REVISTA ANTIOQUEÑA DE LAS CIENCIAS COMPUTACIONALES Y LA INGENIERÍA DE SOFTWARE

ISSN: 2248-7441

Vol. 7, No. 1, Enero – Junio 2017

Publicación semestral
Editorial Instituto Antioqueño de Investigación
Medellín, Antioquia

Investigar – **A**plicar – **I**nnovar
De Antioquia para el mundo

Revista Antioqueña de las Ciencias Computacionales
y la Ingeniería de Software
ISSN: 2248-7441
fundacioniai.org/raccis
raccis@fundacioniai.org

Paraphrase detection method for English language

Método de detección de paráfrasis para el idioma inglés

Yusdanis Feus[1], Leodanys Guerrero[2], Saylin Pompa[3]
[1, 2, 3]Universidad de Granma. Bayamo, Cuba
[1]*yfeusp(AT)udg.co.cu*, [2]*lguerrerogrey(AT)udg.co.cu*, [3]*spompan(AT)udg.co.cu*

Artículo de Investigación
Recibido: 21-03-2017
Revisado: 16-05-2017
Aceptado: 20-05-2017

Abstract
Paraphrase detection is commonly used in various research areas related to the natural language processing, as information retrieval, machine translation, text summarization, automatic identification of text copyright infringement and question answering. Essentially, the methods for paraphrase detection aims at deciding whether two sentences have the same meaning or not. This paper presents a supervised machine learning approach for paraphrase detection, which uses lexical and semantic information. In order to identify paraphrases are used three different machine learning classifiers: support vector machines, k-nearest neighbors and decision trees. In the empirical evaluation, we explore the discriminating power of the lexical and semantic attributes set by separating these types of attributes in two different problems respectively. Moreover, we study the behavior of the three classifiers by combining these lexical and semantic attributes in a single set. We used the Microsoft Research Paraphrase Corpus data set for the empirical evaluation, and our method was compared with respect to related approaches. The experiments showed that the support vector machines classifier in combination with the set of lexical attributes reached the higher performance.

Keywords: natural language processing, method, paraphrase detection, machine learning.

Resumen
La detección de paráfrasis se usa comúnmente en diversas áreas de investigación relacionadas con el procesamiento del lenguaje natural, tales como la recuperación de información, traducción automática, generación de resúmenes, identificación de plagio en textos y búsqueda de respuestas. El objetivo de los métodos de detección de paráfrasis es decidir si dos oraciones tienen el mismo significado. Este artículo presenta un método basado en aprendizaje automático supervisado para la detección de paráfrasis, el cual usa información léxica y semántica. Con el fin de identificar paráfrasis se usan tres clasificadores de aprendizaje automático: máquinas de vectores de soporte, *k*-vecinos más cercanos y árboles de decisión. En la evaluación empírica se examina el poder de discriminación de los conjuntos de atributos léxicos y semánticos de forma separada. Además, se estudia el comportamiento de los tres clasificadores con la combinación de los atributos léxicos y semánticos en un solo conjunto. Se utilizó el conjunto de datos *Microsoft Research Paraphrase Corpus* para la evaluación empírica y se comparó el método propuesto con los acercamientos similares. Los experimentos mostraron que el clasificador máquina de vectores de soporte alcanzó el mayor rendimiento con la combinación del conjunto de atributos léxicos.

Palabras clave: procesamiento del lenguaje natural, método, detección de paráfrasis, aprendizaje automático.

1. Introducción

El Procesamiento de Lenguaje Natural (PLN) incluye retos de investigación en múltiples dimensiones, uno de ellos es la paráfrasis. De forma general, paráfrasis son aquellas expresiones lingüísticas diferentes en la forma, pero con (aproximadamente) el mismo significado [1]. La paráfrasis se ha definido en dependencia del contexto en que se aplica y su función [2-4] y puede ser construida en varios niveles: palabras, oraciones, párrafos o discursos. Desde el punto de vista del PLN, entre las principales áreas de investigación se encuentra: 1) la *generación*, que es la tarea que se encarga de parafrasear automáticamente un texto en cualquiera de los niveles anteriormente mencionados; 2) la *extracción*, que consiste en adquirir paráfrasis o candidatas a partir de un corpus; y 3) la *detección* (identificación o reconocimiento), una tarea dedicada a decidir si dos o más textos están parafraseados o no.

La detección de paráfrasis es un campo de investigación activo que se utiliza para refinar las soluciones a otros problemas del PLN, por ejemplo, la recuperación de información. En este contexto, dada una consulta en lenguaje natural el motor de búsqueda es capaz de identificar y retornar documentos con un significado similar o relacionado con el texto buscado. Otra área importante del PLN que usa intensivamente la detección de paráfrasis es la identificación de plagio en textos [2]. En esta tarea resulta útil reconocer ideas u oraciones que tienen el mismo significado que otras, ya que con el uso de sinónimos y modificaciones sintácticas se puede generar casos de plagio muy difíciles de detectar. Otros campos de aplicación de la paráfrasis son la traducción automática [5], la generación automática de resúmenes [6] y la búsqueda de respuestas [7].

La mayoría de las investigaciones han abordado el problema de la detección de paráfrasis a nivel de oraciones y, generalmente, para el idioma inglés, debido a su universalidad. Además, algunas de las herramientas para el PLN solamente están disponibles para este idioma. Por ejemplo, corpus para la evaluación de los métodos y herramientas para determinar el nivel de similitud semántico entre dos palabras.

Las primeras técnicas para la detección de paráfrasis se basaron en coincidencias léxicas, es decir, el nivel de similitud entre los textos se calcula en función del número de coincidencia entre las palabras [8-10]. Estos métodos no son capaces de identificar paráfrasis entre oraciones que utilizan sinónimos para transmitir el mismo significado, por ejemplo, *consecuencias* y *resultados*. Luego de la disponibilidad de métricas para determinar la similitud entre un par de palabras, basadas en la herramienta WordNet [11], se perfeccionaron las técnicas de detección de paráfrasis con la combinación de las puntuaciones individuales obtenidas entre las palabras de las oraciones [12, 13]. Algunas métricas relativas al área de recuperación de información también se han aplicado directamente a la detección de paráfrasis, tales como la distancia de Manhattan, la distancia euclidiana, similitud del coseno [14] y el uso de modelos probabilísticos [15], además de otras medidas diseñadas específicamente para la paráfrasis como las basadas en *n*-gramas [16] y medidas asimétricas [17].

Últimamente se ha abordado la detección de paráfrasis desde el punto de vista del aprendizaje automático supervisado y se ha utilizado algoritmos de clasificación, tales como máquinas de vectores de soporte, *k*-vecinos más cercanos y máxima entropía [18, 19]. Con la combinación de técnicas del aprendizaje automático y medidas de similitud de textos léxicas, sintácticas y semánticas se ha logrado elevar la efectividad de los métodos de detección de paráfrasis.

Entre las principales limitaciones de los trabajos existentes se puede mencionar que los métodos más precisos en la detección de paráfrasis no son fácilmente reproducibles (por ejemplo, ausencia de códigos fuentes, nivel de detalle inadecuado para su implementación, etc.). Vale destacar que existen potentes herramientas para PLN disponibles en Internet, por ejemplo, para el preprocesamiento del texto, una etapa crucial en la detección de paráfrasis. Para alcanzar niveles adecuados de precisión en la detección, además de realizar la etapa de preprocesamiento del texto, se necesita un algoritmo que detecte la mayor cantidad de tipos de paráfrasis posibles. Su eficacia estará determinada por la cantidad que sea capaz de detectar correctamente y de la complejidad de las mismas.

Este artículo se enfoca al problema de la detección de paráfrasis en el idioma inglés y el objetivo es desarrollar un método para la detección de paráfrasis desde el punto de vista del aprendizaje automático. El método propuesto, como en algunos acercamientos anteriores [14, 18], utiliza información léxica y semántica extraída de los textos para estudiar el comportamiento de tres algoritmos de clasificación automática: máquinas de vectores de soporte, *k*-vecinos más cercanos y el árbol de decisión C4.5. Los experimentos realizados constituyen un aporte de esta investigación. Primero se explora el poder discriminación de los atributos léxicos y semánticos por separado con cada algoritmo de clasificación seleccionado; luego, con el objetivo de incrementar el rendimiento del método propuesto, se analiza el impacto de la combinación de los atributos léxicos y semánticos en un solo conjunto.

2. Acercamientos previos basados en aprendizaje automático

Los métodos desarrollados para la detección de paráfrasis se pueden dividir en dos categorías diferentes dependiendo de las técnicas utilizadas. Por un lado, los acercamientos que utilizan *funciones de similitud* para decidir si un par de oraciones son paráfrasis o no, y por el otro, los que emplean *aprendizaje automático* para combinar varias características extraídas de los pares de textos. Para el desarrollo del método propuesto en la presente investigación se estudiaron las técnicas basadas en aprendizaje automático, debido a que muestran los mejores resultados y dan mayores posibilidades para la experimentación. A continuación, se enuncian algunos de los acercamientos más representativos en el área, que resuelven la detección de paráfrasis desde el punto de vista del aprendizaje supervisado.

El acercamiento desarrollado por Qiu et al. [9] usa un proceso de dos fases: 1) se extraen las unidades de contenido semántico clave desde ambas oraciones, también llamadas minutas de información (*information nuggets*), y 2) se emparejan las minutas de información de cada oración. Si después del proceso alguna minuta permanece desemparejada su contenido es analizado. Si las oraciones no contienen minutas desemparejadas o son insignificantes en su totalidad, el método emite una clasificación positiva de paráfrasis.

La idea principal del acercamiento propuesto por Zhang y Patrick [10] es crear formas canónicas de las oraciones, puesto que los textos con significados similares tienen más probabilidad de transformarse en los mismos textos superficiales (*surface texts*) que aquellos con diferente significado. Solamente se usa un número limitado de técnicas de canonicalización, entre las que se incluye el reemplazamiento de números por etiquetas genéricas, la conversión de voz pasiva a activa y la sustitución de todas las frases en tiempo futuro (tales como *expect to* y *plan to*) por la simple palabra *will*. Un ejemplo de la transformación de voz pasiva a activa es *Those reports were denied by Prince Nayef* a *Prince Nayef denied those reports*.

El método propuesto por Malakasiotis [14] utiliza un clasificador automático de máxima entropía para combinar un conjunto de métricas de similitud de cadenas, tales como las distancias de Levenshtein, de Jaro-Winkler, de Manhattan, entre otras. Este acercamiento explota la base de datos léxica WordNet para realizar la comparación entre palabras que son sinónimos.

La novedad del acercamiento propuesto por Kozareva y Montoyo [18] consiste en su experimentación, debido a que exploran el poder de discriminación de características léxicas y semánticas para identificar paráfrasis. Además, estudian el comportamiento de tres clasificadores de aprendizaje automático: máquina de vectores de soporte, *k*-vecinos más cercanos y máxima entropía. Con el objetivo de mejorar el rendimiento del sistema de detección de paráfrasis, también evalúan el impacto de las características léxicas y semánticas en su conjunto y los resultados de un ensamble de voto.

Las métricas desarrolladas para la traducción automática también han sido aplicadas a la detección de paráfrasis [19] con el objetivo de abordar el problema desde un punto de vista supervisado. El método utiliza tres clasificadores y mediante técnicas de ensamble obtiene el veredicto final. Los clasificadores utilizados son: regresión logística, máquina de vectores de soporte y *k*-vecinos más cercanos.

3. Método propuesto

Un acercamiento para la detección de paráfrasis es el uso de algoritmos de clasificación pertenecientes al área de la minería de datos y el aprendizaje automático. En las tareas de clasificación el objetivo fundamental es estimar, a través de un modelo computacional, una función objetivo desconocido que rige los datos de entrada (datos de entrenamiento). Así, los datos de entrenamiento sirven de entrada al algoritmo de aprendizaje, cuya tarea es estimar la función objetivo por medio del modelo de aprendizaje. Los datos de entrenamiento están estructurados en forma de un vector de atributos, entre los que se encuentra uno especial llamado *clase*. Comúnmente se supone que la función objetivo, a estimar por el algoritmo de aprendizaje, rige la relación entre los atributos y la clase. En la etapa de prueba o aplicación, los algoritmos de aprendizaje son usados para clasificar o predecir la clase de los datos de prueba. En la detección de paráfrasis una solución directa es considerar dos posibles etiquetas de clase (clasificación binaria): paráfrasis y no paráfrasis.

Las ventajas del aprendizaje automático consisten en la habilidad de manipular una gran cantidad de datos y la posibilidad de incorporar nuevas fuentes de información léxicas, sintácticas y semánticas en una sola ejecución. El obstáculo principal del uso de estas técnicas es la disponibilidad y calidad de un corpus de entrenamiento, porque, para aplicar un algoritmo de minería de datos a la detección de paráfrasis, el conjunto de entrenamiento debe incluir ejemplos que representen varias tipologías de paráfrasis. Mientras más tipologías cubiertas, mayor será la posibilidad de obtener un modelo de aprendizaje adecuado. De la misma forma, los atributos de los ejemplos de entrenamiento (las métricas extraídas de pares de oraciones) deben ser capaces de discriminar entre las posibles clases del problema (paráfrasis o no).

La Figura 1 muestra el esquema de entrenamiento del método propuesto, que persigue obtener un modelo capaz de clasificar un nuevo par de oraciones a partir del conocimiento extraído de los datos proporcionados inicialmente. Por su parte, la Figura 2 presenta el esquema para el uso de la minería de datos en el problema en cuestión. Una vez adquirido el modelo de aprendizaje es posible clasificar un nuevo par de oraciones, luego de realizar un proceso de extracción de características.

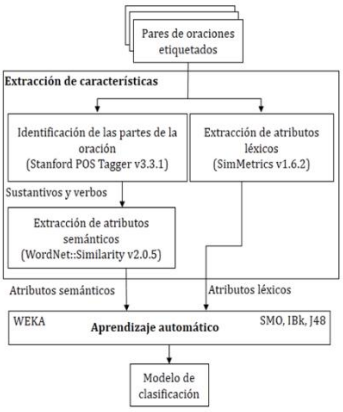

Figura 1. Esquema de entrenamiento

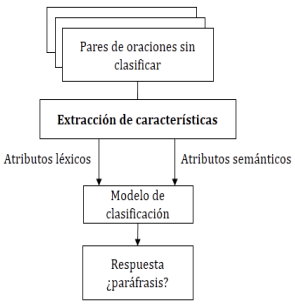

Figura 2. Esquema de clasificación

En los esquemas de entrenamiento (Figura 1) y clasificación (Figura 2) del método propuesto se evidencia que, para resolver el problema inicial, se divide en dos módulos: *extracción de características* y *aprendizaje automático*.

3.1 Extracción de características

La identificación de características es un elemento clave en la precisión de un clasificador automático [20], porque son el conocimiento que tiene el algoritmo para generar el modelo y luego clasificar nuevos casos. Las características deben estar basadas en indicios que permitan decidir si un par de oraciones constituye una paráfrasis o no. Unos rasgos pueden dar buenos resultados con un algoritmo de clasificación y peores con otro, por lo que se debe seleccionar el clasificador adecuado para las características extraídas.

Para obtener buen rendimiento con un clasificador automático es necesario extraer atributos relevantes de los pares de oraciones y, como las paráfrasis aparecen en los niveles léxicos, sintácticos, semánticos, pragmáticos o una combinación de estos, se explora el poder de discriminación de atributos en los niveles léxicos y semánticos. Debido a que las paráfrasis son relaciones bidireccionales [21], todos los atributos diseñados captan la similitud de las oraciones en ambas direcciones.

- *Atributos léxicos*. Se extraen de los pares de oraciones sin realizar un tratamiento previo, es decir, sin la eliminación de las puntuaciones ni las palabras vacías (*stop words*). En muchos casos, cuando dos textos son similares en contenido existe un alto grado de solapamiento de su escritura. Se han desarrollado diversas métricas para calcular la distancia o similitud entre un par de cadenas de texto. Para la obtención de los atributos léxicos se utiliza la librería de código abierto SimMetrics v1.6.2 para Java, que contiene implementaciones de: distancia Levenshtein, distancia de Needleman-Wunsch, distancia de Smith-Waterman, distancia Smith-Waterman-Gotoh, distancia de Jaro, distancia de Jaro-Winkler, longitud de desviación de Chapman, longitud media de Chapman, distancia de *q*-gramas, distancia de Manhattan, similitud del coseno, coeficiente de Dice, distancia euclidiana, similitud de Jaccard, similitud de Monge-Elkan, coeficiente de solapamiento, coeficiente de coincidencias y la distancia Smith-Waterman-Gotoh-Windowed-Affine. Además, con la librería SimMetrics se obtuvo un total de 18 atributos léxicos, normalizados en el intervalo [0; 1] y se incorporó como atributo la cantidad de palabras compartidas entre ambos textos (Words-Shared-Count), para conformar un total de 19 atributos léxicos.

- *Atributos semánticos*. Para la tarea de detección de paráfrasis no es suficiente el análisis de la similitud léxica entre los textos, también se debe tener en cuenta su información semántica, la cual no se puede ver aislada totalmente del análisis léxico. Es común ver que el alineamiento semántico se aborda a nivel léxico [22], con el objetivo es identificar si el significado de un elemento léxico de un texto se expresa también en el otro.

  Con la finalidad de obtener atributos semánticos, primero se identifican las partes de cada oración con la herramienta Stanford POS Tagger v3.3.1 y, posteriormente, se extraen los sustantivos y los verbos de cada una. Con la clase WordStemmer de la librería JWI v2.4.0 (*Java WordNet Interface*), los verbos se transforman a su forma base y los sustantivos al singular. JWI es una interfaz para acceder a WordNet desde el lenguaje de programación Java que soporta desde la versión 1.6 hasta la 3.0 de la base de datos léxica, y se considera la librería más versátil para tal propósito [23]. Para el cálculo de la similitud entre palabras se utiliza las métricas implementadas en el módulo de Perl WordNet::Similarity v2.0.5 y la versión 3.0 de WordNet. El propósito de estas métricas es cuantificar la similitud entre dos palabras [11], lo que resulta útil para la tarea de detección de paráfrasis porque si un par de oraciones comparten muchas palabras similares, se podría suponer como un buen indicador de que tienen un significado análogo en su conjunto.

  Las métricas de WordNet::Similarity utilizadas son: Leacock y Chodorow, Lesk, Wu y Palmer, Resnik, y Jiang y Conrath, y a cada una se le especifica las palabras a comparar, la parte de la oración (sustantivo, verbo, etc.) y el número del significado (una palabra puede tener más de un significado). En el cálculo de la similitud semántica solamente se tuvo en cuenta los sustantivos y los verbos de las oraciones; se consideró como nivel de similitud semántico a la suma de las similitudes individuales de cada par diferente de sustantivos y verbos presentes en las oraciones y, finalmente, se obtuvo seis atributos semánticos, uno por cada métrica seleccionada.

3.2 Aprendizaje automático

El objetivo del módulo de aprendizaje automático es obtener un modelo de clasificación con cada uno de los algoritmos seleccionados, para establecer comparaciones entre los resultados obtenidos con ellos. Un módulo de aprendizaje automático puede estar compuesto por uno o varios clasificadores y, en este caso, para la experimentación se seleccionaron tres de los algoritmos más conocidos en el área de la minería de datos: máquina de vectores de soporte, *k*-vecinos más cercanos y el árbol de decisión C4.5. Por un lado, las máquinas de vectores de soporte y *k*-vecinos más cercanos se han empleado en métodos [18, 19], mientras que los árboles de decisión para la determinación de umbrales a partir de corpus de entrenamiento [13]. Los fundamentos básicos de cada uno de los algoritmos seleccionados son:

- *Máquinas de vectores de soporte*. Las máquinas de vectores de soporte (SVM, *Support Vector Machines*) ejecutan algoritmos de clasificación supervisada de forma rápida y efectiva [20]. Una SVM es un modelo

que representa a los puntos de muestra en el espacio, separa las clases por un espacio lo más amplio posible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, pueden ser clasificadas en una u otra en función de su proximidad. En términos geométricos, una SVM construye un hiperplano o conjunto de estos en un espacio multidimensional que separe de forma óptima los puntos de una clase de la de otra. Las SVM tienen alto rendimiento en problemas de clasificación con dos clases y un espacio de atributos múltiple [18]. Como la detección de paráfrasis se reduce a un problema de dos clases, se considera pertinente su utilización.

- *Algoritmo k-vecinos más cercanos.* El aprendizaje basado en ejemplos es un tipo de aprendizaje supervisado, basado en la hipótesis de que las tareas cognitivas se llevan a cabo contrastando la similitud entre las situaciones nuevas y las situaciones pasadas almacenadas. En el proceso de aprendizaje se memorizan todos los ejemplos en su forma original, sin necesidad de intentar generalizar ninguna regla ni representación más concisa. Uno de los métodos más básicos de aprendizaje basado en ejemplos es *k*-vecinos más cercanos (*k*-NN, *k-Nearest Neighbors*). Este algoritmo es una conocida aproximación estadística que ha sido aplicado a tareas como la clasificación de textos desde sus orígenes [24].
- *Árboles de decisión*. Los métodos de aprendizaje supervisado basados en árboles de decisión son uno de los más populares dentro del área de la IA para tratar el problema de la clasificación [25]. Estos algoritmos permiten la extracción y representación de reglas de clasificación obtenidas a partir de un conjunto de ejemplos. Entre los algoritmos más conocidos pertenecientes a esta familia se encuentran ID3, C4.5 (una extensión de ID3) y C5.0 (una versión extendida de C4.5 de carácter comercial). En la tarea de detección de paráfrasis también se han empleado estos algoritmos, específicamente J48 para la determinación del umbral similitud [13, 26]. J48 es una implementación libre en el lenguaje de programación Java del algoritmo C4.5 en la herramienta WEKA (*Waikato Environment for Knowledge Analysis*) [27]. Para el proceso entrenamiento y clasificación se utilizó la herramienta WEKA v3.7.5, debido a que contiene implementaciones de los algoritmos SVM (como SMO), *k*-NN (como IBk) y C4.5 (como J48).

3.3 Conjunto de datos y métricas para la evaluación

Para entrenar y evaluar el rendimiento de cada clasificador en el método propuesto se empleó el conjunto de datos estándar *Microsoft Research Paraphrase Corpus* (MSRPC) [21], utilizado en entrenamiento y evaluación de los acercamientos previos, a pesar de que algunos investigadores [10] han encontrado indicios para afirmar que no es un corpus rico en relaciones de paráfrasis, por lo menos comparado con la distribución que se encuentra normalmente en los textos. El MSRPC está disponible en dos archivos de texto que contienen los datos de entrenamiento y los datos de prueba. El primero consiste en 4096 pares de oraciones, de los cuales 2753 son paráfrasis, mientras que el conjunto de pruebas contiene 1725 pares de oraciones, de los cuales 1147 constituyen paráfrasis. En los experimentos se utilizaron las medidas de rendimiento más comunes para la evaluación de los algoritmos: exactitud (*accuracy*), precisión, cobertura (*recall*) y la medida-F (F-*score*), tal como se describe en las ecuaciones (1), (2) y (3) respectivamente.

$$exactitud = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

$$precisión = \frac{TP}{TP + FP} \quad (2)$$

$$cobertura = \frac{TP}{TP + FN} \quad (3)$$

$$medida\text{-}F = 2 \cdot \frac{precisión \cdot cobertura}{precisión + cobertura} \quad (4)$$

Donde TP es el conjunto de verdaderos positivos, TN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos. Nótese que TP y TN miden los ejemplos de prueba clasificados correctamente por el algoritmo de aprendizaje, mientras que FP y FN representan los errores cometidos en la clasificación.

4. Resultados y discusión

Se realizaron tres tipos de experimentos encaminados a encontrar cuál algoritmo de aprendizaje automático permite una mejor detección de paráfrasis con los atributos implementados. Primero se experimentó con los atributos léxicos, luego con los semánticos y por último con la combinación de ambos.

Una cuestión clave en la implementación es que las métricas de WordNet::Similarity requieren la especificación del significado de cada palabra, debido a que en WordNet una palabra tiene más de un significado. Los experimentos iniciales para el cálculo de la similitud semántica se realizaron solamente con el primer significado (predominante) de cada palabra. Luego se experimentó con todos los significados de las palabras para el cálculo de la similitud, debido a que una palabra puede tener más de un significado, pero los resultados fueron similares a los alcanzados con los experimentos iniciales. Además, cuando se usaron todos los significados de las palabras el tiempo de cálculo creció considerablemente, por lo tanto, para el cálculo de la similitud semántica solamente se utilizó el primer significado.

Todos los experimentos se realizaron sobre la herramienta WEKA y a los algoritmos SMO y J48 se les dio la configuración por defecto, porque se notó que aportaban los mejores resultados. Sin embargo, los resultados del algoritmo IBk con el valor de k por defecto (k = 1) eran deficientes y se seleccionó un k = 50, después que se analizaron los resultados obtenidos con valores de k entre 1 y 100.

La Tabla 1 muestra los resultados de los tres clasificadores con los atributos léxicos solamente. Como se

puede observar, el algoritmo SMO aportó mejores resultados que IBk y J48 en todos los parámetros de rendimiento con los atributos léxicos. Los algoritmos IBk y J48 obtuvieron resultados similares de exactitud, pero IBk fue más preciso detectando los casos positivos de paráfrasis, sin embargo, J48 fue mejor identificando los casos negativos. La medida-F muestra que IBk fue más preciso de forma general que J48.

Tabla 1. Resultados con los atributos léxicos

Algoritmo	Exactitud	Precisión	Cobertura	Medida-F
SMO	76.2%	78.4%	88.8%	83.2%
IBk	72.6%	75.7%	86.6%	80.8%
J48	72.6%	77.0%	84.0%	80.3%

La Tabla 2 muestra los resultados de los clasificadores usando solamente como atributos las métricas de carácter semántico. Se puede apreciar que, aunque el algoritmo IBk obtuvo mayor exactitud, el resultado no es relevante. El algoritmo SMO no clasificó ningún caso como negativo (no paráfrasis). Además, los resultados de exactitud están cercanos el límite inferior considerado para el conjunto de pruebas del MSRPC. Al ser 1725 pares de oraciones, de los cuales 1147 son paráfrasis, si un método los clasifica a todos como positivos entonces se obtendría una exactitud del 66.5%, como lo hizo el algoritmo SMO.

Tabla 2. Resultados con los atributos semánticos

Algoritmo	Exactitud	Precisión	Cobertura	Medida-F
SMO	66.5%	66.5%	100%	79.9%
IBk	68.2%	70.3%	90.3%	79.1%
J48	67.4%	69.9%	89.5%	78.5%

Dos aspectos del diseño del método inciden directamente en los resultados deficientes de los algoritmos de clasificación con los atributos semánticos: 1) la identificación de las partes de la oración, realizada en este caso con la herramienta Stanford POS Tagger, y que es un campo de investigación que no ha sido resuelto en su totalidad, por lo que hubo errores en la identificación de los sustantivos y verbos en su contexto. 2) Solamente se tuvo en cuenta el primer significado de las palabras (sustantivos y verbos) en el cálculo del nivel de similitud semántico, es decir, no se utilizó ningún procedimiento para la desambiguación de su significado. La Tabla 3 muestra los resultados de los clasificadores con la combinación de los atributos léxicos y semánticos. El resultado de los tres algoritmos, con la combinación de los atributos léxicos y semánticos en un solo conjunto, no superó los resultados obtenidos con los atributos léxicos (Tablas 1 y 3). El algoritmo SMO fue nuevamente superior a los restantes, lo que ratifica su importancia en la detección de paráfrasis.

Tabla 3. Resultados con los atributos léxicos y semánticos

Algoritmo	Exactitud	Precisión	Cobertura	Medida-F
SMO	75.6%	77.9%	88.3%	82.8%
IBk	72.1%	74.4%	88.4%	80.8%
J48	72.6%	76.5%	84.7%	80.4%

En la Tabla 4 se muestra la comparación de los resultados obtenidos con el algoritmo SMO y los atributos léxicos ($SMO_{léx}$) y SMO con atributos léxicos y semánticos ($SMO_{léx\text{-}sem}$). Aunque el mejor resultado obtenido en esta investigación ($SMO_{léx}$) no supera a todos los acercamientos previos que se tuvieron en cuenta (Tabla 4), se considera pertinente hacer algunas valoraciones. En exactitud (Ecuación (1)), que representa el porcentaje de aciertos global del clasificador, $SMO_{léx}$ es superado solamente por dos métodos. En precisión (Ecuación (2)), que indica el porcentaje de pares de oraciones que fueron correctamente predichos en la clase positiva de paráfrasis, $SMO_{léx}$ también es superado por dos métodos. En cobertura (Ecuación (3)), que revela el porcentaje de pares de oraciones paráfrasis que fueron reconocidos como tales, en este caso $SMO_{léx}$ es superado por un acercamiento. Debido a que la precisión y la cobertura dan medidas distintas, ambas deseables, para manejar el compromiso entre ellas se utiliza la medida-F (Ecuación (4)), que involucra de forma armónica a ambas medidas. En medida-F, $SMO_{léx}$ ocupa la segunda posición.

Tabla 4. Comparación con acercamientos previos

Método	Exactitud	Precisión	Cobertura	Medida-F
Canonicalización de textos [10]	71.9%	74.3%	88.2%	80.7%
Clasificación por el significado de disimilitud [9]	72.0%	72.5%	93.4%	81.6%
Uso de información léxica y semántica [18]	76.6%	94.4%	68.7%	79.5%
Combinación de medidas de similitud de palabras [14]	76.1%	79.3%	86.7%	82.8%
Empleo de métricas de traducción automática [19]	77.4%	-	-	84.1%
$SMO_{léx}$	76.2%	78.4%	88.8%	83.2%
$SMO_{léx\text{-}sem}$	75.6%	77.9%	88.3%	82.8%

5. Conclusiones

En este artículo se presenta un método de detección de paráfrasis utilizando técnicas de aprendizaje automático y minería de datos. Se evaluaron tres de los algoritmos de aprendizaje automático más conocidos (SVM, *k*-NN y C4.5) sobre el conjunto de datos MSRPC, para determinar cuál es el más apropiado para la detección de paráfrasis con los atributos implementados. El MSRPC ha sido el conjunto de datos estándar para la detección de paráfrasis, lo que permitió realizar comparaciones de los mejores resultados obtenidos con esta investigación y los métodos previos desarrollados con el mismo fin.

Los experimentos revelaron que los atributos léxicos obtuvieron el mejor resultado utilizando el algoritmo SMO, con un 76.2% de exactitud y una medida-F de un 83.2%. Los atributos semánticos no aportaron resultados relevantes con ningún clasificador y la combinación de atributos léxicos y semánticos en un sólo conjunto tampoco alcanzó los resultados esperados. Con el análisis de los resultados obtenidos se considera pertinente el uso de las SVM en la detección de paráfrasis, debido su capacidad para la clasificación binaria con un espacio de atributos de alta dimensión.

En trabajos futuros se perfeccionarán los atributos semánticos y se incorporarán atributos sintácticos, con el

fin de mejorar el rendimiento del método propuesto. Además, se experimentará con un clasificador basado en redes neuronales, debido a que también han sido empleados en tareas de clasificación de textos.

Referencias

[1] Barrón, A., Vila, M. & Rosso, P. (2010). Detección automática de plagio: de la copia exacta a la paráfrasis. Proceedings of Panorama Actual de la Lingüística Forense en el Ámbito Legal y Policial: Teoría y Práctica. Jornadas (in)formativas de lingüística forense (pp. 76-96). Madrid, España.

[2] Barrón, A. et al. (2013). Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. Computational Linguistics 39(4), pp. 917-947.

[3] Rus, V. et al. (2009). Identification of sentence-to-sentence relations using a textual entailer. Research on Language and Computation 7(2-4), pp. 209-229.

[4] Vila, M., Martí, M. & Rodríguez, H. (2011). Paraphrase concept and typology. A linguistically based and computationally oriented approach. Procesamiento del Lenguaje Natural 46, pp. 83-90.

[5] Onishi, T., Utiyama, M. & Sumita. (2011). Paraphrase lattice for statistical machine translation. IEICE transactions on information and systems 94(6), pp. 1299-1305.

[6] Keck, C. (2014). Copying, paraphrasing, and academic writing development: A re-examination of L1 and L2 summarization practices. Journal of Second Language Writing 25, pp. 4-22.

[7] Fader, A., Zettlemoyer, L. & Etzioni, O. (2013). Paraphrase-Driven Learning for Open Question Answering. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (pp. 1608-1618). Sofia, Bulgaria.

[8] Clough, P. et al. (2002). METER: MEasuring TExt Reuse. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (pp. 152-159). Philadelphia, USA.

[9] Qiu, L., Kan, M. & Chua, T. (2006). Paraphrase recognition via dissimilarity significance classification. Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (pp. 18-26). Sydney, Australia.

[10] Zhang, Y. & Patrick, J. (2005). Paraphrase identification by text canonicalization. Proceedings of the Australasian language technology workshop, (pp. 160-166). Sydney, Australia.

[11] Pedersen, T., Patwardhan, S. & Michelizzi, J. (2004). WordNet::Similarity - Measuring the relatedness of concepts. Proceedings of the Nineteenth National Conference on Artificial Intelligence (pp. 1024-1025). Cambridge, USA.

[12] Corley, C. & Mihalcea, R. (2005). Measuring the semantic similarity of texts. Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment (pp. 13-18). Ann Arbor, USA.

[13] Hsu, M. & Shih, T. (2014). Real-Time Finger Tracking for Virtual Instruments. Proceedings 7th International Conference on Ubi-Media Computing and Workshops (133-138). Ulaanbaatar, Mongolia.

[14] Malakasiotis, P. (2009). Paraphrase recognition using machine learning to combine similarity measures. Proceedings of the ACL-IJCNLP 2009 Student Research Workshop (pp. 27-35). Suntec, Singapore.

[15] Das, D. & Smith, N. (2009). Paraphrase identification as probabilistic quasi-synchronous recognition. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (pp. 468-476). Suntec, Singapore

[16] Cordeiro, J., Dias, G. & Brazdil, P. (2007). A metric for paraphrase detection. Proceedings of the International Multi-Conference on Computing in the Global Information Technology (pp. 7-16). Guadeloupe, French Caribbean.

[17] Cordeiro, J., Dias, G. & Brazdil, P. (2007). New functions for unsupervised asymmetrical paraphrase detection. Journal of Software 2(4), pp. 12-23.

[18] Kozareva, Z. & Montoyo, A. (2006). Paraphrase identification on the basis of supervised machine learning techniques. Advances in Natural Language Processing: 5th International Conference on NLP (pp. 524-533). Turku, Finland.

[19] Madnani, N., Tetreault, J. & Chodorow, M. (2012). Re-examining machine translation metrics for paraphrase identification. Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 182-190). Montréal, Canada.

[20] Feldman, R. & Sanger, J. (2006). The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge: Cambridge University Press.

[21] Dolan, B., Quirk, C. & Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. Proceedings of the 20th international conference on Computational Linguistics (pp. 350). Morristown, USA.

[22] Dagan, I., Glickman, O. & Magnini, B. (2006). The PASCAL Recognising Textual Entailment Challenge. Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment (pp. 177-190). Southampton, UK.

[23] Finlayson, M. (2014). Java Libraries for Accessing the Princeton Wordnet: Comparison and Evaluation. Proceedings of the 7th Global Wordnet Conference (pp. 78-85). Tartu, Estonia.

[24] Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. ACM Computing Surveys (CSUR) 34(1), pp. 1-47.

[25] Quinlan, J. (1986). Induction of decision trees. Machine learning 1(1), pp. 81-106.

[26] Fernando, S. (2007). Paraphrase Identification. MSc. Thesis, Department of Computer Science. University of Sheffield, UK.

[27] Witten, I. & Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques. USA: Morgan Kaufmann.