Visión general de la detección de paráfrasis

Yusdanis Feus-Pérez
Universidad de Granma, Departamento de Desarrollo
yfeusp@udg.co.cu

Resumen

La paráfrasis es un fenómeno importante que puede ser usado para perfeccionar otras tareas en el procesamiento del lenguaje natural, tales como extracción de información, traducción automática, recuperación de información y la identificación de plagio. La detección de paráfrasis ha sido una tarea de interés creciente en los últimos años. El objetivo de esta tarea es determinar si dos frases tienen el mismo significado sin tener en cuenta su escritura. Este artículo describe los principales recursos y técnicas utilizadas para la detección de paráfrasis. Los acercamientos descritos son separados en dos grupos: por un lado, los que usan una función de similitud y por el otro, los que emplean aprendizaje automático supervisado.

Palabras clave: paráfrasis, detección de paráfrasis, función de similitud, aprendizaje automático supervisado.

Abstract

Paraphrase is an important phenomenon that can be used to improve many other natural language process task, such as information extraction, machine translation, information retrieval and automatic identification of copyright infringement. Paraphrase detection has been one task of growing interest during the last years. This task aims to detect whether two sentences have the same meaning regardless of the writing. This paper presents an overview of the main resources and techniques used for paraphrase detection. The described approaches are separated into two groups: on the one hand, those who use a similarity function and on the other hand, those who employ supervised machine learning.

Keywords: *paraphrase, paraphrase detection, similarity function, supervised machine learning.*

1. Introducción

El lenguaje natural es una herramienta muy potente mediante la cual las personas establecemos la comunicación y relacionamos objetos con otros. Hoy en día es posible emplear diferentes palabras o frases para expresar el mismo significado. Esto está muy relacionado con el conocimiento y los hábitos culturales, que influyen directamente en las habilidades para hablar y escribir.

Son paráfrasis aquellas expresiones lingüísticas diferentes en la forma pero con (aproximadamente) el mismo significado [1]. Esta definición es una de las más simples que existe, otros autores han concebido el término en dependencia del contexto en que lo aplican [2, 3, 4, 5]. Las siguientes oraciones constituyen un ejemplo de paráfrasis, ya que a pesar de las modificaciones en la forma, el significado se mantiene:

- Leonardo da Vinci pintó "La Mona Lisa".
- "La Mona Lisa" fue pintada por Leonardo da Vinci.

La paráfrasis 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 encuentran: la generación de paráfrasis, la extracción de paráfrasis y la detección de paráfrasis. La generación de paráfrasis es la tarea que se encarga de parafrasear automáticamente un texto en cualquiera de los niveles anteriormente mencionados [6]. La extracción de paráfrasis consiste en adquirir paráfrasis o candidatos de esta, a partir de un corpus [7]. La detección (identificación o reconocimiento) de paráfrasis es la tarea dedicada a clasificar si dos o más textos están parafraseados o no.

La detección de paráfrasis es un campo de investigación activo, que ha sido aplicado en muchos otros problemas relacionados, como por ejemplo en 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. Por ejemplo, una búsqueda sobre "mascotas" puede retornar documentos acerca del "perro", que es un resultado pertinente porque el perro es una clase de mascota, lo que evidencia la necesidad de un método de detección de paráfrasis.

Otros campos de aplicación de la paráfrasis son la traducción automática [8], generación automática de resúmenes [9], identificación de plagio en textos [10] y búsqueda de respuestas [11].

Debido a su alta complejidad y a su coste computacional, la mayoría de las investigaciones han abordado el problema a nivel de oraciones. Las primeras técnicas para la detección de paráfrasis se basaron en coincidencias léxicas, es decir, el nivel de similitud entre los textos se calculaba en función del número de coincidencia entre palabras [12, 13, 14]. 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 métodos para determinar la similitud entre un par de palabras basados en el recurso WordNet [15], se mejoró el rendimiento de las técnicas de detección de paráfrasis. Varios autores han propuesto la combinación de las puntuaciones individuales obtenidas entre las muestras [16, 17].

También las métricas de recuperación de información han sido aplicadas directamente a la detección de paráfrasis, tales como la distancia de Manhattan, la distancia euclidiana, similitud del coseno [18], y el

uso de modelos probabilísticos [19]. Además de otras medidas diseñadas específicamente para la paráfrasis como las basadas en n-gramas [20] y medidas asimétricas [21].

En los años más recientes se ha abordado la detección de paráfrasis desde el punto de vista del aprendizaje automático supervisado. Se han utilizado algoritmos de clasificación como máquinas de soporte vectorial, k-vecinos más cercanos y máxima entropía [22, 23]. Con la combinación de técnicas del aprendizaje automático y medidas de similitud de textos léxicas, basadas en conocimiento y en corpus, se ha logrado elevar la exactitud de los métodos de detección de paráfrasis.

El objetivo de la presente investigación es hacer un estudio sobre los principales recursos y las tendencias más relevantes que se han desarrollado para la tarea de detección de paráfrasis.

2. Recursos

En esta sección se examinan tres de los principales recursos utilizados en la detección de paráfrasis. Se describe el conjunto de datos *Microsoft Research Paraphrase Corpus* (MSRPC) para la evaluación de los métodos en la sección 2.1. En la sección 2.2 se da una descripción de la base de datos WordNet y en la sección 2.3 las métricas del módulo WordNet::Similarity. Recursos que además de ser utilizados para la detección de paráfrasis, han sido ampliamente utilizados en otras tareas del PLN.

2.1. MSRPC

El MSRPC[1] está compuesto por miles de pares de oraciones que describen eventos similares, tomados de fuentes de noticias en la web [24]. Los pares de oraciones fueron etiquetados por jueces humanos sobre la base de si son semánticamente equivalentes o no, es decir, si comunican exactamente la misma información sin tener en cuenta la escritura.

El objetivo de los investigadores en la construcción de este conjunto de datos fue crear un "corpus monolingüe de dominio amplio de pares de oraciones alineadas" [24]. Los artículos se fueron seleccionando y agrupando durante un período aproximado de ocho meses. Más de 100 000 artículos fueron reunidos y agrupados en aproximadamente 11 000 porciones.

La cantidad de pares de oraciones posibles de esta colección era enorme, por lo que se emplearon dos estrategias para decidir cuáles de ellos serían ejemplos útiles de paráfrasis. La primera usó una métrica de distancia de edición [25] para filtrar los pares de oraciones. Cada oración se convirtió a letras minúsculas y se emparejó con las restantes en el grupo. Luego la oraciones idénticas y aquellas que solamente diferían en signos de puntuación fueron eliminadas, al igual que los pares donde una oración era significativamente más corta que la otra. Además se eliminaron los pares duplicados en cualquier orden. Con este método de filtrado, aplicando una distancia de Levenshtein[2] (con $n \leq 12$), fueron generados 139 000 pares de oraciones aproximadamente. A este conjunto de datos se le denominó L12.

La segunda estrategia se basó en la tendencia de los periodistas de resumir el contenido de un artículo en sus dos primeras de oraciones. El método toma las dos primeras oraciones de cada artículo y las compara

[1] http://research.microsoft.com/en-us/downloads/607D14D9-20CD-47E3-85BC-A2F65CD28042/default.aspx
[2] Número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra [25].

con las dos primeras oraciones de cualquier otro en el corpus. De forma similar a la primera estrategia, se utilizaron filtros adicionales para restringir los pares de oraciones que fueron finalmente aceptados. Uno fue una heurística basada en cadena que asegura que si al menos tres palabras de más de cuatro caracteres son las mismas, entonces se puede dar por hecho una similitud entre las oraciones. El otro fue un filtro que comprueba que la primera oración es al menos la mitad del tamaño de la otra. Después de aplicar estos filtros, 214 000 pares de oraciones fueron generados. Este conjunto de datos fue denominado F2.

La Tasa de Error de Alineamiento (AER, *Alignment Error Rate*), una métrica tomada del campo de la traducción automática estadística [26] fue empleada para medir cuantitativamente la calidad de las paráfrasis generadas en cada conjunto de datos. Así como una pequeña muestra fue analizada manualmente para examinar los tipos de alternancias de paráfrasis encontrados en los pares de oraciones, incluyendo por ejemplo la elaboración (donde una oración tiene más información que otra sobre del mismo tema) y la sinonimia (donde una palabra ha sido sustituida por otra de significado similar).

Sobre la base de la evaluación basada en la AER, se encontró que el conjunto de datos L12 contenía un mayor porcentaje de paráfrasis. Sin embargo, al considerar el análisis manual de las paráfrasis, se encontró que el conjunto de datos F2 contenía ejemplos de paráfrasis más enriquecidos, los cuales son de interés para aplicaciones de similitud, aunque también se generó más ruido.

El próximo paso fue la clasificación de los pares de oraciones por dos jueces humanos. Se aplicaron métodos de filtrado adicionales para reducir el conjunto de datos a 5 801 pares de oraciones. Se les orientó a los jueces que clasificaran los pares de oraciones teniendo en encuenta un conjunto de directrices para determinar la equivalencia semántica. Los desacuerdos fueron resueltos por un tercer juez con la decisión final que se basó en la mayoría de votos. Después de resolver las diferencias, el 67 % (3900) de los pares de oraciones fueron juzgados como semánticamente equivalentes.

Los intentos de imponer directrices estrictas sobre lo que se considera "equivalente" provocó la frustración de los jueces y el colapso del pacto acordado. Las directrices estrictas probaron ser útiles para el caso de la anáfora, donde pronombres como "él" o "ellos" se refieren a entidades mencionadas anteriormente en el texto. Los investigadores se sorprendieron por el alto nivel de acuerdo entre los evaluadores (83 %) a pesar de las directrices relativamente holgadas.

Los investigadores también notaron que el 33 % de los pares de oraciones juzgados como "no equivalentes", a menudo se superponen en el contenido de la información, la redacción y el rango de no tener relación alguna a ser casi equivalentes.

Se pueden extraer algunos puntos relevantes del proceso llevado a cabo para la construcción del MSRPC. En primer lugar, la tasa de acuerdo entre los evaluadores sugieren los valores de la línea base y el límite superior para la clasificación. Debido a que el 67 % de los pares de oraciones se clasificó como positivos (semánticamente equivalentes), una medida básica podría clasificarlos a todos como positivos y tendía una exactitud del 67 %. De igual manera, el acuerdo entre los evaluadores sugiere el 83 % de exactitud como límite superior para cualquier algoritmo de clasificación automática. En segundo lugar, ya que a los pares que se le dio una clasificación negativa están en el rango de ser completamente ajenos a ser casi equivalentes, existe un argumento para no usar estos como datos negativos de entrenamiento [27], aunque muchos de estos pares de oraciones contienen relaciones de paráfrasis interesantes.

A pesar de las intenciones de los autores del corpus, algunos investigadores han encontrado pruebas para insinuar que este no es un recurso rico en relaciones de paráfrasis, al menos comparado con la distribución que se encuentra generalmente en los textos [13]. Específicamente, se encontró que el corpus tiene una tasa de normalizaciones (donde sintagmas nominales son usados en lugar de verbos) considerablemente más baja que la que se encuentran en los escritos generalmente.

El MSRPP se encuentra disponible en dos ficheros de textos que contienen los datos de entrenamiento y los datos de prueba. Existen 1 725 pares de oraciones en el conjunto de datos de prueba y 4 076 en el conjunto de entrenamiento.

2.2. WordNet

WordNet[3] es una base de datos léxica para idioma inglés [28, 29]. Persigue dos objetivos principales: por un lado, construir una combinación de diccionario y tesauro que sea intuitivo y fácil de utilizar; y por otro lado, dar soporte a las tareas de análisis textual y PLN.

La diferencia fundamental de WordNet respecto a otros sistemas con propósitos similares radica en la organización del léxico en torno a cinco categorías: nombres, verbos, adjetivos, adverbios y elementos funcionales. WordNet utiliza los denominados *synonym sets* o *synsets* (conjuntos de elementos léxicos que pueden ser considerados sinónimos entre sí) para la representación de los conceptos, que pueden verse como grupos de elementos de datos semánticamente equivalentes. Un ejemplo de *synset* es {*car*, *auto*, *automobile*, *machine*, *motorcar*}. Al contrario de lo que ocurre con los diccionarios de sinónimos o tesauros tradicionales, un *synset* no tiene una palabra que actúa como identificador del conjunto. El significado del *synset* lo aportan pequeñas definiciones (glosas), que en ocasiones pueden ser ejemplos de oraciones que matizan el significado del concepto. Para el *synset "car"*, la glosa es "*a motor vehicle with four wheels; usually propelled by an internal combustion engine*" y el ejemplo de oración corta: "*he needs a car to get to work*".

Muchas palabras tienen más de un significado (*sense*), que se refieren a los distintos conceptos. Considerando la palabra *"bridge"* o puente en español. El significado más común de esta palabra, en el uso general, es *"a structure that allows people or vehicles to cross an obstacle such as a river, canal, railway, and so on"*. Sin embargo, existen otros significados de "*bridge*", como pueden ser *"the bridge of the nose"* (el puente de la nariz) y *"the bridge card game"* (el juego de cartas puente). Los conceptos están contenidos en diferentes *synset*, lo cual significa que la misma palabra puede aparecer en varios de ellos.

WordNet también proporciona para cada palabra, lo que se ha denominado "cuenta polisémica", una medida del grado en que la palabra se utiliza con cada uno de sus significados. De este modo, si una palabra presenta un valor muy alto para un determinado *synset*, se puede inferir que se trata su acepción más habitual.

Así como una palabra puede aparecer en varios *synset*, también puede aparecer en más de una categoría de las partes de la oración o *part of speech* (sustantivo, verbo, adjetivo, adverbio, etc.). Por tanto, la búsqueda de una palabra en WordNet, listará todos los *synset* que contenga la misma (que corresponden a todos los significados de la palabra), y estos además son agrupados en las categorías de las partes de la oración.

[3] http://wordnet.princeton.edu/wordnet/

Además de la relación léxica de sinonimia, WordNet ofrece otras relaciones semánticas como la antonimia, hiperonimia, hiponimia, meronimia y relaciones morfológicas, que se expresan como punteros entre *synsets*. No obstante, las relaciones se organizan de manera distinta para cada una de las cinco categorías sintácticas en las que se estructura WordNet, aunque todas ellas presentan la relación básica de sinonimia.

WordNet es un recurso muy útil para las investigaciones del PLN. En la tarea de detección de paráfrasis, una de las áreas clave de interés es poder estimar cuantitativamente la similitud de palabras, así los métodos podrían dar una medida más exacta de las similitudes entre textos.

2.3. WordNet::Similarity

El propósito de las métricas que implementa el paquete WordNet::Similarity[4] es dar una medida cuantitativa de la similitud entre dos palabras [15]. Esto es útil para la tarea de detección de paráfrasis, pues si un par de oraciones comparten muchas palabras similares, se podría suponer que este sería un buen indicador de que tienen un significado análogo en su conjunto.

Es importante hacer algunas aclaraciones con respeto a las métricas de similitud basadas en las relaciones de WordNet. En efecto, se está teniendo en cuenta las similitudes entre los conceptos (significado de las palabras) en lugar de palabras, puesto que una palabra puede tener más de un significado. Las métricas de similitud semántica trabajan sobre pares sustantivo-sustantivo y verbo-verbo, pues solo estas estructuras sintácticas se pueden organizar en jerarquías *es un*, así las similitudes solo pueden ser encontradas cuando ambas palabras estén en esta categoría, por ejemplo los sustantivos "perro" y "gato", y los verbos "correr" y "caminar". Estas medidas cuantifican cuán similar es un concepto (*synset*) A a otro B. Por ejemplo, una métrica de esta categoría determinaría que un "gato" es más parecido a un "perro" que a una "silla", puesto que "gato" y "perro" comparten el ancestro "carnívoro" en la jerarquía de sustantivos de WordNet (ver figura 1). Aunque WordNet incluye adjetivos y adverbios, estos no están organizados en una jerarquía *es un*, por lo que las medidas de similitud no pueden ser aplicadas.

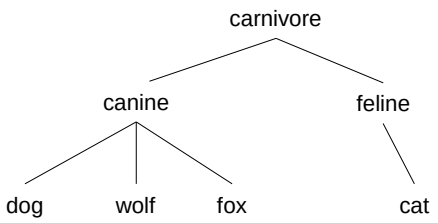

Figura 1: Extracto de la jerarquía de WordNet.

Los conceptos pueden ser relacionados de varias formas, además de la similitud de unos con otros. Incluyen la relaciones *parte de* ("rueda" y "carro"), así como también antítesis ("noche" y "día") y así por el estilo. Las medidas de relación (*relatedness*) hacen uso de este adicional, las relaciones no-jerárquicas en WordNet comprenden las glosas y los *synset*. Como tal, pueden ser aplicados a una gama amplia de pares de conceptos abarcando palabras que son de diferentes partes de la oración, por ejemplo "asesino" y "arma".

[4] http://search.cpan.org/dist/WordNet-Similarity/

Las métricas del recurso WordNet::Similarity se ponen a disposición como un conjunto de módulos del lenguaje de programación Perl. De hecho, el nombre del módulo se podría considerar impreciso, puesto que contiene tres métricas para medir la relación y seis para la similitud.

Se describirán las seis métricas que han sido utilizadas en algunos de los métodos de detección de paráfrasis que se describirán en esta investigación. Una de ellas (*lesk*) es una métrica de relación, mientras que el resto son de similitud.

2.3.1. La métrica *lesk*

La métrica *lesk* utiliza las glosas de las dos palabras y mide la relación como una función de los solapamientos entre estas definiciones [30].

Por ejemplo, los conceptos *"drawing paper"* y *"decal"* tienen las glosas *"paper that is specially prepared for use in drafting*" y *"the art of transferring designs from specially prepared paper to a wood or glass or metal surface"*.

La similitud de dos glosas es calculada por la función $score(G_1, G_2)$, que funciona mediante la búsqueda de la secuencia de superposición de palabras más larga entre las sentencias que no comienzan o terminan con una palabra funcional (pronombre, preposición, artículo o conjunción). En los ejemplos expuestos anteriormente podría ser *"specially prepared"*. La puntuación dada a un solapamiento es n^2 donde n es la longitud de la secuencia, por lo que secuencia de dos palabras tendrá una puntuación de 4. El algoritmo entonces elimina esta secuencia de ambos textos y encuentra la secuencia más larga de lo que queda, y acumula la puntuación. El procedimiento continúa hasta que no queden más solapamientos.

La métrica *lesk* también toma en cuenta todos los conceptos que están directamente relacionados con el concepto en cuestión a través de las relaciones explícitas en WordNet (hiperónimos, hipónimos, entre otros). $RELS$ es definido como un subconjunto de relaciones en WordNet. Por cada relación, una función es definida con el mismo nombre, la cual retorna la glosa de los *synset* relacionados con el *synset* para esa relación. Si más de un *synset* es devuelto, las glosas son concatenadas y retornadas. Por ejemplo $hype(A)$ retornará la glosa del hiperónimo de A.

$RELPAIRS$ es definido como un conjunto reflexivo cerrado de pares de oraciones:

$$RELPAIRS = \{(R_1, R_2) \in RELS| \text{ si } (R_1, R_2) \in RELPAIRS \text{ entonces } (R_2, R_1) \in RELPAIRS\} \quad (1)$$

La restricción reflexiva es impuesta para asegurar que se cumpla que:
$relatedness(A, B) = relatedness(B, A)$.

Finamente, la relación de dos *synset* A y B está dado por:

$$relatedness(A, B) = \sum_{\forall (R_1, R_2) \in RELPAIRS} score(R_1(A), R_2(B)) \quad (2)$$

Por ejemplo, si el conjunto de relaciones $RELS = \{gloss, hypo, hype\}$ y
$RELPAIRS = \{(gloss, gloss), (hypo, hypo), (hype, hype), (gloss, hype), (hype, gloss)\}$ entonces:

$$relatedness(A,B) = score(gloss(A), gloss(B)) + score(hypo(A) + hypo(B)) + score(hype(A) + hype(B)) + score(gloss(A) + hype(B)) + score(hype(A) + gloss(B)) \quad (3)$$

2.3.2. La métrica *lch*

La métrica *lch* determina la similitud de dos nodos mediante la búsqueda de la longitud del camino entre ellos en una jerarquía *es un* [31]. La similitud es calculada como:

$$sim_{lch} = -\log \frac{N_p}{2D} \quad (4)$$

donde N_p es la distancia entre los nodos y D es la profundidad máxima en la taxonomía *es un.*

2.3.3. Ancestro común más específico y contenido informativo

El resto de los métodos usan el concepto del LCS (*least common subsumers* o ancestro común más específico) y el IC (*information content* o contenido informativo).

Dado dos conceptos C_1 y C_2 en una jerarquía *es un*, el LCS es definido como el nodo más específico que ambos comparten como ancestro [32]. Por ejemplo si C_1 es "perro" y C_2 es "gato" , entonces el LCS podría ser "carnívoro" (ver figura 2) .

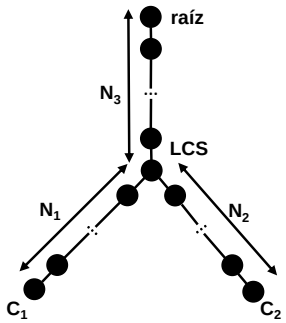

Figura 2: Ilustración de el LCS de dos conceptos C_1 y C_2. Tomada de [33].

El IC de un nodo es una estimación de cuán informativo es el concepto [34]. Frecuentemente, ocurre que los conceptos son estimados con un bajo IC, raramente estos son definidos teniendo en cuenta un alto IC.

Formalmente, el IC de un concepto c es definido como:

$$IC = -\log P(c) \quad (5)$$

donde $P(c)$ es la probabilidad de encontrar a $c$en un corpus extenso.

2.3.4. La métrica *wup*

La métrica *wup* calcula la similitud de dos nodos en función de la trayectoria del LCS de estos [32].

La similitud entre dos nodos C_1 y C_2 es:

$$sim_{wup} = \frac{2 \cdot N_3}{N_1 + N_2 + 2 \cdot N_3} \tag{6}$$

donde N_1 es la cantidad de nodos en la trayectoria desde el LCS a C_1, N_2 es la cantidad nodos en el camino de LCS a C_2, y N_3 es el número de nodos en la trayectoria desde nodo raíz al LCS. Esto es mostrado en la figura 2.

2.3.5. La métrica *resnik*

La métrica *resnik* usa el IC del LCS de dos conceptos [34]. La idea es que la cantidad de información compartida entre dos conceptos indicará el grado de similitud entre estos, y la cantidad de información que estos comparten está indicada por el IC de sus LCS.

Formalmente:

$$sim_{res} = IC(LCS) \tag{7}$$

2.3.6. La métrica *lin*

La métrica *lin* está construida sobre la base de la medida *resnik* por normalización, utiliza el IC de los dos nodos [35].

$$sim_{lin} = \frac{2 \cdot IC(LCS)}{IC(N_1) \cdot IC(N_2)} \tag{8}$$

donde el IC se definió en la ecuación 5

2.3.7. La métrica *jcn*

La métrica *jcn* también usa la idea del IC [36].

$$sim_{jcn} = \frac{1}{IC(N_1) + IC(N_2) + 2 \cdot IC(LCS)} \tag{9}$$

WordNet::Similarity puede utilizarse a través de una interfaz de línea de comandos proporcionada por el programa de utilidad *similarity.pl*. Además, existe una interfaz web[5] que permite a los usuarios ejecutar las métricas interactivamente. El módulo también puede ser incluido en un *script* en Perl para luego invocar sus funcionalidades. En todos los casos, se debe especificar la métrica de similitud y los conceptos sobre los que se desea realizar el cálculo.

[5] http://marimba.d.umn.edu/cgi-bin/similarity/similarity.cgi

3. Trabajos previos en la detección de paráfrasis

En esta sección se examinan algunos de los métodos de detección de paráfrasis más relevantes. Los algoritmos existentes para la detección de paráfrasis se pueden dividir en dos categorías diferentes, en dependencia de las técnicas utilizadas. Por un lado, los acercamientos que utilizan una función de similitud para decidir si un par de oraciones son paráfrasis o no, y por el otro, los que emplean aprendizaje automático supervisado para combinar varias características extraídas de los pares de textos.

Para evaluar la eficacia de los clasificadores (algoritmos, métodos o sistemas) de detección de paráfrasis se han empleado un conjunto de métricas disponibles de la clasificación automática. Para realizar esta evaluación se debe cotejar la predicción del clasificador con la clase real de los objetos que se van a evaluar. Dependiendo de cuál sea el interés particular que tenga la clasificación, esta puede ser binaria con las categorías de "paráfrasis" o "no-paráfrasis", o multiclase donde cada categoría especifica un tipo de paráfrasis.

Para definir las métricas se debe tener en cuenta el conjunto de TP (*true positives*), TN (*true negatives*), FP (*false positives*), y FN (*false negatives*) como en la matriz de confusión mostrada en la tabla 1. En las columnas se tiene la clase que el clasificador ha predicho y, en las filas, las clases a las que realmente pertenecen los objetos. En esta matriz de confusión se registra el número de aciertos y errores (por cada clase) que tuvo el clasificador al ser evaluado con un conjunto de objetos particular.

		Predicción	
		Positiva	Negativa
Real	Positiva	**TP**	**FN**
	Negativa	**FP**	**TN**

Tabla 1: Matriz de confusión para clasificación binaria.

Nótese que TP y TN son los aciertos del sistema; FP y FN representan los errores. Usando estas categorías se puede definir las métricas estándares para medir el rendimiento del sistema, precisión, recuerdo (*recall*), medida-F y exactitud (*accuracy*) como sigue:

$$prec = \frac{TP}{TP + FP} \tag{10}$$

La precisión significa el número pares de oraciones correctamente identificados como pertenecientes a una clase ("paráfrasis" o "no-paráfrasis"), normalizado por el número total de pares correcta o incorrectamente identificados como pertenecientes a esa clase.

$$rec = \frac{TP}{TP + FN} \tag{11}$$

El recuerdo representa el número de pares de oraciones correctamente identificadas como pertenecientes a una clase, normalizado por el número total de pares correctamente identificados y aquellos que no han sido identificados como pertenecientes a esa clase, pero que deberían haber sido.

$$medida\text{-F} = 2 \cdot \frac{prec \cdot rec}{prec + rec} \tag{12}$$

La medida-F es la media armónica entre la precisión y el recuerdo.

$$exact = \frac{TP + TN}{TP + FP + FN + TN} \tag{13}$$

La exactitud da la proporción entre el número total de pares de oraciones correctamente identificados sobre todo el conjunto.

Todos los métodos que se describirán hacen uso del MSRPC para la evaluación, lo que da la posibilidad de establecer comparaciones entre ellos. Sin embargo, también tiene algunas limitaciones, como por ejemplo, algunos investigadores no lo consideran una fuente rica en relaciones de paráfrasis [13].

3.1. Métodos basados en funciones de similitud

Las funciones de similitud sirven para calcular qué tan similares son dos textos y se materializan en algoritmos de computación que realizan esta tarea. Para la detección de paráfrasis se han desarrollado múltiples funciones de similitud, unas han sido más eficaces que otras ante la presencia de ejemplares complejos. Dado un par de oraciones, una función se similitud devuelve un valor real, generalmente en el intervalo $[0; 1]$. El valor 1 significa que las oraciones son idénticas, el 0 que son completamente diferentes y un número intermedio indica el nivel de similitud.

Los acercamientos que emplean una función de similitud obtienen esta a partir de comparaciones léxicas, sintácticas y semánticas de los textos. Para decidir si un par de oraciones son paráfrasis o no, hacen uso de un umbral (valor numérico). Por lo general, los valores por encima del umbral se consideran paráfrasis (ver figura 3).

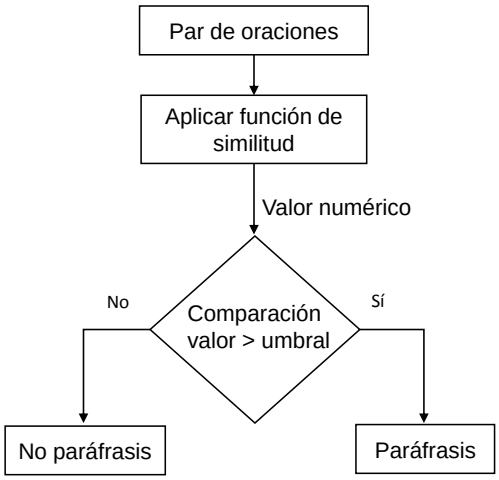

Figura 3: Esquema general de los métodos basados en funciones de similitud.

La definición del umbral es polémico, ya que casos que son paráfrasis se pueden pasar por alto y viceversa. A continuación se describen tres de los trabajos que enfrentan la detección de paráfrasis desde este punto

de vista.

3.1.1. Métricas basadas en corpus y en conocimiento

Mihalcea y otros [37] presentan un método para encontrar la similitud semántica de segmentos cortos de textos a partir de medidas basadas en corpus y en conocimiento.

La motivación del método es que mediante el uso de una similitud entre palabras más sofisticado, los resultados del cálculo de similitudes entre oraciones podrían ser más exactos. La idea es ser capaz de capturar sinónimos cercanos, tales como "perro" y "canino", "herir" y "lastimar", así como coincidencias léxicas exactas.

Para encontrar la similitud de dos fragmentos de textos T_1 y T_2 se usa la función de puntuación siguiente:

$$sim(T_1, T_2) = \frac{1}{2}\left(\frac{\sum_{w\in\{T_1\}}(maxSim(w, T_2)\cdot idf(w))}{\sum_{w\in\{T_1\}} idf(w)} + \frac{\sum_{w\in\{T_2\}}(maxSim(w, T_1)\cdot idf(w))}{\sum_{w\in\{T_2\}} idf(w)}\right) \tag{14}$$

donde $maxSim(w, T_i)$ es la máxima puntuación de similitud encontrada entre la palabra w y las palabras en T_i, de acuerdo con la medida de similitud palabra a palabra empleada, y $idf(w)$ es la frecuencia inversa de documento[6] [38] de la palabra. El valor *idf* es calculado usando el Corpus Nacional Británico [39].

La puntuación de similitud es un valor entre 0 y 1, el valor 0 indica que no existe superposición semántica entre los segmentos de texto y el 1 que son idénticos. Para cada par de textos candidatos, primero se calcula la similitud semántica usando la ecuación 14, luego los textos son declarados paráfrasis si el nivel de similitud excede el umbral 0,5.

En este acercamiento se prueban diferentes medidas de similitud palabra a palabra, basadas en corpus y en conocimiento. Los medidas basadas en corpus se centran únicamente en la información obtenida a partir de extensos ejemplares de estos para estimar la similitud semántica de las palabras. Entre las medidas cabe destacar, la información mutua puntual [40]:

$$\text{PMI-IR}(w_1, w_2) = \log_2 \frac{p(w_1 \& w_2)}{p(w_1)\cdot p(w_2)} \tag{15}$$

donde $p(w)$ es la frecuencia de aparición de la palabra w en el corpus.

Los acercamientos basados en el conocimiento usan las métricas de similitud basadas en WordNet, descritas en la sección 2.3. Los mejores resultados fueron obtenidos a partir de una combinación de estas métricas. Logró una exactitud del 70.3 % una medida-F del 81.3 % sobre el MSRPC [37].

3.1.2. Similitud del coseno

La métrica de similitud del coseno fue originalmente usada en aplicaciones de recuperación de información para encontrar la semejanza entre una consulta y un documento. Los documentos y las consultas son

[6] Medida numérica que expresa cuán relevante es una palabra para un documento en una colección.

representados por vectores. La métrica es una manera de calcular la similitud entre dos vectores. La forma más simple de los vectores es tomar cada elemento $d_i = 1$ si la palabra i-ésima aparece en el documento o tomar $d_i = 0$ en caso contrario.

La forma general de esta métrica es como sigue:

$$cos(q, d) = \frac{\sum_{i=1}^{n} d_i q_i}{\left(\sum_{i=1}^{n} (d_i)^2 \cdot \sum_{i=1}^{n} (q_i)^2\right)^{\frac{1}{2}}} \tag{16}$$

donde q es el vector consulta y d es el vector documento.

Uno de los primeros usos registrados de esta métrica fue en el sistema de recuperación de información SMART, desarrollado por IBM (*International Business Machines*) [41]. La ponderación *tf* (*term frecuency*) es ligeramente más refinada, donde cada elemento d_i se define en función de la frecuencia de la palabra i en el documento d. Trabajos posteriores intentaron incluir una medida de especificidad de la palabra (con palabras más específicas por ser un mejor indicador del contenido del documento que las palabras comunes). La ponderación *idf* modela esto en función de la cantidad de documentos donde aparece la palabra y la cantidad total de documentos. Así la combinación de ponderaciones da la ponderación *tf-idf* [38]:

$$w = \overbrace{(1 + \log(tf_{ij}))}^{tf} \cdot \overbrace{\frac{\log N}{df_i}}^{idf} \tag{17}$$

donde:

- tf_{ij} es el número de veces que la palabra i aparece en el documento j.
- N es la cantidad total de documentos en la colección.
- df_i es el número de documentos en que la palabra i aparece.

Algunas variaciones del método de la similitud del coseno han sido estudiadas para la detección de paráfrasis. El factor variable en estos métodos es la función de ponderación aplicada a cada palabra en el vector [33]:

- cosSim: métrica de similitud del coseno no ponderada.
- cosSimTF: métrica de similitud del coseno ponderada *tf*
- cosSimTFIDF: métrica de similitud del coseno ponderada *tf-idf*.

En los estudios anteriores, se genera una puntuación al comparar un par de oraciones. Para decidir si el par es una paráfrasis se usó un umbral. El umbral se estimó con la parte de entrenamiento del corpus y el algoritmo de aprendizaje automático basado en el árbol de decisión J48 [42]. Los resultados se muestran en la tabla 2, la métrica cosSim aportó mejores resultados que las otras basadas en la similitud del coseno.

Métrica	exactitud	precisión	recuerdo	medida-F
cosSim	**72.7**	**72.6**	94.7	**82.2**
cosSimTF	71.9	72.2	94.0	81.7
cosSimTFIDF	69.0	68.6	**98.3**	80.8

Tabla 2: Evaluación de los métodos basados en la similitud del coseno [33].

3.1.3. Matriz de similitud semántica

El acercamiento propuesto por Mihalcea y otros [37], usó varias métricas de similitud palabra a palabra para encontrar la similitud de oraciones. Para cada palabra en la oración 1, se encuentra la palabra más semejante en la oración 2, de acuerdo a la métrica seleccionada, las puntuaciones máximas se van sumando y asimismo se realiza el procedimiento para las palabras de la oración 2. Una desventaja de este método es que, debido a que solo se encuentra la palabra similar máxima, el resto de las puntuaciones de similitud no se tienen en cuenta para medida final.

El método propuesto por Stevenson y Greenwood [43] para la tarea de extracción de información, aborda el problema mediante una matriz de similitud para calcular la similitud entre los dos vectores de documentos. Conceptualmente, este método es similar al propuesto por Mihalcea y otros [37] (ecuación 14), pero se tienen en cuenta todas las puntuaciones de similitud entre todos los pares de palabras en las oraciones.

Formalmente, si $\vec{a}$ y $\vec{b}$ representan vectores de documentos (con elementos iguales a 1 si una palabra está presente y 0 del lo contrario), entonces la similitud entre $\vec{a}$ y $\vec{b}$ es calculada con la siguiente fórmula:

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot W \cdot \vec{b^T}}{|\vec{a}| \cdot |\vec{b}|} \tag{18}$$

donde W es la matriz de similitud semántica, que contiene la información de la similitud de pares de palabras. Formalmente, cada elemento w_{ij} en W es la similitud de las palabras p_i y p_j de acuerdo con alguna medida de similitud palabra a palabra. Si esta medida es simétrica entonces $w_{ij} = w_{ji}$, es decir, la matriz es simétrica. Los elementos de la diagonal representan las autosimilitudes y por consiguiente deben tener los mayores valores.

La métrica de similitud palabra a palabra puede ser cualquiera de las del paquete WordNet::Similarity (como en [37]). La medida utilizada en Stevenson y Greenwood [43] fue la métrica *jcn*, después de que los experimentos demostraron que era la más efectiva.

La matriz de similitud semántica fue implementada por Fernando y Stevenson [17] para la detección de paráfrasis. Una cuestión clave en la implementación es que las métricas de WorNet::Similarity requieren la especificación del significado de cada palabra, debido a que en WordNet una palabra tiene más de un significado. Los experimentos iniciales 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, pero el procedimiento resultó muy lento, ya que muchas palabras tienen más de un sentido. Por otra parte, los resultados no mejoraron considerablemente con respecto a los primeros experimentos. Por tanto, solo el primer significado de las palabras fue utilizado finalmente para el cálculo de la similitud.

Los experimentos tuvieron en cuenta las siguientes matrices de similitud [17]:

- matrixJcn: matriz de similitud con la métrica *jcn*.
- matrixLch: matriz de similitud con la métrica *lch*.
- matrixLesk: matriz de similitud con la métrica *lesk*.
- matrixLin: matriz de similitud con la métrica *lin*
- matrixWup: matriz de similitud con la métrica *wup*.
- matrixRes: matriz de similitud con *resnik*.

Cuando se compara un par de oraciones se genera una puntuación que significa el grado de similitud. Luego la puntuación se compara con un umbral y los pares de oraciones con puntuaciones superiores a este valor son considerados paráfrasis. Para estimar el umbral también se utiliza la parte de entrenamiento del corpus y el algoritmo de aprendizaje automático basado en el árbol de decisión J48 [42]. Los resultados de los los experimentos se muestran en la tabla 3, el que dio mejores resultados fue matrixJcn.

Métrica	exactitud	precisión	recuerdo	medida-F
matrixJcn	**74.1**	**75.2**	91.3	**82.4**
matrixLch	73.9	74.8	91.6	82.3
matrixLesk	72.9	73.5	**92.6**	82.0
matrixLin	73.7	74.2	92.5	**82.4**
matrixWup	72.2	73.8	90.4	81.2
matrixRes	71.6	75.2	85.4	80.0

Tabla 3: Evaluación de los métodos basados en matriz de similitud [17].

3.2. Métodos basados en aprendizaje supervisado

Los métodos basados en aprendizaje supervisado predicen la clase correspondiente a un par de oraciones a partir de un modelo aprendido de un corpus de entrenamiento. El modelo establece una correspondencia entre las entradas y las salidas deseadas, y es generado a partir de un proceso de extracción de características, que pueden ser léxicas, sintácticas o semánticas. Se ha tratado la detección de paráfrasis como un problema de clasificación, donde los algoritmos aprenden con ejemplos usando datos que han sido organizados en clases en forma manual o a través de algún proceso automático. A través del proceso de entrenamiento, los algoritmos de clasificación determinan las propiedades o características que indican que un objeto pertenece a una clase dada.

Los algoritmos cuando han sido entrenados pueden clasificar datos que no tienen todavía etiquetas. En la detección de paráfrasis la clasificación es binaria, se asigna a un par de oraciones una etiqueta relacionada con una categoría (paráfrasis o no paráfrasis).

En la figura 4 se muestra el esquema general de los métodos de detección de paráfrasis que utilizan aprendizaje supervisado. Se parte de un conjunto de pares de oraciones correctamente clasificados, normalmente por humanos. Luego continúa un proceso de extracción de características de ambos textos que le permitan a un algoritmos de aprendizaje generar un modelo. Una vez generado el modelo, al introducirle al método

un nuevo par de oraciones, este es capaz de predecir la clase correspondiente luego de realizar la extracción de características.

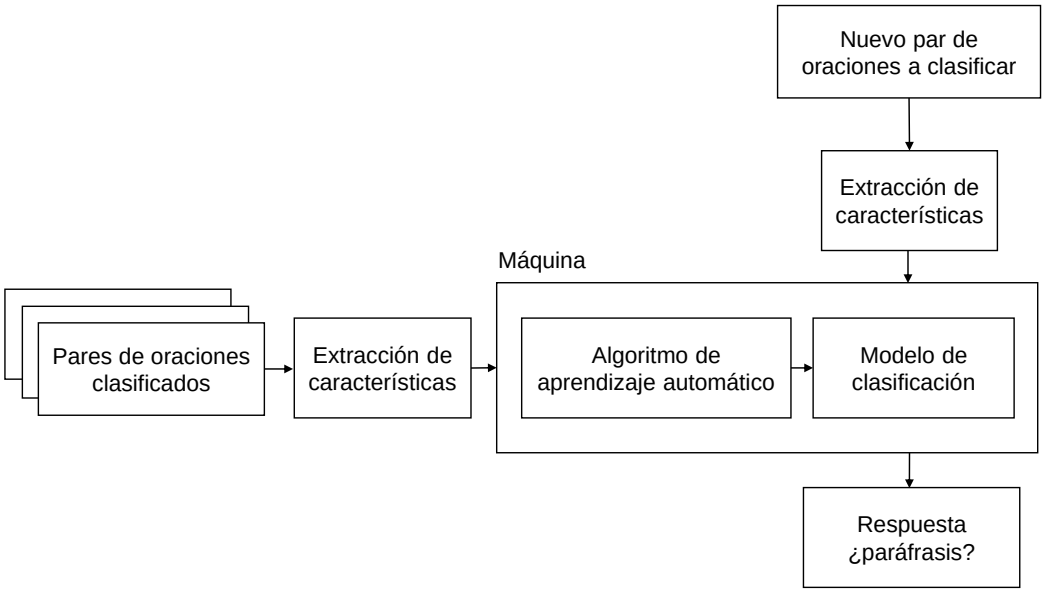

Figura 4: Esquema general de los métodos de detección de paráfrasis con aprendizaje supervisado.

Este grupo de métodos ha resultado ser más exacto, a pesar de la dificultad que conlleva construir un corpus que incluya la mayor cantidad de tipos de paráfrasis posibles para un idioma. A continuación se describen algunos de los acercamientos más representativos en el área, que resuelven la detección de paráfrasis desde el punto de vista del aprendizaje supervisado.

3.2.1. Clasificación por el significado de disimilitud

El acercamiento desarrollado por Qiu y otro*s* [14] usa un proceso de dos fases. Primero, se extraen de ambas oraciones las unidades de contenido semántico clave, también llamadas minutas de información (*information nuggets*). En la segunda fase del método, se emparejan las minutas de información de cada oración. Si después del proceso anterior, alguna minuta permanece desemparejada su contenido es analizado. Si las oraciones no contienen minutas desemparejadas o estas son insignificantes en su totalidad, entonces el método emite una clasificación positiva de paráfrasis.

En trabajos previos [44] se usan palabras representativas como minutas de información para la elaboración de resúmenes. En este método, se usan las tuplas de predicado-argumento como minutas de información. Una tupla de predicado-argumento es una representación estructurada de un verbo predicado junto con sus argumentos. Para la oración "*Richard Miller was injured by a young man*", la tupla de predicado-argumento es:

target (predicate)	:	injure
arg0	:	a young man
arg1	:	Richard Miller

Las tuplas de predicado-argumento se extraen usando un analizador sintáctico [45] y el etiquetador de rol semántico ASSERT [46].

El emparejamiento de tuplas entre oraciones se hace por comparación directa con el objetivo y los argumentos reducidos a sus palabras sintácticas claves. Un factor de peso se aplica a la similitud de los objetivos. Las tuplas con altas similitudes son sucesivamente emparejadas hasta que fallen por tener una similitud por encima de un cierto umbral, o no existan más. Si quedan tuplas desemparejadas, entonces se aplican heurísticas adicionales para intentar emparejarlas. Las heurísticas incluyen el manejo de la forma copular y sintagmas nominales. Las oraciones *"Microsoft rose 50 cents"* y *"Microsoft was up 50 cents"* pueden considerarse que tienen el mismo significado; la segunda oración se encuentra en forma copular. En las oraciones *"Blamed for frequent attacks on soldiers"* y *"Blamed for frequently attacking soldiers"*; la primera oración contiene la información acerca de los ataques en un sintagma nominal, lo que no es detectado por el etiquetador de rol semántico.

Cualquiera de las tuplas desemparejadas restantes son partes diferentes de la oración, por lo que el siguiente paso es estimar su importancia. Para esta tarea se utiliza aprendizaje automático y se experimenta con varios rasgos para el clasificador máquinas de soporte vectorial, así se descubre que dos de ellos fueron especialmente útiles. El primero es la ruta del árbol de análisis sintáctico, la cual es la lista de los componentes sintácticos que unen el objetivo desemparejado con el más cercano emparejado, con la idea de que este camino indica el tipo de relación que tiene el objetivo desemparejado con el resto de la oración. El segundo es el predicado, que es simplemente el texto del verbo objetivo.

Como no existían corpus de entrenamiento para el significado de predicado en ese entonces, se desarrolla un nuevo método que hizo uso de un corpus etiquetado. Después de cada emparejamiento ávido sobre los datos de entrenamiento, cada par de oraciones se clasifica en base a dos dimensiones: si el par es una paráfrasis y la fuente de las tuplas no emparejadas. Esto da lugar a cuatro clases de pares de oraciones, de los cuales solo dos se usan como datos de entrenamiento:

- PS: cuando el par de oraciones es una paráfrasis y las tuplas no emparejadas provienen de una sola oración.
- NS: cuando el par de oraciones no es una paráfrasis y solamente existe una tupla no emparejada.

Los pares de la clase PS se utilizan como tuplas insignificantes, ya que las tuplas no emparejadas no niegan la relación de paráfrasis. Los de clase NS se emplean como tuplas significativas, pues la única tupla no emparejada debe ser responsable de la clasificación negativa de paráfrasis.

Este acercamiento dio buenos resultados para ese entonces sobre el MSRPC, logró un 72.9 % de exactitud y 81.6 % de medida-F [14].

3.2.2. Canonicalización de textos

La idea principal del acercamiento propuesto por Zhang y Patrick [13] es la creación de 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 fueron usadas un número limitado de técnicas de canonicalización. Estas incluyen 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"*.

Una vez que el texto es llevado a la forma canónica, se usan técnicas simples de coincidencia léxica para comparar la transformación del mismo, tales como la subsecuencia común más larga, la distancia de edición y una medida de precisión basada en n-gramas. Estas características léxicas simples son utilizadas para construir un modulo de aprendizaje automático supervisado, basado en árbol de decisión, que clasifica el par de oraciones.

Este método logró el mejor rendimiento general utilizando la conversión de voz pasiva a activa, con una exactitud del 71.9 % y una medida-F del 80.7 % sobre el conjunto de pruebas del MSRPC [13].

3.2.3. Uso de información léxica y semántica

La novedad del acercamiento propuesto por Kozareva y Montoyo [22] consiste en los experimentos realizados. Se explora el poder de discriminación de características léxicas y semánticas para identificar paráfrasis. Además, se estudia el comportamiento de tres clasificadores de aprendizaje automático: máquinas de soporte vectorial, k-vecinos más cercanos y máxima entropía. Con el objetivo de mejorar el rendimiento del sistema de detección de paráfrasis, también se estudia el impacto de las características léxicas y semánticas en su conjunto, y los resultados de un ensamble de voto. Para el esquema de votación se toman las salidas de los tres clasificadores antes mencionados y se clasifica de acuerdo a la clase mayoritaria. El ensamble de voto obtuvo los mejores resultados con una exactitud del 76.6 % y una medida-F del 79.5 % sobre el MSRPC [22].

Las características léxicas que se tuvieron en cuenta en este acercamiento están relacionada con la proporción de n-gramas consecutivos entre los dos textos, llamados *skip-grams*. El objetivo de los *skip-grams* es buscar secuencias no consecutivas (que pueden tener intervalos por medio) entre las dos oraciones. Además, de la subsecuencia común más larga de palabras entre las dos oraciones.

Con el fin de obtener información semántica, primero se identifican las partes de la oración con la herramienta TreeTagger[7]. Las características de similitud de palabras necesitan conocimiento extrínseco, que puede ser extraído de un corpus extenso o del repositorio WordNet. Para establecer la similitud entre los sustantivos y los verbos se utiliza la métrica *lin* del paquete WordNet::Similarity. Se introduce una medida de similitud semántica entre sustantivos y verbos mediante la fórmula:

$$similarity_{lin} = \frac{\sum_{i=1}^{n} sim(T_1, T_2)_{lin}}{n} \tag{19}$$

La fórmula indica la proporción de similitud de sustantivos y verbos con respecto a la similitud máxima para las oraciones T_1 y T_2. Los valores de $sim(T_1, T_2)_{lin}$ son la similitud de los sustantivos y los verbos de los textos T_1 y T_2 de acuerdo con la ecuación 8.

[7] http://www.cis.uni-muenchen.de/∼schmid/tools/TreeTagger/

También se le da tratamiento a los números cardinales. Por ejemplo, "más de 24" se toma como 25, "menos de 24" se toma como 23 y "veinticinco" se transforma en 25.

3.2.4. Combinación de medidas de similitud

Este acercamiento presenta tres métodos para la tarea de detección de paráfrasis. Utiliza un clasificador automático de máxima entropía para aprender cómo combinar un conjunto de medidas de similitud de cadenas. Los métodos son los siguientes [18]:

- INIT. Este método tienen en cuenta 9 medidas de similitud: distancia de Levenshtein, distancia de Jaro-Winkler, distancia de Manhattan, distancia Euclidiana, similitud del coseno, distancia de n-gramas (con $n = 3$), coeficiente de coincidencia, coeficiente de Dice y el coeficiente de Jaccard.
- INIT+WN. Este método trata las palabras que son sinónimos como idénticas, es decir, constituye una mejora de INIT. Explota la base de datos léxica WordNet.
- INIT+WN+DEP. Las características en los dos métodos anteriores solo operan en el nivel léxico. Este método adiciona características que operan sobre las relaciones gramaticales (dependencias). Se adicionan tres medidas para calcular la similitud en el nivel de relaciones gramaticales, llamadas recuerdo de dependencia de $S_1(R_1)$, recuerdo de dependencia de S_2 (R_2) y su medida-F (F_{R_1,R_2}), definidas como sigue:

$$R_1 = \frac{|dependencias\, comunes|}{|dependencias\, de\, S_1|} \tag{20}$$

$$R_2 = \frac{|dependencias\, comunes|}{|dependencias\, de\, S_2|} \tag{21}$$

$$F_{R_1,R_2} = \frac{2 \cdot R_1 \cdot R_2}{R_1 + R_2} \tag{22}$$

El método INIT+WN+DEP al incorporar más información al modelo, obtuvo resultados más exactos (76.1 % de exactitud y 82.8 % de medida-F). De igual manera, INIT+WN fue superior a INIT.

3.2.5. Empleo de métricas de traducción automática

Las métricas desarrolladas para la traducción automática también han sido aplicadas a la detección de paráfrasis [23]. Se aborda el problema desde un punto de vista supervisado. El método utiliza tres clasificadores y mediante técnicas de ensamble obtiene el veredicto final con una exactitud del 77.4 % y una medida-F del 84.1 %. Los clasificadores utilizados son: regresión logística, máquinas de soporte vectorial y k-vecinos más cercanos.

Los atributos utilizados son técnicas de evaluación de traducción automática:

1. BLEU [47]. Es la métrica más común para evaluar traducciones automáticas. Está basada en el traslape de n-gramas entre ambos textos a comparar con diferentes valores de n.

2. NIST [48]. Es una variante de BLEU, también funciona con n-gramas, pero obtiene un promedio aritmético de n-gramas compartidos entre el total y luego uno geométrico para finalmente combinar los resultados.

3. TER [49]. Es una distancia de edición que retorna el mínimo número de operaciones necesarias para hacer idéntica la traducción a evaluar y la traducción ideal. Las operaciones permitidas por esta distancia de edición son: insertar, eliminar y sustituir.

4. TERp [50]. Es una extensión de la medida TER. Agrega operaciones basadas en *stemming*[8], sinonimia y paráfrasis.

5. METEOR [51]. Esta métrica está basada en n-gramas al igual que BLEU, pero toma en cuenta tanto precisión como recuerdo, a diferencia de BLEU que solo toma en cuenta precisión. También lleva a cabo un preprocesamiento donde utiliza *stemming*, sinonimia (a través de Wordnet) y paráfrasis.

6. SEPIA [52]. En este trabajo, el autor propone utilizar n-gramas estructurales, los cuales son capaces de capturar mayor información que los n-gramas tradicionales. Con el conjunto de n-gramas estructurales este método funciona similar a BLEU.

7. BADGER [53]. Es una métrica independiente del lenguaje, basada en la compresión y la teoría de la información. Se calcula una distancia de compresión entre las dos frases que utiliza la Transformación Burrows Wheeler (BWT). La BWT permite tomar en cuenta los contextos de frases comunes sin límites de tamaño.

8. MAXSIM [54]. Esta métrica trata el problema como cotejo de grafos bipartitos emparejando cada palabra de un texto con la más similar en el otro texto.

En los experimentos también se comprobó el rendimiento de cada una de las métricas anteriores de forma individual y TREp resultó ser la de mejores resultados con una exactitud del 74.3 %. La métrica TREp mostró un rendimiento mayor que todos los acercamientos basados en funciones de similitud examinados en esta investigación [37, 33, 17], incluso mejor que algunos métodos que utilizan aprendizaje automático supervisado [14, 13]. De forma general, la técnica de ensamble aplicada con todas las métricas provenientes de la traducción automática citadas anteriormente, constituye uno de los mejores aportes a la detección de paráfrasis [23].

4. Conclusiones

Con la presente investigacion se puedo constatar el conjunto de datos MSRPC ha sido el corpus más empleado para evaluar los métodos de detección de paráfrasis, aunque algunos autores consideran que no es una fuente rica en relaciones de paráfrasis [13]. La base de datos léxica WordNet y las métricas del paquete WordNet::Similarity, han sido ampliamente utilizadas en métodos previos de detección de paráfrasis y sería oportuno valorarlos para el desarrollo de un nuevo acercamiento.

[8] *Stemming*: es un método para reducir una palabra a su raíz (*stem*) o lema.

El estudio de los acercamientos previos, arrojó que el problema se ha enfrentado desde dos puntos de vista generales: a través de una función de similitud o utilizando aprendizaje automático supervisado para combinar características extraídas de los textos. Los métodos basados en aprendizaje automático han sido más exactos y dan más oportunidades para experimentar con varias combinaciones de características extraídas de los textos.

Referencias

[1] Alberto Barrón-Cedeño, Marta Vila, and Paolo Rosso. Detección automática de plagio: de la copia exacta a la paráfrasis. In *Panorama actual de la lingüística forense en el ámbito legal y policial: teoría y práctica. Jornadas (in)formativas de lingüística forense*, pages 76–96, Madrid, España, 2010.

[2] Regina Barzilay and Kathleen R. McKeown. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 50–57. Association for Computational Linguistics, 2001.

[3] Vasile Rus, Philip M McCarthy, Arthur C Graesser, and Danielle S McNamara. Identification of sentence-to-sentence relations using a textual entailer. *Research on Language and Computation*, 7(2-4):209–229, 2009.

[4] Rahul Bhagat. *Learning paraphrases from text*. University of Southern California, 2009.

[5] Marta Vila Rigat, M Antònia Martí, and Horacio Rodríguez. Paraphrase concept and typology. A linguistically based and computationally oriented approach. *Procesamiento del Lenguaje Natural*, (46):83–90, 2011.

[6] Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. Paraphrase generation as monolingual translation: Data and evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 203–207. Association for Computational Linguistics, 2010.

[7] Rahul Bhagat, Eduard Hovy, and Siddharth Patwardhan. Acquiring paraphrases from text corpora. In *Proceedings of the fifth international conference on Knowledge capture*, pages 161–168. ACM, 2009.

[8] Liang Zhou, Chin-Yew Lin, and Eduard Hovy. Re-evaluating machine translation results with paraphrase support. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 77–84. Association for Computational Linguistics, 2006.

[9] Regina Barzilay and Kathleen R McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

[10] Alberto Barrón-Cedeño, Marta Vila, M Antònia Martí, and Paolo Rosso. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. *Computational Linguistics*, 39(4):917–947, 2013.

[11] Fabio Rinaldi, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Mollá. Exploiting paraphrases in a question answering system. In *Proceedings of the second international workshop on*

Paraphrasing, volume 16, pages 25–32, Sapporo, Japan, 2003. Association for Computational Linguistics.

[12] Paul Clough, Robert Gaizauskas, Scott Piao, and Yorick Wilks. METER: MEasuring TExt Reuse. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 152–159. Association for Computational Linguistics, 2002.

[13] Yitao Zhang and Jon Patrick. Paraphrase identification by text canonicalization. In *Proceedings of the Australasian language technology workshop*, pages 160–166, Sydney, Australia, 2005.

[14] Long Qiu, Min-Yen Kan, and Tat-Seng Chua. Paraphrase recognition via dissimilarity significance classification. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 18–26. Association for Computational Linguistics, 2006.

[15] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. WordNet::Similarity - measuring the relatedness of concepts. In *Association for the Advancement of Artificial Intelligence*, pages 1024–1025, 2004.

[16] Courtney Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, pages 13–18. Association for Computational Linguistics, 2005.

[17] Samuel Fernando and Mark Stevenson. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*, pages 45–52. Citeseer, 2008.

[18] Prodromos Malakasiotis. Paraphrase recognition using machine learning to combine similarity measures. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 27–35. Association for Computational Linguistics, 2009.

[19] Dipanjan Das and Noah A Smith. Paraphrase identification as probabilistic quasi-synchronous recognition. In *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*, volume 1, pages 468–476. Association for Computational Linguistics, 2009.

[20] João Cordeiro, Gaël Dias, and Pavel Brazdil. A metric for paraphrase detection. In *Proceedings of the International Multi-Conference on Computing in the Global Information Technology*, pages 7–16. IEEE, 2007.

[21] João Cordeiro, Gaël Dias, and Pavel Brazdil. New functions for unsupervised asymmetrical paraphrase detection. *Journal of Software*, 2(4):12–23, 2007.

[22] Zornitsa Kozareva and Andrés Montoyo. Paraphrase identification on the basis of supervised machine learning techniques. In *Advances in natural language processing*, pages 524–533. Springer-Verlag Berlin Heidelberg, 2006.

[23] Nitin Madnani, Joel Tetreault, and Martin Chodorow. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190, Montréal, Canada, 2012. Association for Computational Linguistics.

[24] Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350, Morristown, NJ, 2004. Association for Computational Linguistics.

[25] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[26] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[27] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *The 3rd International Workshop on Paraphrasing (IWP2005)*, 2005.

[28] George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244, 1990.

[29] George Miller and Christiane Fellbaum. WordNet: An electronic lexical database, 1998.

[30] Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, volume 3, pages 805–810, 2003.

[31] Claudia Leacock and Martin Chodorow. Combining local context and WordNet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[32] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[33] Samuel Fernando. Paraphrase identification. Master's thesis, Department of Computer Science, University of Sheffield, 2007.

[34] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of International Joint Conference on Artificial Intellingence*, pages 448–453, 1995.

[35] Dekang Lin. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304, 1998.

[36] Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, 1997.

[37] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the American Association for Artificial Intelligence*, volume 6, pages 775–780, Boston, 2006.

[38] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[39] Jeremy H Clear. The British National Corpus. *The digital word: text-based computing in the humanities*, pages 163–187, 1993.

[40] Peter D Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–502, 2001.

[41] Gerard Salton and Michael E Lesk. Computer evaluation of indexing and text processing. *Journal of the Association for Computing Machinery*, 15(1):8–36, 1968.

[42] John Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc, 1993.

[43] Mark Stevenson and Mark A Greenwood. A semantic approach to IE pattern induction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 379–386, Morristown, NJ, 2005.

[44] Vasileios Hatzivassiloglou, Judith L Klavans, Melissa L Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen McKeown. SIMFINDER: A flexible clustering tool for summarization. In *Proceedings of NAACL Workshop on Automatic Summarization*, pages 41–49. Association for Computational Linguistics, 2001.

[45] Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139. Association for Computational Linguistics, 2000.

[46] Sameer S Pradhan, Wayne Ward, Kadri Hacioglu, James H Martin, and Daniel Jurafsky. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association of Computational Linguistics*, pages 233–240, Boston, MA, 2004.

[47] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[48] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc, 2002.

[49] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231, 2006.

[50] Matthew G Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2-3):117–127, 2009.

[51] Michael Denkowski and Alon Lavie. Extending the METEOR machine translation evaluation metric to the phrase level. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 250–253. Association for Computational Linguistics, 2010.

[52] Nizar Habash and Ahmed Elkholy. SEPIA: surface span extension to syntactic dependency precision-based MT evaluation. In *Proceedings of the Workshop on Metrics for Machine Translation at AMTA*. Citeseer, 2008.

[53] Steven Parker. BADGER: A new machine translation metric. In *Proceedings of the Workshop on Metrics for Machine Translation at AMTA*. Citeseer, 2008.

[54] Yee Seng Chan and Hwee Tou Ng. MAXSIM: A Maximum Similarity Metric for Machine Translation Evaluation. In *Proceedings of Association for Computational Linguistics*, pages 55–62. Citeseer, 2008.