UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

IDENTIFICACIÓN DE AUTOR UTILIZANDO REPRESENTACIÓN DISPERSA

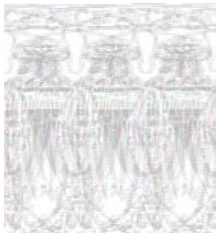
T E S I S

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

P R E S E N T A :

RODRIGO MARTÍNEZ ARZATE



DIRECTORES DE TESIS:
Dr. Ivan Vladimir Meza Ruiz
Dr. Gibrán Fuentes Pineda

Ciudad Universitaria, Cd. Mx., 2016

Índice General

Índice de Tablas

Índice de Figuras

Capítulo 1. Introducción

En esta tesis se propone el modelo de un sistema de identificación de autor que utiliza la clasificación basada en la representación dispersa, un marco de trabajo que ha resultado ser de gran interés debido a su capacidad para reconocer patrones. Lo que queremos lograr en este trabajo es averiguar la identidad del autor de un documento de texto comparándolo contra el estilo de escritura de varios documentos cuyo autor conocemos. Para esto, recurriremos a la hipótesis de que es posible reconstruir un documento con ayuda de la combinación de otros y que durante este proceso, podemos forzar a que la solución también nos indique cuáles son los documentos que están aportando más para la reconstrucción. Al contar con dichos valores de contribución también creemos que es posible descubrir cuál de ellos está reproduciendo con menos diferencias o errores al documento desconocido, con lo que finalmente podríamos determinar quién escribió el documento desconocido.

1.1 Motivación

La historia tal como la conocemos dio inicio en el momento en el que el hombre comenzó a documentar hechos e ideas ayudándose de dibujos y grafos que posteriormente se convirtieron en símbolos y alfabetos que hoy no sólo nos permiten conocer más del pasado sino también descubrir quiénes y cómo fueron los personajes que generaron tales documentos. Así, las formas en que el hombre comenzó a dejar huella y plasmar sus pensamientos fueron evolucionando de la pared de cuevas hasta el papel y la tinta, y de ahí dieron un gran salto hasta los documentos digitales; y no sólo eso, sino que el número de estos documentos se multiplicó exponencialmente con la llegada de computadoras personales más asequibles y la creación de redes de comunicación como Internet.

Con el establecimiento de estas tecnologías, se abrieron las posibilidades a nuevas amenazas como el robo de identidad, fraudes electrónicos y el plagio. Afortunadamente, también la ciencia se ha abierto paso a través de los años y ha llegado a entender algunas de estas amenazas y a crear formas de defendernos de ellas. Como parte de este proceso, se dio la generación de una nueva rama de la ciencia conocida como ciencias forenses para textos digitales.

Para ubicar esta nueva ciencia imaginemos el siguiente escenario, un hombre es encontrado sin vida en un departamento. Junto a él, una carta indica que se trata de un suicidio. Los médicos forenses recorren la escena y recopilan evidencia, entre ésta se encuentra la carta y también el diario personal del mismo hombre. Tras comparar el estilo de escritura de la carta de 'despedida' con el de las entradas del diario utilizando métodos de clasificación automática, los peritos determinan que la carta no fue escrita por él; por consiguiente, podría tratarse de un homicidio. Este es un ejemplo donde la verificación de autor se torna importante y marca una diferencia añadiendo información útil a un problema real.

Un ejemplo donde también se expone la verificación de autor pero en textos más extensos y complejos data del mes de Julio del año 2013, cuando el lanzamiento de la novela ‘El Canto del Cuco’ [1] llamó la atención de la prensa y lectores después de que una llamada anónima al diario inglés Sunday Times diera una pista de que el libro había sido escrito en realidad por la autora de la famosa saga del mago

Harry Potter, la escritora J. K. Rowling, quien habría utilizado el seudónimo de Robert Galbraith. Tras la llamada, el diario habría contactado al experto forense en análisis de textos Patrick Juola [2] para corroborar la identidad del verdadero autor del libro. Después de una intensa investigación y el uso de herramientas para el análisis de textos, el experto llegó a la conclusión de que existía suficiente evidencia como para afirmar que Rowling había escrito el libro y en efecto, la misma Rowling lo habría confirmado días después.

Con estos ejemplos nos damos cuenta de que alrededor nuestro y a diario, nos topamos con colecciones de datos increíblemente grandes que a pesar de estar a nuestro alcance pasan desapercibidos al igual que la información que está contenida en ellos. El extraer esta información también es una de las motivaciones de este trabajo. Los documentos contienen información más allá de lo que las palabras en ellos están contando. Información que nos podría decir algo sobre el autor mismo del texto. Podría decirse que la escritura de cada persona cuenta con un patrón propio, un estilo muy particular que se puede representar de una manera más abstracta para compararlo con el de otras personas.

Aquí es donde comenzamos a lanzar las primeras preguntas. ¿Cómo podemos saber si un documento fue escrito por una persona? ¿Cómo comparamos documentos unos con otros? ¿Cómo definimos lo que es un documento?¿Cómo representamos matemáticamente un documento para manejarlo y operar sobre él?

Recordemos que el trabajo aquí presente así como millones de archivos que viajan por Internet de un servidor a otro, son documentos con texto que fueron concebidos en la creatividad de un usuario, tecleados letra por letra y almacenados digitalmente en el disco duro de una computadora. Entonces ¿por qué no utilizar estas representaciones digitales ya existentes para llevar a cabo estudios periciales y de investigación lingüística?

Aunque es claro que podemos utilizar estas representaciones, comparar un texto con otro, letra por letra o palabra por palabra, nos limita bastante y apenas nos permite saber si estos textos son similares en su totalidad o en fragmentos. Aquí es donde necesitamos cambiar un poco nuestro punto de vista y dejar de comparar documento contra documento y comenzar a comparar objetos, modelos más

abstractos que sean capaces de encapsular la misma información pero de una forma que podamos aprovecharla mejor. Dentro del área de búsqueda y recuperación de información, existen varios modelos que han sido creados con el fin de abstraer información y darle una representación con la que podamos operar. Desde modelos teóricos con bases booleanas y modelos probabilísticos, hasta modelos algebraicos han sido utilizados como estrategia para obtener información relevante a partir de colecciones de datos.

En 1975, Gerard Salton se dio cuenta de que no era práctico intentar representar un documento utilizando la totalidad de sus palabras y junto con A. Wong y C. S. Yang, propuso el modelo espacial de vectores [3] para textos donde por primera vez se estudiaba la idea de comparar dos documentos y cuantificar su similitud con el uso de vectores, pesos y el producto punto. En su artículo titulado 'Un modelo espacial vectorial para indexado automático' [3], Salton define un espacio de documentos compuesto por documentos D_i, cada uno formado por t términos representados por un peso d_{ij} de la siguiente forma:

$$D_i=(d_{i1,}d_{i2,}...,d_{it}) \tag{1}$$

Con esta representación ya definida, Salton explica que es posible medir la similitud entre dos documentos y calcular un valor al que él llama *coeficiente de similitud*, el cual puede ser calculado mediante el producto punto entre dos vectores:

$$s(D_1,D_2)=D_1\cdot D_2=d_{11}d_{21}+d_{12}d_{22}+d_{13}d_{23}+...+d_{1t}d_{2t} \tag{2}$$

El resultado de (2) no es otro vector, sino un valor numérico o escalar que nos indica qué tan parecidos son dos vectores en términos de su magnitud y dirección. Otra alternativa para calcular la similitud entre dos vectores es el coseno, el cual basa la puntuación de similitud únicamente en la dirección a la que están apuntando los dos vectores por lo que para esta decisión no toma en cuenta la magnitud de los vectores de los documentos y no es más que una reinterpretación de la fórmula del producto punto como a continuación se muestra :

$$D_1 \cdot D_2 = \|D_1\| \|D_2\| \cos(\theta) \quad (3)$$

Resolviendo (3) para coseno de θ, obtenemos la segunda fórmula de similitud propuesta por Salton:

$$s(D_1, D_2) = \cos(\theta) = \frac{D_1 \cdot D_2}{\|D_1\| \|D_2\|} \quad (4)$$

La fórmula en (4) nos entrega valores dentro del rango de [-1,1] que son resultado de calcular el coseno del ángulo θ entre los dos vectores. Bajo este esquema, el ángulo θ entre los dos vectores debería ser cero o un valor muy cercano a cero para documentos muy parecidos. En cuyo caso el coseno del ángulo θ resultaría ser 1 o un valor muy cercano a 1. En caso contrario, para documentos muy diferentes el ángulo θ entre ellos aumenta provocando que el coseno de θ tienda a valores cercanos a 0. Para que tanto el cálculo del coseno de θ como el del producto punto coincidan dando resultados dentro del rango de [-1,1], Salton sugiere la normalización unitaria de los vectores. Posteriormente propone la proyección de ellos en la superficie de la esfera unitaria de tal manera que cada documento quede representado por un punto en el espacio donde el documento en cuestión toca la 'cubierta' de la esfera de tal forma que dos documentos con términos similares representados por dos puntos quedarían muy cerca uno del otro.

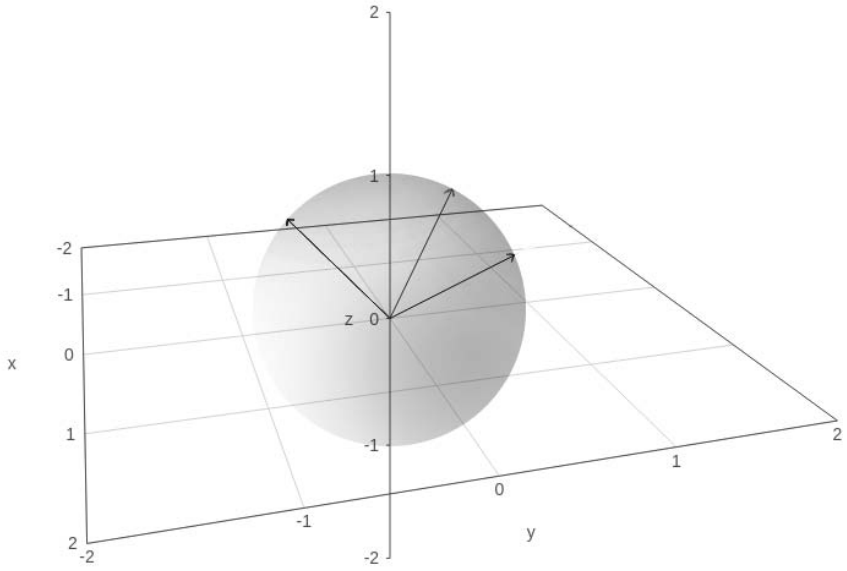


Figura 1: Representación vectorial de documentos y su proyección en la esfera unitaria

El siguiente problema es elegir los términos para representar los documentos. En un inicio, las palabras presentes en un documento eran utilizadas como términos dentro de los vectores de tal manera que cada palabra estaba representada por una entrada en el vector con un valor positivo. Sin embargo, sabemos que dos documentos no contienen siempre las mismas palabras ni la misma cantidad de ellas. De esta forma, si un documento cuenta con 200 palabras y otro con 900, para compararlos es necesario crear un diccionario donde estén contenidas todas las palabras comunes; i.e., que estén presentes en la unión de ambos documentos. Desgraciadamente, para alguno de los dos documentos, usualmente el que cuenta con menos palabras, muchos de sus términos quedarían en cero haciendo que esta representación genere muchos vectores dispersos, que son aquellos cuya mayoría de entradas son cero.

0	0	0	1	0	0	•••	1	0	0	0	0	0

Figura 2: Vector disperso

Otro problema es que regularmente, un documento de texto cuenta con muchas palabras comunes (*stop words*) como artículos, pronombres, conjunciones y signos de puntuación, que si bien ayudan a hilar las ideas en un texto haciéndolo más fácil de entender, es posible obtener buenos resultados en la comparación de textos sin utilizarlas:

Muy lejos, **más allá de las** montañas **de** palabras, alejados **de los** paises **de las** vocales **y las** consonantes, viven **los** textos simulados. Viven aislados **en** casas **de** letras, **en la** costa **de la** semántica, **un** gran océano **de** lenguas. **Un** riachuelo llamado Pons fluye **por su** pueblo **y los** abastece **con las** normas necesarias. Hablamos **de un** país paraisomático **en el que a** uno **le** caen pedazos **de** frases asadas **en la** boca. **Ni** siquiera **los** todopoderosos signos **de** puntuación dominan **a los** textos simulados; **una** vida, **se** puede decir, poco ortográfica.

Figura 3: Palabras comunes marcadas en negritas en un texto de muestra

Este par de problemas sugiere que se debe realizar una elección más frugal y representativa al momento de descomponer un texto y convertirlo en un vector. Es por eso que a través de la constante experimentación, los expertos han llegado a la conclusión de que existen 3 factores importantes que deben ser considerados para la elección de los términos o pesos que forman parte de la representación vectorial de un texto [4]:

- La frecuencia de términos: Es decir, tomar en cuenta el número de veces que una palabra es mencionada en un documento.
- La frecuencia inversa de documento: A veces tomar en cuenta únicamente la frecuencia de términos no es lo más conveniente. Existen casos en los que las palabras más frecuentes se encuentran presentes en todos y cada uno de los documentos de una colección ocasionando que todos los documentos aparentemente resulten similares. Es por eso que es necesario introducir un segundo factor que dependa del contexto particular de la colección con el fin de darle mayor peso a palabras muy frecuentes pero que sólo se encuentran concentradas en una cantidad pequeña de documentos de la colección. Este factor es conocido como la *frecuencia inversa de documento* y usualmente se multiplica con la *frecuencia de términos* con el objetivo de generar un peso

más razonable para cada término.

- La normalización del tamaño de los documentos: Cuando contamos con documentos muy grandes, la probabilidad de que cualquier término o palabra aparezca en uno de ellos es alta. Esto ocasiona que al buscar una palabra en la colección de documentos, los documentos con el mayor número de términos usualmente obtengan mejores puntuaciones de similitud por encima de los documentos con un menor número de palabras. Es por eso que es necesario normalizar la frecuencia de cada término dentro de un documento de acuerdo al número de palabras presentes en el mismo documento.

Ahora que hemos visto que es posible extraer el estilo de escritura de un documento de texto y convertirlo en una representación fácil de manejar, es necesario revisar cómo planeamos comparar estilos y determinar la autoría de los documentos. Para ello, sabemos que por un lado contamos con un documento de texto cuyo autor desconocemos y por otro, un conjunto de documentos cuyos autores sí conocemos. Lo que nos gustaría saber es si existe una manera de combinar todos los documentos conocidos y reproducir al documento desconocido a partir de ellos y además, saber cuál de todos ellos ayudó más en la reproducción. Afortunadamente existe una manera. En el año 2009, Wright et. al. [5] propusieron un modelo en el que describen una forma para reproducir la imagen del rostro de una persona a partir de la combinación de las imágenes de otros rostros y no sólo eso, sino que también una manera de descubrir la identidad del rostro midiendo la contribución de cada uno de los rostros involucrados en la reproducción.

En el método de Wright et. al. [5], se opta por desdoblar cada imagen y colocar un pixel tras otro dentro de arreglos verticales de una dimensión que son colocados uno junto a otro formando una malla bidimensional. Esta malla es una matriz que representa el espacio generado por todos los rostros del conjunto conocido. Tras definir este espacio, se explica que el siguiente paso es descubrir si el rostro desconocido vive en este espacio de muchos rostros. Para esto, es necesario definir un sistema de ecuaciones a partir de la matriz de rostros y resolverlo. De existir una solución, de forma inherente ésta nos indicará qué proporción se requirió

tomar de cada uno de los rostros conocidos para reproducir el rostro desconocido. Encontrar tal solución puede ser fácil, pero aquí es donde entra en acción uno de los puntos centrales de este trabajo, el principio de parsimonia. En pocas palabras, el principio de parsimonia también conocido como la navaja de Ockham, establece que la explicación más sencilla a un problema usualmente es la más probable [6]. Dado que se requiere una solución, pero no sólo eso sino también la más simple; i.e., la que menos contribuciones haya requerido de otros rostros para reproducir el rostro desconocido, Wright et. al. [5] eligieron un método conocido como Homotopía, que es capaz de obtener una respuesta de esta naturaleza. Para efectos de este trabajo, a esta respuesta la denominaremos como la representación dispersa de una señal.

Habiendo definido una forma para representar los documentos de texto y un método para reconstruir documentos de texto a partir de otros documentos, sólo nos queda combinar ambos y utilizar como señales de entrada nuestra representación vectorial de textos en lugar de las imágenes de rostros de Wright et. al. [5] y esto básicamente se convierte en nuestra propuesta inicial para resolver el problema de identificación de autores.

1.2 Objetivo General

El objetivo general del trabajo aquí presente es lograr la identificación de la autoría de un conjunto de documentos distribuidos en diversos géneros literarios e idiomas, utilizando la clasificación basada en la representación dispersa como método principal.

1.3 Objetivos Específicos

Para alcanzar tal objetivo, hemos estudiado el proceso y la serie de pasos por los que deberemos pasar y así se han identificado algunos de los objetivos secundarios que nos ayudarán a cumplirlo.

Definir la forma en que extraeremos y seleccionaremos los atributos más

importantes que serán utilizados para la representación de los datos, que en nuestro caso es un corpus de documentos de texto clasificados bajo diversos géneros literarios e idiomas.

Obtener un conjunto de datos o corpus lo suficientemente grande y con información consistente que sirva para entrenar el algoritmo de clasificación correctamente que al mismo tiempo conserve un rendimiento aceptable con datos reales y nuevos ajenos al corpus de entrenamiento.

Para implementar el algoritmo de clasificación basado en la representación dispersa, deberemos implementar el método de Homotopía que nos permitirá reducir el problema a la elección de una solución dispersa.

1.4 Estructura de la tesis

El trabajo aquí presente tiene la estructura que a continuación presentaremos:

En el Capítulo 2. se presenta y revisa el estado del arte de métodos para la identificación de autores como son el método de Impostores, el método general de Impostores y el método de impostores modificado de auto-verificación ASGALF. También se aborda el estado del arte respecto a la representación vectorial de textos.

En el Capítulo 3. se habla sobre la clasificación basada en la representación dispersa, donde se explica a detalle cada una de las fases de esta técnica, sus condiciones y se aborda brevemente el método de Homotopía. También se describe un ejemplo de la aplicación de este método en la identificación de rostros.

En el Capítulo 4. se describe de forma análoga al capítulo anterior, la aplicación de la clasificación basada en la representación dispersa para la identificación de autores. También se describe la naturaleza del corpus de documentos utilizado durante las fases de desarrollo y pruebas de este trabajo.

En el Capítulo 5. se presentan los resultados de los experimentos realizados aplicando el sistema de identificación de autores sobre el corpus de documentos, así como las métricas utilizadas para calificar los resultados obtenidos.

En el Capítulo 6. se presentan las conclusiones del trabajo realizado, se explican algunas de las mejoras que se podrían realizar al sistema y el trabajo a futuro.

Capítulo 2. Estado del arte

La tarea computacional de responder quién es el autor de un texto determinado se ha convertido en un problema de aprendizaje automático y lingüístico ampliamente estudiado y de gran interés con el surgimiento de nuevos problemas en áreas como el periodismo, seguridad, las ciencias forenses, el derecho y la investigación literaria.

En un lapso menor a 30 años, el hombre visualizó y puso en marcha una de las maravillas modernas del mundo tecnológico: Internet. Con ello, inició una nueva era en la que nuevas amenazas y problemas también llegaron. De un día a otro, cualquier persona podía escribir y documentar información, historias, hechos, ciencia, noticias y generar contenido. La aparición de los medios digitales y la capacidad que ganaron los autores para 'auto' publicarse, también impulsó la creación de herramientas que aumentaron la facilidad con la que se pueden copiar y reproducir de forma ilícita desde documentos oficiales hasta artículos noticiosos y de investigación.

Un ejemplo de esta situación ocurre con las llamadas *granjas de contenidos*, compañías que contratan a un gran número de trabajadores independientes para producir grandes volúmenes de contenido textual como noticias y artículos de entretenimiento con el fin de generar ingresos masivos provenientes de la publicidad colocada en los sitios web de estos contenidos. La forma en que operan estas compañías es a través del reciclaje y mezcla de artículos ya existentes. Estas empresas usualmente no cuentan con reporteros de campo que crean contenidos originales, sino que piden a su equipo editorial recopilar artículos de otros periódicos, leerlos, entenderlos y reescribirlos. En muchas ocasiones, ni siquiera son reescritos, sino únicamente copiados por fragmentos y mezclados con fragmentos de otros artículos relacionados con la misma noticia o tema.

Desgraciadamente, para algunos motores de búsqueda como *Google* o *Bing*, identificar estas *granjas* y excluirlas de sus directorios es complicado debido a que se requiere de herramientas sofisticadas que permitan diferenciar el reciclaje, mezcla y plagio de contenidos de los contenidos originales.

Es por esto que la existencia de conferencias y competencias como el Taller para el Descubrimiento del Plagio, la Autoría y el Abuso del Software Social (PAN por sus siglas en inglés) son necesarias para el avance científico en esta área y presentan metodologías modernas cada vez más eficaces cuya revisión es vital para la explicación del trabajo aquí propuesto. La competencia PAN se ha enfocado en la solución de tres tareas: la Detección de Plagio, la Identificación de Autor y la Evaluación de Perfil de Autor [7]. El tema y aplicación de esta tesis se enfoca en el segundo rubro.

Básicamente, el objetivo de la tarea de identificación de autor es determinar si el autor de un conjunto de documentos dado también es el autor de un documento ajeno al conjunto.

¿El documento A es del Autor B? Sí o No

Los modelos para esta tarea se clasifican en intrínsecas y extrínsecas. Los modelos intrínsecos se encuentran basados únicamente en un conjunto de documentos de autor conocido y un documento de autoría desconocida.

Por su parte, los modelos extrínsecos utilizan recursos externos como documentos adicionales de otros autores tomados de un corpus de entrenamiento o descargados de la red.

En este capítulo revisaremos al modelo ganador de la competencia PAN 2013. Este modelo se encuentra clasificado como extrínseco, se encuentra basado en el método de Impostores y fue desarrollado por Shachar Seidman [8].

En la tabla 1 podemos observar los resultados finales de cada equipo en las evaluaciones de la tarea de Identificación de Autor durante la competencia de PAN 2013 [9]. En la primera columna se muestra el lugar en el que cada equipo concluyó, en la segunda columna se muestra el nombre del equipo que compitió y en la tercera se muestra la puntuación final con la que concluyó cada equipo:

Lugar	**Equipo**	**F_1**
1°	Seidman	0.753
2°	Halvani *et al.*	0.718
3°	Layton *et al.*	0.671
3°	Petmanson	0.671
5°	Jankowska *et al.*	0.659
5°	Vilariño *et al.*	0.659
7°	Bobicev	0.655
8°	Feng&Hirst	0.65
9°	Ledesma *et al.*	0.61
10°	Ghaeini	0.606
11°	van Dam	0.600
11°	Moreau&Vogel	0.600
13°	Jayapal&Goswami	0.576
14°	Grozea	0.553
15°	Vartapetiance&Gillam	0.541
16°	Kern	0.529

Lugar	Equipo	F_1
	LÍNEA BASE	0.500
17°	Veenman&Li	0.417
18°	Sorin	0.331

Tabla 1: Resultados generales y posicionamiento de equipos participantes en Author ID PAN 2013.

Esta puntuación final representada por F_1 en la 1 fue calculada con la siguiente fórmula:

$$F_1 = 2 \cdot \left(\frac{precisión \cdot exhaustividad}{precisión + exhaustividad} \right)$$

donde el valor de precisión se obtiene a través de la fórmula que presentamos a continuación [10]:

$$Precisión = \frac{|documentos\,relevantes \cap documentos recuperados|}{|documentos\,recuperados|}$$

y el valor de exhaustividad está dado por la siguiente fórmula:

$$Exhaustividad = \frac{|documentos\,relevantes \cap documentos recuperados|}{|documentos\,relevantes|}$$

Lo que el resultado del cálculo de estas fórmulas nos indica es qué tan bien un modelo está clasificando documentos de texto en términos de falsos positivos y falsos negativos. Los falsos positivos son aquellos casos en los cuales un sistema determinado concluye que el documento A fue escrito por el autor B, cuando en realidad no lo fue. Mientras que los falsos negativos son aquellos casos en los cuales el sistema concluye que el documento A no fue escrito por B, pero en realidad sí lo fue. Por otro lado, los verdaderos positivos son aquellos casos en los que se determina que el documento A fue escrito por el autor B y la predicción es correcta. Así mismo, los verdaderos negativos son aquellos casos en los que se determina que el documento A no fue escrito por el autor B y la predicción también es correcta.

Así, bajo estos términos, el valor de precisión también puede definirse bajo la siguiente fórmula:

$$Precisión = \frac{verdaderos\ positivos}{verdaderos\ positivos + falsos\ positivos} \quad (5)$$

Y el valor de exhaustividad por la siguiente fórmula:

$$Exhaustividad = \frac{verdaderos\ positivos}{verdaderos\ positivos + falsos\ negativos} \quad (6)$$

Por ejemplo, para entender mejor imaginemos que contamos con un conjunto de 10 documentos cuyos autores, quienes son desconocidos para el sistema, necesitan ser verificados contra el autor conocido de otro documento D. Por otra parte, aunque el sistema lo ignora, nosotros sabemos que cinco documentos de la colección sí fueron escritos por el autor del documento D. Después de analizar los 10 documentos, nuestro sistema concluye que 6 de ellos fueron escritos por el mismo autor del documento D, de esas 6 predicciones sólo cinco resultaron correctas. Entonces podemos decir que nuestro sistema concluyó la tarea con cinco verdaderos positivos, un falso positivo y cero falsos negativos. Lo cual nos da una precisión de 5/6 y un valor de exhaustividad de 5/5. Así, la precisión nos está diciendo 'que tan útiles' son los resultados del sistema y la exhaustividad nos dice 'que tan completos' son los resultados arrojados por el sistema. Con estos dos términos entendidos, podemos interpretar la puntuación F_1 como el promedio ponderado de la precisión y la exhaustividad y donde F_1 alcanza su mejor valor en 1 y su peor valor en 0.

Al multiplicar 5/5 y 5/6 tenemos que $F_1 = 0.83333$, valor cercano al 1 y que nos indica que el rendimiento del sistema del ejemplo es bueno.

2.1 Método de impostores (IM)

En la tarea de Identificación de Autor, se ha observado que el método de los Impostores, propuesto por Moshe Koppel y Yaron Winter en 2011 [11], es uno de los enfoques que mejores resultados y desempeño ha obtenido. Su aparición en las implementaciones ganadoras de las últimas dos ediciones de PAN 2013 y 2014 para la tarea Identificación de Autor, ha demostrado que el acercamiento

extrínseco del método es primordial y bastante prometedor para el desarrollo de futuros sistemas cuya tarea sea identificar autores con mayor precisión y fidelidad.

En la figura 4 es posible apreciar la estructura del método, sus entradas y sus posibles salidas, así como el flujo de la información a través de este. Por otro lado, para explicar la figura 4 así como el contenido de la *caja negra* donde está contenido el método de Impostores, también revisaremos el algoritmo del método utilizando pseudo código y haciendo una breve explicación paso por paso.

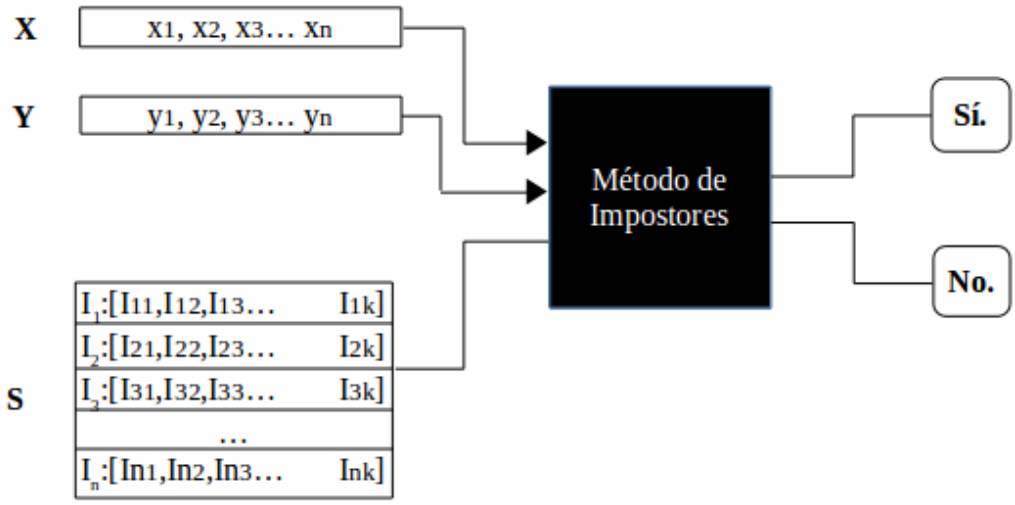


Figura 4: Entradas y salidas del método de Impostores

Algoritmo del Método de Impostores

Entrada: <X,Y>: Un par de documentos. S: Un conjunto de impostores.

Salida: <mismo-autor> o <diferente-autor>

1. Inicializar Puntuación = 0
2. Repetir k veces
   a. Escoger aleatoriamente la tasa porcentual de características de la colección de características completa.
   b. Escoger aleatoriamente n impostores de S: I1,…,In.
   c. Si:

$$similitud(X,Y)*similitud(Y,X) > similitud(X,I_i)*similitud(Y,I_i)\ \forall\, i\in\{1,\ldots,n\}$$

   Entonces: Calcular $puntuación = puntuación + \frac{1}{k}$

```
3. Si: puntuación>Δ*
      Entonces: Regresar <mismo-autor>
      De lo contrario: Regresar <diferente-autor>.
```

A continuación se describen los pasos generales del método de impostores:

- Aplicamos una representación vectorial de texto a todos los documentos. Esta representación, que será explicada en la siguiente subsección de este capítulo, permite la extracción de características especiales conocidas como atributos, que son esenciales para la clasificación, medición y comparación de los textos.
- Sean X y Y un par de documentos. Se desea establecer si X y Y pertenecen al mismo autor. Para ello, se construye un conjunto S de impostores a partir de una búsqueda realizada en la web con la que se reúnen documentos de género e idioma similares al de X y Y.
- El documento impostor I_i ϵ S, es comparado contra el documento X para saber qué tan parecido es uno al otro a través del calculo de la fórmula de similitud de coseno.

$$similitud(X, I_i) = \cos(X, I_i) = \frac{X \cdot I_i}{\|X\| \|I_i\|}$$

- El documento impostor I_i es comparado con el documento Y y se calcula la similitud entre ellos. Así como también se calcula la similitud entre el documento X y Y.
- Si la similitud entre X y Y es mayor que la similitud combinada del par (X, I_i) y el par (Y, I_i), entonces se suma la fracción 1/k a la puntuación, donde *k* es igual al número de iteración actual del algoritmo
- Al término de las *k* iteraciones, si la puntuación supera el umbral Δ*, el algoritmo responde que los documentos X y Y fueron escritos por el mismo autor. En caso contrario, el método responde 'no'.

En la práctica, lo que el método de impostores permite es facilitar la comparación

entre los vectores de los documentos X y Y. Para determinar qué tanto se parecen los vectores, se recurre a la comparación de la distancia entre ellos como podemos observar en la figura 5.

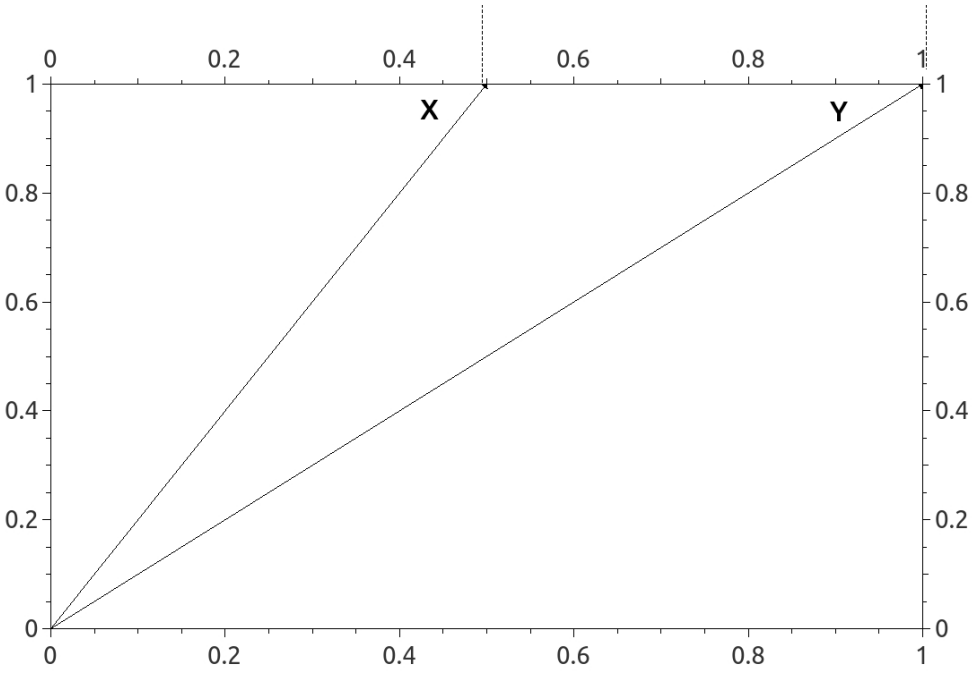


Figura 5: Representación vectorial de documentos X y Y

En la figura 5, observamos dos vectores X y Y y la distancia entre ellos. Desafortunadamente, aunque es posible medir la distancia entre ellos, ésta resulta ambigua; y sin un contexto o referencia, el responder si X y Y son similares o decir si esta distancia es grande o pequeña se convierte en un nuevo problema. Aquí es donde el rol de los impostores se torna importante.

Pongamos atención a la figura 6, donde al integrar los impostores al plano, ahora contamos con diversas referencias y creamos un contexto. Con este contexto ahora es posible decir si la distancia en X y Y es grande o pequeña con relación a la distancia entre X y el resto de los impostores y entre Y y el resto de los impostores. En este caso, la distancia entre X y Y es muy pequeña en comparación al resto de los impostores. Por lo que podríamos suponer que X y Y son muy parecidos y pertenecen al mismo autor.

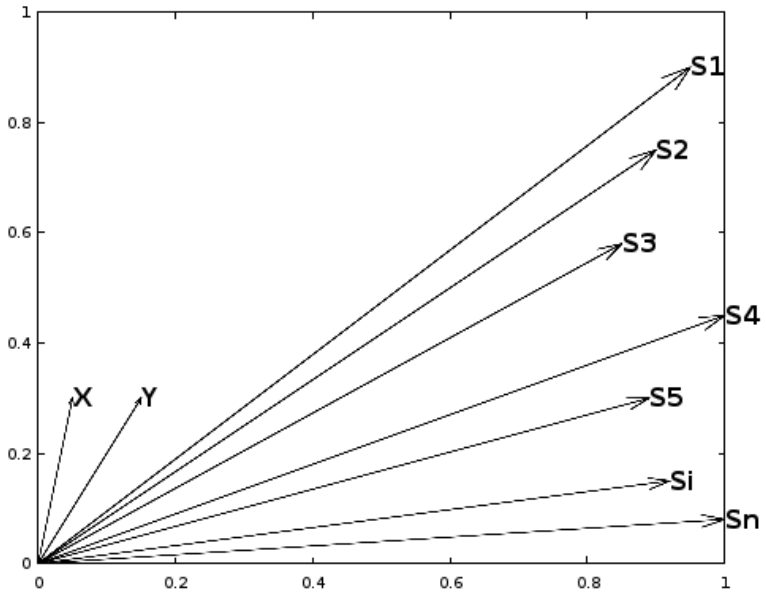


Figura 6: Representación vectorial de documentos X, Y e impostores

En el otro escenario ilustrado en la figura 7, la distancia entre X y Y es grande y el documento X aparentemente se encuentra más cerca de los impostores que de Y, por lo que se podría deducir que los autores de los documentos X y Y son diferentes:

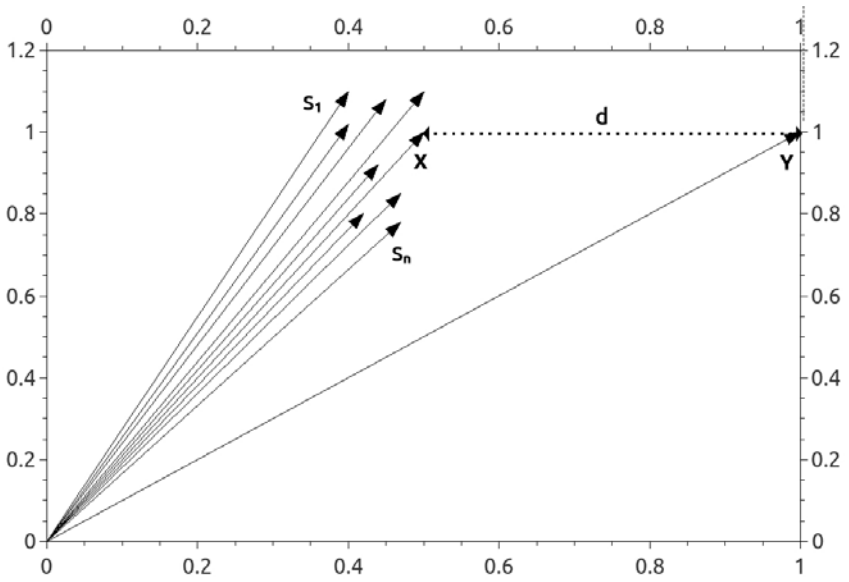


Figura 7: Representación vectorial de documentos X, Y e Impostores

Hasta ahora, el método de Impostores que hemos revisado sólo es capaz de comparar 2 documentos. Para la identificación de autor propuesta en este trabajo, nosotros requerimos de la verificación de un documento contra muchos. Para cumplir con este requerimiento necesitaremos una generalización del método de impostores conocida como el método general de Impostores.

2.2 Método General de Impostores (GenIM)

En la edición 2013 de PAN, el método ganador de la tarea de Identificación de Autor [8] presentado bajo el nombre de Método General de Impostores (GenIM), utilizó como base el algoritmo original de impostores con algunas mejoras que le dieron soporte para diferentes idiomas y para llevar a cabo una comparación generalizada.

En un inicio el equipo asumió que el método original de impostores necesariamente era independiente del lenguaje; sin embargo, se llegó a la conclusión de que cada uno de sus parámetros debía ser optimizado para diferentes idiomas de manera separada. Al mismo tiempo, el equipo generalizó la

implementación del método original para que éste comparara un documento contra un conjunto de documentos de la forma más efectiva posible, en lugar de hacerlo únicamente contra un solo documento.

Como podemos observar en la figura 8, el cambio más notable y simple fue encapsular el Método de Impostores en un ciclo o bucle para la comparación del vector X en cuestión contra cada uno de los vectores de los documentos contenidos en el conjunto Y, cuyo autor es conocido y el mismo para todos ellos.

A grandes rasgos el método general de impostores se describe a continuación:

- Sean X un documento cuyo autor es desconocido y Y un conjunto de documentos cuyo autor es conocido. Se desea conocer si el autor del documento X coincide con el autor de los documentos en el conjunto Y.
- Se ejecuta el método original de impostores sobre cada una de las parejas en turno formadas por el documento X y cada uno de los elementos en Y. Es decir la dupla (X, Yi). Obteniendo la puntuación de similitud entre cada pareja.
- Después se calcula el promedio de todas las puntuaciones de similitud obtenidas con cada pareja.
- Finalmente, si el promedio sobrepasa el umbral θ* predefinido, el algoritmo responde *Sí* y determinamos que el autor del documento X es el mismo autor que el de los documentos del conjunto Y. En caso contrario, responde *No*.

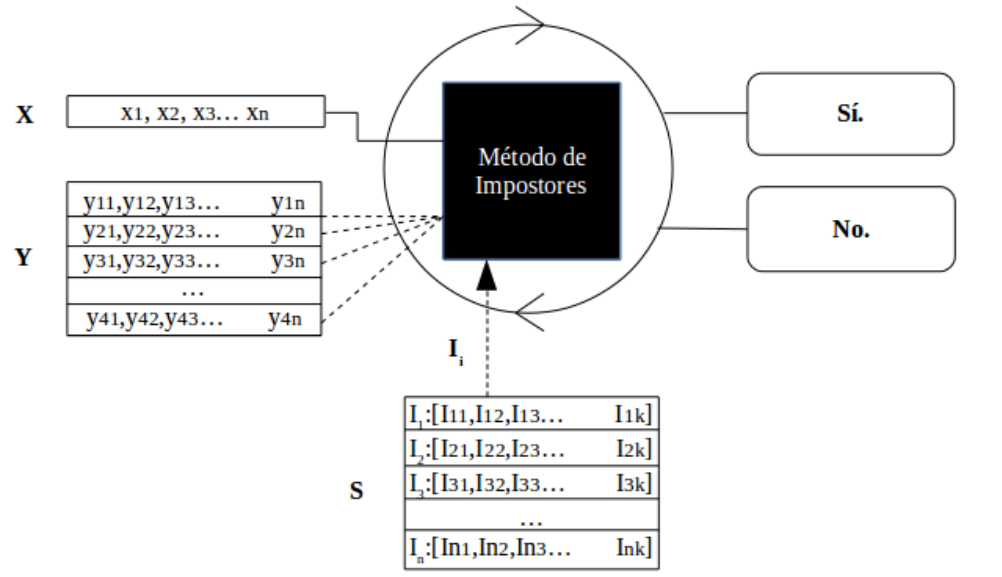


Figura 8: Entradas y salidas del método general de impostores

Algoritmo del Método General de Impostores

Entrada: X: Documento cuyo autor es desconocido . Y = {Y1,…,Yn}: Conjunto de documentos cuyo autor es conocido.

Salida: <mismo-autor> o <diferente-autor>.

1. Para cada par de documentos <X, Yi> en el conjunto D:
   a) Ejecutar el Método de Impostores original para obtener una puntuación binaria de similitud S(X, Yi).
2. Calcular Puntuación = Promedio de las puntuaciones de similitud en ([S(X, Y_1)… S(X, Y_n)]).
3. Regresar <mismo-autor> si puntuación > θ*; en caso contrario <diferente-autor>.

Es importante destacar que Sachar Seidman [8] utilizó el llamado Modelo Espacial de Vectores para representar cada documento como vectores de características, que como ya hemos mencionado veremos a detalle más adelante en este mismo capítulo. También para cada idioma evaluaron el sistema con diferentes conjuntos de características o atributos hasta que obtuvieron el más adecuado; y probaron diferentes medidas de similitud o 'distancias' como la Euclidiana o la Manhattan, con las cuales discernieron qué tan diferentes eran dos documentos observando qué tan distantes se encontraban sus vectores respectivos en el plano.

Para elegir el conjunto S de impostores, el equipo de GenIM [8] recurrió a un motor de búsquedas en Internet y descargó un corpus de impostores por cada idioma. Para hacer las búsquedas, se escogieron de 3 a 4 documentos 'semilla' y de ellos conjuntos aleatorios de 3 a 5 palabras. De los resultados obtenidos en cada búsqueda, eligieron los primeros 10 documentos para añadirlos al corpus de impostores repitiendo el procedimiento hasta obtener un corpus suficientemente grande.

2.3 Método ASGALF

Para la edición de PAN 2014, un equipo propuso una variación del método general de impostores [12] presentado en el 2013, a la que denominaron ASGALF (A Slightly-modified GI Author-verifier with Lots of Features), el cual ganó la competencia.

En este método el equipo ASGALF implementó mejoras en la métrica de similitud que se utiliza para comparar dos documentos y agregaron un gran conjunto de atributos relacionados con el estilo de escritura de los autores, conteos de palabras y prefijos, entre otros.

Gracias a los resultados de ediciones pasadas de PAN y pruebas preliminares, el equipo ASGALF llegó a la conclusión de que el uso de un conjunto de impostores definitivamente agrega información útil que realmente aumenta precisión al momento de decidir qué tan similares son dos documentos.

Como aprendimos en la subsección 2.1, medir simplemente las similitudes entre dos vectores o documentos X y Y no es suficiente ya que es necesaria la introducción de documentos impostores para crear un contexto en el cual sea posible determinar la similitud de X y Y. Cuando agregamos los documentos impostores y los comparamos contra los documentos X y Y, el modelo ahora tiene una mejor idea del contexto sobre el cual se está queriendo determinar qué tan parecidos son X y Y. Por lo que es capaz de dar una respuesta más adecuada al problema.

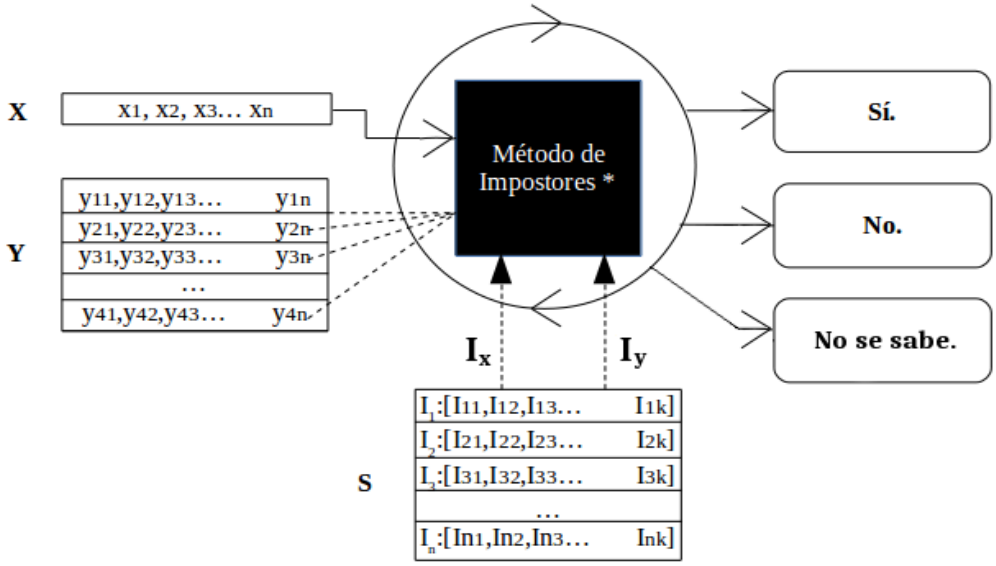


Figura 9: Entradas y salidas del método ASGALF

En la figura 9, podemos distinguir las diferencias entre el método GenIM y ASGALF. La diferencia más notoria es la adición de una tercera salida denominada como 'No se sabe' para los casos en el que el sistema no pudo determinar con certeza una respuesta para el problema. La otra diferencia es el uso de I_x e I_y, vectores que corresponden a los impostores más parecidos a X y a Y respectivamente.

Para aclarar el funcionamiento del algoritmo, a continuación describimos paso a paso el método ASGALF:

- Sean X un documento cuyo autor es desconocido y Y un conjunto de documentos cuyo autor es conocido. Se desea saber si el autor del documento X coincide con el autor de los documentos en el conjunto Y.
- Se ejecuta el método general de Impostores con una modificación en el cálculo de la puntuación que más que determinar si X y Y se parecen lo suficiente, mide qué tanto se parecen. La formula de puntuación utiliza el cálculo de la función min-max y los vectores I_x e I_y, impostores más parecidos a X y Y respectivamente.

- Finalmente, dependiendo de los valores de la puntuación, se determina que el autor del documento X es el mismo autor que el de los documentos del conjunto Y para el caso en el que la puntuación cae en el rango de valores comprendidos en [0,0.5), que no se sabe para el caso en el que la puntuación es igual a 0.5 y que X y Y son de diferentes autores para el caso en el que la puntuación cae en el rango (0.5,1].

Algoritmo del Método ASGALF

Entrada: X: Documento cuyo autor es desconocido . Y = {Y1,…,Yn}: Conjunto de documentos cuyo autor es conocido.

Salida: <mismo-autor>, <diferente-autor> o <no-se-sabe>.

1. Para cada par de documentos <X, Yi> en el conjunto D:

   a) Ejecutar el Método de Impostores original para obtener una puntuación binaria de similitud S(X, Yi).

2. Calcular Puntuación = $Puntuación + \frac{min-max(X,Y)^2}{min-max(X,I_x)\times min-max(Y,I_y)}$

   Donde I_x es el impostor más parecido a X e I_y es el impostor más parecido a Y.

3. Regresar <mismo-autor> si Puntuación ∈ [0,0.5),

   <diferente-autor> si Puntuación ∈ (0.5,1] o

   <no-se-puede-determinar> si Puntuación = 0.5.

Como podemos observar, los dos cambios principales que el equipo ASGALF realizó fueron dentro del método original de impostores. El primer cambio es la forma en que se calcula la puntuación total. En el método ASGALF, la puntuación utiliza específicamente la función *min-max* para representar la similitud entre los dos documentos. Básicamente, *min-max* calcula el valor de similitud que se obtiene comparando qué tanto se parecen los dos documentos realmente contra qué tanto pudieron parecerse.

Así, la función *min-max* está definida como sigue [11]:

$$min-max(X,Y)=\frac{\sum_{i=1}^{n} min(x_i,y_i)}{\sum_{i=1}^{n} max(x_i,y_i)} \quad (7)$$

Al mismo tiempo, es importante añadir que la función *min-max* está basada en la similitud de Jaccard la cual está definida de la siguiente manera [13]:

$$JS(A,B)=\frac{|A\cap B|}{|A\cup B|}=\frac{|A\cap B|}{|A|+|B|-|A\cap B|} \quad (8)$$

Sean A y B dos conjuntos, la similitud de Jaccard está dada por el cociente de la cardinalidad de la intersección de A y B entre la cardinalidad de la unión de A y B.

El segundo cambio importante se encuentra en el número de impostores que sirven como entrada en cada iteración del Método de Impostores original. En GenIM, sólo se utiliza un impostor para calcular la puntuación de similitud en cada iteración. En cambio en ASGALF, se utilizan dos impostores I_x e I_y, tomados de un conjunto de impostores creado aleatoriamente y que al mismo tiempo son los impostores más parecidos al documento X y el impostor más parecido al documento Y respectivamente.

2.4 Representación Vectorial de Textos

Hasta ahora hemos hablado de cómo funcionan los algoritmos y cómo es que estos trabajan con documentos comparándolos entre sí; sin embargo, no hemos explicado la manera en que estos documentos son representados de tal forma que podamos manejarlos dentro de nuestro sistema. En esta sección, revisaremos cómo se representan los documentos de texto y cómo es que esta estructura hace que sea posible medirlos, analizarlos y compararlos.

Cuando hablamos de documentos de texto, nos damos cuenta de que es difícil representar un documento que tiene 100 mil palabras conformadas por alrededor de 500 mil caracteres y más si deseamos que un modelo computacional represente tal documento de una forma bien estructurada y fácil de manejar.

Ahí es donde los vectores toman un papel importante. Los vectores como sabemos, son modelos abstractos que nos permiten representar magnitud y dirección. En nuestro caso representan un contenedor o estructura ideal para los documentos ya que permiten el almacenamiento de muchísimos valores y la aplicación de operaciones entre estos.

Habiendo definido que los vectores serán la estructura en la que estarán contenidos los documentos de texto, ahora debemos pensar en cómo verteremos el contenido de los documentos en los vectores.

Podríamos pensar en insertar cada palabra eligiendo letra por letra como valores que alojaríamos de forma continua en nuestro vector. Sin embargo, almacenar un documento de texto de esta forma podría requerir demasiado espacio y tiempo al momento de intentar comparar dos textos. La forma más utilizada hasta ahora ha sido extraer características especiales del texto a través del *conteo de palabras, sufijos, prefijos, signos de puntuación, número de apariciones de palabras seguidas de otras, letras aisladas, uso de mayúsculas, uso de nombres propios* y algunos otros patrones interesantes, por mencionar algunos; que al final dan lugar a un vector que podemos decir representa el estilo personal de escritura de un autor denotado por un texto particular.

En primer lugar, debido a que no todos los documentos tienen las mismas palabras y no todos los documentos tienen el mismo número de palabras, es necesario crear un vocabulario que sirva como base. Debemos revisar cuáles son las palabras que están presentes en la colección total de documentos, no sólo en uno de ellos, y con base en este vector de vocabulario crear los vectores específicos para cada documento. Este paso nos ayuda a definir la dimensión de los vectores y un orden en su contenido. De no hacerlo, contaríamos con vectores de diferentes tamaños y orden que no nos permitiría la aplicación de operaciones entre ellos.

Para aclarar el concepto, veamos en la figura 10 cómo se representaría un trabalenguas popular bajo un esquema de características en el que contaremos cuántas veces aparece la misma palabra repetida en el texto y en el que para fines prácticos permitiremos la presencia de palabras comunes (*stopwords*):

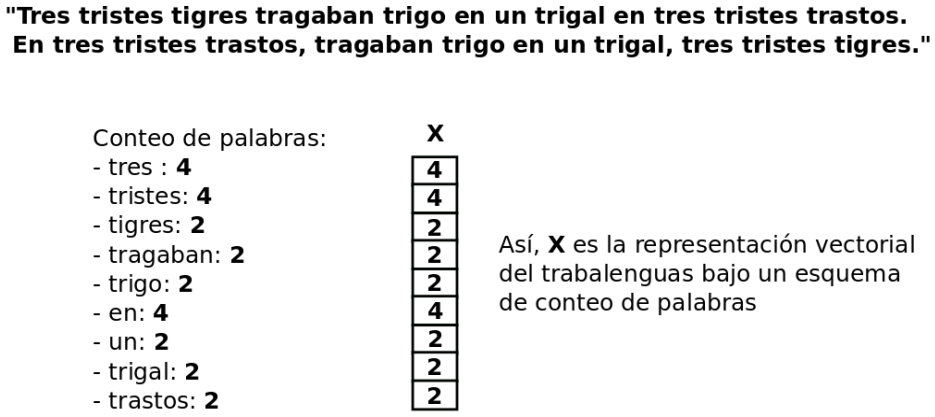


Figura 10: Ejemplo de representación vectorial de un trabalenguas

Ahora, cuando queremos comparar un texto contra otros textos e incluimos documentos nuevos a nuestra base de datos de documentos, es necesario revisar y generar un vocabulario común en el que se encuentren todas las palabras presentes en todos los documentos. Así, sobre este vocabulario es que entonces creamos los vectores finales para cada documento. En la figura 11 se muestra la representación vectorial de un nuevo trabalenguas Y y del trabalenguas X de la figura 10, bajo un vocabulario nuevo que contiene las palabras presentes en ambos trabalenguas:

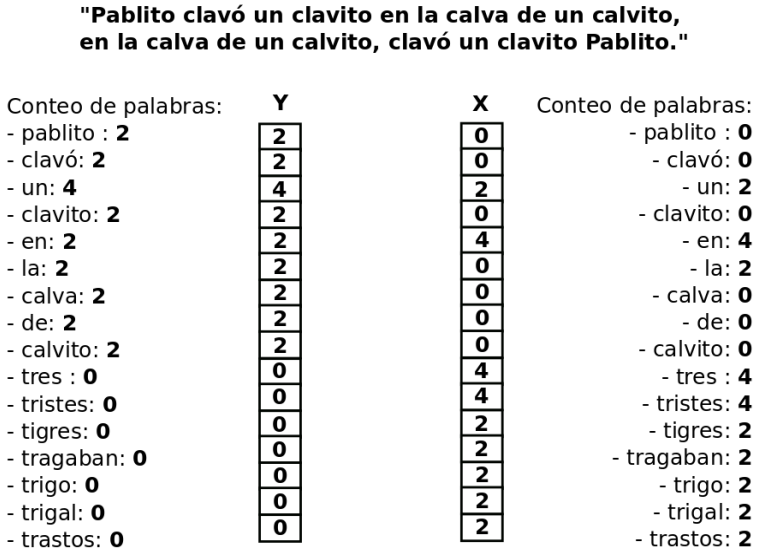


Figura 11: Representación vectorial de dos trabalenguas utilizando un vocabulario común

Ya que sabemos cómo se puede representar cualquier texto en un primer paso, ahora toca el turno a revisar qué otros tipos de características son las más comunes en el área del procesamiento del lenguaje natural.

Bolsa de Palabras (Bag of words)

En ésta se calcula la frecuencia o peso de una palabra por el número de apariciones de la misma en el documento. Esta característica es la misma que empleamos en nuestro ejemplo anterior. Es la más básica y comúnmente se utiliza para problemas de clasificación de documentos donde la frecuencia de las palabras es usada como un atributo para el entrenamiento del clasificador.

Bigrama

En esta característica se calcula la frecuencia o número de apariciones de dos palabras consecutivas en un documento dado. Básicamente se trata de una bolsa de palabras que en lugar de contabilizar una palabra, registra la frecuencia de la aparición de dos palabras adyacentes. La estructura de los bigramas permite conservar un poco del orden secuencial del texto que se pierde al convertir los documentos en vectores. Otra ventaja es que también elimina ambigüedad al recuperar parte del contexto en el que se está utilizando cada palabra, a diferencia de la bolsa de palabras, donde se aísla cada palabra. Veamos cómo quedaría nuestra representación vectorial bajo este esquema:

Bigrama	**Frecuencia**
tristes tigres	2
tigres tragaban	2
tragaban trigo	2
trigo en	2
en un	2
un trigal	2
tres tristes	3

tristes trastos	2

Tabla 2: Ejemplo parcial de representación vectorial de un texto utilizando bigramas con frecuencia mayor o igual a 2

Trigrama

Similar al bigrama. Tal como su nombre lo indica, en esta característica se calcula la frecuencia o aparición de tres palabras consecutivas en el texto. Al igual que los bigramas, los trigramas son utilizados para la predicción de la aparición de palabras y recupera aún más del contexto del documento. Es decir que podemos decir con cierta probabilidad cuál será la siguiente palabra en una secuencia de palabras. El trigrama es un caso particular de un n-grama cuando n es igual a 3. Por consiguiente, debemos notar que los bigramas son el caso particular cuando n es igual a 2.

Trigrama	**Frecuencia**
tragaban trigo en	2
trigo en un	2
en un trigal	2
tres tristes trastos	2

Tabla 3: Ejemplo parcial de representación vectorial de un texto utilizando trigramas con frecuencia mayor o igual a 2

Prefijo

En esta representación se calcula la frecuencia o número de apariciones de prefijos en las palabras del texto. Este atributo es utilizado usualmente cuando se desea capturar el estilo de escritura del autor basándose únicamente en la raíz de las palabras que utiliza, ocasionando que esta representación sea independiente del género o número de las palabras que utiliza. Usualmente se utilizan los prefijos definidos por preposiciones latinas antiguas para el caso del español. Para términos prácticos, también podemos definir secuencias de letras como en nuestro ejemplo de la figura 10 de donde extraemos prefijos de tres letras:

Prefijo	Frecuencia
tre	4
tri	8
tig	2
tra	4

Tabla 4: Ejemplo parcial de representación vectorial de un texto utilizando prefijos con frecuencia mayor o igual a 2

Sufijo

Esta representación es similar a la del prefijo, con la diferencia de que se cuenta la aparición de sufijos. Este atributo se utiliza para descubrir si existe cierta tendencia en las terminaciones de las palabras del autor. Si un escritor suele utilizar con frecuencia adverbios de modo como por ejemplo el adverbio *frecuentemente* o diminutivos como *carrito*, este atributo agregará a la representación esta tendencia en su estilo. Para nuestro ejemplo, escojamos todos los sufijos de tres letras del texto de la figura 10:

Sufijo	Frecuencia
res	6
tes	4
ban	2
igo	2
gal	2
tos	2

Tabla 5: Ejemplo parcial de representación vectorial de un texto utilizando sufijos con frecuencia mayor o igual a 2

Bigramas de prefijos

Esta representación combina las representaciones de bigramas y prefijos. Ésta se utiliza para identificar el uso recurrente de dos prefijos consecutivos.

Prefijo bigrama	Frecuencia
tre tri	2
Tri tig	2
tra tri	2

Tabla 6: Ejemplo parcial de representación vectorial de un texto utilizando bigramas de prefijos con frecuencia mayor o igual a 2

Bigramas de sufijos

En esta representación se combinan el uso de bigramas y sufijos y se cuenta el número veces que aparecen dos sufijos consecutivos dentro de un documento. Por ejemplo, algunos autores utilizan frecuentemente un verbo en presente seguido de un adverbio de modo de forma como en: *corren apresuradamente*. En este caso, esta representación se enfocaría únicamente en las terminaciones *en* y *mente* contando el número de veces que esta combinación aparece en el texto independientemente del verbo o el adverbio que estén siendo utilizados. A continuación un ejemplo de esta representación basado en el texto de la figura 10:

Sufijo bigrama	Frecuencia
res tes	4
tes res	2
ban igo	2
tes tos	2

Tabla 7: Ejemplo parcial de representación vectorial de un texto utilizando bigramas de sufijos con frecuencia mayor o igual a 2

Palabras comunes (*stop words*)

En esta característica se toman en cuenta las palabras que tienen la frecuencia más alta dentro del texto y que no comunican o aportan tanto como otras palabras, al mensaje que el autor desea dar al lector. Las palabras comunes son aquellas palabras que suelen ser las más utilizadas en cada lenguaje.

Palabras comunes	Frecuencia
en	4
un	2

Tabla 8: Representación vectorial de un texto utilizando palabras comunes

Bigramas de palabras comunes

En ésta se cuentan las palabras comunes consecutivas en el texto. Esta representación también nos ayuda a enfocarnos en ciertos aspectos del estilo de escritura del autor como cuántas palabras de las que utiliza realmente están transmitiendo un mensaje o si el autor utiliza muchas muletillas en sus textos como por ejemplo: *es que* o *así que*.

Bigramas de palabras comunes	Frecuencia
en un	2

Tabla 9: Ejemplo parcial de representación vectorial de un texto utilizando bigramas de palabras comunes con frecuencia mayor o igual a 2

Puntuación

En esta característica se cuentan las apariciones de diferentes signos de puntuación como el *punto*, la *coma*, el *punto y coma*, *signos de exclamación* y de *interrogación*, *dos puntos*, *comillas*, *paréntesis*, *puntos suspensivos*, *guión*, entre otros. Estos elementos que a simple vista podrían carecer de importancia para la tarea de verificación de autor, también nos permiten crear un perfil de cada autor. Algunos autores utilizan muchas *comas*, mientras que otros prefieren utilizar frases largas sin pausas y concluir con *punto*.

Puntuación	Frecuencia
Coma	2
Punto	1

Tabla 10: Representación vectorial de un texto utilizando puntuación

Palabras por oración

En esta característica se cuenta el número de palabras que un autor utiliza en cada oración. Esta representación consta sólo de un valor numérico que se obtiene de calcular el promedio de palabras utilizadas en cada uno de las oraciones dentro de un documento o varios de un mismo autor.

Palabras por oración	Frecuencia
Tres tristes tigres tragaban trigo en un trigal en tres tristes trastos.	12
En tres tristes trastos, tres tristes tigres tragaban trigo en un trigal.	12
Promedio	12

Tabla 11: Representación vectorial de un texto utilizando palabras por oración

Generalización a diferentes niveles de categorías gramaticales

El equipo ASGALF utilizó la representación de categorías gramaticales también conocidas como *etiquetado de partes del habla* (del inglés POS o *part-of-speech tagging*). En esta representación se etiqueta cada palabra del documento de acuerdo a su categoría gramatical, ya sea ésta un *sustantivo, verbo, adjetivo, adverbio, preposición, pronombre, conjunción o artículo,* y posteriormente sobre esta representación se aplica alguna de las previamente mencionadas. Por ejemplo, *bigramas de adverbios o sufijos de verbos*.

Riqueza del documento

El equipo ASGALF también dio gran importancia a la riqueza del documento; es decir, la característica representada por el número total de palabras únicas por documento normalizado, según el número total de palabras en el mismo documento. Esta característica básicamente sirve para medir la creatividad del autor observando la cantidad de palabras únicas en el vocabulario de sus documentos.

Características	GenIM	ASGALF	Este trabajo
Bolsa de palabras	*	*	*

Características	GenIM	ASGALF	Este trabajo
Bigramas		*	*
Trigramas		*	*
N-gramas	*	*	
Prefijos			*
Sufijos			*
Bigramas de prefijos			*
Bigramas de sufijos			*
Palabras comunes			*
Bigramas de palabras comunes			*
Puntuación			*
Palabras por oración			*
Riqueza del documento		*	
N-gramas de letras con N={1,2, ...,10}		*	
N-gramas de palabras con N={1,2, ...,10}		*	
N-gramas de palabras comunes con N={1, 2, ...,10}		*	
N-gramas de formas de palabras con N={1,2, ...,10}		*	
N-gramas de etiquetas POS con N={1,2, ...,10}		*	
N-gramas de palabras POS con N={1,2, ...,10}		*	

Tabla 12: Características utilizadas por los diferentes métodos para la identificación de autor

En la tabla 12, se resumen las diferentes características más utilizadas para representar textos y los métodos que las utilizan. De la tabla 12 y el rendimiento de los diferentes métodos, es posible deducir que el uso de n-gramas en sus diferentes formas (como es en el caso del sistema ASGALF) sustituye al uso de características más variadas como la puntuación y palabras por oración.

Para la representación de textos en este trabajo, se lleva a cabo la extracción de características de documentos en dos etapas. En la primera, se crea un diccionario de todas las palabras presentes en el texto que se desea representar vectorialmente. Una vez que se cuenta con este vocabulario, se transforma en un vector donde cada una de sus entradas está ocupada por una de las palabras del vocabulario. Posteriormente, utilizando el modelo de Bolsa de palabras, se realiza el conteo de las apariciones de cada palabra del vocabulario en el texto dado y se coloca dicho valor en la entrada correspondiente a la palabra en turno.

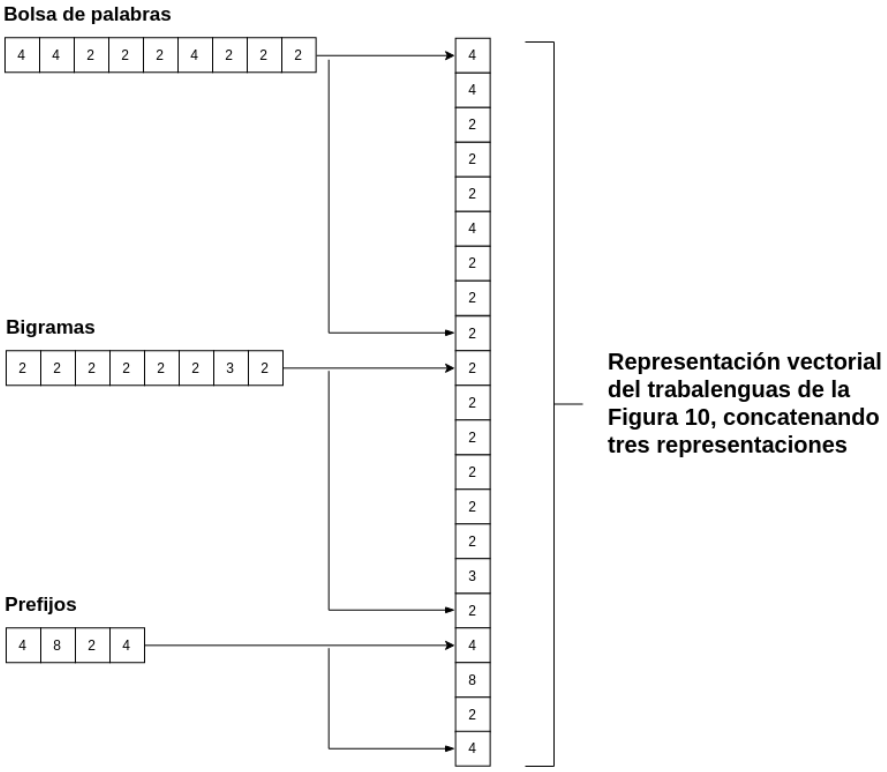


Figura 12: Vector final resultado de la concatenación de diferentes representaciones vectoriales correspondientes a tres características del mismo documento

Una vez que se ha realizado este procedimiento, en entradas contiguas del mismo vector se añaden uno tras otro, valores para otros esquemas como la frecuencia de *bigramas, trigramas, prefijos, sufijos, bigramas de prefijos, bigramas de sufijos, palabras comunes, bigramas de palabras vacías, puntuación y palabras por oración.*

En la figura anterior, es fácil notar que los vectores podrían no contar con una dimensión fija, ya que ésta depende del número de palabras diferentes presentes en cada texto y el número de los diferentes esquemas seleccionados para la representación vectorial de cada texto. Es por eso que es importante definir un diccionario de palabras y un esquema de características, ya que la dimensión del vector del diccionario será la que determine el tamaño de los vectores de cada documento. De esta manera, la representación vectorial de un documento de texto queda definida formalmente de la siguiente manera:

$$d=(w_{1,}w_{2,}\ldots,w_m) \tag{9}$$

Donde cada w_i es una de las m características elegidas para la definición de la representación vectorial, donde al mismo tiempo cada w_i puede estar conformada por una secuencia de frecuencias de las diferentes palabras presentes en el diccionario o bien, el valor singular de una característica como es el caso de las palabras por oración.

En este capítulo se presentaron los métodos para la identificación de autores más exitosos en la competencia de la tarea de identificación de autor PAN CLEF. Con ayuda de estos, ahora nos será posible ubicar el rendimiento de nuestro sistema y de esta manera, medir la precisión de nuestros resultados frente al de sistemas en el estado del arte de este campo de investigación. Además, revisamos la representación vectorial que utilizaremos para manipular nuestros documentos de texto, misma que también está siendo utilizada por otras propuestas en el estado del arte de la identificación de autor.

Capítulo 3. Clasificación basada en la Representación Dispersa

En el núcleo de nuestro sistema propuesto descansa la clasificación basada en la Representación Dispersa. Este tipo de clasificación ha sido utilizada con éxito en diversos campos que requieren métodos de identificación como son la identificación de rostros y de sonidos. Es por ello que creemos que dará resultados interesantes en el área de identificación de autores.

Sencillamente, nuestro método se basa en la suposición de que una señal, ya sea una imagen o un documento de texto, puede ser representada por la combinación lineal de diversas señales en forma de vectores y que es posible medir cuál de estos vectores está contribuyendo más para la generación de dicha señal con el fin de concluir cuál de ellos es más parecido a nuestra señal original. Sin embargo, encontrar dicho vector bajo estas condiciones no es fácil.

Por su parte, la Clasificación basada en la Representación Dispersa nos indica que

en verdad existe una manera de lograrlo con ayuda de la minimización de norma L1 y el método de análisis conocido como Homotopía.

3.1 Formulación del problema

Antes de sumergirnos por completo en la solución, primero plantearemos el problema y estudiaremos sus componentes para entender un poco más de qué estamos hablando cuando nos referimos al término de clasificación basada en Representación Dispersa.

Siguiendo el planteamiento de Wright et. al., se asume que es posible construir una matriz A de m renglones por n columnas a partir de un conjunto suficientemente grande de señales de 'entrenamiento', una por cada columna en A, cuya identidad o categoría se conoce; así mismo, que esta matriz genera un subespacio vectorial de señales, y que en este subespacio de señales posiblemente también se encuentra la señal y, que se desea reconstruir, y cuya identidad también se intenta descubrir [5].

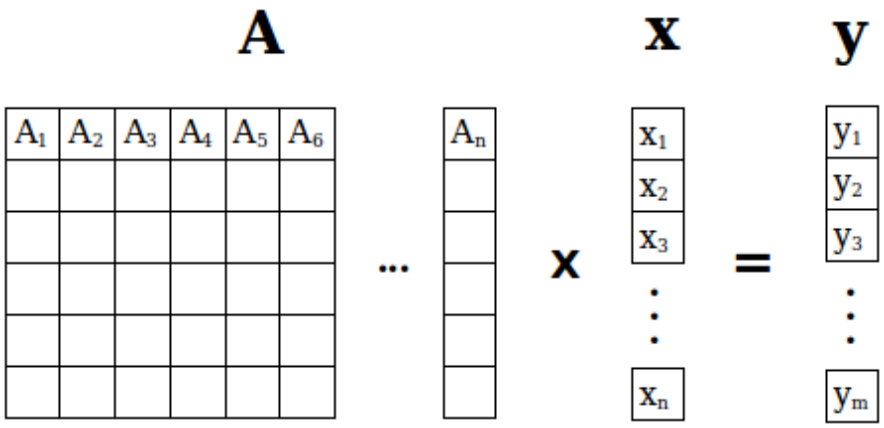


Figura 13: Representación gráfica del sistema de ecuaciones Ax = y

Observando la figura 13, lo natural es suponer que el siguiente paso para encontrar la identidad de la señal y, es resolver el sistema de ecuaciones. Para esto replantearon el problema de la siguiente forma:

$$x_2 = arg\,min\|x\|_2 \qquad sujeto\ a \qquad Ax = y \tag{10}$$

En otras palabras, el nuevo planteamiento en (10) está diciendo que lo que se busca es resolver el sistema de ecuaciones $Ax=y$, y que al mismo tiempo que la solución cuente con la norma $L2$ más pequeña de entre todas las posibles soluciones. A esto se le conoce como un problema de minimización de norma $L2$ y se puede resolver fácilmente por medio de la pseudo inversa de A. Sin embargo, tal como lo apunta Wright et. al. [5], con este método la solución resultaría ser muy densa; i.e., con muchas entradas diferentes de cero, valores muy grandes y distribuidos en muchas categorías o identidades de nuestro conjunto de entrenamiento, lo cual ocasionaría que la solución careciera de información relevante para decidir cuál de las señales de entrenamiento en A estaría contribuyendo más para la reconstrucción de la señal de entrada y.

Para evitar que esto suceda, se debe suponer que para el problema se cuenta una señal válida y; i.e., que es posible reconstruirla utilizando vectores contenidos en la matriz A. Así entonces, debe existir un conjunto de vectores en A de la misma categoría con los que es posible representar la señal y con la precisión suficiente. Además, esta representación también resulta ser dispersa si el número de categorías en la matriz A es lo suficientemente grande.

Es por esto que otra alternativa sería utilizar la minimización de la norma $L0$ como se expone de la siguiente forma:

$$x_0 = \arg\min \|x\|_0 \qquad sujeto\ a \qquad Ax = y \tag{11}$$

La norma $L0$ más que calcular la norma de un vector, nos indica el número de entradas diferentes de cero que un vector posee. Así, de encontrar el vector solución con la norma $L0$ más pequeña posible se estaría encontrando la solución más dispersa al sistema de ecuaciones. Sin embargo, el problema para encontrar la solución más dispersa a un sistema de ecuaciones subdeterminado es *NP-duro* y no existe un procedimiento más eficiente que reducir por fuerza bruta probando cada una de las combinaciones posibles para cada entrada en x [14].

Por otro lado, también se sabe que si la solución x_0 es lo suficientemente dispersa, entonces la solución al problema de minimización utilizando la norma $L0$ es

equivalente a la solución al problema de minimización utilizando la norma *L1* [15], cuya solución puede ser encontrada en tiempo polinomial utilizando métodos optimización convexa como Homotopía [5]. De esta manera, optamos por el uso de la minimización de norma *L1* para resolver el problema quedando definido de la siguiente forma:

$$x_1 = arg\,min \|x\|_1 \quad sujeto\ a \quad Ax = y \tag{12}$$

A continuación en la tabla 13, realizamos una comparación de las tres normas revisadas en esta sección con el fin de exponer sus diferencias:

Norma	**Suma**	**Sujeto a**
L_0	$\|x\|_0 = \sqrt[0]{\sum_i x_i^0} = 1+1+1+1+\ldots+1 = \text{card}(x_i \neq 0)$	$Ax = y$
L_1	$\|x\|_1 = \sqrt[1]{\sum_i x_i^1} = x_{11}+x_{12}+x_{13}+\ldots+x_{1n}$	$Ax = y$
L_2	$\|x\|_2 = \sqrt[2]{\sum_i x^2} = \sqrt{x_{11}^2+x_{12}^2+x_{13}^2+\ldots+x_{1n}^2}$	$Ax = y$

Tabla 13: Cálculo de diferentes tipos de normas

En la siguiente sección abordaremos y explicaremos el método de Homotopía para resolver el problema de minimización de norma *L1* en la ecuación (12).

3.1.1 Homotopía

Encontrar la solución de un sistema subdeterminado de ecuaciones lineales y al mismo tiempo restringir ésta a que sea la más dispersa es considerado por los expertos como un problema NP-Duro [16]. Sin embargo, existe un método que nos permite transformar este problema en uno cuya solución sea más fácil de encontrar y que al mismo tiempo nos de una solución dispersa. Este método se conoce como Homotopía y su propósito es transformar la función objetivo de un problema de minimización de norma *L2* a uno de norma *L1* a medida que se busca mantener un equilibrio donde la solución parcial es la más aproximada y también la más simple, que en nuestro caso significa dispersa.

$$x_1 = arg\,min \|x\|_1 \quad sujeto\ a \quad Ax = y \tag{13}$$

A grandes rasgos, el algoritmo de homotopía funciona de manera similar a otros métodos que inician con un vector solución de coeficientes *x'* vacío. En cada paso, el algoritmo busca y añade al vector solución *x'* el coeficiente con mayor correlación al vector desconocido denominado como *y* en la ecuación (13). Posteriormente, se incrementa este mismo coeficiente en dirección del signo del valor de su correlación con *y*, y se detiene hasta que otro coeficiente tenga al menos la misma correlación con *y* que el coeficiente anterior. Se continua incrementado el valor de los coeficientes que han sido añadidos a *x'* en la dirección de la suma mínima de los cuadrados de los coeficientes en *x'*. Se detiene este incremento hasta que otro coeficiente tenga tanta correlación con *y* como los coeficientes en *x'*. De esta manera, el algoritmo continua hasta que todos los coeficientes están en *x'*.

Debido a que el algoritmo únicamente avanza en dirección de los coeficientes con mayor correlación con *y*, descartar un coeficiente con baja correlación implica la aparición de un cero en la solución final *x'*, y estos valores en cero permanecen sin cambio hasta que el algoritmo concluye induciendo un alto nivel de dispersión en el vector solución final [17] .

El algoritmo de Homotopía cuenta con una complejidad del orden de $O(dm^2+dmn)$ cuando logra recuperar una solución dispersa en *d* pasos con un número *d* de entradas diferentes de cero; donde *n* es la dimensión del vector solución (número de coeficientes de cada señal) y *m* es el número de observaciones (ecuaciones) presentes en el sistema. En el peor caso se da cuando la dispersión *d* y el número de observaciones *m* crecen proporcionalmente con la dimensión *n*. En este caso, la complejidad de Homotopía se ubica en el orden de $O(n^3)$ [5].

3.2 Marco de Clasificación

Tal como lo observamos en diversos trabajos de reconocimiento de señales como el de Wright et. al. [5], uno de los primeros pasos para definir el marco de clasificación sobre el cual se va a trabajar es el de reunir una cantidad

considerable de señales y etiquetarlas de acuerdo al tipo de problema que deseamos resolver.

Para entender mejor esto y ponerlo en práctica, pensemos en esto como la creación de un diccionario de señales que resulta de mapear señales en etiquetas que servirán posteriormente para reconocer la identidad de cada señal durante el proceso de clasificación. A estas etiquetas también se les conoce como *categorías* o *clases.*

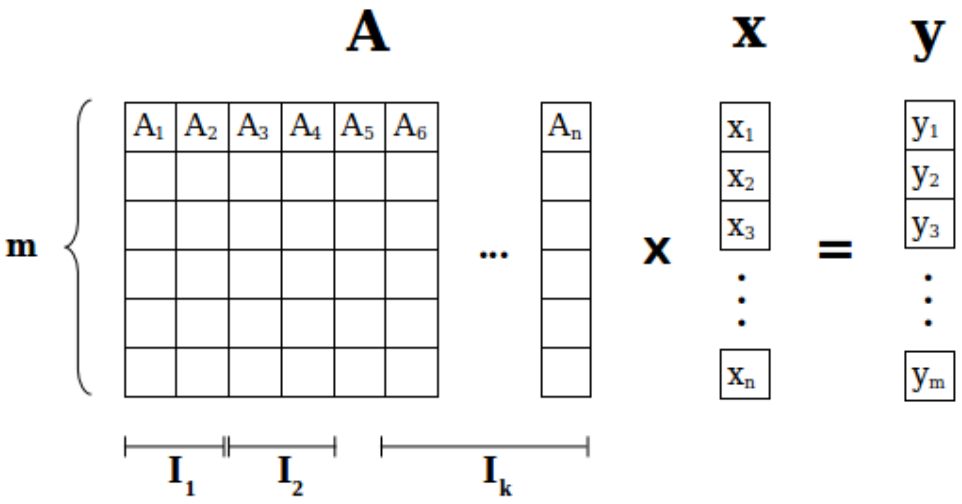


Figura 14: Las N señales etiquetadas dentro de k categorías identificadas con la letra I que concatenadas como vectores transpuestos forman la matriz A.

En la figura 14 es posible apreciar cómo queda finalmente constituida la matriz *A* de señales. En ésta, cada columna de *A* es representada por una señal del diccionario antes descrito, sólo que con el detalle adicional de que aquellas señales que cuentan con características similares bajo un marco de referencia común, son agrupadas en conjuntos I_k y clasificadas bajo la misma identidad. Por ejemplo, para el caso de un marco para la identificación de rostros, las imágenes o señales de rostros de una misma persona quedarían etiquetadas bajo la misma identidad.

Con *A* organizada de esta manera, en el momento en el que el sistema de ecuaciones logra ser resuelto utilizando el método de Homotopía, es posible tener una idea de la identidad de los vectores que contribuyeron más para la reconstrucción de la señal de entrada *y*, simplemente con revisar la etiqueta I_k que

se les otorgó en el diccionario de señales en un principio.

3.3 Clasificación basada en Homotopía

Basándose en la forma en que el diccionario A fue construido a inicio de esta misma subsección, el paso más lógico a realizar es revisar valor por valor cada una de las entradas en el vector disperso *x'*, fijarse en la entrada con el valor más grande y ver con cuál señal de entrenamiento en A está relacionado utilizando el diccionario de categorías. Y de esta manera daríamos con la identidad de la señal de entrada *y*.

Sin embargo, como nos lo marcan *Wright et. al.,* el ruido en las señales y errores en nuestro modelo causarían que al concluir la ejecución del método Homotopía, la mayoría de las entradas diferentes a cero en el vector disperso *x'* esté formada por valores muy pequeños asociados a múltiples señales de entrenamiento en *A* complicando la selección de la categoría correcta, contrario a lo que esperábamos y desaprovechando la estructura lineal del sistema.

Así, una vez que el sistema es resuelto utilizando Homotopía y que se cuenta con una solución, es decir el vector disperso *x',* el siguiente paso es medir la contribución de cada una de las entradas diferentes de cero presentes en el vector disperso *x'* y determinar cúal reproduce con mayor fidelidad al vector de entrada *y*. Para esto, se requerirán dos funciones.

La primera la denominaremos δ_i, la cual recibirá al vector disperso *x'* como entrada y entregará un vector x'_i como salida, donde x'_i es el mismo vector *x'* pero con cada una de sus entradas en cero a excepción de la i-ésima clase.

La segunda denominada *r*, calculará los residuales que se obtienen de evaluar el vector x'_i obtenido en δ_i de la siguiente forma:

$$r_i = \left\| y - A\delta_i(x'_i) \right\|_2 \tag{14}$$

Ahora que tenemos una forma de evaluar cada una de las categorías *i* que contribuyeron en el vector disperso x_i, es posible clasificar la señal *y* basándose en

el valor más pequeño de entre todas las evaluaciones de r_i:

$$min \quad r_i = \left\| y - A\delta_i(x'_i) \right\|_2 \tag{15}$$

Finalmente, revisemos el algoritmo de clasificación dispersa basado en Homotopía tal como Wright et. al. [5] lo propone:

Algoritmo de Clasificación basada en Representación Dispersa

Entrada: Una matriz A formada por señales de entrenamiento

$A = [A_{1,} A_{2,} A_{3,} \ldots A_k] \in R^{mxn}$ donde k es el número de categorías

Una señal de entrada $y \in R^m$

1. Normalizar las columnas de A en norma L2.
2. Resolver el problema de minimización utilizando homotopía:

$$x' = arg\,min \|x\|_1 \quad \text{sujeto a Ax = y}$$

3. Calcular los residuales:

$$r_i(y) = \left\| y - A\delta_i(x') \right\|_2 \quad \text{para i = 1, 2, 3,… k}$$

Salida:

$$Identidad(y) = arg\,min_i\, r_i(y)$$

3.4 Identificación de rostros

Ahora que hemos visto el algoritmo para la clasificación basada en Representación Dispersa, podemos revisar el método pero en esta ocasión aplicado a la identificación de rostros con el fin de observar cómo funciona de manera práctica ejecutado en un problema real; para posteriormente dar el salto a la identificación de autores.

La automatización del procesamiento de imágenes y el reconocimiento e identificación objetos y rostros utilizando nuevas tecnologías se ha convertido en un reto crucial para los científicos ya que de vencerlo paso por paso, sería posible entender cómo funciona la memoria del ser humano, cómo es que el cerebro

memoriza las caras de nuestros conocidos o si es verdad que juzgamos a otro humano utilizando los atributos que percibimos a partir de su rostro; al mismo tiempo, también se requieren herramientas más precisas y automáticas para resolver problemas prácticos que suceden en el mundo a diario que requieren de velocidad y operación continua en temas relacionados con la seguridad, criminalística, psicología, redes sociales, medicina, comunicación, entretenimiento, entre otros. Es por eso que consideramos que vale la pena abordar de forma didáctica la clasificación dispersa aplicada al reconocimiento de rostros.

Emulando lo que hicimos en la subsección 3.1, volvamos a definir una matriz A de señales de m renglones y n columnas, que en esta sección comenzaremos a denominar como nuestro diccionario de caras de entrenamiento y donde cada columna de A estará formada por los píxeles de la imagen del rostro de un individuo. A nuestro vector o señal de entrada y, lo conoceremos de ahora en adelante como el rostro desconocido que queremos identificar con ayuda de nuestro sistema. Finalmente, el vector x' continuará siendo el vector disperso que se obtiene al resolver el problema de optimización ya conocido:

$$x' = arg\,min \|x\|_1 \quad sujeto\ a \quad Ax = y \tag{16}$$

En la figura 15 se observa que imágenes del mismo individuo se organizan de forma contigua para el etiquetado y la identificación del individuo. Por su parte, el algoritmo de optimización nos devolverá un vector disperso x' que esperamos, únicamente se concentrará en las columnas de los individuos que contribuyeron más para la generación del vector y.

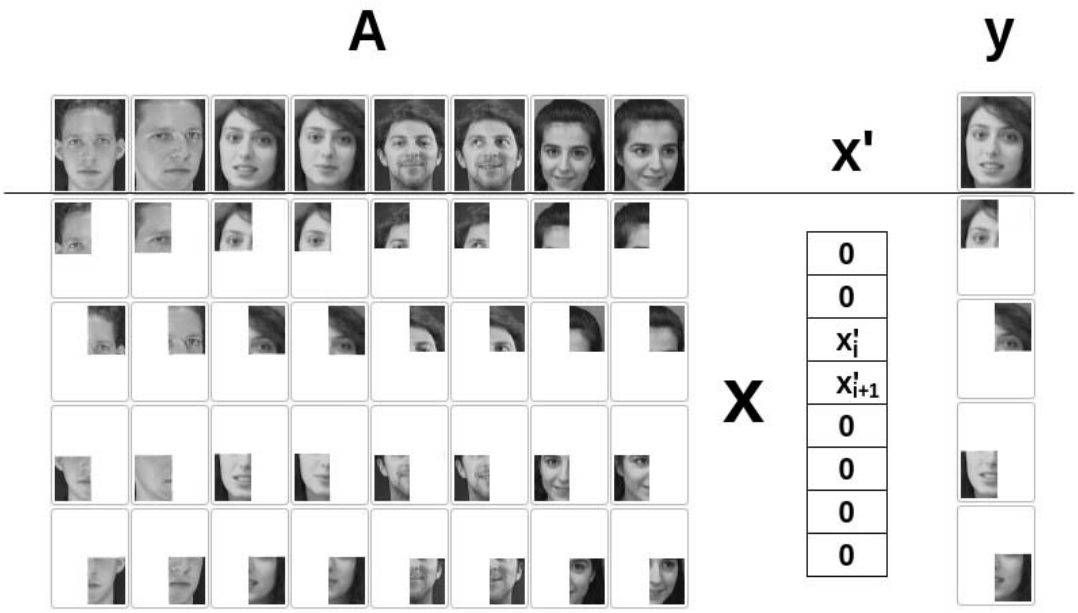


Figura 15: Visualización de la matriz A de rostros, vector disperso *x'* y rostro desconocido Y. Imágenes tomadas de la base de datos de rostros de AT&T Laboratories Cambridge.

Para formar el diccionario de caras *A*, se recurrió a la base de datos de caras de AT&T Laboratories Cambridge. La cual consta de 10 fotografías del rostro de 40 individuos diferentes, 400 fotografías en total, con un tamaño de 92 píxeles de ancho por 112 píxeles de alto.

Cada imagen puede visualizarse como una matriz de 112 renglones por 92 columnas. Sin embargo, para acoplar las imágenes en forma de señales para el sistema, conviene 'desdoblar' los píxeles de cada fotografía con el fin de formar un vector transpuesto de 10304 entradas o desde otra perspectiva, una matriz de una columna por 10304 renglones.

Aplicando el mismo procedimiento de forma iterativa a cada una de las 400 imágenes, obtenemos 400 vectores transpuestos de una columna por 10304 renglones. Con todas las imágenes transformadas en vectores, el paso siguiente es concatenar todos los vectores columna de tal forma que cada uno quede junto a otro de forma continua dentro de la nueva matriz.

La nueva matriz está identificada como A en el sistema ilustrado en la figura 15 y está formada por 10304 renglones y 400 columnas. Por su parte, el rostro desconocido a identificar está también representado por una imagen de 92 píxeles de ancho por 112 de alto y es transformado en un vector columna de 10304 renglones de la misma manera que cada uno de los rostros en A. Este vector de entrada está identificado como y en la figura 15. Es importante destacar que para el reconocimiento de señales ya sea de imágenes u otro tipo de señales, el tamaño de los vectores debe ser el mismo para todas las señales incluyendo las señales de entrada. Es decir que el sistema fallará si se le alimenta con una imagen más pequeña o más grande que las contenidas en el diccionario de imágenes.

Ahora que el preprocesamiento de los rostros está hecho, el sistema de ecuaciones está completo y es posible resolverlo y encontrar el vector disperso x' utilizando el método de Homotopía. Una vez que el método de Homotopía resuelve el sistema de ecuaciones, se utiliza el vector disperso x' resultante para encontrar la identidad del rostro representado por el vector y.

Para esto, se realiza el cálculo iterativo de residuales sobre el vector x' utilizando la función r_i, mencionado en la subsección 3.3. De esta manera, en la iteración i únicamente las entradas diferentes de cero en x' que están relacionadas con la identidad i permanecen *encendidas* mientras que al resto se le da un valor de 0.

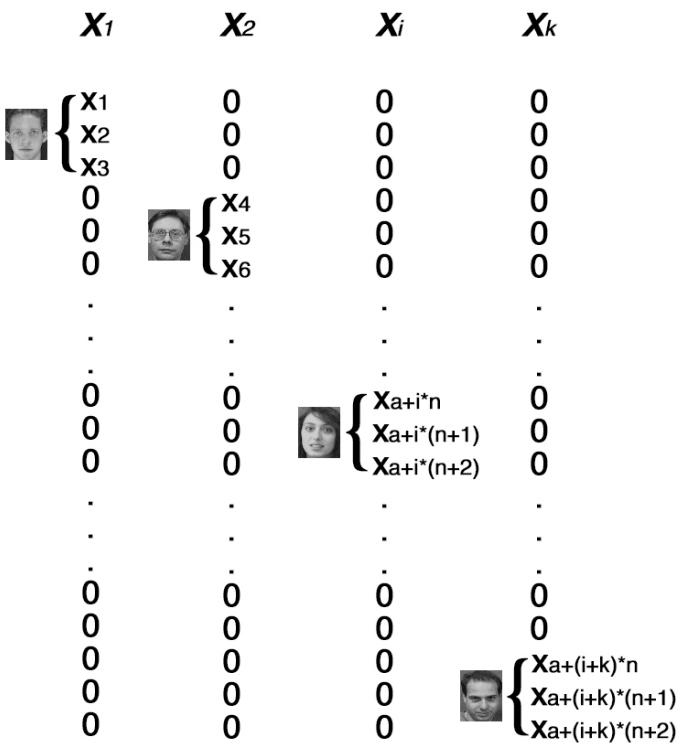


Figura 16: Ejemplos de cómo queda constituido el vector x’ en cada iteración para el cálculo de residuales. Imágenes tomadas de la base de datos de rostros de AT&T Laboratories Cambridge.

Después de obtener los residuales para las k identidades contribuyentes y presentes en el vector disperso x', se prosigue escogiendo el residual más pequeño de entre todos debido a que este representa la diferencia y distancia más pequeña entre el rostro de entrada y y una de las identidades presentes en x'. Finalmente, hemos encontrado la identidad del rostro en nuestra base de datos, más parecido o que coincide más con la verdadera identidad del rostro representado por el vector y.

Por otro lado, cuando el método de homotopía recupera un vector x' que no es tan disperso como se requiere y obstaculiza la selección de una categoría o cuando el cálculo de residuales no revela diferencias considerables para definir un candidato, entonces se determina que no fue posible clasificar la señal de entrada, i.e., el rostro de entrada es desconocido.

Capítulo 4. Identificación de autor

Hasta ahora, hemos revisado a fondo el funcionamiento de la clasificación basada en la representación dispersa y su aplicación al problema práctico de la identificación de rostros en el capítulo 3. En este capítulo, finalmente abordaremos la aplicación de la clasificación basada en la representación dispersa a la identificación de autores.

En la introducción de este trabajo, se habló por primera ocasión de la motivación que tenemos para desarrollar tecnología que identifique y clasifique documentos de texto con el fin de encontrar al autor de dichas obras; así también, se habló de la existencia del taller para el *Descubrimiento del plagio y el uso incorrecto de la autoría y el software social*, mejor conocido como PAN por sus siglas en inglés, mismo en el que también presentamos el sistema descrito en este trabajo como una propuesta para la identificación de autores siguiendo un marco de Clasificación basada en la Representación Dispersa.

Formalmente la tarea para la identificación de autores fue definida en la edición de PAN 2014 de la siguiente manera:

Dado un pequeño conjunto de documentos de texto "conocidos" pertenecientes a un mismo autor (de no más de 5 y no menos de un documento) y un documento en "cuestión", la tarea es determinar si el documento en "cuestión" fue escrito por la misma persona que escribiera los documentos "conocidos" dentro del conjunto dado.

El conjunto de documentos, también conocido como corpus y provisto por los organizadores del taller para la construcción de los sistemas de verificación, fue presentado en cuatro lenguajes. Al mismo tiempo fue clasificado en cuatro géneros distintos bajo las siguientes combinaciones: Ensayos y reseñas en Holandés, ensayos y novelas en Inglés y artículos tanto en Griego como en Español. De acuerdo con la convocatoria del PAN 2014, cada equipo debía enviar su propuesta de software para la identificación de autores considerando que el software requiere como entradas el género y el lenguaje del documento de texto a identificar; y como salida, una puntuación representada por un número real dentro del rango [0,1] correspondiente a la probabilidad de una respuesta positiva, donde un resultado con valor 1 denota una similitud idéntica y un valor 0 una similitud nula. Para el caso en el que el valor cae cerca de 0.5, el nivel de similitud se considera no concluyente y se deja "sin respuesta" para dicho problema de verificación.

Como el lector recordará, para implementar la identificación de señales utilizando la clasificación basada en representación dispersa, es necesario procesar previamente las señales que en este caso se trata de los documentos de texto. Este procedimiento es realizado en dos fases.

En la primera, tal como observamos en la figura 10 de la subsección 2.4, es necesario escoger un esquema de características que nos permita extraer arbitrariamente diferentes atributos de los textos a procesar con el fin de que estos constituyan a los diferentes vectores que formarán parte de nuestro diccionario, es decir, la matriz A de la figura 13 en la subsección 3.1.

En el caso de este sistema, decidimos utilizar el modelo espacial de vectores formado por las siguientes características: bolsa de palabras, bigramas de palabras, trigramas de palabras, prefijos, sufijos, bigramas de prefijos, bigramas de sufijos, palabras comunes, bigramas de palabras comunes, frecuencia de puntuación y palabras por oración.

En una segunda etapa, sumamos todos los vectores correspondientes a los textos escritos por el mismo autor. Este vector acumulativo es normalizado dividiendo cada una de sus entradas por el número total de documentos escritos por el autor y el vector resultante es utilizado como la representación final del estilo de dicho autor.

4.1 Planteamiento del problema

La metodología dispersa utilizada exitosamente en tareas de reconocimiento de rostros, se basa en encontrar la identidad de una cara a partir de un conjunto de caras conocidas. En este trabajo, adaptamos la metodología para determinar la identidad del autor de un texto a partir de un conjunto de documentos cuyos autores son conocidos.

El método consiste en identificar cuáles son los componentes que contribuyen a la generación del documento en cuestión a partir de un conjunto de documentos de muestra. El razonamiento básico consiste en revisar estos componentes y encontrar aquel que cuenta con la contribución más grande. De encontrarlo, significaría que dos documentos con grandes similitudes en el mismo tipo de características, deben pertenecer al mismo estilo de escritura; y por lo tanto al mismo autor.

Para identificar los componentes, el método reduce el procedimiento a la resolución de la misma ecuación vista en la subsección 3.1:

$$x' = arg\,min \|x\|_1 \quad \text{sujeto a} \quad Ax = y \tag{17}$$

Donde y es el documento en cuestión, A es la matriz de N muestras de documentos de texto clasificados entre k autores y donde x es el vector a minimizar que representa la contribución de cada candidato. De esta forma, la multiplicación de

la matriz *A* de candidatos por el vector de contribución puede generar el documento en cuestión:

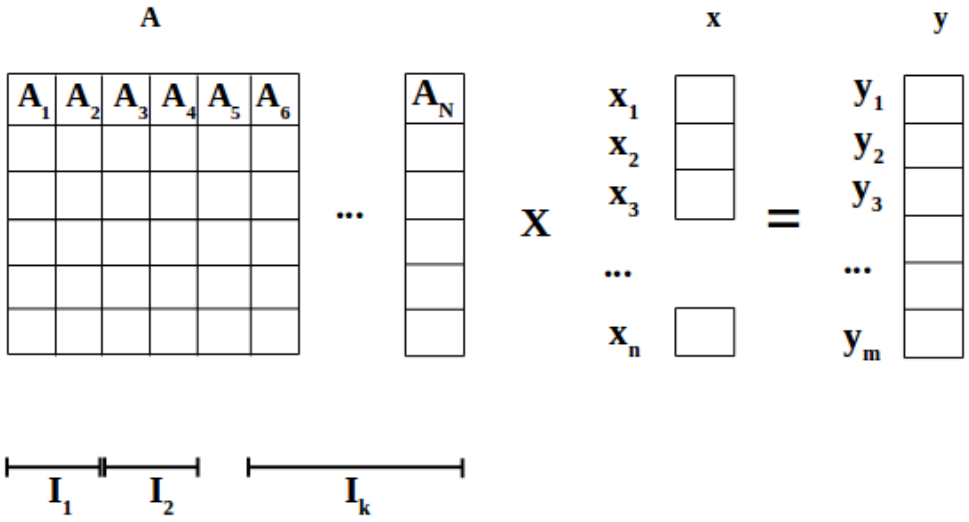


Figura 17: Representación gráfica del sistema de ecuaciones para el caso de autores.

Para nuestro sistema hicimos algunas modificaciones a la estructura final visualizada en la figura con el fin de incluir una adaptación del método de impostores que revisamos en el capítulo 2. Esta modificación inicialmente se hizo con base en la estructura del corpus de entrenamiento provisto por los organizadores de la competencia PAN CLEF 2014.

Para la adaptación de los impostores, se realizó una búsqueda aleatoria sobre el mismo corpus para elegir 10 documentos que contuvieran texto del mismo género e idioma que los del problema en turno. También se verificó que estos no fueran tomados de la misma colección del problema en turno debido a que una de las características que buscamos en los impostores es que fueran ajenos al problema (i.e., que pertenecieran a otro autor y que el autor no tuviera relación alguna con los documentos del problema), pero que al mismo tiempo se mantuvieran dentro del dominio del tema e idioma del problema.

Una vez elegidos, los impostores se integraron como vectores columna al diccionario de señales representado por la matriz *A* y concatenado, a su derecha, el

vector columna correspondiente al estilo del autor conocido tal como podemos ver en la figura 18:

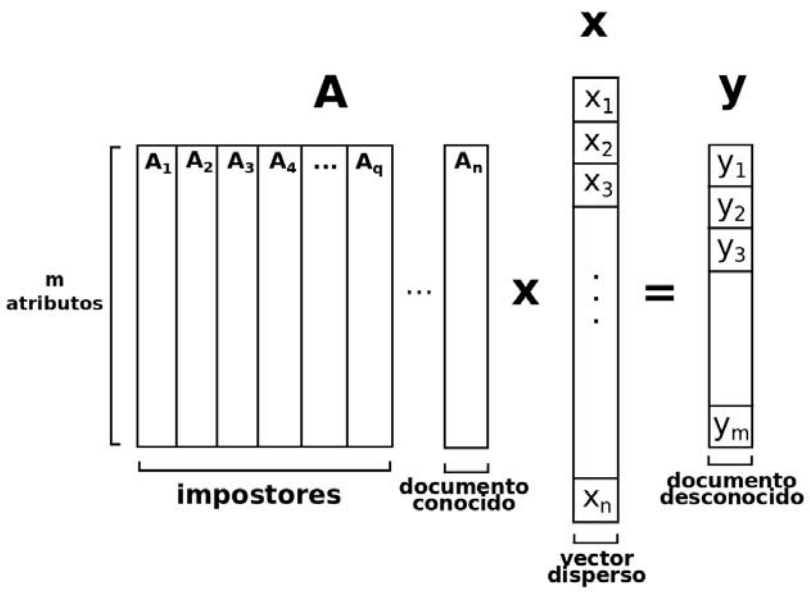


Figura 18: Representación gráfica del sistema de ecuaciones para verificación de autores con impostores.

Después de resolver el sistema de ecuación por minimización, a partir del vector disperso resultante *x'* podemos comparar los residuales dados por la función *r* vista en la subsección 3.3:

$$r_i = \left\| y - A\delta_i(x') \right\|_2$$

Y así decidir cuál autor está contribuyendo con más componentes escogiendo el resultado con el menor residuo:

$$\min r_i = \left\| y - A\delta_i(x') \right\|_2$$

Posteriormente, se adaptó el método para producir una probabilidad como resultado de iterar un número arbitrario de veces (en este caso 10) sobre el método en general, valor que fue elegido tras experimentar con diversos valores y medir el rendimiento para cada uno. De esta manera el algoritmo se define como

sigue:

Algoritmo de identificación de autor basado en Representación Dispersa

Entrada: Documentos conocidos, documentos 'impostores' y el documento desconocido

1. Preprocesar todos los documentos y proyectarlos a vectores de tal forma que formemos la matriz A:

$$A=[A_0 A_{1,} A_{2,} A_{3,} \ldots A_k] \in R^{mxn}$$

donde los vectores columna A_i para i = 0, 1, 2, …, c son los documentos conocidos y A_i para i = c+1, c+2, …, n son los documentos 'impostores' de k diferentes autores.

2. Preprocesar el desconocido de entrada y proyectarlo en un vector

$$y \in R^m$$

3. Realizar un muestreo sobre cada vector columna en A que consiste en revolver las palabras contenidas en cada representación, recortar la lista de palabras y tomar sólo un porcentaje de ésta, y volver a hacer un conteo sobre esta nueva muestra.

4. Resolver el problema de minimización utilizando homotopía:

$$x' = arg\,min_x \|x\|_1 \quad \text{sujeto a} \quad Ax = y$$

5. Calcular los residuales:

$$r_i(y) = \left\| y - A\delta_i(x') \right\|_2 \quad \text{para i = 1, 2, 3,… k}$$

6. Elegir el residuo más pequeño

$$Identidad(y) = arg\,min_i\, r_i(y)$$

7. Si el residuo más pequeño pertenece al del documento del autor conocido, entonces sumamos 1 a un contador. En caso contrario, no sumamos nada.

8. Repetir desde el paso 3 hasta el 7 del algoritmo por diez ocasiones generando diferentes muestreos en cada ocasión.

9. Dividir el valor final del contador entre las 10 iteraciones.

$$final = contador / iteraciones$$

9. Responder si un documento desconocido fue escrito o no por el autor conocido de acuerdo con la siguiente función:

```
mismoAutor (final) = Sí          si final > 0.5
                     No          si final < 0.5
                     No sé       si final = 0.5
```

Salida: True si el documento desconocido fue escrito por el autor conocido, **False** en el caso contrario y **None** si el sistema no pudo determinarlo

4.2 Corpus

La codificación de los algoritmos con los que funciona el sistema es sin duda la parte más importante de este trabajo; sin embargo, sin una base de datos o un corpus robusto es imposible probar el sistema bajo condiciones reales y así, determinar su capacidad de verificación así como los ajustes que se requieren.

Para este sistema, utilizamos el corpus provisto por los organizadores la competencia PAN CLEF 2014. El corpus está dividido en dos partes, la primera parte es la de entrenamiento y consiste en 696 problemas de identificación distribuidos en cuatro lenguajes y cinco géneros representados por ensayos y reseñas en holandés, ensayos y novelas en inglés, artículos en griego y en español.

Cada problema de identificación consta desde uno y hasta 5 textos pertenecientes a un solo autor conocido y un documento desconocido. El sistema utiliza los documentos conocidos para determinar si el documento desconocido también fue escrito por el autor de estos. El corpus de entrenamiento fue utilizado para la calibración de nuestro sistema previamente a la competencia. Posteriormente, durante la competencia, los organizadores dieron acceso a la segunda parte de los datos denominada como corpus de evaluación con el objetivo de poner a prueba a nuestro sistema con datos desconocidos y sobre los cuales se calificó el rendimiento de nuestro sistema.

De la tabla 14, donde se resume una descripción del corpus en números, observamos que el género con el mayor número de documentos en el corpus de entrenamiento es el de ensayos en inglés, de lo cual podemos inferir que es una de las categorías donde se pueden obtener buenos resultados al clasificar debido a la gran cantidad de datos disponibles para el entrenamiento del sistema. Esta categoría también es una de las que más problemas por género tiene. Las novelas en inglés en el corpus de entrenamiento es el género con menos documentos por problema, con lo que se espera que al sistema se le dificulte más la clasificación de documentos en este género por contar con menos información contra la cual comparar los documentos desconocidos.

	Lenguaje	Género	No. de Problemas	No. de Documentos	Prom. documentos conocidos por	Prom. palabras por documento
Entrenamiento	Holandés	Ensayos	96	268	1.8	412.4
	Holandés	Reseñas	100	202	1.0	112.3
	Inglés	Ensayos	200	729	2.6	848.0
	Inglés	Novelas	100	200	1.0	3137.8
	Griego	Artículos	100	385	2.9	1404.0
	Español	Noticias	100	600	5.0	1135.6
		Total	**696**	**2384**	**2.4**	**1091.0**
Evaluación	Holandés	Ensayos	96	287	2.0	398.1
	Holandés	Reseñas	100	202	1.0	116.3
	Inglés	Ensayos	200	718	2.6	833.2
	Inglés	Novelas	200	400	1.0	6104.0
	Griego	Artículos	100	368	2.7	1536.6
	Español	Noticias	100	600	5.0	1121.4
		Total	**796**	**2575**	**2.2**	**1714.9**
Total			**1492**	**4959**	**2.3**	**1415.0**

Tabla 14: Propiedades generales del corpus utilizado tanto para entrenamiento como para las pruebas del sistema.

Por otro lado, el género de noticias en español tiene el mayor promedio de documentos conocidos por problema y eso resulta benéfico para la eliminación de incertidumbre y de nuevo, un mejor entrenamiento del sistema. Las reseñas en holandés tienen el promedio total de palabras por documento más bajo completando la peor configuración de entre todos los géneros ya que también cuenta con un número bajo de documentos conocidos por problema. El número promedio de documentos conocidos es importante ya que a mayor número de documentos, contamos con más información para la clasificación y por tanto, la tarea de determinar si el documento desconocido fue escrito por el mismo autor se torna más fácil. En casos contrarios, en problemas donde la comparación es de uno a uno como en el de las reseñas en holandés, la comparación se vuelve complicada por la poca información con la que se cuenta. Cabe destacar que el número de palabras por documento también ocasiona que el sistema no pueda determinar el estilo de escritura de los documentos. En el caso de las novelas en inglés se observó que a pesar de contar con un promedio bajo de documentos conocidos por

problema, los documentos desconocidos en los problemas de este género, tienen más palabras por documento que los documentos de autor conocido, proveyendo al sistema de más información y contrarrestando la falta de información o documentos para la comparación.

En resumen, una de las configuraciones más completas en información es la del género de noticias en español, de la que se espera que el sistema aproveche toda la información disponible y clasifique los documentos mejor que en el resto de los géneros; mientras que para el género de reseñas en holandés se espera que el sistema logre una de las puntuaciones más bajas o la más baja.

Capítulo 5. Experimentos

En este capítulo revisaremos el proceso de entrenamiento y los resultados de las ejecuciones que se realizaron con el sistema de verificación de autor propuesto. Para medir el rendimiento del sistema, se evaluó utilizando las métricas:

- AUC
- C@1

propuestas por la competencia PAN CLEF 2014. En el entrenamiento se consideraron diferentes configuraciones de representaciones, así como número de documentos impostores. También presentamos un experimento de tipo extirpación para evaluar la contribución de las características textuales.

5.1 Métricas

Para interpretar los resultados, primero explicaremos las métricas utilizadas en la tarea de identificación de autor de la competencia PAN CLEF 2014. Como el lector recordará, como parte de los requerimientos de la competencia, los organizadores solicitaron que cada sistema evaluara el corpus de evaluación descrito en el capítulo pasado y para cada problema calculara la probabilidad de que un documento desconocido perteneciera al mismo autor de los documentos conocidos en dicho caso. De esta manera, nuestra predicción fue evaluada utilizando estas mismas métricas y al final comparada con el *patrón oro*, el cual es el conjunto de todas las respuestas verdaderas a los problemas de la competencia y que fue provisto también por los organizadores.

La primer métrica denominada AUC, es el área bajo la curva ROC. ROC, acrónimo en inglés de Característica Operativa Relativa, es una representación gráfica de la razón entre los verdaderos positivos y los falsos positivos obtenidos en un sistema dado de clasificación. Por su parte AUC, es el área contemplada bajo la curva ROC que se obtiene de calibrar el sistema para diferentes umbrales de clasificación y de observar los verdaderos positivos y falsos positivos obtenidos bajo cada nuevo umbral. En pocas palabras, mientras menos falsos positivos y más verdaderos positivos se obtengan, el área bajo la curva será mayor indicando que un sistema está clasificando de forma correcta la mayoría de los documentos.

En PAN CLEF 2014, AUC fue utilizado como una medida escalar de medición debido a que ha demostrado ser una medida efectiva para evaluar una clasificación binaria, ya que no depende de un umbral en específico.

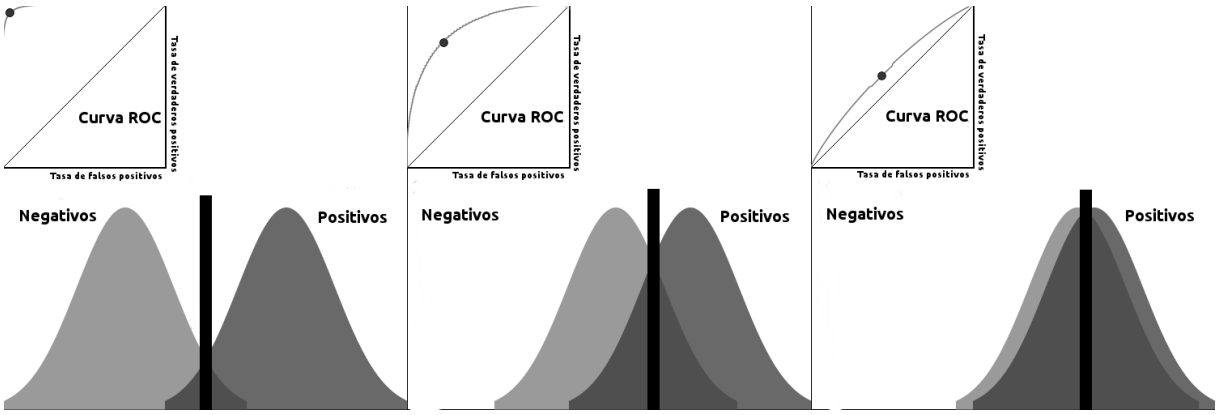


Figura 19: Visualización de un ejemplo de clasificación y su curva ROC correspondiente. Imagen tomada, editada y traducida de navan.name/roc

La segunda métrica utilizada es C@1, la cual está definida por la siguiente fórmula:

$$c@1 = \frac{1}{n} \cdot \left(n_c + \frac{n_u * n_c}{n} \right)$$

donde:

- n = número de problemas
- n_c = número de respuestas correctas
- n_u = número de problemas sin decisión

Esta métrica fue propuesta por Peñas A. & Rodrigo A. [18] y aceptada en la competencia de PAN, debido a que en muchas tareas como la presentada en este trabajo a veces es preferible responder que no se sabe la respuesta a un problema que responderlo de forma incorrecta. Esta respuesta se puede apreciar cuando un sistema responde a un problema con una probabilidad de 0.5. C@1 es una medida que busca premiar o dar mejor puntuaciones a sistemas que buscan mantener el mismo número de respuestas correctas y al mismo tiempo minimizar el número de respuestas incorrectas dando respuestas sin decisión.

De esta manera, la puntuación final para cada caso está dada por el producto de AUC y C@1.

5.2 Experimentos

Para este trabajo se llevó a cabo una serie de experimentos que sirvieron para encontrar la mejor configuración formada por las diferentes representaciones disponibles que estimamos serían capaces de dar la mejor clasificación de documentos en nuestro sistema.

Las variables que se modificaron para cada ejecución fueron el número de atributos en diferentes combinaciones presentes (ver tabla 12 del Capítulo 2.) y el número de impostores agregados a la matriz de entrenamientos, ambos factores para cada tipo de documento tanto por lenguaje como por género. El corpus utilizado para esta fase fue el mismo que los organizadores de la competencia PAN CLEF 2014 pusieron a disposición de los competidores para el entrenamiento de los sistemas.

Género	Atributos	AUC	C@1	Puntuación
Holandés: Ensayos	Todos	0.9572	0.8916	0.85348
Holandés: Reseñas	Todos	0.5936	0.5400	0.32054
Inglés: Ensayos	Todos	0.5002	0.5022	0.25123
Inglés: Novelas	Todos	0.8466	0.6344	0.53708
Griego: Artículos	Todos	0.7802	0.7176	0.55987
Español: Noticias	Todos	0.7910	0.6825	0.53986

Tabla 15: Puntuación de la fase de entrenamiento para cada género utilizando todos los atributos.

La puntuación obtenida en la fase de entrenamiento, disponible en la tabla 15, confirma diversas suposiciones realizadas en el Capítulo 4. como es el caso de las reseñas en holandés, género que obtuvo una de las puntuaciones más bajas tal como se esperaba dado que los textos de este género son demasiado cortos . Por su parte, el género de los ensayos en inglés demostró que el sistema requiere de algunas modificaciones ya que a pesar de contar con una buena configuración de datos, concluyó con la puntuación más baja de todos los géneros. Comparando los resultados de las dos categorías en inglés, observamos que las novelas obtuvieron mejores resultados. Con esto llegamos a la conclusión de que a mayor número de

palabras por documento, este idioma arroja mejores resultados. Adicionalmente, el género de noticias en español también concluyó con resultados menos sobresalientes de lo esperado al mismo tiempo que el género de ensayos en holandés obtuvo los mejores resultados de todos los géneros.

Género	Atributos	AUC	C@1	Puntuación
Holandés: Ensayos	Todos	93%	88%	82%
Holandés: Reseñas	Todos	57%	52%	30%
Inglés: Ensayos	Todos	57%	56%	32%
Inglés: Novelas	Todos	66%	61%	41%
Griego: Artículos	Todos	82%	75%	62%
Español: Artículos	Todos	75%	71%	54%

Tabla 16: Resultados oficiales de la fase de evaluación obtenidos en la competencia PAN CLEF 2014.

En la tabla 16, se ilustra la puntuación final obtenida en una segunda fase de pruebas, la cual se llevó a cabo como parte de la competencia PAN CLEF 2014 y en la que el sistema fue puesto a prueba en un escenario real con un corpus totalmente nuevo y desconocido. Es de notar que el sistema mejoró en general a pesar de haber obtenido puntuaciones diferentes a las obtenidas en la fase de desarrollo, en especial en el género de artículos en griego el cual mejoró 7% para la fase de prueba. Se cree que esto se debe principalmente a dos razones, la primera es el cambio de corpus donde diferencias importantes en el estilo de escritura de los artículos entre el corpus de entrenamiento y el corpus de prueba demostraron que el sistema es un tanto más sensible a la variación de ciertos atributos de lo que debería; las diferencias se pueden atribuir a un cambio drástico en el tamaño de los textos o los diferentes temas y palabras utilizadas en los documentos. La segunda sería la inestabilidad del algoritmo de homotopía, que si bien intenta encontrar la solución más dispersa y la que cuenta con el error mínimo en todos los casos, esta solución es una aproximación y nunca es la misma entre una ejecución y otra.

Debido a la premura de la competencia y falta de tiempo, nos fue imposible realizar más pruebas al sistema. Sin embargo, dos años después y para fines de

este trabajo y un análisis más detallado, se retomó la fase de entrenamiento y se llevaron a cabo pruebas de extirpación. Para estas pruebas, también se utilizó el corpus de entrenamiento provisto inicialmente por los organizadores de la competencia PAN CLEF 2014. Sobre este corpus, se llevó a cabo una validación cruzada equivalente en la que por cada prueba de extirpación en una categoría, se dejó fuera uno de los problemas de la misma para ser utilizado como corpus de evaluación. En estas pruebas se 'extirparon' una a una las 11 representaciones que fueron elegidas para el conjunto final del sistema definitivo que participó en la competencia PAN CLEF 2014. El objetivo de estas pruebas es el de medir la contribución que la presencia de un atributo dado está generando para la clasificación del documento. En la tabla 17, se observa que el 'extirpar' o quitar el atributo de trigramas a la representación vectorial de los documentos en el género de artículos en griego, aumenta en un 4% la puntuación final por encima de la reportada en la fase de entrenamiento utilizando todas las representaciones; en contraste, la eliminación de la representación de prefijos en el género de ensayos en holandés aumenta la pérdida concluyendo con una puntuación 6.39% más baja que la reportada en la fase de entrenamiento.

Representación extirpada	**Holandés: Ensayos**	**Holandés: Reseñas**	**Inglés: Ensayos**	**Inglés: Novelas**	**Griego: Artículos**	**Español: Noticias**
Bigramas	1.30%	3.03%	-1.20%	1.64%	3.85%	0.37%
Prefijos	6.39%	2.88%	-1.89%	3.70%	6.29%	1.66%
Puntuación	-1.29%	0.83%	0.84%	-2.41%	-0.68%	0.58%
Stopwords	1.26%	4.93%	-2.08%	0.49%	0.67%	0.13%
Sufijos	5.03%	-0.34%	-1.23%	4.88%	1.93%	0.91%
Trigramas	2.16%	2.45%	0.61%	-1.59%	-4.11%	1.28%
Palabra por oración	3.35%	1.49%	0.64%	2.45%	0.75%	1.47%
Bolsa de Palabras	0.52%	4.36%	-1.17%	-4.18%	0.22%	0.40%
Bigramas de Sufijos	0.73%	4.84%	0.42%	3.80%	3.98%	-1.35%
Bigramas de Prefijos	1.96%	0.28%	-1.35%	2.21%	-0.45%	0.48%
Bigramas de Stopwords	1.59%	2.31%	-2.53%	-1.55%	-0.29%	3.82%

Tabla 17: Porcentaje de pérdida por representación extirpada durante las pruebas del

sistema realizadas con validación cruzada equivalente sobre el corpus de entrenamiento.

Revisando de forma más detallada los resultados finales de extirpación de la tabla 17, suponemos que una mejor configuración para los ensayos en holandés hubiera sido posiblemente alcanzada únicamente con la extirpación del atributo de puntuación, aumentando en 1.29% la puntuación final. En el caso de las reseñas en holandés, eliminar de la representación final el atributo de sufijos habría aumentado la puntuación 0.34%, siendo ésta la única modificación disponible para la mejora del sistema en relación con este género. Por otro lado, ésta también fue la categoría que reportó la puntuación más baja de entre todos los géneros. En el caso de los ensayos en inglés, es en el que se observa que al eliminar el mayor número de atributos de entre todos los géneros, aparentemente debería aumentar la puntuación final. En este género donde el hecho de extirpar individualmente hasta siete atributos (de los 11 disponibles) nos reporta aumento en la puntuación, una de nuestras hipótesis fue que la puntuación final podía aumentar aún más al hacer la extirpación simultánea de los siete atributos, dejándonos con una representación formada únicamente por cuatro atributos que son el de puntuación, trigramas, palabras por oración y bigramas de sufijos.

En el género de novelas en inglés, al remover individualmente los atributos de puntuación trigramas, bolsa de palabras y los bigramas de palabras comunes, efectivamente observamos una mejora en los resultados de identificación de autor de hasta 4.18% en el mejor caso, el cual se trata de la bolsa de palabras. Para el caso de los artículos en griego también se reportan mejoras con la eliminación individual de cuatro atributos donde el de trigramas es el que más incrementa la puntuación, con un 4.11%. Finalmente, en el género de artículos en español, se reporta únicamente el atributo de bigramas de sufijos, cuya eliminación aumenta en 1.35% el rendimiento del sistema.

Sin embargo, si bien la extirpación de una sola representación resulta en una mejora, la extirpación simultánea de varias representaciones que individualmente lo son, no necesariamente lo es. Después de varias pruebas en las que intentamos eliminar más de una representación a la vez, concluimos que la hipótesis realizada

en el caso de los ensayos en inglés (y en general para cualquier otro) resultó equivocada. Tal como lo observamos en las pruebas realizadas a otras categorías, la mejora sólo se logró obtener con la extirpación individual y excluyente por categoría; i.e., que únicamente es posible observar un incremento en la puntuación final cuando se extirpa una y sólo una de las representaciones a la vez. Por ejemplo, para el caso de los ensayos en inglés, realizamos la extirpación simultánea de las siete representaciones y realizamos una validación cruzada equivalente con la representación formada por puntuación, trigramas, palabras por oración y bigramas de sufijos. Después de probar esta configuración de extirpación y promediar todos los casos para este mismo género, obtuvimos una puntuación final de 0.27 (27%) o una pérdida en el rendimiento de 0.02 (2%) respecto a lo reportado sin extirpación alguna.

En general, el atributo de puntuación aparentemente es el que más daño hace al rendimiento del sistema al ser incluido en la representación final; ya que al ser removido en 3 de los 6 géneros, los resultados de estos géneros mejora. Por otro lado, el atributo de palabras por oración es el único cuya presencia demuestra ser una mejora constante para el sistema en todos los géneros. Debemos destacar que la realización de estas pruebas de extirpación sirvió más para entender la naturaleza de los atributos y su impacto en los diferentes géneros e idiomas y no como una herramienta para la calibración perfecta del sistema. Esto es debido a que al intentar adaptar nuestro sistema lo mejor posible al corpus de la competencia PAN CLEF podríamos caer en un sobre ajuste del modelo que posiblemente nos haría obtener muy buenos resultados para el caso específico de la competencia PAN CLEF, pero al mismo tiempo nos daría pésimos resultados al ser trasladado a otro problema con un corpus totalmente nuevo y diferente.

Capítulo 6. Conclusiones

En este trabajo pusimos en práctica el uso de un método para la identificación de señales conocido como clasificación basada en la representación dispersa. Con este logramos identificar la autoría de textos clasificándolos en diversos géneros (ensayo, artículo, reseña, novela) e idiomas (Inglés, Español, Griego, Holandés) con una tasa de éxito aceptable.

Para alcanzar el final del trabajo, se definieron y atendieron diversos problemas. El primero que alcanzamos fue la implementación de un preprocesador necesario para proyectar o convertir los datos a analizar, es decir, los documentos de texto a una representación vectorial. Esta representación facilita a nuestro sistema manipular los datos textuales. Para tal efecto, se hizo una investigación profunda respecto a la representación de texto más utilizada actualmente en el campo, llegando a la conclusión de que el modelo de abstracción basado en la concatenación de atributos estadísticos y de frecuencias, como el conteo de palabras, funcionaría bien.

Posteriormente, ya que contamos con una forma de representar el texto y manipularlo, nos enfrentamos al problema de resolver el sistema de ecuaciones de la forma $Ax=y$ buscando minimizar la norma L1 del vector solución a través del método de Homotopía. Lo anterior con el fin de generar un vector disperso con el cual seríamos capaces de clasificar el texto de entrada y finalmente definir la autoría de este. Para este paso recurrimos a la búsqueda de una implementación del método en código que pudiéramos adaptar para nuestras necesidades. Tras la búsqueda, logramos encontrar una implementación de Homotopía en el lenguaje 'M' de MATLAB desarrollado por la Universidad de Berkeley [19], que resultó fácil de modificar e integrar como un módulo al sistema, el cual se encuentra implementado en Python con el apoyo de las bibliotecas NumPy, Scikit-Learn y Oct2py. En un inicio y tras algunas modificaciones, la integración no resultó como esperábamos entregando resultados incoherentes e inconsistentes; por lo que se requirió de depuración, pruebas y modificaciones adicionales como la corrección del rango de índices de los vectores, dimensión de las matrices y el manejo de valores numéricos. También, se ajustó el número de iteraciones para la convergencia del algoritmo, así como el umbral de error que controla la minimización de la norma en los vectores resultantes.

Una vez que estuvieron listos los módulos de proyección de documentos y el de Homotopía, proseguimos a decidir qué tipo de atributos y qué número de ellos funcionaría mejor para el sistema partiendo de que una buena selección y extracción de atributos hace la diferencia al momento de iniciar la construcción de un modelo de verificación. Tras diversas pruebas que llevamos a cabo de forma individual y que denominamos de 'extirpación', comparamos las puntuaciones obtenidas de aplicar diferentes combinaciones de atributos dependiendo del género e idioma en turno. Finalmente, con base en resultados preliminares obtenidos durante la fase de entrenamiento del sistema, decidimos utilizar los atributos de frecuencia:

- Bolsa de palabras
- Bigramas

- Trigramas
- Prefijos
- Sufijos
- Bigramas de prefijos
- Bigramas de sufijos
- Palabras comunes
- Bigramas de palabras comunes
- Signos de puntuación
- Palabras por oración

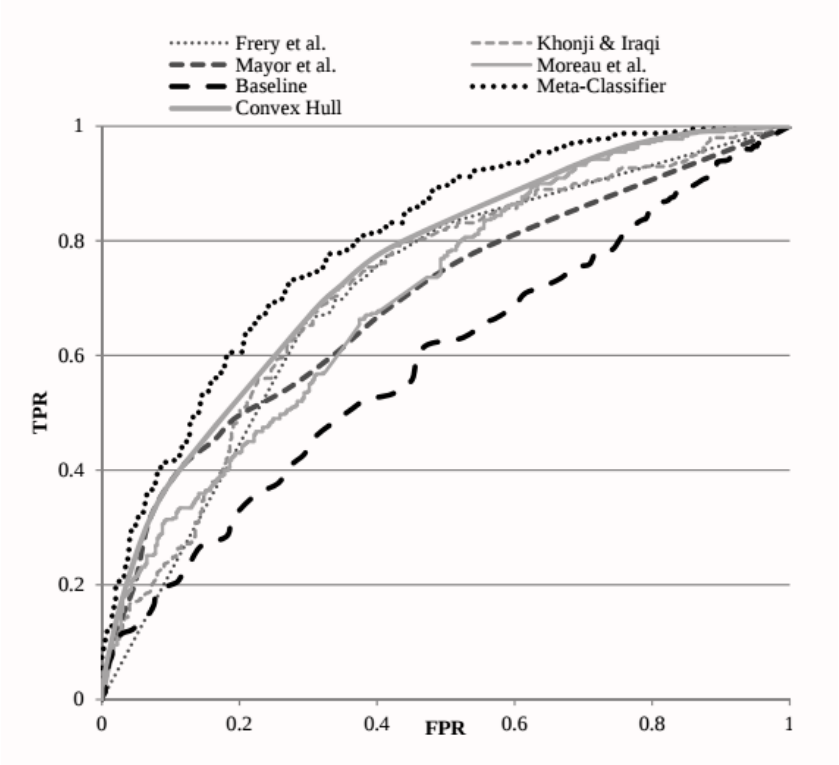


Figura 20: Curvas ROC de diferentes sistemas participantes en la competencia PAN CLEF 2014 comparadas contra la línea base y el meta-clasificador. Imagen modificada y tomada del artículo Overview of the Author Identification Task at PAN 2014.

En la figura 20, se observa el rendimiento del sistema identificado por la leyenda

Mayor et al., junto con el de otros sistemas participantes en la competencia PAN CLEF 2014 donde se visualizan las curvas ROC de los sistemas con el mejor rendimiento, del casco convexo, de la *línea base* y del *meta-clasificador* para esta competencia. En un inicio, la *línea base* había sido representada por una certeza aleatoria de 0.5 tanto para la métrica de AUC como para C@1; es decir, una con la misma tasa de éxito que el hecho de intentar adivinar lanzando una moneda al aire, si un documento fue escrito por un autor o no. Sin embargo esta línea fue descartada por los orgnizadores de la competencia por no representar reto alguno. Debido a que se requería una *línea base* más completa y acorde al estado del arte de la tarea, se eligió como *línea base* al desempeño de uno de los sistemas participantes y ganador de la competencia PAN CLEF 2013 [8] ya que contaba con un enfoque independiente de los idiomas, además de que este sistema era capaz de generar puntuaciones tanto binarias como reales y dichas puntuaciones ya se encontraban calibradas de tal forma que podían utilizarse como probabilidades cuyos valores mayores a 0.5 eran considerados como respuestas afirmativas.

En la misma figura 20, la curva etiquetada con la leyenda *meta-clasificador* representa el rendimiento del *meta-modelo* resultante de combinar todas las respuestas dadas por los sistemas participantes en cada problema y sacar un promedio. Por su parte, la curva del casco convexo está formada por los resultados de los mejores sistemas para cada combinación de género e idioma. Por ejemplo, nuestro sistema identificado por la leyenda *Mayor et al. [20]*, forma parte del casco convexo en algunos puntos ubicados a la izquierda de la gráfica de la figura 20 debido a que obtuvo el mejor resultado general en la evaluación de problemas de ensayos en holandés. Al mismo tiempo, el sistema siempre se mantiene por encima de la línea base a excepción del segmento relacionado con las reseñas en Holandés, donde su desempeño fue el más bajo en comparación con los demás géneros evaluados por nuestro mismo sistema.

En conclusión, el sistema obtuvo una tasa de éxito de 74.1% en la fase de desarrollo y de 71.6% en la fase de competencia. Lo cual significa 7 de cada 10 son respuestas acertadas en las que el sistema es capaz de identificar y contestar correctamente si un texto ha sido escrito por un autor determinado o no.

Tras intentar emular los buenos resultados observados en otras áreas y utilizar como punto de referencia trabajos análogos de visión computacional con representación dispersa para el reconocimiento de rostros [5], en el que gracias al uso de la representación dispersa la elección de los atributos pasó a un segundo plano y se tornó menos crucial que el número de atributos utilizados debido a que la propiedad de dispersión es aprovechada correctamente, podemos concluir que aún nos falta mucho por desarrollar en nuestro sistema y que si bien, el uso de esta metodología no alcanzó las expectativas hablando en una connotación general de reconocimiento de patrones donde otros sistemas robustos de reconocimiento de rostros [5] han reportado tasas de éxito de 93.6%, los resultados obtenidos en este trabajo significan un gran avance en el área de verificación de textos donde aún se está trabajando por superar diversos obstáculos.

Por otro lado, vale la pena destacar que una de las motivaciones para la realización de este trabajo fue la competencia de PAN CLEF 2014 donde se concursó con este sistema en la tarea de identificación de autor obteniendo el quinto lugar de entre 13 participantes. En esta misma competencia, donde se da a conocer el estado del arte de sistemas utilizados para problemas de lingüística forense, el primer lugar fue para el equipo ASGALF [12] cuyo sistema reportó una tasa de éxito de 76.5% en la identificación de autores; demostrando que la tarea de identificación de autores, aún es un reto para la comunidad científica y de la misma manera, que el rendimiento del sistema descrito en el presente trabajo es bastante competitivo.

Cabe destacar que gracias al desarrollo de este trabajo, se logró la publicación de un par de artículos de investigación. En el artículo:

- *Mayor, C., Gutierrez, J., Toledo, A., Martinez, R., Ledesma, P., Fuentes, G. y Meza, I.: A Single Author Style Representation for the Author Verification Task. Working Notes for CLEF 2014 Conference, 2014* [20].

se expone este mismo sistema y el uso de la clasificación basada en la representación dispersa propuesta para la identificación de textos, así como su rendimiento durante su participación en la competencia PAN CLEF del año 2014. Por otro lado y en un campo ajeno al de la verificación de autores, también participé en la publicación del artículo:

- *Martinez, R., Silva, L., Villarreal, T., Fuentes, G. y Meza, I.: SVM Candidates and Sparse Representation for Bird Identification.Working Notes for CLEF 2014 Conference, 2014* [21].

en el cual se propone un sistema para la identificación de cantos de aves utilizando la misma metodología de clasificación dispersa abordada en este trabajo y que en lugar de utilizar documentos de textos como señales de entrada, se utilizan grabaciones de cantos de aves.

Adicionalmente, después de haber estudiado los resultados y revisado los obstáculos que surgieron del desarrollo de este sistema, creemos que diversas modificaciones mejorarían el rendimiento del sistema en una segunda iteración de desarrollo. La adición y eliminación exhaustiva de diferentes atributos para la representación vectorial es una de ellas y las pruebas de extirpación de atributos lo demostraron reportando aumentos en la tasa de éxito de identificación con diferentes combinaciones que no fueron contempladas para la versión final del sistema. También, creemos que la elección de mejores candidatos a impostores podría aumentar la afinidad de género y contenido entre documentos eliminando ruido innecesario en los datos.

A futuro, consideramos que la inclusión de técnicas de análisis automático más específicas y acordes al idioma y género serán de gran importancia para la extracción y elección de atributos. También, nos mantenemos firmes en la idea de que el uso de impostores deberá ser una constante en el desarrollo de futuras versiones. Esto concuerda con que la mayoría de las propuestas en el estado del arte, usan impostores reportando mejores resultados que aquellos que no los utilizan.

Apéndice

De forma complementaria, presentamos la tabla completa de extirpación de representaciones con los resultados desglosados para cada métrica:

Género	Atributo Extirpado	AUC	C@1	Puntuación
Holandés: Ensayos	Bigramas	0.94466	0.88976	0.84052
Holandés: Ensayos	Prefijos	0.92817	0.85069	0.78959
Holandés: Ensayos	Puntuación	0.95052	0.91146	0.86636
Holandés: Ensayos	Stopwords	0.94314	0.8916	0.84091
Holandés: Ensayos	Sufijos	0.93251	0.86133	0.8032
Holandés: Ensayos	Trigramas	0.95074	0.8750	0.8319
Holandés: Ensayos	Palabra por oración	0.93294	0.87891	0.81997
Holandés: Ensayos	Bolsa de Palabras	0.94813	0.89464	0.84824
Holandés: Ensayos	Bigramas de Sufijos	0.95877	0.8826	0.8462
Holandés: Ensayos	Bigramas de Prefijos	0.93359	0.89323	0.83391
Holandés: Ensayos	Bigramas de Stopwords	0.94141	0.88976	0.83762
Holandés: Reseñas	Bigramas	0.5634	0.5151	0.29021
Holandés: Reseñas	Prefijos	0.545	0.5353	0.29174
Holandés: Reseñas	Puntuación	0.583	0.5356	0.31225
Holandés: Reseñas	Stopwords	0.5164	0.5252	0.27121
Holandés: Reseñas	Sufijos	0.6108	0.5304	0.32397
Holandés: Reseñas	Trigramas	0.553	0.5353	0.29602
Holandés: Reseñas	Palabra por oración	0.5762	0.5304	0.30562
Holandés: Reseñas	Bolsa de Palabras	0.5074	0.5459	0.27699
Holandés: Reseñas	Bigramas de Sufijos	0.5084	0.5353	0.27215
Holandés: Reseñas	Bigramas de Prefijos	0.5826	0.5454	0.31775
Holandés: Reseñas	Bigramas de Stopwords	0.572	0.5200	0.29744
Inglés: Ensayos	Bigramas	0.5187	0.50738	0.26318
Inglés: Ensayos	Prefijos	0.52225	0.51728	0.27015
Inglés: Ensayos	Puntuación	0.4892	0.49638	0.24283
Inglés: Ensayos	Stopwords	0.5284	0.5148	0.27202
Inglés: Ensayos	Sufijos	0.53555	0.4920	0.26349
Inglés: Ensayos	Trigramas	0.49855	0.49163	0.2451
Inglés: Ensayos	Palabra por oración	0.48485	0.5050	0.24485
Inglés: Ensayos	Bolsa de Palabras	0.5237	0.50197	0.26288
Inglés: Ensayos	Bigramas de Sufijos	0.48695	0.50738	0.24707
Inglés: Ensayos	Bigramas de Prefijos	0.5116	0.5175	0.26475
Inglés: Ensayos	Bigramas de Stopwords	0.529	0.52275	0.27653
Inglés: Novelas	Bigramas	0.8130	0.6405	0.52073
Inglés: Novelas	Prefijos	0.8504	0.5880	0.50004
Inglés: Novelas	Puntuación	0.8620	0.6510	0.56116
Inglés: Novelas	Stopwords	0.8416	0.6324	0.53223
Inglés: Novelas	Sufijos	0.8194	0.5959	0.48828
Inglés: Novelas	Trigramas	0.844	0.6552	0.55299
Inglés: Novelas	Palabra por oración	0.8376	0.612	0.51261
Inglés: Novelas	Bolsa de Palabras	0.8516	0.6798	0.57892
Inglés: Novelas	Bigramas de Sufijos	0.7998	0.624	0.49908

Género	Atributo Extirpado	AUC	C@1	Puntuación
Inglés: Novelas	Bigramas de Prefijos	0.8772	0.5871	0.5150
Inglés: Novelas	Bigramas de Stopwords	0.8688	0.6360	0.55256
Griego: Artículos	Bigramas	0.7686	0.6784	0.52142
Griego: Artículos	Prefijos	0.7466	0.6656	0.49694
Griego: Artículos	Puntuación	0.7974	0.7107	0.56671
Griego: Artículos	Stopwords	0.7748	0.714	0.55321
Griego: Artículos	Sufijos	0.7772	0.6955	0.54054
Griego: Artículos	Trigramas	0.8218	0.7313	0.60098
Griego: Artículos	Palabra por oración	0.7772	0.7107	0.55236
Griego: Artículos	Bolsa de Palabras	0.7896	0.7062	0.55762
Griego: Artículos	Bigramas de Sufijos	0.7726	0.6732	0.52011
Griego: Artículos	Bigramas de Prefijos	0.8022	0.7035	0.56435
Griego: Artículos	Bigramas de Stopwords	0.7958	0.7072	0.56279
Español: Noticias	Bigramas	0.8082	0.6634	0.53616
Español: Noticias	Prefijos	0.7962	0.6572	0.52326
Español: Noticias	Puntuación	0.7922	0.6741	0.53402
Español: Noticias	Stopwords	0.7938	0.6784	0.53851
Español: Noticias	Sufijos	0.7874	0.6741	0.53079
Español: Noticias	Trigramas	0.8044	0.6552	0.52704
Español: Noticias	Palabra por oración	0.7916	0.6634	0.52515
Español: Noticias	Bolsa de Palabras	0.7852	0.6825	0.5359
Español: Noticias	Bigramas de Sufijos	0.808	0.6848	0.55332
Español: Noticias	Bigramas de Prefijos	0.7902	0.6771	0.53504
Español: Noticias	Bigramas de Stopwords	0.7634	0.6572	0.50171

Bibliografía

[1] David Zax. How did computers uncover J. K. Rowling's pseudonym? *Smithsonian Magazine*, 2014.

[2] Patrick Joula. How a computer program helped show J. K. Rowling write a Cuckoo's calling? *Scientific American*, 2013.

[3] Gerard Salton, Andrew Wong y Chungshu Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 1975.

[4] Gerard Salton y Chris Buckley. Term-weighting approaches in automatic text retrieval. *InformationProcessing and Management*, 1988.

[5] John Wright, Allen Yang, Arvind Ganesh, Shankar Sastry. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[6] Robert Audi. *The Cambridge Dictionary of Philosophy*, 1999.

[7] Efstathios Stamatos, Martin Potthast, Francisco Rangel, Paolo Rosso y Benno Stein. *Overview of the PAN/CLEF 2015 Evaluation Lab*, 2015.

[8] Shachar Seidman. Authorship Verification Using the Impostors Method. *PAN-CLEF Authorship Identification Workshop*, 2013.

[9] Patrick Joula y Efstathios Stamatatos. Overview of the Author Identification Task at PAN 2013. P*AN-CLEF Authorship Identification Workshop*.

[10] Nicholas Jardine y Cornelis Joost van Rijsbergen. The use of hierarchical clustering in information retrieval. *Information Storage and Retrieval,* 1971.

[11] Moshe Koppel y Yaron Winter. Determining if Two Documents are by the Same Author. *Journal of the American Society for Information Science and Technology*, 2014.

[12] Mahmoud Khonji y Youssef Iraqi. A Slightly-modified GI-based Author-verifier with Lots of Features (ASGALF). *PAN-CLEF Authorship Identification Workshop*, 2014.

[13] Paul Jaccard. Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 1901.

[14] Edoardo Amaldi y Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 1998.

[15] David L. Donoho. For Most Large Underdetermined Systems of Linear Equations, the Minimal l1-Norm Solution Approximates the Sparsest Near-Solution. *Communications on Pure and Applied Mathematics*, 2006.

[16] David L. Donoho, Yaakov Tsaig, Iddo Drori y Jean-Luc Starck. Sparse Solution of Undetermined Linear Equations by Stagewise Orthogonal Matching Pursuit. *Technical Report (Stanford University. Dept. Of Statistics)*, 2006.

[17] David L. Donoho y Yaakov Tsaig. Fast Solution of L1-norm Minimization Problems When the Solution May be Sparse. *Technical Report (Stanford University. Dept. Of Statistics),* 2006.

[18] Anselmo Peñas, Yusuke Miyao, Álvaro Rodrigo, Eduard Hovy y Noriko Kando. Overview of CLEF QA Entrance Exams Task 2014. *Working Notes for CLEF 2014 Conference*, 2014.

[19] Allen Yang, Arvind Ganesh, Shankar Sastry y Yi Ma. Fast L1-Minimization Algorithms for Robust Face Recognition. *Technical Report (Electrical Engineering and Computer Sciences University of California at Berkeley)*, 2010.

[20] Cristhian Mayor, Josué Gutierrez, Angel Toledo, Rodrigo Martínez, Paola Ledesma, Gibrán Fuentes y Ivan Meza. A Single Author Style Representation for the Author Verification Task. *PAN-CLEF Authorship Identification Workshop*, 2014.

[21] Rodrigo Martínez, Laura Silva, Toaki Esaú Villarreal Olvera, Gibrán Fuentes e Ivan Vladimir Meza Ruiz. SVM Candidates and Sparse Representation for Bird Identification. *Working Notes for CLEF 2014 Conference*, 2014.