

II Jornadas Internacionales de Estadística Aplicada 5 y 6 de diciembre de 2019

Minería de Textos aplicada a la Atribución de autoría utilizando los clasificadores SVM y kNN

Autores: Pablo Nicolás Ramos, Melisa Rocío Valdiviezo, José Humberto Farfán, Mariela Ester Rodríguez, Ariel Alejandro Vega

Facultad de Ingeniería, Universidad Nacional de Jujuy,
San Salvador de Jujuy

Datos de contacto: pablonicolasr777@gmail.com (3884207007), melisan8012@gmail.com (3884586212), jhfarfan@gmail.com (3884804982), maru972@gmail.com (3884719818), arielalejandrovega@gmail.com (3885209480)

RESUMEN.

En las últimas décadas los escritos, novelas, artículos científicos, entre otros, pasaron de su formato clásico de publicación al electrónico. Debido al incremento de material plagiado en la web se deben suministrar herramientas automáticas que permitan detectarlo. Entonces, en este trabajo se propone abordar la tarea de atribución de autoría automática de textos digitales a través del Aprendizaje Automatizado. Se presentan dos clasificadores de textos, uno basado en Support Vector Machine, y otro basado en los k vecinos más cercanos, ambos representados mediante el modelo de la bolsa de 3-grams a nivel de caracteres. Estos clasificadores fueron evaluados a través de un conjunto de textos que provee el PAN, demostrando que el kNN obtiene mejor rendimiento en el Dataset elegido.

Palabras Claves: Minería de Textos, Atribución de Autoría, SVM, kNN, Similitud.

1. Introducción

La tarea de Atribución de Autoría (AA) es un área de investigación que ha ganado interés creciente en los últimos años principalmente por sus potenciales (y actuales) aplicaciones en problemas de seguridad nacional e inteligencia, lingüística forense, análisis de mercados e identificación de rasgos de personalidad, entre otros. El AA se enfoca en la clasificación automática de textos basándose fundamentalmente en las elecciones estilísticas de los autores de los documentos, e incluye distintas tareas de análisis como por ejemplo la atribución de autoría, la verificación de autor, la detección de plagios, la determinación del perfil del autor y la detección de inconsistencias estilísticas. Los enfoques predominantes en esta área están basados en el aprendizaje automático/de máquina supervisado. En pocas palabras, estos enfoques derivan, a partir de un conjunto de datos etiquetados (conjunto de entrenamiento) y un proceso inductivo de aprendizaje/entrenamiento, un clasificador que puede generalizar sus predicciones a otros datos no observados previamente. La representación clásica de los textos/documentos en estos casos, incluye tanto atributos basados en el contenido (palabras) como en el estilo de escritura de los autores. A partir de la disponibilidad de volúmenes inmensos de información en la Web, se reconoce cada día más el rol de la AA como una herramienta fundamental para hacer un uso adecuado y

ventajoso de esta información, lo que ha quedado plasmado en un incremento de Workshops y Competencias Internacionales específicos de esta temática. Más allá de la relevancia y ventajas que pueden tener este tipo de tareas existe, actualmente, un desarrollo limitado en nuestro país de trabajos y grupos de investigación especializados en la problemática del AA. En este contexto, en la presente línea de investigación se enfoca en la atribución de autoría como un problema de clasificación multiclase.

2. Metodología

La metodología que se adopta en el presente trabajo es la descrita por las investigaciones de análisis de autoría realizadas por Stamatatos (2009) y Castillo-Juarez (2012). La misma se centra en 3 etapas:

1. **Selección de rasgos y características de redacción:** El análisis de AA se lleva a cabo a nivel de caracteres y se utilizan rasgos léxicos. La elección de este enfoque se basa en la disponibilidad de herramientas del Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés) para varios idiomas. Este enfoque obtiene una buena performance, según lo reportado en los trabajos seleccionados para la selección de la metodología aplicada en este documento. También se podrían utilizar rasgos sintácticos y semánticos, pero no aportan significativos aumentos de precisión, según resultados experimentales y consideraciones de los autores de los trabajos mencionados.
2. **Construcción computacional de un modelo de escritura:** Esta propuesta utiliza el modelo Bag-of-n-grams para la representación de documentos. Este modelo de representación registra la cantidad de veces que aparece cada n-gram en cada documento de una colección. Un n-gram es una secuencia de n objetos sucesivos; esos objetos pueden ser palabras o caracteres. La frecuencia de aparición de cada n-gram se puede representar mediante un histograma de frecuencia.
3. **Método de aprendizaje para la clasificación e identificación del autor:** es la etapa en la que se toma la decisión sobre la autoría de un documento sospechoso o anónimo. Se responde a la pregunta: “¿Quién fue el autor de dicho documento sospechoso?”. Los clasificadores basados en instancias responden adecuadamente en tareas de clasificación de documentos y, la Atribución de Autoría puede considerarse una sub-tarea de clasificación de documentos en la que se debe hacer especial énfasis (Castro, 2019).

3. Desarrollo

3.1. Análisis Automático de Atribución de Autoría

Este trabajo se enfoca en la Atribución de Autoría (AA) basada en métodos estadísticos y computacionales. Esto se basa en estudiar atributos textuales que al medirlos nos permitan diferenciar entre documentos escritos por distintos autores (Stamatatos, 2009). La tarea de AA es considerada como un problema de clasificación multiclase; y comparte procesos similares con otras tareas íntimamente relacionadas con el tratamiento automático de texto (por ejemplo, la clasificación temática). No obstante, existen importantes diferencias entre la AA y otros problemas de clasificación de documentos, sobre todo en el tipo de características textuales que se extraen de los textos. Un ejemplo de ello, es que en la Atribución de Autoría son más relevantes los atributos de estilo de escritura que los de contenido (López-Monroy, 2012).

3.1.2. Tareas principales de la AA

La tarea de Atribución de Autoría es un subcampo de la Minería de Datos, que comprende dos enfoques principales, ambos con el objetivo de conocer la autoría de un documento de texto en un idioma dado, pero en contextos distintos.

El primer enfoque se conoce como **Identificación de Autoría (IA)**; y consiste en predecir el

autor de un documento de texto, dado un conjunto de autores candidatos para los cuales se tienen disponibles textos de su autoría (Stamatatos, 2009). Esta tarea, se puede considerar como un problema de clasificación multiclase de una etiqueta; donde cada documento pertenece a un autor, y la cantidad de autores determinan la cantidad de clases a distinguir. Este enfoque presenta dos subtareas; por un lado, tenemos, **1) Clase Cerrada**, donde se puede asumir que el documento a clasificar pertenece a algunos de los autores candidatos; y por el otro lado, tenemos, **2) Clase Abierta**, donde el documento a clasificar puede no pertenecer a ninguno de los autores candidatos.

En el segundo enfoque tenemos la tarea de **Verificación de Autoría (VA)**. En este enfoque solamente se cuenta con el autor y sus documentos (López Monroy, 2012), el objetivo es determinar si los documentos de prueba pertenecen o no a dicho autor (Argamon y Juola, 2011). Este caso se considera como un problema de clasificación uniclase (Koppel y Schler, 2004). Este enfoque está íntimamente relacionado con la detección intrínseca de plagio (Pérez Afonso, 2013).

Este trabajo se enfoca en la tarea de Identificación de Autoría (IA) de Clase Cerrada. Entonces se puede definir formalmente la tarea de atribución de autoría como un problema de clasificación multiclase. La misma se define en la sección 3.2.

3.2. Atribución de Autoría como un problema de clasificación multiclase

Desde el punto de vista de los métodos de clasificación, la Atribución de Autoría (AA) se puede definir formalmente de la siguiente manera ():

Se tiene un conjunto de documentos $D = \{d_1, \dots, d_n\}$, donde cada documento d_i se expresa como $d_i = \{t_1, \dots, t_m\} \in \mathcal{R}^m$; d_i es un vector en un espacio m dimensional; \mathcal{R}^m se denomina feature space, espacio de variables o espacio de características. Cada documento tiene asignada una clase o etiqueta real $l(d_i) = \mathfrak{A}_i$ conocida de antemano; \mathfrak{A} representa el conjunto de autores (cantidad de clases). Para un problema con \mathfrak{A} autores, $l(d_i)$ puede tomar \mathfrak{A} valores discretos distintos.

3.2.1. Obtención de un modelo de AA

En el trabajo realizado por Stamatatos (2009), se describen tres métodos para obtener un modelo de Atribución de Autoría, y se describen a continuación:

1. **Basados en el perfil:** La idea consiste en modelar el estilo de escritura basándose en una cantidad de texto representativa del autor. Éste podría ser obtenido a partir de la concatenación de todos sus documentos. La idea principal es ignorar las pequeñas diferencias entre sus documentos, y extraer características del estilo general (perfil) de escritura.
2. **Basados en instancias:** Se basan en la utilización de múltiples instancias de texto del autor. La idea es extraer características de estilo comunes a nivel documento. Los textos se representan como vectores de atributos, para luego utilizar algún algoritmo de clasificación.
3. **Híbridos:** Éstos combinan características de los dos anteriores. Por ejemplo, representar de manera individual cada documento, pero utilizando características obtenidas a nivel clase. Es decir, se aplica algún algoritmo de clasificación tal como en los métodos basados en instancias, pero sobre vectores de documentos cuyas características textuales fueron extraídas a partir del perfil de escritura de cada autor, tal como en los métodos basados en perfil.

Este trabajo se enfoca en los modelos de AA basados en instancias. En la siguiente sección se describe el proceso genérico de este enfoque.

3.2.2. Obtención de un modelo de AA

En el trabajo realizado por Cagnina et al (2011), se mencionan 3 tareas estilográficas fundamentales en aplicaciones de atribución de autoría:

1. Caracterización del estilo de escritura del autor, \mathcal{A}_i , de un modelo único.
2. Detección de Similitudes.
3. Y finalmente, la Atribución de Autoría.

3.2.2.1. Construcción de un modelo de escritura

Para la construcción del modelo de estilo de escritura de un autor \mathcal{A}_i , resulta importante seleccionar un conjunto de características del documento, d_i , que permitan determinar aquellos fragmentos t_j cuyo estilo de escritura presenten una variación significativa respecto a los demás fragmentos. Las medidas estilísticas más importantes según Pérez Afonso (2013), son:

1. **Longitud media de las Palabras.** Esta medida, es el promedio de la longitud de las palabras calculada en caracteres.
2. **Longitud media de las Sentencias.** Esta medida, es el promedio de la longitud de las sentencias del documento calculada en palabras.
3. **Índice de Gunning Fox.** Esta medida se ha desarrollado para medir el grado de legibilidad que tiene un documento escrito. El resultado que se obtiene, indica la cantidad aproximada de años que necesita una persona en formación para comprender un texto.

$$I_G = 0,4 \left(\frac{|palabras|}{|sentencias|} + 100 \frac{|palabras\ complejas|}{palabras} \right) \quad (1)$$

Donde palabras complejas: son las que tienen al menos tres sílabas (menos nombres propios y sufijos, como es, ed o ing), $|palabras|$: número de palabras en el texto evaluado, y $|sentencias|$: número de sentencias en el texto evaluado.

4. **Función R.** Esta medida obtiene la variedad de vocabulario de un documento escrito.

$$R = \frac{100 \times \log(M)}{M^2} \quad (2)$$

Donde M es el número de palabras en el documento.

5. **Función K.** Esta función es una variación de la función R, y mide el cálculo de la riqueza de vocabulario.

$$K = \frac{10^4 (\sum_{i=1}^{\infty} i^2 V_i - M)}{M^2} \quad (3)$$

Donde M es el número de palabras en el documento y V_i es el vector que representa el número de veces que aparece cada palabra en el documento.

6. **Histograma global de palabras** (Funes y Errecalde, 2012): La ocurrencia de las palabras del documento es representada por un único histograma.
7. **Histograma local de palabras** (Escalante et al., 2011): La ocurrencia de las palabras de cada fragmento t_i de un documento de texto son representadas por un histograma.

3.2.2.2. Detección de Similitud Textual

La tarea de similitud textual se encarga de comparar textos para determinar la similitud entre ellos. Existen diversas métricas y medidas para obtener el grado de similitud textual entre dos documentos. Estas medidas tradicionalmente se clasifican en dos grandes grupos: medidas compuestas y medidas no compuestas (Álvarez Carmona, 2014). Las medidas compuestas dividen los textos regularmente en palabras, luego evalúan las palabras de ambos textos realizando una comparación 1 a 1. Al finalizar el proceso se obtiene un único resultado que determina la similitud de entre todos los fragmentos en los cuales se dividió los textos. Por otro lado, las medidas no compuestas, primero representan los textos mediante algún modelo abstracto, como los descritos en la sección 3.2.2.1, para posteriormente, realizar la comparación de textos.

3.2.2.2.1. Modelos de Similitud basadas en Teoría de Conjuntos

Las medidas de similitud basadas en el Modelo de Teoría de Conjuntos (Tversky, 1977) más utilizadas son:

1. **Coefficiente de Jaccard:** mide el grado de similitud entre dos conjuntos (en este caso, documentos), y se obtiene al dividir la intersección de términos entre la unión de los mismos [6]. Se denota:

$$sim_j(t_1, t_2) = \frac{|t_1 \cap t_2|}{|t_1 \cup t_2|} \quad (5)$$

Siempre toma valores entre 0 y 1, correspondiente este último a la igualdad total entre ambos conjuntos.

2. **N-grams:** Los n-grams son subsecuencias de **n** palabras, tomados de las cadenas de textos a comparar. La medida de similitud mediante n-grams, consiste en dividir el número de n-grams que comparten ambas cadenas, entre el número total de n-grams (Barrón-Cedeño, 2008). Se denota:

$$sim_{ngrams}(t_1, t_2) = \frac{|ngrams(t_1) \cap ngrams(t_2)|}{|ngrams(t_1) \cup ngrams(t_2)|} \quad (6)$$

3. **Similitud Coseno:** Se asume que los fragmentos de texto se encuentran representados en forma vectorial, entonces la similitud coseno se define como:

$$sim_c(t_1, t_2) = \frac{|t_1 \cap t_2|}{\sqrt{|t_1| \cdot |t_2|}} \quad (7)$$

Esta métrica toma valores entre -1 y 1. Cuanto más cercano a 1 sea el valor de la métrica, más similares son los textos.

La medida de similitud que se emplea en este trabajo es el coseno de similitud.

3.2.2.3. Atribución de autoría: enfoque supervisado

Se emplean técnicas de Aprendizaje Automático Supervisado para crear una función capaz de clasificar correctamente cualquier documento de entrada después de aprender una serie de ejemplos (mediante un conjunto de datos (dataset, en inglés) de entrenamiento). Para lograrlo, el modelo generado debe ser capaz de clasificar correctamente, documentos que no ha visto anteriormente (Moreno et al., 1994). Para resolver un problema de aprendizaje supervisado se tienen que considerar varios pasos (Álvarez Carmona, 2014):

1. **Determinar el tipo de ejemplos de entrenamiento.** Este paso es fundamental, hay que decidir qué tipo de datos se van a utilizar para entrenar un buen modelo de clasificación.
2. **Reunir un conjunto de entrenamiento.** Se debe crear cuidadosamente una selección de un conjunto de objetos de entrenamiento que se recopila junto con sus clasificaciones correspondientes.
3. **Determinar el modelo de clasificación para generar una función aprendida que determinará las clases de los elementos de prueba.** La precisión de la función aprendida depende en gran medida de cómo el objeto de entrada está representado. Normalmente, el objeto de entrada se transforma en un vector que contiene una serie de características que son descriptivas del objeto. El número de características debe ser lo suficientemente grande como para predecir con precisión la salida.

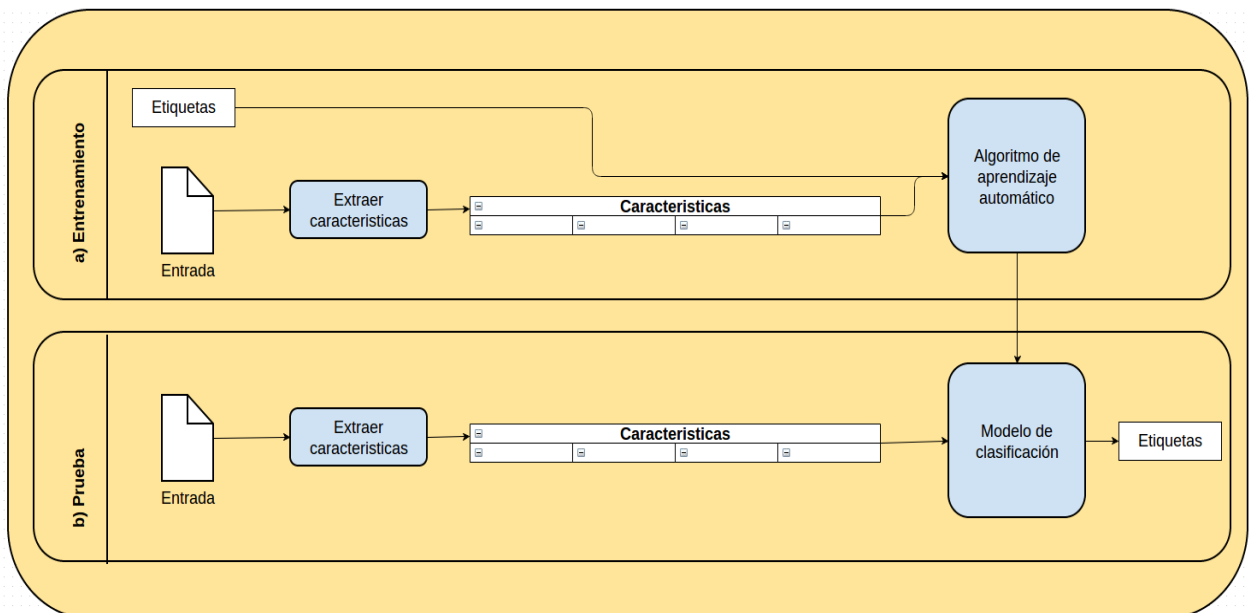


Figura 1. Representación del Aprendizaje Supervisado

En la figura 1, se observa que Aprendizaje Supervisado consta de dos partes importantes (Moreno et al., 1994): 1) Fase de Entrenamiento, y 2) Fase de Prueba. En la fase de entrenamiento a partir de una determinada entrada y un conjunto de etiquetas (que también se les conoce como clases) se extraen las características de cada objeto en la colección de datos; posteriormente estas características con sus respectivas etiquetas son los parámetros que utiliza un algoritmo de aprendizaje automático cuyo resultado es un modelo de clasificación. El modelo creado es usado para clasificar nuevos datos a los cuales se les extraen las mismas características que se extrajeron a los datos de entrenamiento. Finalmente, el modelo obtendrá el resultado de la clasificación.

Los métodos de clasificación más populares en el estado del arte, son: máquinas de vectores de soporte (SVM), clasificador bayesiano (NB), vecinos más cercanos (kNN) y regresión logística (LR).

En el marco de investigación de este trabajo se comparan los resultados del clasificador SVM y vecinos más cercanos (kNN).

3.2.2.3.1. Máquinas de soporte vectorial (SVM, por sus siglas en inglés)

SVM, o bien, las llamadas Máquinas de Soporte Vectorial (Support Vector Machine), son métodos de clasificación capaces de trabajar en espacios de alta dimensionalidad.

Cuando se comenzó a utilizar estos algoritmos sólo resolvían problemas de clasificación binaria, y actualmente se utilizan para múltiples tipos de problemas, como por ejemplo la atribución de autoría (AA), y en general, la clasificación de documentos. Para entender la esencia intrínseca de estas técnicas, se deben tener en cuenta los siguientes supuestos: 1) se tienen dos dimensiones y, 2) se tienen dos categorías.

Sea un conjunto de datos como el que se puede observar en la figura (2). Estos datos están representados como un vector n -dimensional. El caso más simple del SVM, consiste en tratar de encontrar un hiperplano lineal que separe las dos categorías, aquí representadas como cuadrados y círculos. En este caso el hiperplano de separación es una recta y la dimensión $n = 2$.

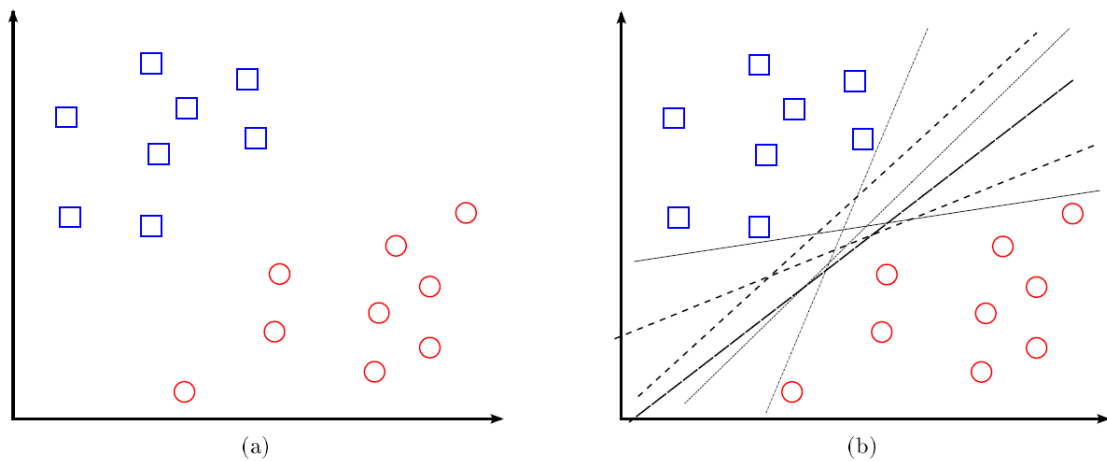


Figura 2: (a) Conjunto de datos del ejemplo. (b) Posibles hiperplanos de separación de las dos categorías.

Existe un número infinito de posibles hiperplanos (líneas) que realicen la clasificación pero, ¿cuál es la mejor y cómo la definimos?. Para responder esta cuestión, lo que se busca es seleccionar un hiperplano de separación óptima, es decir, aquella solución que permita un margen lo más amplio posible entre las categorías, debido a esto, también se denomina clasificador de máximo margen. El hiperplano que cumple esa condición es el equidistante a los ejemplos más cercanos de cada clase. Estos puntos determinan el margen y el hiperplano, y se denominan vectores de soporte. En la figura 3, son los que tienen el color compacto.

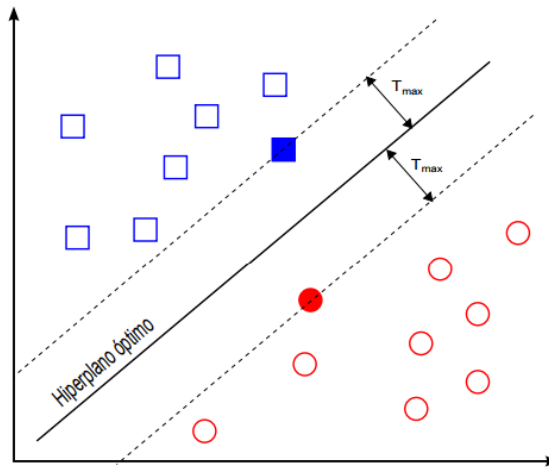


Figura 3: Imagen donde se encuentra el hiperplano óptimo y los vectores de soporte para el ejemplo dado.

En este caso, se clasifica como cuadrado cualquier documento d_i que estuviese por encima del hiperplano, y como círculo, cualquiera que estuviese por debajo.

Usualmente, los problemas que desean resolver (clasificar) no son tan simples como se ilustra en los ejemplos anteriores. Existen casos en los que las dimensiones y las categorías son más de dos y donde los conjuntos de datos no son separables por un hiperplano lineal, como es el caso de la figura 4.

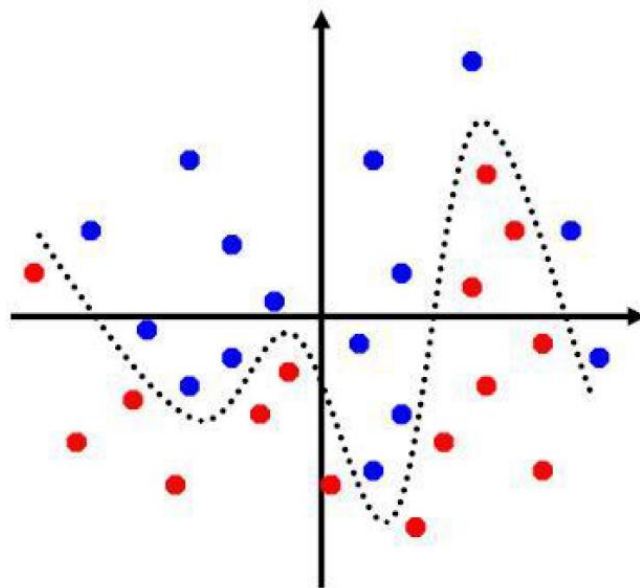


Figura 4: Ejemplo en el que no existe ningún hiperplano lineal que divida las dos categorías

Cuando las categorías no son linealmente separables, se busca un clasificador de “margen débil”, el cual permite que algunos puntos no estén en su lado del hiperplano, es decir, es más flexible. Para lograr esto, se tiene una restricción de que el total de las distancias de los errores de entrenamiento son menores con una constante $C > 0$. Cabe destacar, que ciertos problemas linealmente separables pueden ser resueltos con este clasificador para obtener cierta flexibilidad y evitar el overfitting (sobreajuste).

Si los datos no son linealmente separables en el espacio original, se puede realizar una

transformación de estos mediante una función núcleo (*kernel*).

Definición 1 (Función *kernel*). Sea X el espacio de entrada, H el de características dotado de un producto interno y una función $F: X \rightarrow H$, con H el espacio inducido de Hilbert, se define la función núcleo como $K: X \times X \rightarrow \mathbb{R}$ como:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Estas funciones proyectan la información a un espacio de características de mayor dimensión, lo que aumenta la capacidad de clasificación de SVM lineal. Los núcleos más utilizados son el lineal, el polinómico, el radial y el sigmoideal.

- Lineal: $K(x_i, x_j) = x_i \cdot x_j$
- Polinómico: $K(x_i, x_j) = (x_i' \cdot x_j + c)^d$, siendo d el grado del polinomio y c un parámetro.
- Radial: $K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$, siendo σ un parámetro.
- Sigmoideal: $K(x_i, x_j) = \tanh(\gamma \cdot x_i' \cdot x_j + \delta)$, siendo γ y δ parámetros.

La búsqueda de un hiperplano óptimo es un problema de optimización cuadrática con restricciones lineales que se resuelve utilizando técnicas estándar. La propiedad de convexidad que se exige garantiza una única solución. Una vez que se obtiene el hiperplano, se puede clasificar cualquier nuevo documento comprobando en qué lado de los vectores se encuentra, y así, estimar la categoría a la que pertenece.

3.2.2.3.2. Los k vecinos más próximos (k-nearest neighbour classification, kNN)

El algoritmo del vecino más próximo (Nearest Neighbour, NN) calcula la similitud entre el documento que se desea clasificar y todos los documentos que pertenecen al conjunto de datos de entrenamiento. Una vez localizado el documento de entrenamiento más similar, se le asigna la misma categoría a este nuevo documento.

Una de las variantes más utilizadas de este algoritmo es la de los k vecinos más próximos (k-nearest neighbour, kNN). En este caso, se toman los k documentos más parecidos y se estudian las categorías a las que pertenecen. La categoría que tenga más representantes será asignada al nuevo documento.

Este método es de simple aplicación y resulta muy eficaz, especialmente cuando el número de categorías posibles es alto y los documentos son heterogéneos. El principio básico de funcionamiento es el siguiente: cada muestra del conjunto de test se compara con un número de muestras de entrenamiento preclasificados, y se evalúa su semejanza de acuerdo con una medida de similitud, en este caso, el coseno de similitud, con el fin de encontrar la clase de salida asociada. El parámetro k permite especificar el número de vecinos, es decir, la formación de muestras a tomar en cuenta para la clasificación. En este trabajo nos centramos en tres modelos kNN con k igual a 3.

3.2.2.4. Estimación del error

En problemas reales, el conjunto de datos es de tamaño limitado; entonces se debe buscar una estrategia que permita dividir el dataset en 3 conjuntos. Un subconjunto de datos de entrenamiento que permitan construir el modelo; un subconjunto de datos de validación que se utiliza para "validar" el modelo, es decir, prevenir el overfitting y el underfitting; y por último, un subconjunto de datos de prueba sobre el cual se evalúa el modelo. Usualmente, se reporta la precisión del modelo en base al test set.

Existen diferentes técnicas que permiten estimar el error de clasificación; este trabajo utiliza el

método *Hold-out* que obtiene una buena estimación del error a un bajo costo computacional. Entonces, sea \mathcal{D} un conjunto de datos con N elementos, se describen:

3.2.2.4.1. Validación simple (Hold-Out)

El método Hold-out es el más sencillo de los métodos de validación, debido a su bajo costo computacional. Este separa el conjunto de datos (documentos) disponibles \mathcal{D} , en dos subconjuntos, uno utilizado para entrenar el modelo y otro para realizar el test de validación (Arlot y Celisse, 2010). De esta manera se crea un modelo únicamente con los datos de entrenamiento. Con el modelo creado se generan datos de salida que se comparan con el conjunto de datos reservados para realizar la validación, es decir, aquellos que no han sido utilizados en el entrenamiento, por lo que no han sido utilizados para generar el modelo (Hawkins et al., 2003). Los estadísticos obtenidos con los datos del subconjunto de validación son los que dan validez a un método dado, en términos de error. La precisión del modelo está dada por la relación entre la cantidad de clasificaciones correctas obtenidas durante la prueba y, el número total de ejemplos del subconjunto de prueba. Para obtener una estimación más confiable, se debe repetir el proceso hold-out, tomando distintos conjuntos de datos de entrenamiento (aleatorio), un determinado número de veces.

3.2.2.4.2. Validación cruzada (K-fold crossvalidation)

Este método, k-fold, basado en el método hold-out, tiene mayor utilidad cuando el conjunto de datos es pequeño (Yang y Huang, 2014). En este caso, el total de los datos se divide en k subconjuntos, de manera que se aplica el método hold-out k veces, utilizando cada vez un subconjunto distinto para validar el modelo entrenado con los otros $k - 1$ subconjuntos (Jung y Hu, 2015), es decir, en cada iteración uno de los subconjuntos se deja para prueba. El error medio obtenido de los k análisis realizados nos proporciona el error cometido por el método, permitiendo así evaluar su validez. Si este proceso se repite n veces, se construyen y evalúan $k * n$ clasificadores, pero ello se traduce en un gran costo computacional.

Si se comparan los dos métodos, hold-out y k-fold, el método k-fold tiene la ventaja de que todos los datos son utilizados para entrenar y validar, por lo que se obtienen resultados más representativos a priori. Mientras que, para el método hold-out, se realiza el proceso n veces de manera aleatoria, lo que no garantiza que los casos de entrenamiento y validación no se repitan.

3.3. Dataset utilizado

El PAN¹ es un foro importante donde se discuten los avances en la detección de plagio y la atribución de autoría. En el mismo, se crean corpus sobre estos temas y se les pide a los participantes que desarrollen nuevas metodologías para resolverlos. En la edición del PAN 2012, se presentaron tres corpus balanceados para la tarea de atribución de autoría. Los mismos se denominan Problema A, Problema C y Problema I. Los corpus se conformaron con fragmentos de novelas escritas en lengua inglesa. En esta investigación se utiliza el dataset suministrado por el Problema C (se eligió resolver dicha propuesta); del mismo se conocen ocho autores \mathcal{A}_i con $i = 1, \dots, 8$ y los datos de entrenamiento se componen de dos ejemplos para cada uno de ellos, mientras que los datos de prueba constan de un ejemplo de cada autor para ser clasificado (ocho ejemplos en total). La longitud de los textos llega a 13000 como máximo.

3.4. Métricas de Evaluación

Evaluar la performance de un modelo de Atribución de Autoría requiere de un conjunto de datos etiquetados, es decir, cada documento asociado a su autor correspondiente (resultado correcto). De esta forma el modelo obtenido clasifica el dataset y sus resultados son comparados con los reales. Se puede pensar que un modelo es mejor que otro, si su cantidad de aciertos es mayor; pero no es la única métrica que nos define la calidad de clasificación (Silva, 2018). Las métricas

¹ PAN: es una serie de eventos científicos y tareas compartidas sobre texto forense digital y estilometría.

usadas en la competencia del PAN, son:

- **Precisión:** Expresa la cantidad de falsos positivos, ya que se calcula para una categoría dada, como la división entre la cantidad de observaciones que fueron correctamente clasificadas sobre el total de las que fueron clasificadas bajo dicha clase.

$$precision_{ci} = \frac{clasificados_{ci} \cap etiquetados_{ci}}{etiquetados_{ci}} \quad (6)$$

- **Recall o Cobertura:** A diferencia de la precisión, recall es el nombre de la métrica que refleja falsos negativos. Es decir, para una clase dada, se divide la cantidad de observaciones que han sido clasificadas correctamente bajo esta categoría, sobre la cardinalidad real de la clase.

$$recall_{ci} = \frac{clasificados_{ci} \cap etiquetados_{ci}}{clasificados_{ci}} \quad (7)$$

- **F₁ Score:** El desbalance que puede existir entre precisión y recall en una tarea de clasificación, hace que el clasificador no sea efectivo. Entonces para solucionar este problema, se combinan ambas métricas, lo que da origen a la ecuación de la Media Harmónica, que se expresa como sigue:

$$F_1score = 2 \cdot \frac{1}{\frac{1}{recall} + \frac{1}{precision}} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (8)$$

3.5. Experimentación

El entrenamiento y evaluación de los clasificadores propuestos se realizó a partir, del Dataset del problema C, suministrado por el PAN 2012. Para validar los modelos se agregaron 4 documentos confeccionados en forma manual. Los clasificadores se enfrentan a un problema de clasificación multiclase.

Este trabajo tiene la finalidad de comparar los desempeños de cada clasificador propuesto, en este caso, SVM y el de los k-vecinos más cercanos. En esta sección, se muestran los resultados del método supervisado (clasificación). Se utiliza 3-grams de caracteres, sin muestreo, y utilizando los valores que se consideran adecuados para cada clasificador (Eder et al., 2016).

Tabla 5. Precisión de los clasificadores (3-grams de 'c')

Clasificador	SVM	3-NN
150 MFW	0.80	0.89
200 MFW	0.50	0.89
250 MFW	0.6875	0.89

Tabla 6. Recall de los clasificadores (3-grams de 'c')

Clasificador	SVM	3-NN
150 MFW	0.50	0.75
200 MFW	0.50	0.75
250 MFW	0.50	0.75

Tabla 7. F-Measure de los clasificadores (3-grams de 'c')

Clasificador	SVM	3-NN
150 MFW	0.62	0.81
200 MFW	0.50	0.81
250 MFW	0.58	0.81

Se observa una mejor performance promedio en el clasificador kNN. La baja performance de SVM se debe a dos razones: la baja cantidad de datos de entrenamiento y la corta longitud de los textos. Se piensa que el tamaño de los textos representa una limitación en estos experimentos.

Por lo tanto, estos experimentos preliminares indican que, en textos simples y cortos, el clasificador de los k vecinos más cercanos, a pesar de ser particularmente sencillo, es efectivo.

4. Conclusión

Con este trabajo se presentan los resultados del estudio realizado sobre algoritmos de clasificación utilizados en la atribución de autoría, más precisamente, SVM y kNN.

Los métodos utilizados han sido configurados de tal manera que sean simples, para reducir la complejidad, ya que planeamos implementarlos en el marco de la Facultad de Ingeniería de la UNJu.

En este trabajo, se han estudiado dos métodos para la tarea de atribución de autoría, en textos cortos. Los métodos estudiados son capaces de modelar el estilo de escritura particular de cada documento de un determinado autor (AA, basada en instancias). Se utilizó el paquete stylo, desarrollado para el Lenguaje de Programación R, que provee diferentes técnicas de aprendizaje automático y minería de texto, para el análisis y clasificación de textos. Estas técnicas se han analizado y ajustado, con el fin de obtener un método con buen rendimiento para la tarea de atribución de autoría. Se demostró la superioridad del algoritmo kNN (con $k=3$), que logró un F_{measure} mayor que el de SVM. Esto último confirma lo descrito en la sección 3.2.2.3.2, debido a que el número de candidatos (clases) es considerable.

Para proponer métodos generalizados que podrían usarse con todo tipo de textos, la validación del modelo es un problema clave. En nuestro trabajo futuro, estudiaremos este problema en detalle.

5. Referencias

1. Álvarez Carmona, M. A. (2014). "Detección de similitud en textos cortos considerando traslape, orden y relación semántica de palabras" (tesis de maestría). Instituto Nacional de Astrofísica, Óptica y Electrónica, México. Obtenido de: <https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/168/1/AlvarezCMA.pdf>.
2. Argamon, S., y Juola, P. (2011). "Overview of the international authorship identification competition at pan-2011". Notebook for PAN at CLEF.
3. Arlot, S. y Celisse, A. (2010). "A Survey of Cross-validation Procedures for Model Selection". Statistics Surveys, 4, 40-79. Obtenido de: <http://dx.doi.org/10.1214/09-SS054>
4. Barrón Cedeño, L. A. (2008). "Detección automática de plagio en texto" (tesis de maestría). Universidad Politécnica de Valencia, España. Obtenido de:

<https://riunet.upv.es/handle/10251/12186>.

5. Cagnina, L., Errecalde, M., Rosso, P. (2011). "Algoritmos bio-inspirados aplicados a tareas de clasificación de textos cortos". IV Jornadas TIMM (Temática en Tratamiento de Información Multilingüe y Multimodal). Obtenido de: http://timm.ujaen.es/wp-content/uploads/2014/03/timm2011_submission_8.pdf
6. Castro, D. (2019). "Verificación de autoría, modelos intrínsecos basados en semejanza" (tesis doctoral. Universidad de Alicante, España. Obtenido de: <http://hdl.handle.net/10045/91047>
7. Eder, M., Rybicki, J. and Kestemont, M. (2016). "Stylometry with R: a package for computational text analysis". R Journal 8(1): 107-121. Obtenido de: <https://journal.r-project.org/archive/2016/RJ-2016-007/index.html>
8. Escalante, H. J., Solorio, T., Montes y Gómez, M. (2011) "Local histograms of character n-grams for authorship attribution". In ACL, pages 288-298. The Association for Computer Linguistics. Portland, Oregon. Obtenido de: <https://www.aclweb.org/anthology/P11-1030>
9. Funez, D. G., Errecalde, M.L. (2012). "Detección de plagio intrínseco basada en Histogramas". XVIII Congreso Argentino de Ciencias de la Computación. San Luis, Argentina. Obtenido de: <http://sedici.unlp.edu.ar/handle/10915/23743>
10. Koppel, M., y Schler, J. (2004). "Authorship verification as a one-class classification problem". Proceedings of the 21st International Conference on Machine Learning, 7p.
11. López Monroy, A. P. (2012). "Atribución de Autoría utilizando distintos tipos de características a través de una nueva representación" (tesis de maestría). INAOE, Puebla, México. Obtenido de: <http://ccc.inaoep.mx/~pastor/resources/papers/tesis-master.pdf>
12. Moreno, A., Armengol, E., Bejar, J., Belanche, Ll., Cortés, U., Gavaldá, R., Gimeno, J. M., López, B., Martín, M., Sánchez, M. (1994). "Aprendizaje Automático". Universitat Politècnica de Catalunya, España. ISBN: 84-7653-460-4. Obtenido de: <https://upcommons.upc.edu/bitstream/handle/2099.3/36157/9788483019962.pdf>
13. Pérez Afonso, J. (2013). "Detección Intrínseca de Plagio" (tesis de maestría). Universidad Politécnica de Valencia, España. Obtenido de: <https://riunet.upv.es/handle/10251/43831>
14. Raschka, S. (2014). "Naive Bayes and Text Classification I. Introduction and Theory". ArXiv Preprint arXiv:1410.5329. Obtenido de: <https://arxiv.org/abs/1410.5329>
15. Salinas, J. y Izetta, J. (2016). "Clasificación automática de textos periodísticos usando Random Forest". CACIC, Universidad Nacional de San Luis, Argentina. Obtenido de: <http://sedici.unlp.edu.ar/handle/10915/55733>
16. Silva, M. G. (2018). "Predicción de Tendencias en Redes Sociales basada en características sociales y contenido" (tesis de grado). Universidad Nacional de Córdoba, Argentina. Obtenido de: https://rdu.unc.edu.ar/bitstream/handle/11086/6245/Tesis-Final_Silva.pdf?sequence=1&isAllowed=y
17. Stamatatos, E. (2009). "A survey on modern authorship attribution methods". Journal of the American Society for Information Science and Technology, 60, 538-556. Obtenido de: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.440.1634&rep=rep1&type=pdf>

18. Tversky, A. (1977). "Features of Similarity". Psychological review, 84(4):327–352, 1977.
Obtenido de: <http://dx.doi.org/10.1037/0033-295X.84.4.327>