documentos de prueba a clasificar, y C la cantidad de categorías donde se quieren clasificar los documentos. Si la función es 1, se dice que el documento d_j pertenece a la clase c_i, por el contrario, si la función es 0 se dice que el documento d_j no pertenece a la clase c_i [16].

Las representaciones utilizadas para representar un texto consideran la presencia o ausencia de términos o palabras independientes en el documento. Esta representación denominada bolsa de palabras, es procesada por los algoritmos clasificadores generalmente de 2 tipos: Maquinas de Soporte Vectorial (SVM) y clasificadores con métodos Bayesianos. Ambos métodos tratan a las variables predictoras (palabras) como independientes. Este supuesto de independencia se utiliza en diversos ámbitos y permite obtener buenos desempeños de clasificación. Sin embargo, considerando que la naturaleza de los datos reales es muchas veces relacional, en esta investigación se propone una representación y una clasificación que se hace cargo de esta simplificación considerando las relaciones entre variables predictoras y utilizando para esto redes de palabras co-ocurrentes. El método propuesto plantea una representación y una clasificación de textos distintas a las tradicionalmente utilizadas en clasificación automática.

4.1.4. Medidas de Similitud entre Grafos

Se centra en conseguir lo siguiente:

Dado: dos grafos, G1 (n1; e1) y G2 (n2; e2), con la posibilidad de diferentes nodos y vértices; y el mapeo entre los nodos de los grafos.

Encontrar: un algoritmo que calcule la similitud de los dos grafos, que retorne una métrica de la similitud, por lo general un número real entre 0 y 1.

La Similitud de Grafos se basa en lo siguiente; el mismo nodo en ambos grafos va a ser similar si sus vecinos son similares, y los vecinos de sus vecinos y así sucesivamente.

La Similitud de Grafos posee numerosas aplicaciones en diversos campos, como las redes sociales, procesos de imágenes, redes biológicas, componentes químicos y visión de computadores; y para cada uno de esos campos existen diferentes algoritmos y medidas de similitud. Existen diferentes propuestas de para la clasificación, que se dividen en tres categorías principales; Isomorfismo de Grafos, Extracción de Características y Método de Intervalos.

Isomorfismo de Grafos

Dos grados son similares si son isomorfos, o uno de ellos es isomorfo de un subgrafo del otro, o ambos grafos poseen subgrafos isomorfos. El inconveniente del Isomorfismo de Grafos es que la versión exacta del algoritmo es un problema de complejidad exponencial y no es aplicable a Grafos de tamaño grande, los que son el interés de estudio en la actualidad.

Durante las últimas décadas se han realizado trabajos de medidas de distancia, o similitud, entre grafos. Estas medidas pueden agruparse en tres categorías.

Métodos Iterativos

La filosofía detrás de este método es que dos nodos son similares si sus vecinos son similares, en cada iteración los nodos intercambian medidas de similitud y este proceso termina cuando la convergencia es alcanzada. Un ejemplo de algoritmo exitoso que pertenece en esta categoría, el cuál es explicado con mayor detalle más adelante.

El método en la literatura que resuelve la Similitud de Grafos tiene el problema que entrega resultados que no son muy intuitivos.

4.2. Técnicas Utilizadas

4.2.1. Red de Palabras

Una red de palabras, es un grafo donde las palabras son los vértices y las palabras co-ocurrentes tienen un enlace que los une, no se consideran los signos de puntuación, por lo tanto dos palabras que estén separadas por una coma, o un punto, no se consideraran co-ocurrentes por lo tanto no tendrán un enlace entre ellos que los una.

A continuación se mostraran algunas propiedades de las redes complejas presentes en las redes de palabras.

- **Ley de Zipf:**

En los años cuarenta un filósofo y lingüista estadounidense George K. Zipf [4] demostró que dentro de un texto escrito la frecuencia de las palabras, de acuerdo a su ranking de aparición, decaía según una ley de potencia. Afirma que un pequeño número de palabras son utilizadas con mucha frecuencia, mientras que frecuentemente ocurre que un gran número de palabras son poco empleadas.

La ley de potencia descrita por Zipf tomaba la forma $P_n \sim 1/n^a$ donde P_n es la frecuencia de una palabra ubicada en el lugar *n* del ranking y *a* la constante con un valor cercano a 1, Así, la segunda palabra se repetirá aproximadamente con una frecuencia de ½ de la primera, la tercera con una frecuencia de 1/3 y así sucesivamente. La distribución de frecuencia de palabras en un texto era entonces invariante de escala.

Estos resultados dieron paso a lo que hoy se conoce como lingüística cuantitativa, no dice nada sobre el sentido de un texto, de hecho, un texto hecho al azar puede tener una distribución de frecuencia de aparición que también se ajuste a una ley de potencia.

- **Small World:**

Ferrer i Cancho y Solé [17] descubrieron que al tratar un conjunto de palabras de la base de datos de *British National Corpues* como una red, al asumir enlaces entre palabras próximas (co-ocurrentes) en frases, era posible observar propiedades de redes complejas en el idioma inglés. La red de la lengua inglesa mostraba una distribución invariante de escala de las palabras según su conectividad, algo muy similar a los resultados de Zipf ya que de forma indirecta indican la frecuencia de una palabra dentro de un texto. Pero, lo más interesante era que la red mostraba otras propiedades complejas similares a las redes tipo *small world* [18] un alto clustering medio y, en promedio, una corta distancia entre palabras.

4.2.2. Representación de redes de palabras

Sea $\Omega = (W, E)$ el grafo del lenguaje escrito donde $W = \{w_i\}, (i = 1, \dots, N)$ es el conjunto de N palabras y $E = \left\{\{w_i, w_j\}\right\}$ el número de conexiones entre palabras. Podemos definir palabras adyacentes como un par i y j, pertenecientes a Ω, entre las cuales existe un enlace. Ahora, para el grafo Ω podemos definir un vector de palabras $A(\Omega)$ como una representación $N \; x \; M_i$ palabras, donde M_i son subvectores de largo variable que contiene las palabras relacionadas a la palabra en la posición i. De forma que, cuando existe el enlace entre dos palabras i y j $A_{ij} = 1$, en caso contrario $A_{ij} = 0$, si existe más de 1 enlace se define como $A_{ij} = A_{ij} + 1$.

El enlace entre palabras se construye a partir de entradas y salidas que tiene una palabra en una frase. Se considera la puntuación de un texto como un separador de ideas por lo que las palabras que están antes o después de una puntuación no se consideran vecinos en la red. Por ejemplo el micro cuento “Asesinato en la 5ta avenida” contiene 59 palabras distintas, donde el número entre los enlaces representa la cantidad de aparición de ese par de palabras juntas.

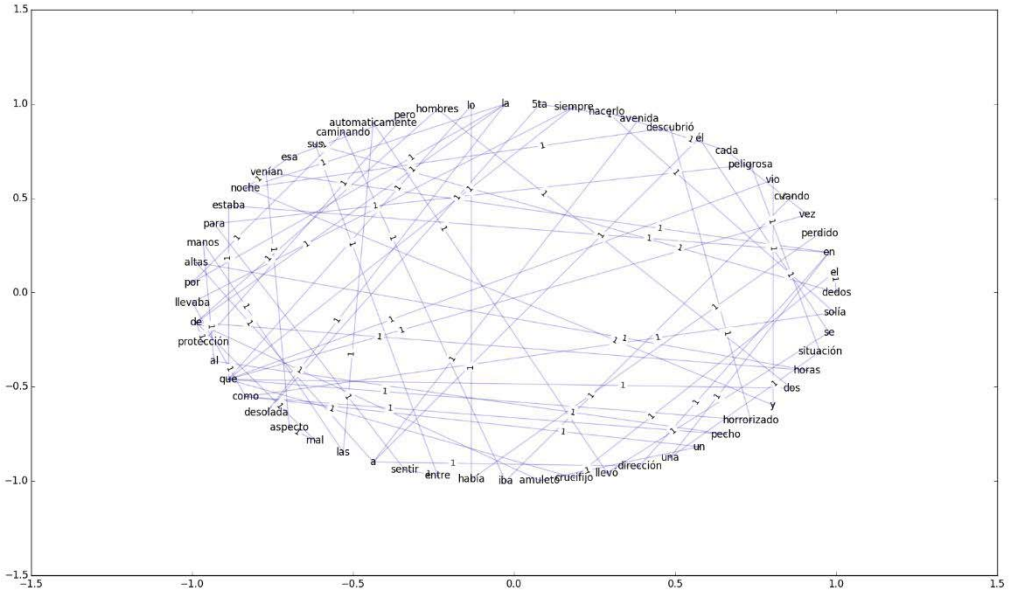

Figura 4.2 Red de palabras del micro cuento "Asesinato en la 5ta avenida”, N=59

Las redes de palabras [19] son objetos matemáticos del tipo *G(N,E)*, donde *G* es el grafo compuesto por *N* palabras distintas y *E* enlaces que las relacionan por co-ocurrencia en el texto. G_i Corresponde a la red de palabras co-ocurrentes construida a partir de un conjunto de textos perteneciente a un autor *i* clasificado por un humano. De esta forma habrán tantos grafos *G* como autores de textos hayan sido entrenadas. Por otro lado, *h* corresponde a la red de palabras construida a partir de un nuevo texto sin clasificar.

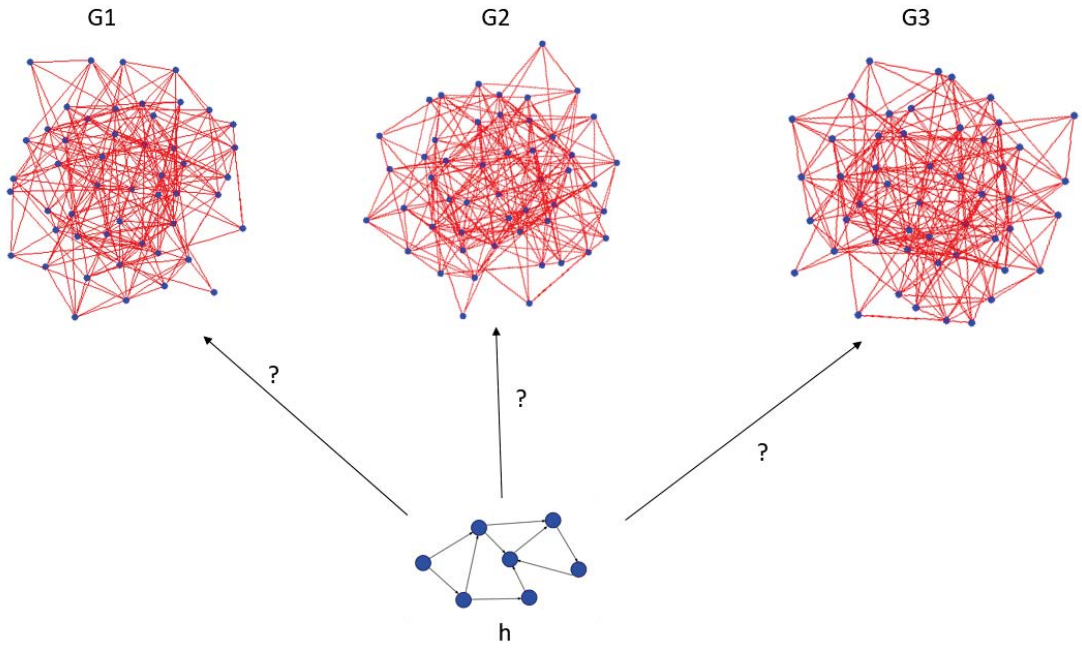

Figura 4.3 Grafo de las palabras *h*, que representa el texto sin clasificar, evaluando su reconstrucción en tres grafos de entrenamiento *G1 , G2 y G3*, construidos por textos clasificados por un humano como autor 1 , autor 2 y autor 3 respectivamente.

Este método se diferencia de los algoritmos de clasificación supervisada tradicionales en dos aspectos fundamentales, la representación y el clasificador. La primera es totalmente distinta al tradicional concepto de bolsa de palabras que asume independencia entre las variables predictoras y distribución uniforme de éstas. De hecho, al relacionar textos de autores *i* para formar un sistema G_i único por autor significa considerar el orden de las palabras, su frecuencia, su frecuencia de co-ocurrencia y además el contexto, la gran mayoría dejadas de lado por las aproximaciones tradicionales.

Este método se diferencia del algoritmo SVM en que los elementos entrenados no son tratados de forma independiente sino que forman una gran estructura G por autor.

4.2.3. SimRank [20]

La base detrás de este algoritmo es que, en muchos dominios, objetos similares están relacionados con otros objetos similares. Más precisamente, los objetos A y B son similares si están relacionados con los objetos C y D, respectivamente, y C y D son en sí mismos similares. El caso base es que los objetos son similares a sí mismos.

Utilizando una descripción de recursividad básica detrás de este enfoque se puede decir que “dos objetos son similares si están referenciados por objetos similares“. Como el caso base, consideramos un objeto máximamente similar cuando este se compara con sí mismo, al cual se le asigna el valor de similitud igual a 1.

Para un nodo *v* en un grafo, denotamos por $I(v)$ al conjunto de in-vecinos de *v*, estos son los nodos que tienen una unión direccionada a *v*. Los vecinos individuales se denotan como $I_i(v)$, para $1 \leq i \leq |I(v)|$.

Denotemos la similitud entre objetos a y b por $s(a,b) \in [0,\infty[$. Dado esto definimos como $s(a,b)$ de la siguiente manera.

$$s(a,b) = \frac{C}{|I(a)| * |I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b))$$

Donde C es una constante entre 0 y 1. Una tecnicidad aquí es que, a o b no pueden no tener ningún in-vecino. Puesto que no se tendría manera de inferir la similitud entre a y b, para este caso ocurre que $s(a,b) = 0$, por lo que definimos la suma en la ecuación pasa a ser 0 cuando $I(a) = 0$ o $I(b) = 0$.

4.2.4. SimRank Adaptado

Sea δ el conjunto de grafos de los autores procesados, sea $G \in \delta$ un grafo de palabras para un autor procesados, sea H como el grafo de palabras al cual se quiere establecer su posible autor.

Para la utilización del SimRank se define como, *a* un nodo de G, $Ig(a,G)$ como los in-vecinos de *a* en G, los vecinos individuales se denotan como $Ig_i(a,G)$, para $1 \leq i \leq |Ig(a,G)|$ y se define R1 como una lista de nodos recorridos en G que haya tomado el valor *a* y R2 como una lista de nodos recorridos en H que haya tomado el valor de *b*.

Denotemos la similitud entre los nodos $a \in$ G y $b \in$ H por $sa(a,G,b,H) \in [0,1]$. Dado esto definimos como $sa(a,G,b,H)$ de la siguiente manera.

$$sa(a,G,R1,b,H,R2) = \frac{C}{|Ig(a,G)| * |Ig(b,H)|} \sum_{i=1}^{|Ig(a,G)|} \sum_{j=1}^{|Ig(b,H)|} sa(Ig_i(a,G), Ig_j(b,H))$$

Al igual que el SimRank original C es una constante entre 0 y 1, el término del algoritmo está dado por la comparación de $|Ig(a,G)| = 0$ *o* $|Ig(b,H)| = 0$ la suma pasa a ser 0 y también está dado por la comparación de *a* y *b* si estos valores son iguales, sa(a,G,b,H)=1.

Los listados R1 y R2 son utilizados para eliminar los ciclos en la búsqueda del SimRank, impidiendo que el algoritmo termine en un bucle infinito, estos también afectan al resultado final, ya que si $Ig(a,G) \in R1$ *y* $Ig(b,H) \in R2$ entonces el valor de la suma seria 0, excepto para el caso que a y b sean iguales el valor pasa a ser 1.

Dada esta definición, se establece como el algoritmo de comparación, al promedio del SimRank adaptado para las palabras de H presentes en G:

$$C(G,H) = \frac{\sum^{|G \cap H|} sa(a,G,\{\},b,H,\{\})}{|G \cap H|} \; donde \; a \in G \; , b \in H \; y \; a = b$$

El valor de $C(G,H) \in [0,\infty[$ donde 0 representa que no es igual. La particularidad de este algoritmo es que a y b al inicio son iguales, por ende en la primera iteración del SimRank adaptado no debe tomar en cuenta la igualdad de las palabras, los campos {} representan una lista vacía.

4.2.5. Árboles de Decisión (C4.5)

C4.5 es un algoritmo usado para generar un árbol de decisión desarrollado por Ross Quinlan, el cual es una exterior del algoritmo ID3, creado por él mismo.

C4.5 construye árboles de decisión desde un grupo de datos de entrenamiento usando el concepto de entropía de información. Los datos de entrenamiento se consideran $S=s_1, s_2,\ldots$ de datos ya clasificados, cada uno $s_i = x_1, x_2,$ es un vector, donde x_i representan los atributos o características de los datos. Luego los datos de entrenamiento son aumentados con un vector $C=c_1, c_2,\ldots$, donde c_i representan la clase a la cual pertenecen.

En cada nodo que posee el árbol se escoge un atributo de los datos que discrimina de mejor manera el conjunto, dividiéndolo así en subconjuntos pertenecientes a una clase u otra, de esta forma, una vez entrenado los datos se procede a clasificar los nuevos datos a partir de las decisiones que tenga que ir tomando en cada nodo, llegando así a determinar a qué clase debe pertenecer.

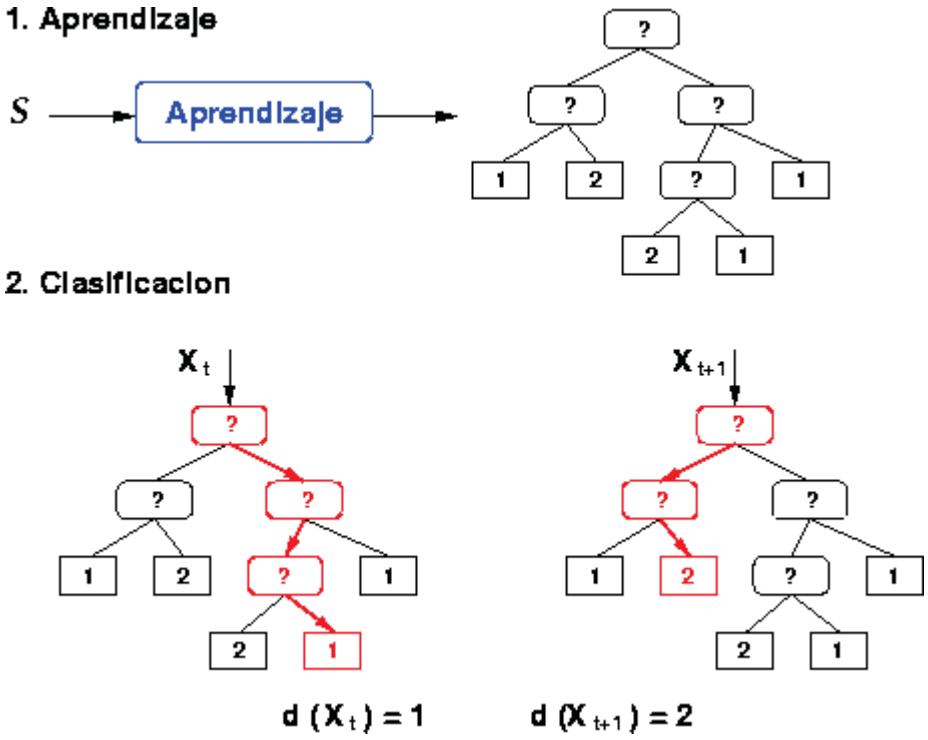

Figura 4.4 Esquema algoritmo C4.5

Existe una implementación open source de este algoritmo en el lenguaje de programación java, denominado J48, utilizado por la herramienta weka.

4.2.6. Máquinas de Soporte Vectorial

Máquinas de Soporte Vectorial han mostrado un buen desempeño en general en una gran variedad de problemas de clasificación, más recientemente en clasificación de textos.

En términos geométricos, el problema que resuelve las SVM (Support Vector Machine) es identificar una frontera de decisión linear entre dos clases, a través de una línea que los separe, maximizando el espacio del hiperplano. Sin embargo, las SVM incluyen una función llamada kernel, la cual permite realizar separaciones no lineales de los datos, proyectando la información a un espacio de características de mayor dimensión. Esto se logra cambiando la representación de la función, mapeando el espacio de entradas D a un nuevo espacio de características $F = \{\varphi(d) | \ d \in D\}$. Esto es:

$$d = \{d_1, d_2, \ldots, d_n\} \rightarrow \varphi(d) = \{\varphi(d)_1, \varphi(d)_2 \ldots, \varphi(d)_n\}$$

En la figura 6.1 se muestra un mapeo de un espacio de entradas de dos dimensiones a un nuevo espacio de características de dos dimensiones, donde la información no puede ser separada por una máquina lineal mientras que en el nuevo espacio de características esto resulta sencillo.

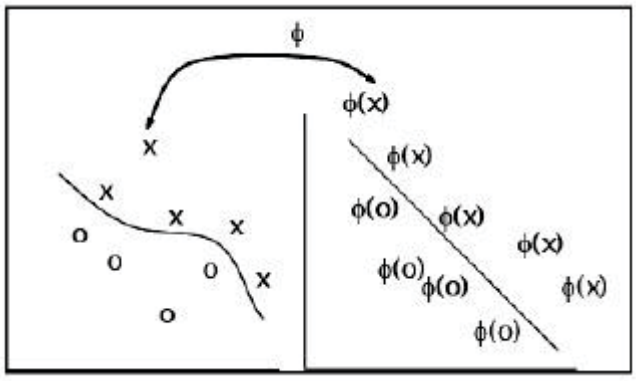

Figura 4.5 Mapeo del espacio de entradas a un espacio de mayor dimensión.

La función real φ no necesita ser conocida, es suficiente tener una función kernel k, la cual hace posible realizar el mapeo de la información de entrada al espacio de características de forma implícita y entrenar a la máquina lineal en dicho espacio.

4.2.7. Naive Bayes Multinomial

Fundamentado en el teorema de Bayes con algunas hipótesis simplificadoras adicionales; con las simplificaciones añadidas se resume en la hipótesis la independencia entre las variables utilizadas para predecir. El clasificador suele ser bien entrenado bajo un ambiente de aprendizaje supervisado.

El clasificador Naive Bayes (Bayes Ingenuo) se construye usando el conjunto de entrenamiento para estimar la probabilidad de cada clase dados los valores de atributos

(palabras) del documento de una nueva instancia. Usando el Teorema de Bayes para estimar las probabilidades:

$$P(c_j|d) = \frac{P(c_j)P(d|c_j)}{P(d)}$$

El denominador en la ecuación anterior no distingue entre categorías y puede ser eliminado. Este método asume que los atributos son condicionalmente independientes, dada la clase. Esto simplifica los cálculos.

$$P(c_j|d) = P(c_j)\prod_{i=1}^{M} P(d_i|c_j)$$

Una estimación $\hat{P}(c_j)$ para $P(c_j)$ puede ser calculada de la fracción de documentos de entrenamiento que es asignada a la clase c_j:

$$\hat{P}(c_j) = \frac{N_j}{N}$$

Donde N_j es el número de documentos de entrenamiento para los cuales la clase es c_j y N es el número total de documentos de entrenamiento.

Una estimación $\hat{P}(d_i|c_j)$ para $P(d|c_j)$ está dada por:

$$\hat{P}(d_i|c_j) = \frac{1 + N_{ij}}{M + \sum_{k=1}^{M} N_{kj}}$$

Donde N_{ij} es el número de veces de la palabra i ocurrida dentro de los documentos de la clase c_j en el conjunto de entrenamiento. Para evitar el problema de la probabilidad cero se utiliza Laplace (agregar un 1). M es el número de términos en el vocabulario.

A pesar de que la suposición de independencia condicional es generalmente falsa para la aparición de la palabra en documentos, el clasificador Naive Bayes es sorprendentemente efectivo.

5. Experimentación

A continuación se va a especificar todo lo que fue necesario para el desarrollo de la experimentación.

5.1. Datos a Utilizar

El primer dataset se compone de una base de datos de textos de opinión llamada Guardian Corpus, con un total de 844 textos de distintos tópicos y con 13 autores distintos.

Para el periodo de entrenamiento se establece 2/3 del total, esto quiere decir que 563 textos serán utilizados como textos de autores y el resto será utilizado como prueba, como se define en la siguiente tabla:

Tabla 5.1 Distribución de los datos a probar dataset 1

Autor	Cantidad de textos	Textos de entrenamiento	Textos de prueba
Catherine Bennett	41	28	13
George Monbiot	53	36	17
Hugo Young	54	18	18
Jonathan Freedland	126	84	42
Martin Kettle	46	31	15
Mary Riddell	67	44	23
Nick Cohen	48	32	16
Peter Preston	91	60	31
Polly Toynbee	70	46	24
Roy Hattersley	44	30	14
Simon Hoggart	116	79	37
Will Hutton	50	34	16
Zoe Williams	38	25	13

Se crearon tres sets de pruebas, con el fin de poder comparar cómo se comporta el algoritmo con la data mezclada, y poder obtener conclusiones más acertadas.

Para disminuir la variabilidad de los resultados se ha conformado un segundo dataset que contiene solamente textos de un solo tópico por autor, este se conforma de 346 textos en total, la división de entrenamiento y prueba está conformada igual que el primer dataset.

Tabla 5.2 Distribución de los datos a probar dataset 2

Autor	Cantidad de textos	Textos de entrenamiento	Textos de prueba
Catherine Bennett	11	7	4
George Monbiot	41	27	14
Hugo Young	35	23	12
Jonathan Freedland	68	45	23
Martin Kettle	36	24	12
Mary Riddell	23	15	8
Nick Cohen	9	6	3
Peter Preston	66	44	22
Polly Toynbee	12	8	4
Roy Hattersley	3	2	1
Simon Hoggart	5	3	2
Will Hutton	22	14	8
Zoe Williams	14	9	5

5.2. Medidas de Evaluación

Para las medidas de evaluación se utilizó la siguiente métrica:

- SimRank adaptado.

Este es un algoritmo propuesto en este informe, consta de una variación de SimRank, adaptándolo a la comparación de 2 grafos que no están unidos.

La explicación de este algoritmo ya fue abordada, por lo que solo describirá la formula final utilizada.

$$C(G,H) = \frac{\sum^{|G\cap H|} sa(a,G,\{\},b,H,\{\})}{|G \cap H|} \; donde\; a \in G\;, b \in H\; y\; a = b$$

Donde:

$C(G,H) \in [0,1]$: El valor 0 indica que no existe similitud y el valor 1 indica una similitud total.

$sa(a,G,\{\},b,H,\{\})$: Función SimRank adaptada

$|G \cap H|$: Cardinalidad de las palabras en común entre G y H

G: Es la red de palabras de un autor conocido.

H: Es la red de palabras con autor desconocido

{}: Representa una lista vacía.

5.3. Experimentos

Los experimentos se realizaron utilizando la formula anteriormente planteadas, programadas en Python, utilizando la librería NetworkX para el tratamiento de grafos y funciones especiales.

En un computador con las siguientes características:

Procesador: AMD A8-6600K APU.

Memoria RAM: 8 GB con 7,21 utilizables, a una frecuencia de 1600 MHz.

Disco Duro: 1 TB a 7200 RPM.

Para el algoritmo de SimRank especificado anteriormente, se realizó la programación en Python, utilizando como datos de entrada los especificados en la sección 5.1.

También se definió la constante C, para la experimentación se utilizaron los 3 set de datos definidos anteriormente y con los valores de C igual a {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}.

La entrada de datos se definió como archivos de texto plano, en la cual cada autor puede tener varios textos y de la misma manera se definieron los textos de prueba.

Dentro de la experimentación se debió realizar un cambio en el algoritmo, debido a que antes se trabajaba de manera recursiva y Python no soporta un nivel muy grande de recursividad, teniendo que realizar el algoritmo de manera iterativa.

Debido a que estas pruebas son con textos en inglés se hizo uso de una lista de Stop Words, obtenidas de Ranks.nl (véase anexo), con el fin de eliminar palabras poco significativas que ralentizaban el proceso y que no entregaban información relevante.

5.4. Resultados dataset 1

Para presentar los resultados se realizó la tabulación de ellos, mostrando el porcentaje de aciertos para cada autor en los set de datos.

5.4.1. Set 1 con Simrank Adaptado

Tabla 5.3 Resultados Dataset 1 - Set 1

Autor	C 0,1	C 0,2	C 0,3	C 0,4	C 0,5	C 0,6	C 0,7	C 0,8	C 0,9	C 1,0
Catherine Bennett	0%	0%	7%	7%	7%	7%	7%	14%	14%	7%
George Monbiot	6%	6%	0%	0%	0%	0%	6%	6%	0%	0%
Hugo Young	0%	0%	0%	0%	6%	6%	6%	0%	0%	0%
Jonathan Freedland	0%	0%	0%	2%	2%	2%	17%	36%	40%	31%
Martin Kettle	13%	13%	13%	13%	7%	7%	7%	0%	0%	7%
Mary Riddell	0%	0%	0%	0%	0%	0%	9%	27%	45%	50%
Nick Cohen	0%	0%	0%	0%	0%	0%	6%	6%	0%	0%
Peter Preston	0%	0%	0%	0%	0%	0%	3%	17%	10%	7%
Polly Toynbee	9%	9%	9%	9%	9%	9%	9%	4%	17%	13%
Roy Hattersley	7%	7%	7%	7%	13%	20%	7%	7%	0%	0%
Simon Hoggart	5%	5%	5%	8%	8%	8%	5%	5%	8%	3%
Will Hutton	0%	0%	0%	0%	0%	6%	0%	0%	0%	0%
Zoe Williams	8%	15%	15%	15%	17%	17%	23%	8%	8%	15%
Total	3%	4%	4%	4%	5%	5%	8%	12%	14%	12%

5.4.2. Set 2 con Simrank Adaptado

Tabla 5.4 Resultados Dataset 1 - Set 2

Autor	C 0,1	C 0,2	C 0,3	C 0,4	C 0,5	C 0,6	C 0,7	C 0,8	C 0,9	C 1,0
Catherine Bennett	0%	7%	7%	7%	7%	7%	21%	14%	29%	7%
George Monbiot	17%	17%	17%	17%	17%	11%	6%	0%	6%	0%
Hugo Young	0%	0%	0%	0%	0%	0%	6%	0%	6%	6%
Jonathan Freedland	0%	0%	0%	0%	0%	0%	26%	45%	45%	19%
Martin Kettle	0%	0%	0%	0%	0%	0%	7%	7%	0%	0%
Mary Riddell	9%	9%	9%	14%	14%	14%	18%	32%	45%	45%
Nick Cohen	13%	13%	13%	13%	19%	19%	19%	6%	0%	0%
Peter Preston	0%	0%	0%	0%	0%	0%	7%	10%	3%	0%
Polly Toynbee	0%	0%	0%	0%	0%	0%	9%	13%	13%	9%
Roy Hattersley	13%	13%	13%	13%	13%	20%	20%	0%	0%	0%
Simon Hoggart	5%	5%	5%	5%	3%	3%	5%	13%	13%	18%
Will Hutton	12%	12%	12%	12%	12%	6%	6%	6%	0%	0%
Zoe Williams	8%	17%	25%	25%	25%	25%	25%	0%	25%	17%
Total	5%	6%	6%	6%	6%	6%	13%	15%	17%	11%

5.4.3. Set 3 con Simrank Adaptado

Tabla 5.5 Resultados Dataset 1 - Set 3

Autor	C 0,1	C 0,2	C 0,3	C 0,4	C 0,5	C 0,6	C 0,7	C 0,8	C 0,9	C 1,0
Catherine Bennett	0%	0%	0%	8%	8%	8%	8%	8%	8%	0%
George Monbiot	6%	6%	6%	6%	6%	6%	0%	18%	24%	0%
Hugo Young	0%	0%	0%	0%	0%	0%	0%	6%	0%	0%
Jonathan Freedland	0%	0%	0%	0%	0%	0%	12%	29%	45%	17%
Martin Kettle	0%	0%	0%	0%	0%	0%	0%	6%	6%	6%
Mary Riddell	4%	4%	4%	4%	4%	4%	4%	13%	9%	48%
Nick Cohen	0%	0%	0%	0%	0%	6%	6%	19%	19%	6%
Peter Preston	3%	3%	3%	3%	3%	3%	6%	13%	13%	3%
Polly Toynbee	0%	0%	0%	0%	0%	0%	4%	17%	17%	13%
Roy Hattersley	13%	20%	20%	20%	20%	20%	20%	7%	0%	0%
Simon Hoggart	3%	3%	3%	3%	3%	3%	8%	3%	11%	26%
Will Hutton	0%	0%	0%	0%	0%	0%	18%	18%	18%	6%
Zoe Williams	25%	25%	25%	25%	25%	25%	17%	17%	8%	0%
Total	3%	4%	4%	4%	4%	4%	8%	14%	16%	12%

5.4.4. Análisis de resultados Simrank Adaptado

Como se puede ver en los resultados en promedio los mejores fueron con C=0.9 para los 3 casos, sin embargo no superan el 20% de acierto, un punto importante al ver los resultados es que al variar la constante se puede observar un cambio no predecible en estos porcentajes, ya que hasta cierto rango el valor de acierto aumenta en alguno y disminuye en otros, superando este rango puede ocurrir una caída brusca de aciertos o un aumento y luego puede ocurrir otra caída o aumento, de esta forma no es posible predecir un comportamiento, también se debe considerar que para algunos valores de C aumenta la cantidad de aciertos de un autor y disminuye para otros, con esto se podría establecer un nuevo valor de C que se tendría que definir para cada autor, dependiendo de la complejidad del texto o alguna otra característica.

5.4.5. Resultado de algoritmos WEKA

En la siguiente tabla se muestran los porcentajes de aciertos para cada uno de los algoritmos probados en WEKA, estos resultados son el promedio de los set de datos definidos anteriormente.

Tabla 5.6 Resultados WEKA - Dataset 1

Algoritmo	J48	WLSVM	Naive Bayes Multinomial
Aciertos	37%	24%	88%

5.4.6. Análisis de resultados WEKA

Para tener resultados con los cuales contrastar los del algoritmo Simrank se emplearon los algoritmos J48, WLSVM y Naive Bayes Multinomial, todos ellos se encuentran en la herramienta WEKA y pueden ser empleados para la clasificación de textos. De los resultados obtenidos se puede destacar que solo Naive Bayes Multinomial fue satisfactorio para la clasificación automática de los textos; los otros tienen un desempeño muy similar al de la propuesta.

5.4.7. Comparación de resultados

En la siguiente tabla se muestra la comparación de los resultados de WEKA contra el promedio de los mejores resultados del Simrank Adaptado, por coincidencia el mejor resultado del algoritmo se da con C=0.9, dando como promedio un 16%.

Tabla 5.7 Comparación resultados WEKA y Simrank Adaptado - Dataset 1

Comparación	J48	WLSVM	Naive Bayes Multinomial
algoritmos	37%	24%	88%
Promedio C=0.9	16%	16%	16%
Comparación alg/prom	230%	153%	555%

En comparación a los resultados obtenidos por el algoritmo Simrank Adaptado, los obtenidos por WEKA son superiores, obteniendo hasta un 555% de diferencia, lo que lleva a plantear que la propuesta de este trabajo, hasta el momento no es apta para determinar la autoría de textos debido al bajo porcentaje de acierto, para ningún valor de C.

5.5. Resultados dataset 2

Al igual que el dataset 1 se tabularon los datos mostrando el porcentaje de aciertos para cada autor en los set de datos.

5.5.1. Set 1 con Simrank Adaptado

Tabla 5.8 Resultados Dataset 2 - Set 1

Autor	C 0,1	C 0,2	C 0,3	C 0,4	C 0,5	C 0,6	C 0,7	C 0,8	C 0,9	C 1,0
Catherine Bennett	0%	0%	0%	0%	0%	0%	0%	25%	0%	0%
George Monbiot	0%	0%	0%	0%	0%	0%	0%	7%	7%	14%
Hugo Young	0%	0%	0%	0%	0%	0%	0%	0%	17%	33%
Jonathan Freedland	0%	0%	0%	0%	4%	9%	4%	26%	22%	17%
Martin Kettle	0%	0%	0%	0%	0%	0%	0%	25%	0%	8%
Mary Riddell	0%	0%	0%	0%	0%	0%	13%	13%	38%	50%
Nick Cohen	0%	0%	0%	0%	33%	0%	0%	33%	0%	0%
Peter Preston	5%	5%	5%	5%	5%	0%	9%	32%	23%	23%
Polly Toynbee	0%	0%	0%	0%	0%	0%	0%	0%	25%	25%
Roy Hattersley	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Simon Hoggart	50%	50%	50%	100%	100%	100%	100%	100%	0%	0%
Will Hutton	0%	0%	0%	0%	0%	0%	13%	13%	13%	0%
Zoe Williams	20%	20%	20%	20%	20%	20%	20%	0%	20%	0%
Total	3%	3%	3%	3%	5%	4%	7%	19%	16%	18%

5.5.2. Set 2 con Simrank Adaptado

Tabla 5.9 Resultados Dataset 2 - Set 2

Autor	C 0,1	C 0,2	C 0,3	C 0,4	C 0,5	C 0,6	C 0,7	C 0,8	C 0,9	C 1,0
Catherine Bennett	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
George Monbiot	0%	0%	0%	0%	0%	0%	0%	29%	29%	29%
Hugo Young	0%	0%	0%	0%	0%	0%	0%	8%	0%	17%
Jonathan Freedland	0%	0%	0%	0%	0%	0%	4%	17%	22%	17%
Martin Kettle	0%	0%	0%	0%	0%	0%	0%	0%	0%	33%
Mary Riddell	0%	0%	0%	0%	0%	0%	0%	13%	50%	63%
Nick Cohen	33%	33%	33%	0%	33%	0%	0%	0%	0%	0%
Peter Preston	0%	0%	0%	5%	0%	0%	5%	27%	36%	14%
Polly Toynbee	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Roy Hattersley	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Simon Hoggart	50%	50%	50%	100%	50%	50%	50%	0%	0%	0%
Will Hutton	0%	0%	0%	0%	0%	0%	0%	0%	13%	13%
Zoe Williams	20%	20%	40%	20%	40%	40%	0%	20%	20%	20%
Total	3%	3%	3%	3%	3%	3%	3%	14%	19%	20%

5.5.3. Set 3 con Simrank Adaptado

Tabla 5.10 Resultados Dataset 2 - Set 3

Autor	C 0,1	C 0,2	C 0,3	C 0,4	C 0,5	C 0,6	C 0,7	C 0,8	C 0,9	C 1,0
Catherine Bennett	0%	0%	0%	0%	0%	0%	0%	0%	50%	25%
George Monbiot	0%	0%	0%	0%	0%	0%	0%	7%	7%	14%
Hugo Young	0%	0%	0%	0%	0%	0%	0%	0%	8%	0%
Jonathan Freedland	0%	0%	0%	0%	0%	0%	4%	13%	17%	9%
Martin Kettle	0%	0%	0%	0%	0%	0%	0%	8%	0%	0%
Mary Riddell	0%	0%	0%	0%	0%	0%	13%	25%	63%	75%
Nick Cohen	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Peter Preston	0%	0%	0%	0%	0%	0%	5%	9%	14%	5%
Polly Toynbee	75%	75%	75%	75%	75%	50%	50%	50%	25%	25%
Roy Hattersley	100%	100%	100%	100%	100%	100%	0%	0%	0%	0%
Simon Hoggart	100%	100%	100%	100%	100%	100%	100%	100%	0%	0%
Will Hutton	0%	0%	0%	0%	0%	0%	13%	13%	0%	0%
Zoe Williams	20%	20%	20%	20%	20%	20%	20%	0%	20%	40%
Total	6%	6%	6%	6%	6%	5%	8%	12%	15%	13%

5.5.4. Análisis de resultados Simrank Adaptado

Al observar los resultados obtenidos, podemos apreciar que estos, al igual que el dataset 1 poseen un comportamiento aleatorio, ya que al variar el valor de C no se puede estimar un porcentaje de aciertos, sin embargo existen autores que poseen el 100% de aciertos, se presume que estos resultados se deben a la poca cantidad de textos de prueba y de entrenamiento, dado que al tener menor cantidad de textos de entrenamiento la red de palabras se vuelve más acotada, ya que esta se forma con menos palabras generando una menor complejidad de conexión entre ellas.

5.5.5. Resultado de algoritmos WEKA

En la siguiente tabla se muestran los porcentajes de aciertos para cada uno de los algoritmos probados en WEKA, estos resultados son el promedio de los set de datos definidos anteriormente.

Tabla 5.11 Resultados WEKA – Dataset 2

Algoritmo	J48	WLSVM	Naive Bayes Multinomial
Aciertos	35%	23%	79%

5.5.6. Análisis de resultados WEKA

Al igual que el dataset anterior se utilizaron los algoritmos J48, WLSVM y Naive Bayes Multinomial, a través de la herramienta WEKA. De los resultados obtenidos se puede destacar que solo Naive Bayes Multinomial fue satisfactorio para la clasificación automática de los textos; los otros tienen un desempeño muy similar al de la propuesta.

5.5.7. Comparación de resultados

En la siguiente tabla se muestra la comparación de los resultados de WEKA contra el promedio de los mejores resultados del Simrank Adaptado, este es con C=1.0, dando como promedio un 17%.

Tabla 5.12 Comparación resultados WEKA y Simrank Adaptado – Dataset 2

Comparación	J48	WLSVM	Naive Bayes Multinomial
algoritmos	35%	23%	79%
Promedio C=1.0	17%	17%	17%
Comparación alg/prom	205%	135%	465%

Los resultados obtenidos a través de WEKA, al igual que el dataset anterior, muestran un comportamiento similar, es decir que los porcentajes de diferencia son sobre el 100%, llegando

a un máximo de 465% para el algoritmo Naive Bayes Multinomial. En comparación con el dataset anterior, se puede ver una mejora no significativa de un 1%, hay que tomar en cuenta que solo se alteraron los datos de prueba, sin modificar el algoritmo.

5.6. Comparación de resultados entre Dataset 1 y Dataset 2

Al analizar los resultados obtenidos anteriormente, se esperaba que el Dataset 2 tuviese mejores resultados que el Dataset 1, debido a que en el Dataset 2 solo se incluyeron textos de un mismo tópico, disminuyendo de esta manera la variabilidad de las palabras contenidas en los textos, ya que todos ellos se centrarían en un solo contexto, para el caso de las pruebas se seleccionó el tópico "World", el cual contiene la mayor cantidad de textos por autor. Al comparar los 2 resultados hubo una mejora de los estos, pero no de una manera significativa, alrededor de un 5%.

Uno de los motivos por el cual se cree que hubo pocos aciertos en los resultados, tanto para el Dataset 1 como para el Dataset 2, es debido a que algunos autores poseen mucha variabilidad en las palabras que escogen a la hora de crear sus textos; o que hacen uso de palabras que son muy utilizadas. El primer caso hace que las redes de palabras de los autores se vuelvan complejas, y en el segundo caso hace que las redes de palabras entre los distintos autores sea más homogénea, causando mayor cantidad de falsos positivos.

La comparación de los valores para C para ambos Dataset llevó al siguiente resultado; en el Dataset 1 el mayor valor de este en promedio fue de 0.9, en cambio en el Dataset 2 su valor varía entre 0.8 y 1.0., por lo que se estima que el valor óptimo para C estaría entre C= 0.8 +-0.1, considerando textos del mismo tamaño y contexto.

6. Trabajos Futuros

Lo que todavía falta para completar el trabajo es encontrar una alternativa al Simrank que siga utilizando las redes de palabras para determinar la autoría de los textos, debido al bajo desempeño del Simrank; además de identificar las causas del bajo desempeño de este. También se considera que hay que realizar varias pruebas con datasets de diversas complejidades con el fin de encontrar si se puede utilizar el Simrank para analizar algún tipo de texto.

.

7. Conclusiones

En la actualidad la determinación de Autoría ha ganado cada vez mayor importancia debido a que soluciona los problemas de propiedad intelectual, como los derechos de autor, la detección de plagio, ver a quién pertenecen textos anónimos, entre otros. Una detección automática y certera puede ayudar a evitar que las situaciones de acoso, mensajes terroristas, plagios escalen a mayores niveles, castigando de manera más rápida a los responsables.

La resolución de la Atribución de Autoría mediante un algoritmo que utilice las Redes de Palabras abre un nuevo enfoque a la hora de realizar caracterizaciones de textos; ya que hasta el momento no se ha realizado. Una aproximación para realizar la determinación de Autoría es a través de las características que ofrece la Estilometría, añadiendo las características que ofrecen las Redes de Palabras, como cálculo de caminos más cortos, costo de reconstrucción de una red, ver la conectividad de los vértices, entre otros, permitiendo generar un perfil más amplio de un autor. Como propuesta se desarrolló un algoritmo basado en Simrank, con unas adaptaciones llamándose Simrank Adaptado.

De los resultados obtenidos se llegó a la conclusión que la propuesta planteada no es apta para determinar la autoría de los textos de forma automática, debido a que todos los resultados tienen un porcentaje de acierto muy bajo, ninguno supera el 20% de acierto; considerando que en las pruebas realizadas con el set 2 de datos se limitó el contexto a una sola categoría, World, con el objetivo de reducir la variabilidad de las palabras obtenidas de los textos. La causa del bajo desempeño aún no se encuentra identificada, se ha pensado en la complejidad de los textos como una causa posible, ya que esto hace que la red de palabras crezca de manera exponencial, lo que hace que sea más costoso el procesar cada texto.

8. Referencias

[1] T. C. Mendenhall, «The characteristic curves of composition,» *Science, IX*, p. 237–49, 1887.

[2] G. Yule, «On sentence-length as a statistical,» *Biometrika*, vol. 30, pp. 363-390, 1938.

[3] G. Yule, The statistical study of literary vocabulary, Cambridge University Press, 1944.

[4] G. L. Zipf, Human behavior and the principle of least effort, Addison-Wesley, 1965.

[5] F. Mosteller y . D. Wallace, Inference and disputed autorship: The Federalist, Addison-Wesley, 1964.

[6] D. Holmes, «Authorship attribution. Computers and the Humanities,» 1994, p. 87–106.

[7] D. Holmes, «The evolution of stylometry in humanities scholarship. Literary and Linguistic,» 1998, pp. 111-117.

[8] J. Rudman, «The state of authorship attribution studies: Some problems and solutions.,» *Computers and the Humanities*, nº 31, pp. 351-365, 1998.

[9] L. a. B. Cedeño, «Detección automática de plagio en texto,» Valencia, 2008.

[10] D. R. Amancio, «Autorship recognition via fluctuation analysis of network topology and word intermittency,» Sao Paulo, 2015.

[11] T. Solorio, S. Pillay, S. Raghavan y M. Montes y Gómez, «Modality Specific Meta Features of Authorship Attribution in Web Forums Post,» de *Proceeding of the 5th International Joint conference of natural language processing*, Chiang Mai, Thailand, 2011.

[12] A. P. L. Monroy, «Atribución de Autoría utilizando Distintos tipo de Características A través de una Nueva Representación,» 2012.

[13] S. E., «A survey on modern authorship attribution methods.,» Journal of the American Society for Information Science and Technology, 2009.

[14] C. Z. A. &. B. ,. J. Figueroa, categorización automática de documentos en español, 2000.

[15] D. &. M. i. n. J. Jurafsky, Speech and language processing: An introduct ion to natural language, New Jersey: Prentice-Hall, 2000.

[16] F. Sebastiani, Machine Learning in Automated Text Categorization, 2002.

[17] R. F. i. C. y. R. Solé, The Small-World of Human Language, 2001.

[18] Milgran, The Small World Problem, 1967.

[19] J. M. A. T. I. &. b. R. cárdemas, Topological Cpmplexity in Natural and Formal Languages., 2011.

[20] G. Jeh y J. Widom, «SimRank: A Measure of Structural-Context Similarity,» Stanford University, 2002.

Anexo

Lista de Stop Words obtenidas de Ranks.nl; la lista empleada es la default del habla inglesa.

"a"	"did"	"herself"	"not"	"the"	"we've"
"about"	"didn't"	"him"	"of"	"their"	"were"
"above"	"do"	"himself"	"off"	"theirs"	"weren't"
"after"	"does"	"his"	"on"	"them"	"what"
"again"	"doesn't"	"how"	"once"	"themselves"	"what's"
"against"	"doing"	"how's"	"only"	"then"	"when"
"all"	"don't"	"i"	"or"	"there"	"when's"
"am"	"down"	"i'd"	"other"	"there's"	"where"
"an"	"during"	"i'll"	"ought"	"these"	"where's"
"and"	"each"	"i'm"	"our"	"they"	"which"
"any"	"few"	"i've"	"ours"	"they'd"	"while"
"are"	"for"	"if"	"ourselves"	"they'll"	"who"
"aren't"	"from"	"in"	"out"	"they're"	"who's"
"as"	"further"	"into"	"over"	"they've"	"whom"
"at"	"had"	"is"	"own"	"this"	"why"
"be"	"hadn't"	"isn't"	"same"	"those"	"why's"
"because"	"has"	"it"	"shan't"	"through"	"with"
"been"	"hasn't"	"it's"	"she"	"to"	"won't"
"before"	"have"	"its"	"she'd"	"too"	"would"
"being"	"haven't"	"itself"	"she'll"	"under"	"wouldn't"
"below"	"having"	"let's"	"she's"	"until"	"you"
"between"	"he"	"me"	"should"	"up"	"you'd"
"both"	"he'd"	"more"	"shouldn't"	"very"	"you'll"
"but"	"he'll"	"most"	"so"	"was"	"you're"
"by"	"he's"	"mustn't"	"some"	"wasn't"	"you've"
"can't"	"her"	"my"	"such"	"we"	"your"
"cannot"	"here"	"myself"	"than"	"we'd"	"yours"
"could"	"here's"	"no"	"that"	"we'll"	"yourself"
"couldn't"	"hers"	"nor"	"that's"	"we're"	"yourselves"