Detección de plagio intrínseco usando la segmentación de texto

Dario G. Funez and Marcelo L. Errecalde

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)
Facultad de Cs. Físico, Matemáticas y Naturales, Universidad Nacional de San Luis
Tel: (02652) 420823 / Fax: (02652) 430224
{dgfunez,merreca}@unsl.edu.ar

Resumen La detección de plagio intrínseco utiliza la estilografía para delimitar secciones de un documento, que se sospecha que fueron escritas por un autor diferente. En este trabajo, se analiza si la *segmentación de texto* es una opción viable como estrategia de descomposición de texto y se implementa un algoritmo simple para la detección de outliers, una de las componentes básicas en este tipo de tareas. Se provee además de un ambiente integrado de ejecución que permite ingresar un texto en inglés, y muestra las secciones de ese texto que exiben un estilo de escritura distinto al autor del texto. El algoritmo de detección fue testeado en el corpus de la Competencia de Detección de Plagio *Pan 2009*, donde se obtienen resultados comparables a los obtenidos con otros algoritmos representativos del estado del arte en el área.

Palabras Claves: detección de plagio intrínseco, estilografía, segmentación de texto, índices de legibilidad.

1. Introducción

Cuando se tiene un texto cuya autoría, una persona se atribuye sin serle propia, estamos en presencia de *plagio* [9]. El plagio está presente en distintas áreas como la música, la política, el procesamiento de imágenes y la literatura [2]. También es un problema que deben enfrentar las empresas ya que la competencia puede plagiar su inovación [1]. Esta actividad, se ve favorecida por la gran cantidad de información disponible hoy en día, y a que la Web ha facilitado su acceso suministrando buscadores, enciclopedias online (Wikipedia), sitios con monografías, etc. En muchos casos, un autor comete plagio de texto por desconocer cómo se debe realizar correctamente la cita y su paráfrasis. En el caso que un escritor utilice una frase exacta, debe colocarla entre comillas y cuando se parafrasea un texto, se debe incorporar la fuente con su cita bibliográfica.

La detección de plagio no es una tarea sencilla, debido a que un texto se puede reescribir de forma tal que es muy dificultoso encontrar similitudes ente ellos. La detección de plagio puede ser caracterizada como *extrínseca* e *intrínseca*. La detección extrínseca usa una colección de documentos como referencia y tiene

como desventaja su gran costo computacional, ya que requiere buscar el plagio en cada uno de los archivos de dicha colección. La detección que analiza un texto sin tener en cuenta un potencial archivo fuente, se denomina *intrínseca* y es el tipo de detección en que nos centraremos en este trabajo. Este tipo de detección no detecta plagio en el sentido estricto de la palabra, es decir, no suministra el documento de donde se extrajo la información y solamente provee los pasajes de texto que se sospecha de plagio. La simple lectura de un texto, nos puede indicar sobre la presencia de distintos estilos de escritura, y por lo tanto que otros autores constribuyeron en el texto sin mencionarlos.

Para la detección intrínseca, en [9] se sugiere seguir las siguientes etapas en secuencia: 1) selección de una estrategia de *descomposición* del texto, 2) definición del *modelo estilográfico* que comprende la definición de las medidas estilográficas que serán recuperadas del texto y 3) la *identificación de los outliers* que discrimina las secciones plagiadas de las no plagiadas. Con respecto a la manera en que se va a descomponer el texto, en [9] se considera que la detección de límites por tópicos o *segmentación de texto*, puede dar buenos resultados pero destacan la dificultad de la misma. La segmentación de texto es utilizada en el procesamiento del lenguaje natural, para textos que no tienen límites estructurales como capítulos, párrafos o secciones [7]. La esencia de este algoritmo es que detecta cambios de vocabulario, los cuales son muy importantes para la detección de plagio. Un cambio abrupto de vocabulario, nos informa sobre un posible estilo de escritura distinto. De acuerdo a nuestro conocimiento, no se han realizado hasta el momento experiencias concretas con este tipo de técnicas en la detección de plagio intrínseco.

Nuestra contribución en este trabajo se enfoca en este último aspecto, realizando una primera experiencia en el uso de una estrategia de segmentación particular propuesta por Freddy Choi [4] para la etapa de descomposición de texto. Además, se provee un ambiente de ejecución amigable que soporta todas las etapas involucradas en la detección de plagio intrínseco con este enfoque. Los resultados preliminares obtenidos con el corpus Pan09 muestran que este enfoque es una alternativa competitiva, con respecto a otros algoritmos representativos del estado del arte en el área.

El resto del trabajo se organiza de la siguiente manera. En la sección 2 se describe la detección intrínseca de plagio. La sección 3 explica el algoritmo de segmentación de texto empleado en este trabajo. En la sección 4 se detalla la implementación de los módulos que componen el ambiente. En la sección 5 se describen los experimentos realizados y el análisis de los resultados. Finalmente, en la sección 6 se ofrecen las conclusiones y el trabajo futuro.

2. Detección de Plagio Intrínseco

Se comete plagio cuando una persona se atribuye como propio un trabajo o idea sin reconocer la propiedad intelectual del autor legítimo [3]. El plagio de texto ocurre cuando un texto utiliza secciones de texto de otro autor, sin proporcionar la fuente [2]. Un caso muy frecuente es el uso del *cut-and-paste* que

incluye un fragmento sin modificaciones, siendo este tipo de plagio el más fácil de detectar. La tarea de detección se dificulta cuando se oculta un texto, de tal manera que es difícil encontrar similitud entre ellos. Los trucos más usados son utilizar sinónimos, cambiar el orden de las oraciones, adicionar palabras, etc.

Para poder afirmar que un texto ha incurrido en plagio, se debe proporcionar la sección plagiada y la sección del documento fuente. El detector debe realizar comparaciones del texto sospechoso con todos los posibles documentos. Como este conjunto puede ser impráctico de procesar y existe información que no está disponible en formato digital, es que existen métodos de detección que solamente necesitan del texto a chequear, pero no exhiben la prueba del plagio. Este método, denominado *intrínseco*, analiza un texto t de un único autor y devuelve las secciones de t redactadas por otros autores [9]. El fundamento del análisis intrínceso es que un escritor mantiene su estilo de escritura en todo el texto. Para tal fin se realiza un análisis de estilo, que extrae información que no es tan evidente en el texto [3]. El análisis de estilo se basa en el hecho que cada autor tiene un estilo de escritura personal, que no sufre modificaciones en un texto de su autoría y que es difícil de describir, pero puede ser representado por medidas estilográficas que se expondrán más adelante.

En el análisis intrínseco se cuenta con la información de estilo de autor del texto, es decir, un fragmento de texto se lo puede clasificar como perteneciente a él o no. Este es un problema de clasificación de una única clase, En estos problemas, se cuenta con una clase objetivo y todos aquellos objetos que no pertenecen a la clase objetivo son denominados *outliers* [11].

Para la detección intrínseca, se necesitan definir los siguientes componentes del proceso de verificación de plagio [9]: 1) la *estrategia de descomposición*, 2) la construcción del *modelo de estilo* y 3) la identificación de *outliers*. Cada una de ellas, se describe brevemente en las siguientes subsecciones.

2.1. Estrategia de descomposición

La estrategia más sencilla y rápida es dividir el documento en *secciones* de igual longitud. En [5] se elige un tamaño de entre 40 y 200 palabras para delimitar una sección. Una alternativa, es descomponer el texto empleando algún límite estructural como los capítulos, secciones, párrafos, etc. Estos pueden ser complementados utilizando como limitadores objetos del texto como tablas, notas al pie, referencias bibliográficas etc. Otra posibilidad es dividir el texto por tópicos, utilizando algún algoritmo de segmentación de texto. Esta última alternativa, utilizada en el presente trabajo, se describe con mayor detalle en la sección 3.

2.2. Construcción del modelo de estilo

La *estilografía* es el estudio estadístico de un estilo de escritura [9]. Un escritor utiliza el mismo patrón para la redacción de oraciones y tiene un vocabulario específico que depende de su formación personal y esto se puede caracterizar con

un *modelo*. Existen distintas medidas de características lingüisticas para modelar un estilo de escritura que corresponden a alguna de la siguientes categorías [6]:

- *Estadísticas del texto*: Se basan en características léxicas, a nivel de palabra y/o caracter. Calculan por ejemplo, las frecuencias de caracteres especiales, como símbolos de puntuación y palabras en el texto. Las siguientes medidas corresponden a esta categoría:
  - Promedio de la clase de frecuencia de las palabras: Las palabras son agrupadas por *clases*, determinadas por su frecuencia de aparición en un texto (función f), determinándose la clase de una palabra de acuerdo a la siguiente fórmula [6]: $c(\ ) = \lfloor \log_2(f(\ )/f(\ ))\rfloor$, donde es la palabra de mayor frecuencia y $\lfloor\ \rfloor$ denotan la función piso. El promedio es calculado teniendo en cuenta la cantidad de las palabras de cada clase. Esta medida contiene información sobre la complejidad y la riqueza del vocabulario y se ha demostrado que es robusta y no depende del tamaño del texto para el cual se aplica.
  - Función R: Esta medida se relaciona directamente con el vocabulario del escritor y se define como [2]: $R = (100 \times \log(CP\ ))/CP\ ^2$, donde CP es el total de palabras en el texto.
  - Función K: Tiene el mismo objetivo que la función R, asumiendo en este caso una distribución Poisson en la frecuencia de las palabras, utilizando en este caso la fórmula [1]: $K = 10^4 \times (\sum_{i=1}^{\infty} {}^2v_i - CP\ )/CP\ ^2$, donde v_i denota la cantidad de palabras con frecuencia .
- *Características sintáticas*: Estas medidas capturan estilos de escritura a nivel de *sentencias* como por ejemplo el promedio de oraciones cortas o el promedio de oraciones que comienzan con pronombres interrogativo (por ejemplo *what*). Otro ejemplo es el uso del promedio de sentencias en *voz pasiva*.
- *Características "Part-of-Speech"*: Comprende la frecuencia de clases de palabras como adjetivos, pronombres, sustantivos, adverbios etc.
- *Características estructurales*: Contienen información a nivel de la *organización del texto* (usos de saludos, longitud de párrafos y/o capítulos, etc.).
- *Indices de legibilidad*: Estiman el grado de comprensibilidad requerido para el entendimiento de un texto. La definición de estos índices se basan en distintas medidas calculadas en un fragmento del texto: COr , la cantidad de oraciones, CS , el total de sílabas y $CL\ n$ es el total de números y letras. Algunos de los índices más empleados son:
  - Indice de Flesh: Utilizado en textos en general, toma todo el rango de valores entre 0 y 100. Los valores más bajos significan que el texto es más difícil de comprender. La fórmula de Flesh se expresa como: $_F = 206,\ 3\ - (1{,}01\ \times CP\ )/COr\ - (\ 4{,}6 \times CS\ )/CP\ .$
  - Indice de Flesh-Kincaid: Este índice, modificación del anterior, es comúnmente aplicado a textos técnicos e indica un grado que estima la cantidad de años (edad) requerida para entender el texto. Su fórmula es: $= (0{,}3\ \times CP\ )/COr\ + (11,\ \times CS\ )/CP\ - 1\ ,$
  - Índice de Coleman-Liau: Toma valores entre 0,4 y 16,3 y se calcula como: $_C = (\ ,\ \ \times CL\ n)/CP\ - (30{,}0 \times COr\ )/CP\ - 1\ ,$

2.3. Identificación de outliers

En las fases anteriores, se obtiene un texto divido en n secciones $_1 \dots _n$ sobre el que luego se computa, en cada una de ellas, el modelo definido y se produce como salida n vectores de características, cuya dimensión depende de la cantidad de medidas utilizadas [9]. La única información con que se cuenta es el estilo de escritura del escritor en cuestión, se conoce sólo esta clase, planteándose de esta manera un problema de clasificación de *una clase*. En estos casos se caracteriza los elementos objetivos, de tal manera que se pueda distinguir si un nuevo elemento pertenece o no a dicha clase objetivo. La *detección de outliers* es un problema con estas características. La tarea en este caso es detectar aquellos objetos que no se asemejan a un conjunto de objetos predeterminados, es decir detectar los *outliers* [11]. En el contexto de nuestro trabajo, serán considerados como outlier aquellas secciones cuyos vectores presenten una alta variabilidad en las medidas respecto a la clase objetivo. Entre los métodos más difundidos para resolver problemas con estas características, podemos mencionar:

- *Métodos de densidad*: Aproximan la función de densidad de probabilidad de la clase objetivo. Se considera a los outliers uniformente distribuidos y la regla de Bayes se puede utilizar para diferenciar objetos outliers de los objetivos. Proveen buenos resultados con tamaños de muestra grandes.
- *Métodos de límite*: Tratan de delimitar una región utilizando distancias entre los elementos objetivos. Los outliers son aquellos objetos que no están comprendidos en esa región.
- *Métodos de Reconstrucción*: Necesitan conocimiento previo sobre la generación de los elementos objetivos. Los outlier son aquellos objetos que son difícil de reconstruir.
- *Método basado en la MEDA*: La MEDA se define como la mediana del conjunto de las diferencias absolutas entre cada medida $_i$ y su media $^-$, en donde el n es la cantidad de medidas y se expresa de la siguiente manera: $M \quad A = med \quad n \; [\quad (\ _1 - ^-), ..., \quad (\ _n - ^-)]$. Se conoce que el 50 % de las medidas están contenidas en el siguiente intervalo: $[^- - M \quad A, ^- + M \quad A]$. Una medida $_i$ se puede clasificar como outlier si $_i \notin [^- - 4, \ * M \quad A, ^- + 4, \ * M \quad A]$. Una sección se la clasifica como outlier si contiene un número mínimo de medidas outliers.

En la siguiente subsección, se definen las medidas de evaluación generalmente empleadas para cuantificar la performance de un detector de plagio.

2.4. Medidas de evaluación

Para evaluar el comportamiento de un algoritmo de detección de plagio, se deben computar la *precisión* (en inglés *precision*), la *cobertura* (*recall*) y la *granularidad* de las detecciones realizadas. En la definición de estas medidas seguiremos las siguientes convenciones de notación: 1) representa una sección plagiada del conjunto S de todas las secciones plagiadas, 2) r denota una sección detectada del conjunto R de detecciones, 3) S_R son las secciones plagiadas que han sido

detectadas, 4) $|\ _i\ |$ y $|\ r_i\ |$ denotan el tamaño (en cantidad de caracteres) de la sección correspondiente y $|\ S\ |$ y $|\ R\ |$ denotan la cardinalidad de los conjuntos respectivos [1]. Finalmente, $\alpha(\ _i)$ es la cantidad de caracteres detectados de $\ _i$, $\beta(r_i)$ es la cantidad de caracteres plagiados de r_i y $\gamma(\ _i)$ es la cantidad de caracteres plagiados detectados de $\ _i$. En base a estos valores, la precisión, cobertura, granularidad y evaluación global (*overall*), se definen de la siguiente manera[2]: $rec\quad = 1/\ |\ S\ |\ \sum_{i=1}^{|S|} \alpha(\ _i)/\ |\ \ _i\ |$, $prec\quad on = 1/\ |\ R\ |\ \sum_{i=1}^{|R|} \beta(r_i)/\ |\ r_i\ |$, $gr\ \ nu\quad r\ d\ \ d = 1/\ |\ S_R\ |\ \sum_{i=1}^{|S_R|} \gamma(\ _i)$, y $over\quad = F/(\log_2(1 + gr\ \ nu\quad r\ d\ \ d))$

Estas medidas se interpretan de la siguiente manera. La precisión cuantifica el porcentaje de detecciones correctas, el recall el porcentaje de plagio detectado, una granularidad cercana a 1 significa que el algoritmo detectará cada plagio a lo sumo una vez. En todos los casos, valores cercanos a 1 indican que el algoritmo de detección tiene buena performance.

3. Segmentación de texto

La segmentación de texto divide un texto en unidades con el mismo tópico [4]. La implementación de Freddy Choi es realizada en dos fases sobre el texto completo. En la primer etapa las *stops words* (artículos, preposiciones, conectores etc.) son removidas ya que no aportan información relevante del texto. La raíz de cada palabra se obtiene mediante un algoritmo de *stemming* y se almacena su frecuencia en el texto en un vector. Cada oración tiene asociada un vector y la frecuencia de la palabra en la oración se denota como $f_{i,j}$.

La matriz S resultante de aplicar la similitud coseno a cada par de vectores es llamada *matriz de similitud* [4]. Dado que no es sencillo determinar los límites de los segmentos directamente sobre S, esta matriz es sometida a un proceso de *ranking* que obtiene una nueva matriz S' a partir de S, denominada *matriz de rango*. Cada elemento (valor) de la matriz S' resulta de desplazar una *máscara* (matriz cuadrada) sobre S. Cada valor r en S' se determina en base al conjunto de valores que cubre la máscara en S ( ) y al valor central de la máscara en S (c). La fórmula para obtener r es: $r = \ v\ \ ue/|\ \ |$, donde $v\ \ ue$ es el número de elementos en con menor similitud que c.

La última etapa del algoritmo de segmentación, utiliza los valores obtenidos en S' y aplica un método de clustering divisivo, basado en el algoritmo de maximización de Reynar [8] para detectar los límites de los segmentos. Este algoritmo se basa en el concepto de *densidad interna* donde, dado un segmento delimitado por las sentencias y (inclusive); si $\ _{i,j}$ es la suma de los valores rango de las sentencias en el segmento y $\ _{i,j}$ es el área interna que abarca el segmento dada por la fórmula: $\ _{i,j} = (\ -\ + 1)^2$. Si $B = \ _1 \ldots\ _m$ es una lista de m segmentos coherentes y y denotan la suma de valores rango y área respectivamente,

[1] Información obtenida de: http://www.uni-weimar.de/medien/webis/research/workshopseries/pan-10/task1-plagiarism-detection.html.

[2] En la evaluación global, refiere a la tradicional medida , la media harmónica de precisión y recall: $2 \times (\ e\ i\ io\ \times\ e\ \ )/(\ e\ i\ io\ +\ e\ \ )$

correspondiente al segmento en B, la densidad interna de B se define como: $= \sum_{i=1}^{m} / \sum_{i=1}^{m}$.

El proceso comienza inicializando B con un único segmento que representa todo el documento. Cada paso del algoritmo separa uno de los segmentos en B y el punto de corte se elige de tal manera que maximiza . La cantidad de segmentos m se determina de forma automática y queda establecida cuando el gradiente tiene variaciones inusuales. Si $^{(n)}$, es la densidad interna de n segmentos, el gradiente se define como: $\delta\ ^{(n)} = \ ^{(n)} - \ ^{(n-1)}$.

Para un documento con límites potenciales, si u,v denotan la media y varianza de $\delta\ ^{(n)}$ con $n \in 2, \ldots, \ + 1$, el m queda definido al aplicar el threshold $u + \sqrt{v}$ a δd. A menudo, un valor de $= 1,$ es utilizado en la práctica.[3]

4. Arquitectura del ambiente

El ambiente está compuesto por los módulos que se muestran en la Figura 1:

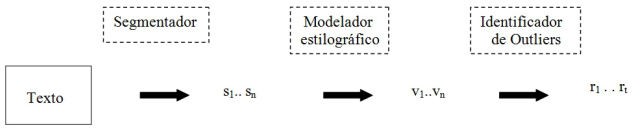

Figura 1. Módulos que conforman el ambiente

- *Segmentador*: Este módulo descompone el texto de entrada en una secuencia de segmentos cohesivos $_i$, donde cada $_i$ puede tener distinta cantidad de oraciones y constituye una unidad indivisible. Utiliza la segmentación de texto explicada en la sección 3 e implementada por la libreria *Morphadorner*.[4] La salida del segmentador es una secuencia de secciones $_1 \ldots \ _n$.
- *Modelador estilográfico*: Este módulo recibe como entrada la secuencia de secciones de texto del segmentador y, para cada sección, se obtienen las medidas estilográficas que componen el modelo. El modelo, en este caso incluye las siguientes medidas: promedio de sentencias pasivas, índices de legibilidad (Kincaid, Flesh y Coleman-Liau), promedio de clase de palabras, riqueza de vocabulario (funciones R y K), promedio de cada clase de palabras (sustantivos, adjetivos, adverbios) y cantidad de símbolos de puntuación. *Morphadorner* suministra toda la información necesaria sobre las componentes de una oración para calcular las medidas de estilo utilizadas. Estas medidas, ya han sido aplicadas exitosamente en trabajos sobre *atribución de autoría* [6]. La salida de este módulo es una secuencia de n vectores v_i de dimensión M, donde el M es la cantidad total de medidas extraídas del texto. Actualmente, el detector implementado utiliza un total de 24 medidas estilográficas.

[3] Restricciones de espacio impiden una explicación más detallada del enfoque descripto en esta sección. El lector interesado puede encontrar en [4] más detalles y ejemplos de los procedimientos involucrados en este método.

[4] *Morphadorner* es una libreria Java para PLN de acceso libre, suministrada por la Universidad de Northwestern.

- *Identificador de Outliers*: Este módulo recibe como entrada los vectores de características de cada sección y devuelve las secciones que se sospecha de plagio. El módulo elige las secciones que no se corresponden con el estilo del autor del texto. Para tal fin, se implementó el método de detección de outliers basado en la MEDA, el cual fue fue seleccionado por ser simple, ya que la detección no debe insumir demasiado tiempo. Una sección se la considera un outlier si el 30 % de sus medidas es un outlier. Una medida $_i$ es considerada un outlier, si no se encuentra contenida en el siguiente intervalo: $[^{-} - \alpha * M \quad A, ^{-} + \alpha * M \quad A]$, donde el parámetro $\alpha = 1$, fue elegido experimentalmente. Una vez que se dispone de las secciones outliers, las secciones adyacentes se agrupan en una única sección.

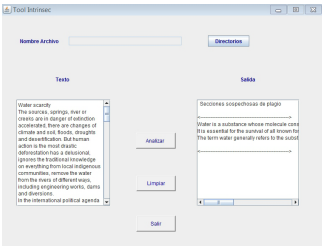

Figura 2. Ejemplo de ejecución del ambiente.

El Ambiente de Ejecución

Un ambiente de ejecución define el contexto en el cual se ejecutan ciertas tareas y, en este trabajo, está compuesto de los módulos definidos previamente. De acuerdo a nuestro conocimiento, no se encuentran disponibles actualmente herramientas de libre acceso que permitan realizar la detección intrínseca, existiendo en cambio herramientas online como *Stylysis* [5] que tienen características similares al ambiente propuesto, pero sólo muestran las secciones con cambios de estilo, es decir, marca en el texto completo los puntos donde se producen alteraciones en el estilo, pero no aisla las secciones que no se corresponden con el autor del texto como se hace en nuestro caso.

El ambiente propuesto en este trabajo, denominado *ToolIntrinsec*[6], tiene las características deseables de una interface de usuario amigable, ya que es fácil de usar, muestra los resultados de forma sencilla y es portable por estar codificado en el lenguaje Java. Permite que un usuario pueda comprobar si un texto contiene secciones que se sospecha de plagio y luego se puede buscar la fuente utilizada por otro método (extrínseco). El usuario tiene dos opciones de ingreso de un texto para el análisis de plagio intrínseco. Una de ellas, consiste en especificar el archivo donde se encuentra el texto, y la otra es ingresando directamente el texto en la sección dedicada a tal fin. Así, por ejemplo, en la Figura 2 se muestra que en la sección *Texto* se ha ingresado el texto (traducido al inglés) de una nota sobre *"El Problema de Escasez de Agua"* [7], a cuyo texto se le adicionó otro

[5] http://memex2.dsic.upv.es:8080/StylisticAnalysis

[6] Disponible en forma gratuita en http://sites.google.com/site/merrecalde/resources para aquellos investigadores que deseen profundizar en el tema.

[7] http://www.solociencia.com/ecologia/problematica-global-agua-escasez-agua.htm

fragmento (de un autor distinto) sobre el agua [8], mostrando en la sección *Salida* una fracción del fragmento utilizado para realizar el plagio artificial.

5. Experimentos

Para los experimentos se seleccionaron de forma aleatoria una colección de 1000 archivos de un total de 6000 archivos extraídos del corpus Pan09. El corpus es una colección de documentos en inglés del proyecto Gutenberg, que contienen fragmentos plagiados generados artificialmente [1]. Para evaluar el comportamiento del algoritmo de detección, éste debe producir como salida un archivo *xml* con las anotaciones de las detecciones realizadas. Para computar las medidas de evaluación se utilizó el script Pyhton *perfmeasure.py* suministrado por los organizadores de la competencia, que devuelve la precisión, el recall, la granularidad y el *Plagdet score* u overall.

En la siguiente tabla, se muestran los valores totales obtenidos con el detector propuesto en este trabajo:

Precisión	Recall	Granularidad	plag-det
0,1204	0,2430	1,27	0,1361

El único detector intrínseco que compitió en Pan10, obtuvo los siguientes resultados para el corpus Pan09 [10]:

Precisión	Recall	Granularidad	plag-det
0,0752	0,1852	1.71	0,0743

En la comparación entre ambos, se observa que el detector implementado supera la performance de este analizador con mejores valores de precisión, recall y granularidad. Por otra parte, los siguientes son los resultados de la competencia Pan09 en la tarea de detección intrínseca:

Puesto	Precisión	Recall	Granularidad	plag-det
1	0,2321	0,4607	1,3839	0,2462
2	0,1091	0,9437	1,0007	0,1955
3	0,1968	0,2724	1,4524	0,1766
4	0,1036	0,5630	1,7049	0,1219

Como se puede observar en la tabla anterior, la performance del detector es ligeramente superior a los valores obtenidos en el cuarto puesto, con mejores valores de precisión y granularidad. Esto muestra, que si bien los resultados experimentales son aún preliminares, el enfoque propuesto es altamente competitivo con respecto a otros algoritmos representativos del estado del arte en el área, y nos motiva para continuar profundizando y perfeccionando la propuesta. En particular, se ha observado que el detector tiene un mejor comportamiento cuando el archivo tiene muchas secciones plagiadas pero, no presenta un buen desempeño cuando el texto no tiene plagio, ya que muestra secciones con cambio de estilo de manera incorrecta.

[8] Disponible en: http://es.wikipedia.org/wiki/Agua

6. Conclusiones y Trabajo Futuro

Los experimentos realizados han brindado evidencia aceptable, de que la segmentación de texto es una buena opción para la descomposición de un texto en el análisis intrínseco y que las medidas de estilo utilizadas son útiles para caracterizar un estilo de escritura. Además, el ambiente que soporta este tipo de técnica, es un aporte interesante para la detección de plagio ya que su facilidad de uso permite a usuarios novatos en el tema realizar sus propias verificaciones de situaciones de plagio.

Como trabajo futuro, se planea realizar una comparación entre el algoritmo de segmentación de texto implementado y otros enfoques como el *TextTiler* creado por Marti Hearst, que también forma parte de la librería Morphadorner. Se podría implementar además, un algoritmo híbrido que combine las dos grandes categorías de detección de plagio. De esta manera, en una primera etapa se podría realizar la detección intrínseca y luego aplicar un enfoque extrínseco.

Para mejorar la performance del detector, una posibilidad sería incorporar al modelo estilográfico nuevas medidas de estilo y utilizar otro método de detección de outlier más efectivo. Finalmente, y como ya fuera mencionado en la sección anterior, un objetivo inmediato es mejorar el desempeño del detector, en aquellos casos de archivos que no tienen secciones plagiadas.

Referencias

1. Enrique Vallés Balaguer. Empresa 2.0: Detección de plagio y análisis de opiniones. Master's thesis, Universidad Politécnica de Valencia, 2011.
2. Luis Alberto Barrón Cedeño. Detección automática de plagio en texto. Master's thesis, Universidad Politécnica de Valencia, 2008.
3. Chien-Ying Chen, Jen-Yuan Yeh, and Hao-Ren Ke. Plagiarism detection using rouge and wordnet. *CoRR*, abs/1003.4065, 2010.
4. Freddy Y.Y. Choi. Advances in domain independent linear text segmentation. In *Proc. of 1st Conf. of ANLP*, pages 26–33. Morgan Kaufmann, 2000.
5. Sven Meyer Zu Eissen and Benno Stein. Intrinsic plagiarism detection. In *ECIR*, Lecture Notes in Computer Science, pages 565–569. Springer, 2006.
6. Sven Meyer Zu Eissen, Benno Stein, and Marion Kulig. Plagiarism detection without reference collections. In *GfKl*, pages 359–366. Springer, 2006.
7. José E. Medina Pagola and Laritza Hernández Rojas. Segmentación por tópicos en documentos de múltiples párrafos. *ACIMED [online]*, 15(6), 2007.
8. Jeffrey C. Reynar and Adwait Ratnaparkhi. A maximum entropy app. to identifying sentence boundaries. In *Proc. of 5th Conf. of ANLP*, pages 16–19, 1997.
9. Benno Stein, Nedim Lipka, and Peter Prettenhofer. Intrinsic plagiarism analysis. *Language Resources and Evaluation*, 45(1):63–82, 2011.
10. Pablo Suárez, José Carlos González Cristóbal, and Julio Villena-Román. A plagiarism detector for intrinsic plagiarism. In *Report for PAN at CLEF 2010*, 2010.
11. David M. J. Tax. *One-class classification; Concept-learning in the absence of counter- examples*. PhD thesis, Delft University of Technology, 2001.