UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

UNIDAD ACADÉMICA PROFESIONAL TIANGUISTENCO

"Inducción de medidas de similitud utilizadas en tareas de procesamiento de lenguaje natural, mediante regresión simbólica"

Tesis

para obtener el grado de

Maestro en Ciencias de la Computación

Que presenta:

Ing. Eder Vázquez Vázquez

Asesora:

Dra. Yulia Ledeneva Nikolaevna

Tutores adjuntos:

Dr. René Arnulfo García Hernández

Dr. José Luis Tapia Fabela

TIANGUISTENCO, MÉXICO. Diciembre de 2017

UAEM | Universidad Autónoma del Estado de México

DICTÁMEN DE AUTORIZACIÓN Y OBTENCIÓN DE GRADO DE MAESTRÍA

Tianguistenco, Méx., a 12 de noviembre de 2017

Título del proyecto:

Inducción de medidas de similitud utilizadas en tareas de procesamiento de lenguaje natural, mediante regresión simbólica

Tesista:

Ing. Eder Vázquez Vázquez

Dictamen: **No. de revisión:** 5

- ☐ **Rechazado**
- ☐ **Sujeto a modificaciones**
- ☐ **Aceptado, condicionado**
- ■ **Aceptado**

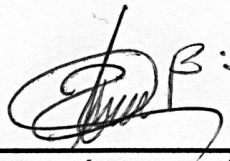
Observaciones generales:

Aceptado para la impresión
Aceptado para la defensa de grado

Tutor Adjunto	Tutor Académico	Tutor Adjunto
Dr. Rene Arnulfo García Hernández	Dra. Yulia Nikolaevna Ledeneva	Dr. José Luis Tapia Fabela

Declaración de originalidad del trabajo escrito

Mediante esta carta hago constar que el trabajo de tesis presentado en este documento es original porque cita debidamente los contenidos utilizados como soporte a la investigación presentada, por lo que exonero a la Universidad Autónoma del Estado de México de cualquier problema de derechos de propiedad intelectual.



Eder Vázquez Vázquez

TEMET NOSCE

Dedicatoria

A mis padres, por apoyarme en esta travesia, otra vez, - GRACIAS-

A mis hermanas, por soportarme.

A todas aquellas personas a las que conté mis temas de maestría,
a quien comenté mis avances sobre mis investigaciones (experimentos y artículos),
sé que los aburrí más de una vez, gracias por escucharme, y soportarme.

Agradecimientos

A mis asesores, Dra. Yulia Ledeneva, René Gárcia Hernandez, por compartir sus conocimientos conmigo y alentarme a terminar y seguir mis investigaciones.

A mis maestros de postgrado, sus conocimientos fueron de gran ayuda
para lograr concluir esta meta.

Al Mtro. Jose Rafael Cruz Reyes, por siempre estar al tanto de mi investigación,
de mis avances y de mis atrasos, así como por apoyarme cuando lo necesitaba.

A mis compañeros de postgrado, maestría, doctorado y postdoctorado,
por compartir los momentos.

Agradecimientos especiales a Consejo Nacional de Ciencia y Tecnología (*CONACYT*) por el apoyo otorgado a través de la beca para estudios del Programa Nacional de Posgrado de Calidad (*PNPC*) en la Maestría en Ciencias de la computación para el CVU - 712303

Resumen

El procesamiento de lenguaje natural es un conjunto de tareas capaces de procesar el lenguaje oral y escrito mediante técnicas y métodos computacionales que permitan la manipulación de lenguajes naturales. Algunas de las tareas creadas para el procesamiento de lenguaje natural son: Recuperación de información, Detección de plagio, Desambiguación del sentido de las palabras, Generación automática de resúmenes, Detección de nombres de medicamentos confusos, Detección de palabras clave, Clasificación de tópicos, Clasificación de documentos, entre otras.

A pesar de que el objetivo de las tareas del procesamiento de lenguaje natural es especifico para cada una de ellas, estas tareas comparten algunas características en común. Las características que comparten la mayoría de las tareas de procesamiento de lenguaje natural son: 1) Precisan una forma de representación de la información, 2) Requieren una función de similitud, 3) Necesitan un paradigma de evaluación. Estos tres elementos son de gran importancia al momento de desarrollar una aplicación de procesamiento de lenguaje natural, pero el elemento que más impacto tienen en su desarrollo es la función de similitud que se utiliza.

Existe una gran cantidad de funciones de similitud que pueden ser aplicadas al procesamiento de lenguaje natural, y aunque estas funciones han demostrado generar buenos resultados, aún no existe una "mejor" función de similitud que genere resultados competitivos para todas las tareas de procesamiento de lenguaje. Existen investigaciones que tratan de resolver el problema de la "mejor" función de similitud, pero centrándose en generar una función de similitud especifica a cada aplicación de procesamiento de lenguaje natural.

Una de las maneras de crear funciones de similitud especificas es a través de la inducción de los valores generados por funciones de similitud conocidas. A este proceso se le conoce como inducción de funciones de similitud. Existen diversos métodos de inducción, entre ellos análisis de regresión (técnica estadística), algoritmos genéticos, redes neuronales, regresión simbólica (técnicas computacionales), entre otras.

Es esta tesis se propuso la aplicación de un método de inducción de funciones de similitud a través de regresión simbólica. El método propuesto genera funciones de similitud a través de la combinación inducida de valores de similitud generados por funciones conocidas. El método propuesto fue probado en dos tareas del procesamiento de lenguaje natural: detección de nombres de medicamentos confusos y desambiguación del sentido de las palabras. Los resultados del método propuesto aplicado a ambas tareas del procesamiento de lenguaje natural mencionadas generan buenas funciones de similitud, y los resultados al

evaluar las tareas con sus respectivos paradigmas de evaluación, muestran resultados superiores a otros métodos del estado del arte de dichas tareas.

Los resultados finales de la evaluación de las tareas de procesamiento de lenguaje natural utilizando la función de similitud inducida por el método propuesto general resultados superiores a otros trabajos, por lo cual se comprueba la eficacia del método propuesto.

El método propuesto está diseñado de tal forma que puede ser utilizado por diversas tareas del procesamiento de lenguaje natural, siempre y cuando estas cumplan con los tres componentes antes mencionados (una forma de representación de la información, función de similitud y paradigma de evaluación). En esta tesis se demuestra la aplicación del método a la detección de nombres de medicamentos confusos y desambiguación del sentido de las palabras, y se deja abierta la futura aplicación del método a otras tareas del procesamiento de lenguaje natural.

Contenido

Índice de Figuras

Índice de Tablas

CAPÍTULO 1.

Introducción

Tradicionalmente, y desde tiempos antiguos, el conocimiento de la humanidad ha sido almacenado en diversas formas escritas (tales como libros, papiros, revistas, cartas, etc.) pero actualmente, este conocimiento ha ido creciendo a pasos enormes provocando que los libros y demás formas escritas de información pasen a almacenarse en forma digital (Friedman *et al.*, 2013) (Wang *et al.*, 2013). El término *digital* describe a la tecnología que genera, almacena y procesa datos de manera electrónica (Chu, 2003). Con la llegada de la computadora, de Internet y otras tecnologías de la información, la información digital y de cualquier otro medio electrónico está aumentando de manera exponencial (Gantz *et al.*, 2013) (Malhotra *et al.*, 2013).

Internet es la colección más grande de información digital (Bordignon *et al.*, 2007), y cada día millones de personas acceden a ella para realizar búsquedas que les ayuden a satisfacer sus necesidades de información (Jaffri, 2007), así como para resolver problemas relacionados a distintos factores como salud, educación, finanzas, agricultura, etc. (Marcus *et al.*, 2016). Debido a la enorme cantidad de información en internet, es sumamente difícil recuperar información relevante de esta colección (Fan *et al.*, 2009), por lo cual, surge la necesidad de contar con herramientas, que, de manera automática, permitan manejar la información y lograr una forma de comprensión de la misma (Malhotra *et al.*, 2013).

El procesamiento del lenguaje natural surge como una sub-área dentro de las ciencias de la computación capaz de procesar el lenguaje oral (Avello, 2005) y escrito (Jackson *et al.*, 2007) mediante métodos, técnicas y herramientas de computación que permitan la manipulación de lenguajes naturales (Bharati *et al.*, 1996). Un lenguaje natural es aquel que permite la comunicación entre una o varias personas (Cañon *et al.*, 2007), y así expresar conocimientos, ideas, opiniones, órdenes y demás acciones lingüísticas, por lo cual, es una propiedad exclusivamente humana (Eifring *et al.*, 2005). Una característica peculiar de los lenguajes naturales es que continúan su evolución sin considerar la gramática, y cualquier regla sobre ellos se desarrolla después de sucedido el hecho (Vásquez *et al.*, 2009).

El procesamiento de lenguaje natural es una sub-disciplina de la inteligencia artificial y de la lingüística computacional (Hogenboom *et al.*, 2010) que trata de extraer la representación más significativa de un texto de manera automática o semi-automática mediante modelos computacionales (Kao, 2007). La razón de la importancia de las técnicas de procesamiento de lenguaje natural es que la mayor parte de la información digital (almacenada por usuarios o en internet) se encuentra descrita precisamente en un lenguaje natural (inglés, español, alemán, etc.) (Langer, 2001) (Cambria *et al.*, 2014). El análisis automático de un lenguaje implica una profunda comprensión del mismo mediante técnicas computacionales (Cambria *et al.*, 2014) (PARC, 2016).

A pesar de que el procesamiento de lenguaje natural facilita el desarrollo de técnicas y herramientas que permiten la manipulación de información descrita en lenguaje natural, cuenta con algunas limitaciones, por ejemplo:

- no puede entender palabras nuevas
- no puede construir inferencias
- no puede concluir desambiguaciones
- no puede generar conceptos

Las cuales son acciones que realiza un humano de manera espontánea al utilizar su lenguaje (Liddy, 1998b) (Cambria *et al.*, 2014). Sin embargo, la principal ventaja del procesamiento de lenguaje natural es que, puede convertir el lenguaje natural en representaciones formales que la computadora puede interpretar y manipular fácilmente (Collobert *et al.*, 2008) (PARC, 2016). Algunas de las tareas que involucran la aplicación de técnicas de procesamiento de lenguaje natural pueden ser: Recuperación de Información (Baeza-Yates *et al.*, 2011), Detección de Plagio (Alzahrani *et al.*, 2012), Desambiguación del sentido de las palabras (Vargas, 2016), Generación Automática de Resúmenes (Ledeneva, 2008), Detección de nombres de medicamentos confusos (Millán, 2016), entre otros.

Dos de las apliaciones de procesamiento de lenguaje natural más estudiadas en los últimos años son la Desambiguación del sentido de las palabras (Vargas, 2016) y la Detección de nombres de medicamentos confusos (Millán, 2016).

La desambiguación del sentido de las palabras es la capacidad de identificar el significado correcto de las palabras, a partir del contexto en el que se emplean, mediante técnicas computacionales (Navigli, 2009). Mientras que, desambiguar palabras es una tarea fácil para los humanos, para una computadora es un proceso complejo (Yuan *et al.*, 2016), es por eso que los últimos años se han incrementado las investigaciones relacionadas al desambiguación del sentido de las palabras (Navigli, 2009), y algunos de los trabajos más resaltantes son los de (Mihalcea *et al.*, 2004) y (Vargas, 2016).

La detección de nombres de medicamentos confusos es la tarea del procesamiento de lenguaje natural que se encarga de analizar la confusión de nombres de medicamentos se da como consecuencia por su parecido ortográfico o fonético (OMS, 2007) (Medicine, 2007). Cuando dos nombres de medicamentos son confundibles con base en su parecido fonético u ortográfico, se dice que son un par de nombres LASA (Look-Alike & Sound-Alike). De acuerdo a la (FDA, 2017), alrededor de 1.3 millones de personas al año son afectadas debido a la confusión de nombres de medicamentos, y el costo monetario asciende a los $3 millones de dólares anuales. Ante esta situación, se han llevado investigación que permiten identificar pares de nombres de medicamentos confusos, y de esta manera ayudar al sector salud reduciendo los casos de error de medicación. Algunas de estas investigaciones son (Kondrak *et al.*, 2006) (Lambert *et al.*, 2004) (Nagata *et al.*, 2014) y recientemente (Millán, 2016).

La mayoría de las tareas de procesamiento de lenguaje natural comparten algunas características en común:

1) Precisan una forma de representación de la información (generalmente textual) (Hartigan, 1975) (Lee *et al.*, 2014).

2) Requieren una función de asociación (o función de similitud) (Huang *et al.*, 2012) (Carmona, 2014) (Niewiadomski *et al.*, 2015).

3) Necesitan un paradigma de evaluación (Clark *et al.*, 2013).

La información textual generalmente es representada mediante el modelo espacio-vectorial (Salton *et al.*, 1975). En esta representación, cada documento de texto es representado mediante un vector de características, donde cada característica corresponde a un valor del mismo, tal como sus palabras, conceptos, longitud. A pesar de que el modelo espacio-vectorial es una de las representaciones más antiguas, sigue siendo una de las más utilizadas (Yih *et al.*, 2010) (Huang *et al.*, 2012) (Mehrbod *et al.*, 2014) (Gadge *et al.*, 2015).

La función de similitud es una propiedad que mide el grado de semejanza entre dos documentos de texto. Dadas dos secuencias de texto, la similitud entre ellas es representada por un valor numérico, regularmente entre 0 y 1, donde los valores más cercanos a 1 indican una similitud más alta entre ambas secuencias (Goshtasby, 2012) (Baccour *et al.*, 2014). La medición de similitud entre distintos textos es una parte fundamental para la mayoría de las formas de análisis de información y de procesamiento de lenguaje natural (Liu *et al.*, 2015a) (Metzler *et al.*, 2007) (Yih *et al.*, 2007) (Huang, 2008).

Las tareas de procesamiento de lenguaje natural necesitan ser evaluadas bajo un método estándar que permita la comparación entre distintos sistemas. Existen diversos métodos para evaluar las tareas de procesamiento de lenguaje natural (Clark *et al.*, 2013), pero generalmente se realiza mediante una evaluación intrínseca (Jones *et al.*, 1996). Una evaluación intrínseca mide la calidad de una aplicación de procesamiento de lenguaje natural con respecto a un archivo etiquetado manualmente. Regularmente esta evaluación es realizada en un entorno de laboratorio (Resnik *et al.*, 2006) (Elliott *et al.*, 2003).

Estos tres elementos (representación de información, función de similitud, y paradigma de evaluación) han sido ampliamente estudiados, pero en los últimos años, la similitud textual entre dos textos ha sido una tarea exhaustiva en diversas tareas del procesamiento de lenguaje natural (Yih *et al.*, 2007) (Liu *et al.*, 2015b).

Como se menciona anteriormente, la definición de una función de similitud es importante para la mayoría de las tareas de procesamiento de lenguaje natural (Hillel *et al.*, 2007), y por lo general solo expertos del dominio (expertos en cierta tarea de procesamiento de lenguaje) son capaces de proporcionar los conocimientos necesarios para crear una nueva medida de similitud (Cheng *et al.*, 2008) (Stahl, 2004) (Liu *et al.*, 2015a).

Existen diversas medidas de similitud en la literatura de procesamiento de lenguaje natural (Thada *et al.*, 2013) (entre ellas destacan: similitud Coseno, Jaccard, Dice, Euclidiana, entre otras), cada una con diferentes características y diversos objetivos, que han sido aplicadas en

diversas tareas de procesamiento de lenguaje natural (Chen *et al.*, 2009), y aunque han otorgado buenos resultados, no existe una "mejor" función de similitud (Thada *et al.*, 2013). Stahl afirma que la correcta definición de mejores medidas de similitud podría mejorar la calidad de las tareas de procesamiento de lenguaje natural (Stahl, 2004). Investigaciones en el tema (Maggini *et al.*, 2008) (Chopra *et al.*, 2005) (Yih *et al.*, 2010) (Kumar *et al.*, 2008) (Carmona, 2014) aseguran que la solución consiste en calcular, mediante aprendizaje supervisado, un modelo que mejore la similitud entre dos documentos de texto de un conjunto de datos, ya sea mediante la definición de nuevas medidas de similitud, o mediante la inducción de los valores generados por las funciones de similitud existentes.

La definición de medidas de similitud o inducción de ellas mediante el aprendizaje automático es conocida como Aprendizaje de Métricas de Similitud (*Learn similarity measure*) (Maggini *et al.*, 2008) (Bellet *et al.*, 2015), y consiste en buscar un modelo que mejore la similitud a través de del aprendizaje mediante un conjunto de datos, de manera que, el modelo aprendido, satisfaga las restricciones de similitud (o distancia) y mejore los resultados al evaluar cierto proceso de lenguaje natural (Stahl, 2004) (Liu *et al.*, 2015a). Básicamente, el aprendizaje de métricas de similitud busca aprender los aspectos que comparten los datos para llegar a un objetivo determinado y, representar porteriormente, los aspectos mediante un modelo (Hillel *et al.*, 2007) (Bellet *et al.*, 2015). Debido a que el aprendizaje de medidas de similitud involucra la incorporación de diversos datos (que pueden estar en distintas formas), es necesario inferir sobre ellos, esta técnica es un problema de optimización (Gabel, 2003) (Maggini *et al.*, 2008) (Qiong *et al.*, 2013) (Chen *et al.*, 2014) que puede resolverse a través de un proceso iterativo (Xing *et al.*, 2002) (Kumar *et al.*, 2008, Jin *et al.*, 2009) (Yin *et al.*, 2010).

La optimización es la maximización o minimización de una función objetivo sujeta a las restricciones de sus variables para darle solución a un problema (Nocedal *et al.*, 2006). Existen diferentes métodos de resolución de problemas mediante técnicas de optimización, pero (Baquela *et al.*, 2013) las clasifica en tres:

1) Resolución mediante cálculo
2) Resolución mediante técnicas de búsqueda
3) Resolución mediante técnicas de convergencia de soluciones.

Dentro de este último grupo, existe un paradigma computacional que se encarga de la resolución de problemas difíciles, donde la búsqueda y la optimización son su objetivo principal (Sette *et al.*, 2001) (Fogel, 2006) (Can *et al.*, 2011). Este paradigma se le conoce como Algoritmos Evolutivos (Koza, 1992b) (Banzhaf *et al.*, 1998).

Los algoritmos evolutivos son técnicas basadas en la teoría de la selección natural de Darwin (Darwin, 1859), en la que los individuos que mejor se adapten a su entorno tienen una mayor

probabilidad de sobrevivir y pasar sus características genéticas a sus descendientes. Estos algoritmos combinan la supervivencia del más apto con métodos de intercambio de información (basado en la reproducción sexual) (Fogel, 2006) para simular la evolución de una población de individuos, con el fin de descubrir iterativamente mejores soluciones(Can *et al.*, 2011). Aunque existen diversas clases de algoritmos evolutivos, los más utilizados son los Algoritmos Genéticos y la programación genética (Berlanga, 2010).

La programación genética es una rama de la inteligencia artificial (Parra, 2007) que pretende resolver problemas mediante la inducción de programas[1] y algoritmos que los resuelvan (Koza, 1992b) (Tomassini, 1995) (Banzhaf *et al.*, 1998) (Sette *et al.*, 2001). Estos algoritmos fueron diseñados para encontrar de forma automatizada programas de computadora (Goldberg, 1989) (Brameier *et al.*, 2006) que, a partir de un conjunto de valores de entrada, puedan definir un modelo que describa el comportamiento de los datos (Koza, 1992b) (Gestal, 2010) (Do *et al.*, 2008) (Ragalo *et al.*, 2012). El propósito de la programación genética es inducir una población de programas de computadora que vayan mejorando de forma automática con los datos con los que se han entrenado y generar una expresión matemática basada en esos datos (Banzhaf *et al.*, 1998) (Brameier *et al.*, 2006) (Do *et al.*, 2008) (Dabhi *et al.*, 2011).

Una de las principales tareas de la programación genética es la Regresión Simbólica, la cual es conocida principalmente como una técnica de aprendizaje para producir ecuaciones (funciones o modelos) matemáticas que sean capaces de definir modelos de datos para una gran diversidad de problemas (Koza, 1992b) (Koza, 1994) (Sette *et al.*, 2001).

Existen trabajos que se han encargado de demostrar la relación que existe entre la definición y optimización de funciones (medidas) de similitud y la regresión simbólica, pues como se explicó anteriormente, estas dos tareas trabajan bajo la optimización de valores que permitan la resolución a un problema en específico (Gabel, 2003) (Sette *et al.*, 2001). En esas investigaciones se trabaja sobre la definición automática de medidas (funciones o modelos) de similitud mediante inducción, y como resultado han sido beneficiadas algunas técnicas de procesamiento de lenguaje natural.

En algunos de estos trabajos se encuentran las aportaciones que realizó Stahl (Stahl *et al.*, 2003), el cual hace uso de un programa evolutivo con el fin de optimizar medidas de similitud para la tarea de Razonamiento Basado en Casos, cuyo objetivo fue estimar la utilidad de las consultas realizadas por un usuario en la búsqueda de computadoras personales (PC) con base en las características de las mismas PC. Sullivan (Sullivan *et al.*, 2007) por su lado

[1] Programa puede ser entendido como un conjunto de pasos (algoritmos), reglas gramaticales, funciones o modelos.

evolucionó las funciones de *kernel* dentro de una Máquina de Soporte Vectorial mediante programación genética. El objetivo de ese trabajo fue optimizar el *kernel* de la máquina con el fin de mejorar los resultados en la tarea de clasificación. Dentro del área textual, (Metzler *et al.*, 2007) y (Bär *et al.*, 2012) realizan el aprendizaje de medidas de similitud para la detección de similitud semántica en textos. En ambos trabajos, los resultados finales indican una mejora frente a resultados obtenidos por otras investigaciones del estado del arte. En el trabajo de Carmona (Carmona, 2014) también se hace uso de un programa genético con el fin de ensamblar medidas de similitud a partir de medidas ya existentes para la tarea de Detección de Paráfrasis. En el trabajo de Vázquez (Vázquez, 2015) se hizo uso de un algoritmo basado en la programación genética el cual se utilizó para modelar la relevancia de la posición de las oraciones que son utilizadas en la generación automática de resúmenes.

Como se mencionó anteriormente, el aprendizaje de medidas de similitud es una técnica cuyo objetivo principal es inducir automáticamente medidas de similitud sofisticadas, donde esto puede ser visto como un proceso de optimización y búsqueda dentro de un espacio de soluciones (Gabel, 2003) (Qiong *et al.*, 2013). La programación genética ha sido conocida principalmente como una técnica de aprendizaje (basada en la optimización) para producir funciones matemáticas (Koza, 1992b) que sean capaces de definir modelos de datos (Do *et al.*, 2008). Por otra parte, algunas de las tareas de procesamiento de lenguaje natural (tales como Recuperación de Información, Detección de plagio, Clasificación de documentos) hacen uso de medidas de similitud con el fin de mejorar la calidad de los resultados en tales sistemas (Stahl, 2004).

1.1 Planteamiento del problema

Las investigaciones en torno al aprendizaje de medidas de similitud dentro del área de procesamiento de lenguaje natural han sido variadas. Cada una de las investigaciones dentro de esta área han propuesto métodos diferentes de aprendizaje de medidas de similitud con base en la inducción de las medidas existentes, y su aplicación se ha visto limitada al problema específico que tratan en esa investigación, es decir, su método propuesto no ha sido probado en otras tareas de procesamiento de lenguaje natural. Basándose en el estado del arte relacionado a este tema, y debido a que la desambiguación del sentido de las palabras y la detección de nombres de medicamentos confusos son dos de las principales tareas del procesamiento de lenguaje natural, el problema encontrado fue el siguiente:

¿Cómo mejorar los resultados de las tareas de procesamiento de lenguaje natural, específicamente para desambiguación del sentido de las palabras y detección de nombres de medicamentos confusos, mediante la inducción de los resultados generados por diversas medidas de similitud existentes, mediante la aplicación de regresión simbólica?

1.2 Objetivos

A partir del estado del arte y del planteamiento anterior surgieron los siguientes objetivos para respaldar el presente trabajo de tesis.

1.2.1 Objetivo general

Inducir medidas de similitud actuales mediante programación genética con el fin de mejorar los resultados en las tareas de desambiguación del sentido de las palabras y detección de nombres de medicamentos confusos pertenecientes al área de procesamiento de lenguaje natural.

Para lograr este objetivo, y por la aplicación de la metodología que más adelante se detalla, surgieron los siguientes objetivos particulares.

1.2.2 Objetivos particulares

1. Conocer los resultados generados por las medidas de similitud existentes en las tareas de procesamiento de lenguaje natural desambiguación del sentido de las palabras y detección de nombres de medicamentos confusos.
2. Evaluar los resultados generados por las medidas de similitud para la desambiguación del sentido de las palabras y detección de nombres de medicamentos confusos.
3. Desarrollar el programa de inducción basado en programación genética.
4. Aplicar el método propuesto basado en programación genética para inducir los resultados de las medidas de similitud existentes con el fin de mejorar los resultados en las tareas de desambiguación del sentido de las palabras y detección de nombres de medicamentos confusos.
5. Evaluar los resultados obtenidos por el método propuesto en cada las tareas de desambiguación del sentido de las palabras y detección de nombres de medicamentos confusos.
6. Comparar los resultados obtenidos por el método propuesto frente a los resultados de las medidas de similitud existentes y frente a los trabajos relacionados a las tareas de desambiguación del sentido de las palabras y detección de nombres de medicamentos confusos.

1.3 Hipótesis

Ante la problemática planteada, la hipótesis es:

Si la programación genética permite inducir modelos o funciones matemáticas, y si el aprendizaje de medidas de similitud genera medidas de similitud más sofisticadas a partir de conocimiento previo, entonces será posible inducir los resultados de las medidas de similitud existentes, mediante programación genética, con el fin de mejorar los resultados obtenidos en tareas de Procesamiento de lenguaje natural.

1.4 Justificación

Debido a la gran cantidad de medidas de similitud existentes aplicadas en el estado del arte para resolver problemas dentro del procesamiento del lenguaje natural, y debido a la dificultad de crear nuevas medidas de similitud específicas, fue conveniente realizar esta investigación donde se demuestra que la aplicación de un método de inducción basado en programación genética, donde se utilizan los valores generados por las medidas de similitud existentes, puede mejorar los resultados en la tareas de procesamiento de lenguaje natural.

El principal beneficio que aporta esta tesis es el de demostrar que la inducción de las medidas de similitud actuales puede mejorar los resultados en las tareas de procesamiento de lenguaje natural haciendo uso de la programación genética.

1.5 Delimitación

En esta tesis se presenta un método de inducción de funciones de similitud mediante regresión simbólica, dicho método esta diseñado para poder ser aplicado a diversas tareas de procesamiento de lenguaje natural, pero en esta tesis solo se prueba y verifica su aplicación a la tarea de desambiguación del sentido de las palabras, específicamente al método propuesto en (Vargas, 2016), y la tarea de detección de nombres de medicamentos confusos utilizando el método de (Millán, 2016). La aplicación del método propuesto en esta tesis a otras tareas de procesamiento de lenguaje natural queda fuera del alcance de la misma.

1.6 Estructura de la tesis

El resto de la tesis está organizado de la siguiente forma. En el capítulo 2 se presentan los conceptos necesarios para entender esta tesis. En el capítulo 3 se presenta una descripción detallada de las funciones de similitud, sus características, tipos y algunos ejemplos. En el capítulo 4, es presentado el estado del arte, donde se analizan diversos trabajos similares a la

investigación central de este trabajo. El método propuesto para dar solución al problema planteado en esta investigación es mostrado en el capítulo 5. En el capítulo 6 se muestran los resultados de los experimentos que permiten validar al método propuesto, además de dar solución al problema planteado. Finalmente, en el capítulo 7, se describen las conclusiones generadas a partir de este trabajo de investigación.

CAPÍTULO 2.

Marco Teórico

En este capítulo se describen los conceptos y definiciones necesarias para la correcta comprensión de esta investigación.

Se describen primeramente la rama de investigación sobre la cual se basa, procesamiento de lenguaje natural, así como varias de sus características y la razón de su estudio actual. Después se describe el aprendizaje máquina (*Machine Learning*) como herramienta de ayuda al procesamiento de lenguaje natural, así como algunos de sus componentes y algoritmos. Finalizando con la descripción de algoritmos genéticos y programación genéticas, donde ésta última es la técnica principal utilizada en esta investigación.

2.1 Lenguaje natural

El lenguaje es uno de los aspectos fundamentales del comportamiento humano y es un elemento crucial de nuestras vidas (Allen, 1994). El lenguaje ha ido evolucionando a través del tiempo con fines de la comunicación humana (Brookshear *et al.*, 1993). De manera escrita, el lenguaje ha permitido mantener el conocimiento de generación en generación (Allen, 1994), mientras que de manera hablado permite la expresión de sentimientos y la comunicación (Cañon *et al.*, 2007). Al uso del lenguaje que permite la comunicación natural entre personas con el fin de permitir la transmisión y recepción de información se le conoce como *lenguaje natural* (Brookshear *et al.*, 1993) (Allen, 1994) (Pathak *et al.*, 2012).

La investigación del entendimiento del lenguaje natural tiene dos principales motivaciones (Allen, 1994) (Bharati *et al.*, 1996): la tecnológica y la lingüística. La motivación tecnológica se basa en la construcción de modelos computacionales que permitan entender y reproducir varias tareas del lenguaje natural. De esta manera, se puede revolucionar la forma en cómo la interacción de una persona con una computadora es entendida (Allen, 1994). La motivación lingüística es también cognitiva, ya que se enfoca en entender el cómo los humanos se comunican entre ellos haciendo uso de su lenguaje natural (Fowler, 1974) (Bharati *et al.*, 1996).

El estudio del lenguaje natural está abarcado por diferentes campos disciplinarios (Allen, 1994), de la cuales destacan: Lingüística, Psico-lingüística, Filosofía y Lingüística Computacional.

- Lingüística: permite el estudio científico del lenguaje humano a partir de la creación de modelos que describan y expliquen las lenguas de manera teórico-descriptiva (Pinker, 1994) (Cabarcos *et al.*, 2005). Estudia el uso del lenguaje y cómo lo utilizamos para pensar, sentir y conocer el mundo que nos rodea (Villena, 2005).
- Psicolingüística: sus principales intereses se localizan en torno a dos grandes temas: la adquisición de la habilidad lingüística por parte del niño, y la relación entre el comportamiento lingüístico y los procesos mentales de codificación y decodificación que anteceden a dicho comportamiento (Codesido, 1999).
- Lingüística computacional: Es la rama que estudia el procesamiento del lenguaje natural mediante técnicas computacionales (Gelbukh, 2010) y mediante la aplicación de algoritmos, y modelos que representen el lenguaje, logrando un razonamiento del mismo (Allen, 1994).

2.2 Procesamiento de lenguaje natural

El *procesamiento de lenguaje natural* es una sub-disciplina de la inteligencia artificial y de la lingüística computacional (Allen, 1994) (Hogenboom *et al.*, 2010) (Cañon *et al.*, 2007) (Pathak

et al., 2012), la cual es entendida como la habilidad que tiene una computadora para procesar (de manera automática o semiautomática), analizar y sintetizar el lenguaje natural (Bharati *et al.*, 1996) (Kao, 2007) (Gelbukh, 2010). El área del procesamiento de lenguaje natural se encarga del desarrollo de técnicas y modelos que permitan la transformación de información en lenguaje natural a un formato entendible por una computadora (Collobert *et al.*, 2008) (Hogenboom *et al.*, 2010) (Friedman *et al.*, 2013).

Dentro de esta área se han creado sistemas, técnicas y herramientas que facilitan el acceso y manipulación de información, algunas de estas son: Manejo de conocimiento, Traducción automática, Minería de Texto, Identificación de autoría, Identificación de plagio (Gelbukh, 2010), Recuperación de información, Extracción de información, Generación de resúmenes (Vásquez *et al.*, 2009), Reconocimiento óptico de caracteres, clasificación de documentos, Generación de informes, Sistemas de diálogo, entre otras más (Copestake, 2003).

Los sistemas, técnicas y herramientas del procesamiento del lenguaje natural son afectadas por diversas áreas de investigación, tales como la Lingüística, Inteligencia Artificial, *Machine Learning*, Estadística computacional y Ciencias Cognitivas. Sin embargo, la base del procesamiento del lenguaje natural es ampliamente impulsada por técnicas de cálculo para analizar y representar documentos descritos en lenguaje natural en uno o varios de los niveles de análisis de lenguaje, y de esta manera simular el proceso que realiza un humano en ciertas tareas (Liddy, 2001) (Behzadi, 2015).

2.3 Niveles de análisis de lenguaje

El procesamiento de lenguaje natural es un conjunto de técnicas computacionales que sirven para analizar y representar textos en uno o más niveles de análisis lingüístico, con el propósito de conseguir un procesamiento similar al de los humanos en alguna de las tareas de esta área (Liddy, 1998a), por lo cual, cualquier sistema de procesamiento de lenguaje puede incluir uno o varios niveles de procesamiento (Liddy, 2001). Los niveles de análisis de lenguaje son los siguientes (Carbonell, 1992) (Liddy, 1998a) (Allen, 1994) (Hogenboom *et al.*, 2010) (Kumar, 2011):

- Nivel Fonológico: Interpretación del sonido y de habla. Trata de cómo las palabras se relacionan con los sonidos que representan. Descompone el texto en unidades de palabra.
- Nivel Morfológico: Trata de cómo las palabras se construyen a partir de unas unidades de significado más pequeñas. Tiene que ver con la formación de palabras y su análisis para extraer raíces, rasgos flexivos, unidades léxicas compuestas y otros fenómenos (incluidos los prefijos, sufijos y raíces).
- Nivel Léxico: Tiene que ver con la validez de las palabras y su pertenecen a una categoría como: nombre, pronombre, verbo, adverbio, y así sucesivamente.

- Nivel Sintáctico: Trata de cómo las palabras pueden unirse para formar oraciones, fijando el papel estructural que cada palabra juega dentro de una oración. Se ocupa de la gramática y la estructura de las oraciones.
- Nivel Semántico: Se encarga del significado de las palabras y de cómo estos significados se unen para dar significado a una oración. El nivel semántico determina los significados posibles de una sentencia (significado de cada palabra dentro de la oración), incluyendo la desambiguación de palabras.
- Nivel Pragmático: Se ocupa del conocimiento que proviene del mundo exterior, es decir, desde fuera del contenido del documento. Se centra en como las palabras son usadas en diferentes situaciones y en cómo su uso afecta a la interpretación de la oración.
- Nivel de Discurso: Se centra en cómo una oración afecta y es afectada por la interpretación de las oraciones precedentes y siguientes a ella.
- Nivel Mundial: Incluye el conocimiento general acerca de la estructura del mundo del lenguaje de los usuarios y el cómo lo utilizan, por ejemplo, al mantener una conversación.

2.4 Aproximaciones del procesamiento de lenguaje natural

Existen dos aproximaciones principales bajo las cuales opera el procesamiento de lenguaje natural: basados en reglas y basados en modelos estadísticos.

2.4.1 Procesamiento de lenguaje natural basado en reglas

También conocido como "enfoque simbólico", y se basa principalmente en reglas determinísticas descritas por un grupo de expertos, las cuales funcionan de acuerdo al uso de los componentes de procesamiento de lenguaje natural. Las reglas son una manera excelente de encapsular diversos tipos de conocimiento adquirido por un experto, por ejemplo, la manera en cómo una persona expresa una negación o afirmación (Jackson *et al.*, 2007).

La ventaja más importante que tiene el uso del procesamiento de lenguaje natural basado en reglas es que permite el desarrollo rápido de sistemas que cubran los casos más comunes (para los cuales ya hay reglas descritas), y luego mejorar la precisión del sistema con el tiempo. Además, las reglas son escritas por expertos en el tema, por lo cual, identificar un error en el sistema en más sencillo (Wolniewicz, 2015).

La fuente primaria de evidencia en sistemas basados en el enfoque simbólico proviene de las reglas desarrolladas por humanos y lexicones (Liddy, 2001).

2.4.2 Procesamiento de lenguaje natural basado en modelos estadísticos

Como su nombre lo indica, el procesamiento de lenguaje natural estadístico se basa en el análisis estadístico del lenguaje, también conocido como "enfoque empírico", debido a que trata de aprender las relaciones estadísticas para los componentes de procesamiento de lenguaje natural mediante el procesamiento de grandes cantidades de ejemplos. Debido a esto, la precisión de un modelo estadístico depende del volumen de datos disponibles para generar el aprendizaje (Liddy, 2001) (Jackson *et al.*, 2007).

Los enfoques estadísticos emplean diversas técnicas matemáticas y estadísticas para generar modelos aproximados generalizados de fenómenos lingüísticos de fenómenos reales, generalmente encontrados en corpus etiquetados. Por lo tanto, estos enfoques utilizan los datos observables y disponibles como la principal fuente de aprendizaje (Liddy, 2001).

La principal ventaja de este enfoque es que puede ir aprendiendo los fenómenos del lenguaje con costos más bajos (ya que no depende de la supervisión de un experto) (Wolniewicz, 2015).

2.5 Componentes de las tareas de procesamiento de lenguaje natural

La mayoría de las tareas de procesamiento de lenguaje natural comparten algunas características en común:

1) Precisan una forma de representación de la información (generalmente textual) (Hartigan, 1975) (Lee *et al.*, 2014).
2) Requieren una función de asociación (o función de similitud) (Huang *et al.*, 2012) (Carmona, 2014) (Niewiadomski *et al.*, 2015).
3) Necesitan un paradigma de evaluación (Clark *et al.*, 2013).

2.5.1 Representación de la información

La selección adecuada para representar la información depende de las características que lingüísticas que serán extraídas (Grace *et al.*, 2016).

Una de las características más comunes que se extraen de los documentos son las palabras (*bag-of-words*) (Le *et al.*, 2014). Aunque, otros autores proponen la extracción de características más concretas y que no sean tan ambiguas (McDonald *et al.*, 2001) (Huang, 2011), como por ejemplo conceptos o frases que se encuentran dentro del documento a analizar. Otra de las características más importantes a extraer de un documento, son los *n-gramas*. Un *n-grama* es una secuencia de n palabras (o caracteres) consecutivos: un 2-grama (o bi-grama) es una secuencia de dos palabras consecutivas.

Existen diversas formas para representar las características que son extraídas de los documentos, pero la más utilizadas es el modelo espacio-vectorial (Salton *et al.*, 1975) (Yih *et al.*, 2010). En esta representación, cada característica extraída es representada en cada uno de los elementos del vector, donde cada característica tiene asociado un valor numérico el cual indica su importancia (Yih *et al.*, 2010). A pesar de ser el modelo más utilizado, por ser eficaz, existen otros modelos de representación, tales como grafos o árboles.

La representación basada en grafos es una construcción matemática que permite modelar la relación de la información de manera efectiva. En esta representación un texto se puede representar mediante en un grafo, dónde un vértice puede ser una característica, y las aristas representar su tipo de relación (Sonawane *et al.*, 2014).

En la representación basada en árboles, cada sentencia (oración o frase) del documento es representado mediante un árbol sintáctico, donde las hojas del árbol (terminales) son las palabras de la oración, y los nodos internos (padres de las hojas) son las etiquetas del habla (*Part-of-Speech*) (Massung *et al.*, 2013).

El detalle con el modelo de representación de texto es que, éste depende de las características extraídas y del objetivo de la aplicación de procesamiento de lenguaje natural. Por lo cual es recomendable realizar diferentes extracciones de características y representarlas desde diferentes perspectivas (Yih *et al.*, 2010) (Massung *et al.*, 2013).

2.5.2 Función de similitud

La función de similitud mide el grado de semejanza entre dos documentos de texto. Dadas dos secuencias de texto, la similitud entre ellas es representada por un valor numérico, regularmente entre 0 y 1, donde los valores más cercanos a 1 indican una similitud más alta entre ambas secuencias (Goshtasby, 2012) (Baccour *et al.*, 2014).

La medición de similitud entre distintos textos es una parte fundamental para la mayoría de las formas de análisis de información y de procesamiento de lenguaje natural (Liu *et al.*, 2015a) (Metzler *et al.*, 2007) (Yih *et al.*, 2007) (Huang, 2008).

Una característica importante de las funciones de similitud es que generalmente no proporcionan una dependencia entre las diversas tareas en las que son empleadas y sus respectivas soluciones, por lo cual, solo representan una forma de heurística de trabajo, y la aplicación de las mismas depende del objetivo de la aplicación (Stahl, 2004).

Otra característica de las funciones de similitud, es que éstas dependen del contexto de aplicación y de la forma de representación de la información (Nowak *et al.*, 2007). Además, cabe destacar que el uso de una función de similitud dentro de una aplicacion de

procesamiento de lenguaje natural es un paso intermedio, entre la representación de información y la generación de resultados.

En el Capítulo 3, se presenta una descripción más detallada de las funciones de similitud, su definición formal, tipos y aplicaciones. A partir de este momento, utilizaremos los términos *función de similitud* o *medida de similitud* para referirnos al mismo término antes descrito.

2.5.3 Paradigma de evaluación

Como otros sistemas informáticos, las tareas de procesamiento de lenguaje natural necesitan ser evaluadas bajo un método estándar que permita la comparación entre distintas tareas, por lo cual, la evaluación de éstas juega un papel fundamental para la orientación y enfoque de los investigadores en procesamiento de lenguaje natural (Clark *et al.*, 2013) (Yang, 2015).

Debido a la existencia de diversas tareas dentro del procesamiento de lenguaje natural, existen criterios específicos bajo los cuales se evalúan cada una de éstas. Sin embargo, existen paradigmas generales que describen algunos principios básicos de evaluación (Jones *et al.*, 1996).

Los criterios de evaluación aplicados al procesamiento de lenguaje natural, generalmente caen bajo dos paradigmas principales: evaluación intrínseca y evaluación extrínseca (Jones *et al.*, 1996) (Clark *et al.*, 2013).

Las evaluaciones intrínsecas están directamente relacionadas con el objetivo principal del sistema de procesamiento de lenguaje, por lo cual, su desempeño se mide principalmente con respecto a un *gold standard* (archivo de oro) que fue previamente definido por evaluadores o expertos humanos (Clark *et al.*, 2013) (Bär *et al.*, 2015b).

En las evaluaciones extrínsecas, los resultados de las tareas de procesamiento de lenguaje natural son evaluados con datos externos a los del sistema, considerando a la aplicación dentro de un entorno complejo, o como un sistema de usuario final. Estos tipos de evaluaciones generalmente miden el desempeño del propósito de la aplicación (Clark *et al.*, 2013).

La mayoría de las tareas de procesamiento de lenguaje natural hacen uso de la evaluación intrínseca, debido a que permite una comparación directa con otras tareas. Para llevar a cabo una correcta comparación con otras tareas, es necesario que las evaluaciones intrínsecas utilicen un *gold standard*. Un *gold standard* es una colección de textos, documentos, sentencias o palabras que fueron etiquetadas manualmente por expertos humanos bajo categorías previamente definidas (Poibeau *et al.*, 2008) (Hinze *et al.*, 2012).

El archivo *gold standard* tiene un impacto directo en el desarrollo de las tareas de procesamiento de lenguaje natural (Wissler *et al.*, 2014), por lo cual, los desarrolladores de estas tareas se enfocan en mejorar los resultados de sus tareas al ser evaluados por el *gold standard*.

Estos tres elementos (representación de información, función de similitud, y paradigma de evaluación) han sido ampliamente estudiados, pero en los últimos años, la similitud textual entre dos textos ha sido una tarea muy estudiada en diversas áreas del procesamiento de lenguaje natural (Yih *et al.*, 2007) (Liu *et al.*, 2015b).

Dentro del procesamiento de lenguaje natural existen diversas funciones de similitud que han sido aplicada (Thada *et al.*, 2013) (Chen *et al.*, 2009), y aunque estas han otorgado buenos resultados, no existe una "mejor" función de similitud que genere los mejores resultados para las tareas de procesamiento de lenguaje natural (Thada *et al.*, 2013). Stahl (Stahl, 2004) afirma que la correcta definición de mejores medidas de similitud podría mejorar la calidad en los resultados de los sistemas de procesamiento de lenguaje natural. Investigaciones en el tema (Maggini *et al.*, 2008) (Chopra *et al.*, 2005) (Yih *et al.*, 2010) (Kumar *et al.*, 2008) (Carmona, 2014) aseguran que la solución consiste en calcular, mediante aprendizaje supervisado, un modelo que mejore la similitud entre dos documentos de texto de un conjunto de datos, ya sea mediante la definición de nuevas medidas de similitud, o mediante la inducción de los valores generados por las funciones de similitud existentes. Estos autores aseguran que la definición de nuevas medidas o modelos de funciones de similitud se puede lograr mediante *machine learning* (aprendizaje automático).

2.6 Machine learning

El término *machine learning* (aprendizaje automático) hace referencia a la detección automática de patrones significativos dentro de un conjunto de datos por medio de programas de computadora. Estos programas deben de "aprender" a partir de las entradas disponibles para ellos. Donde el termino aprendizaje hace referencia al proceso de convertir la experiencia en conocimiento (Shai *et al.*, 2014). El objetivo principal del aprendizaje automático es tratar de inducir de manera automática conocimiento (reglas, modelos, etc.) a partir de los datos de entrada con los que se está entrenando (Gabel, 2003).

La entrada a un algoritmo de *machine learning* es conocida como "conjunto de entrenamiento", y representa el conjunto de experiencia que se necesitan aprender. La salida producida por estos algoritmos es el conocimiento adquirido que puede ser aplicado para resolver el problema para el cual se ha entrenado (Shai *et al.*, 2014).

El principal desafío del proceso del aprendizaje automático es generar un algoritmo que posea buena capacidad de generalización, es decir, que no solo aprenda de los ejemplos conocidos utilizados en su proceso de entrenamiento, sino también, que sea capaz de construir un modelo general que permita solucionar ejemplos desconocidos (Cárdenas *et al.*, 2014).

El aprendizaje automático es un surgimiento natural a partir de la intersección de las ciencias de la computación y la estadística, donde el fin común es entender el comportamiento de

cierto conjunto de datos. Por un lado, las ciencias de la computación se han encargado de construir programas o algoritmos que ayuden a la resolución de problemas intratables, mediante el aprendizaje de datos de entrenamiento, además de entender qué algoritmos y métodos son más eficaces para esa solución. Por el otro lado, la estadística se centra en saber qué se puede deducir del conjunto de datos con un grupo de modelos, saber la fiabilidad de la deducción y las conclusiones a las que se llega con sus métodos (Mitchell, 2006). Del primer grupo destacan las técnicas de análisis de regresión, las cuales estudian la relación entre variables dependientes y sus variables independientes (Palacios-Cruz *et al.*, 2013), mientras que del segundo grupo destacan los modelos de inducción basados en enfoques algorítmicos (algoritmos evolutivos y programación genética) (Banzhaf *et al.*, 1998).

2.6.1 Análisis de regresión

El análisis de regresión es un método simple para investigar las relaciones funcionales entre distintas variables. La relación se expresa en forma de una ecuación o un modelo matemático que conecta la respuesta o variable dependiente y una o más variables explicativas o predictoras (Chatterjee *et al.*, 2013). Dicho de otra forma, trata de predecir el valor de una variable en función de los valores conocidos de variables explicativas (Milton, 2007).

Existen tres tipos principales bajo los cuales se clasifica el análisis de regresión: Regresión lineal para datos cuya distribución puede ajustarse a una línea recta; Regresión no-lineal para datos cuya tendencia se ajuste a una curva, y Regresión logística para modelar datos cuando la salida es de tipo binario (Nettleton, 2012).

La regresión lineal es una de las técnicas estadísticas más poderosas y utilizadas en diversas aplicaciones. Esta técnica permite cuantificar la relación entre una variable de respuesta x, y una o más variables predictorias y, siempre y cuando éstas sean cuantitativas y cuenten con una distribución normal, es decir, en la regresión lineal debe existir una relación en la que el incremento (o decremento) de una variable sea proporcional en cada punto (Milton, 2007) (Palacios-Cruz *et al.*, 2013).

La regresión no lineal permite cuantificar la relación entre una variable de respuesta x, y una o más variables predictoras y para conjunto de datos que traten de ajustarse a curvas, modelos cóncavos, convexos, sigmoidales o modelos con máximos y mínimos (Braga *et al.*, 2015). Los modelos de regresión no lineales son menos utilizados que los lineales, debido a que la solución no es encontrada a partir de una expresión matemática explicita, sino que necesita un proceso iterativo para encontrar una solución que se ajuste a los datos, y también debido a que es necesario elegir el modelo de posible solución a encontrar antes de seleccionar las variables y comenzar el proceso iterativo (Nettleton, 2012) (Braga *et al.*, 2015).

La regresión logística trabaja para problemas donde la variable dependiente es de tipo binario (1 o 0, $Sí$ o No), y produce una fórmula que predice la probabilidad de que ocurra un evento en función de los valores de las variables de entrada (Nettleton, 2012). El modelo de regresión logística ha sido utilizado para resolver problemas de clasificación supervisada. La regresión logística contempla variables de respuesta categóricas (Braga *et al.*, 2015).

Sea cual sea el tipo de análisis de regresión que se utilice (lineal, no lineal, logística), el proceso de resolución al problema es el mismo, y consiste en ocho pasos principales (Chatterjee *et al.*, 2013):

1) Declaración del problema
2) Selección de variables potencialmente relevantes
3) Recopilación de datos
4) Especificación del modelo a utilizar
5) Elección del ajuste del modelo
6) Ajuste del modelo
7) Validación del modelo
8) Empleo del modelo creado para solucionar el problema

Declaración del problema: Es el primer paso, y quizás el más importante dentro del análisis de regresión, ya que es donde la formulación del problema y el objetivo son definidos. Debido a que la incorrecta formulación del problema puede conducir a la generación errónea de un modelo, se debe poner especial énfasis en este primer paso.

Selección de variables: Es este paso, los expertos en el área de estudio, seleccionan el conjunto de variable que son más propensas a resolver el problema. Debido a que el problema puede contener demasiadas variables, el experto debe saber discriminar las menos relevantes, y seleccionar solo las más importantes.

Recopilación de datos: Los datos recopilados consisten en observaciones sobre n elementos del problema a resolver, donde cada uno de estos elementos contiene los valores para cada una de las variables que fueron seleccionadas en el paso anterior. Generalmente los datos son obtenidos de entornos reales, lo cual provoca que muy pocas veces los datos puedan ser controlados por el investigador.

Especificación del modelo a utilizar: En este paso se selecciona la forma del modelo que se va a obtener. El tipo de modelo también que se va a utilizar tiene que ser especificado por los expertos en el área. El tipo de modelo puede ser lineal, no lineal o logístico.

Elección del ajuste del modelo: Una vez que se tiene definido el modelo, la siguiente tarea consiste en estimar los parámetros del modelo basado en los datos recogidos. El método de estimación más comúnmente utilizado es el método de mínimos cuadrados.

Ajuste del modelo: El siguiente paso es estimar los parámetros de regresión, así como las valores, y ajustar el modelo de datos utilizando el método de ajuste seleccionado.

Validación del modelo: La validez de un método en el análisis de regresión, depende de ciertos supuestos. Los supuestos se hacen generalmente con los datos recolectados y el modelo creado. La exactitud del análisis y las conclusiones derivadas de un análisis depende fundamentalmente de la validez de estos supuestos. El análisis de regresión es considerado como un proceso iterativo, en el cual, las salidas generadas por el modelo se utilizan para validar la resolución del problema, por lo cual, cuando el experto considera que el modelo obtenido produce una salida satisfactoria que se ajuste de manera razonable a los datos, el modelo validado es pasado a la última fase.

Empleo del modelo creado para solucionar el problema: Una vez que un modelo obtiene cierto nivel de validación, el proceso iterativo del análisis de regresión es terminado, modelo obtenido es empleado para dar solución al problema. La Figura 2.1 representa el proceso iterativo del análisis de regresión.

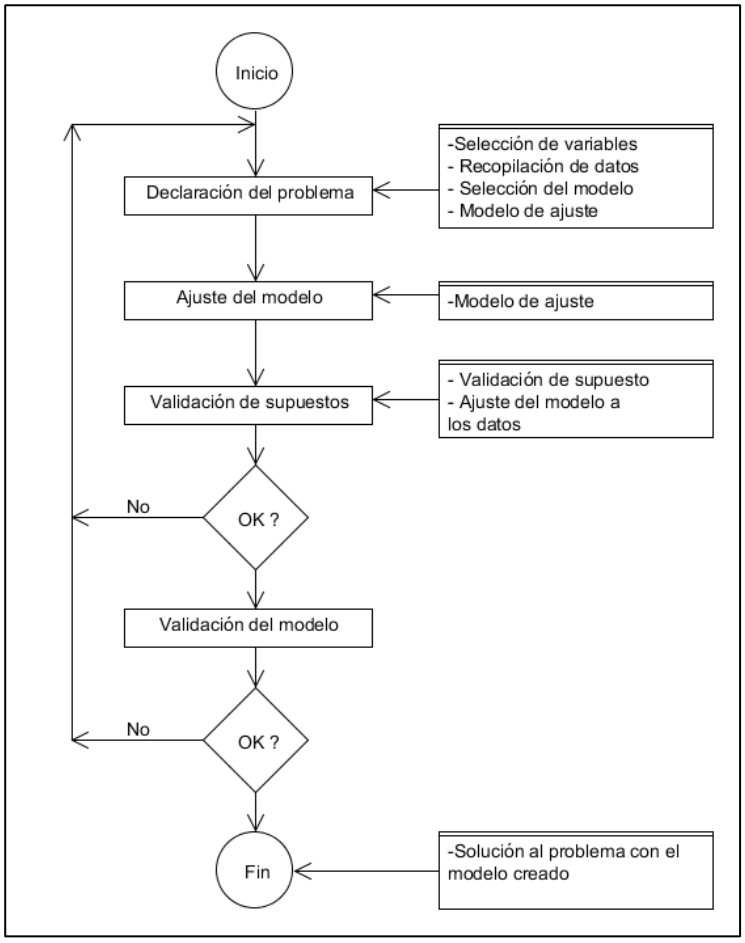
El detalle del uso de herramientas de análisis de regresión tradicionales depende del modelo a usar (lineal, no lineal, logístico) antes de la selección de variables, y es difícil justificar esta selección en modelos no lineales (Braga *et al.*, 2015). En problemas donde el análisis incluye más de una variable el proceso se vuelve más complicado (Bernal *et al.*, 2015). Por lo tanto, el análisis de regresión para grandes conjuntos de datos habitualmente se realiza por computadora (Milton, 2007).

2.6.2 Algoritmos evolutivos

Son técnicas algorítmicas basadas en la teoría de la selección natural de Darwin (Darwin, 1859) para la resolución de problemas difíciles. La teoría de la selección natural indica que los individuos que mejor se adapten a su entorno, tienen mayor probabilidad de sobrevivir y pasar sus características (las cuales le permitieron sobrevivir) a sus descendientes. En los algoritmos genéticos, al estar basados en esta teoría, los individuos son las posibles soluciones, y las características son la manera en solucionar el problema (Gestal, 2010) (Goldberg, 1989).

Estos algoritmos combinan la supervivencia del individuo más apto con métodos de intercambio de información (entre individuos) para, de esta forma, simular la evolución de una población de individuos con el fin de descubrir, de manera iterativa, mejores soluciones (Can *et al.*, 2011).

Figura 2.1 Proceso de resolución de problemas mediante el análisis de regresión



Los algoritmos evolutivos permiten dar solución a problemas difíciles, principalmente a aquellos donde el espacio de búsqueda (de posibles soluciones) es muy amplio, donde la búsqueda manual sería prácticamente imposible (Sette *et al.*, 2001) (Fogel, 2006).

Existen diferentes clases de algoritmos evolutivos:

- Algoritmos genéticos (Holland, 1975) (Goldberg, 1989): Modelan la solución del problema mediante la representación genómicas de las características de la posible solución. Cada elemento del individuo solución en una características que es o no seleccionada para general la mejor solución al problema (Holland, 1992).
- Estrategias de evolución (Schwefel, 1993): Se utilizan para modelar al mismo tiempo las variables del problema, así como los parámetros de evolución (Berlanga, 2010).
- Programación genética (Koza, 1992b) (Koza, 1994): En estos algoritmos, los individuos son representados mediante árboles sintácticos, los cuales representan programas, algoritmos o modelos que dan una solución al problema.

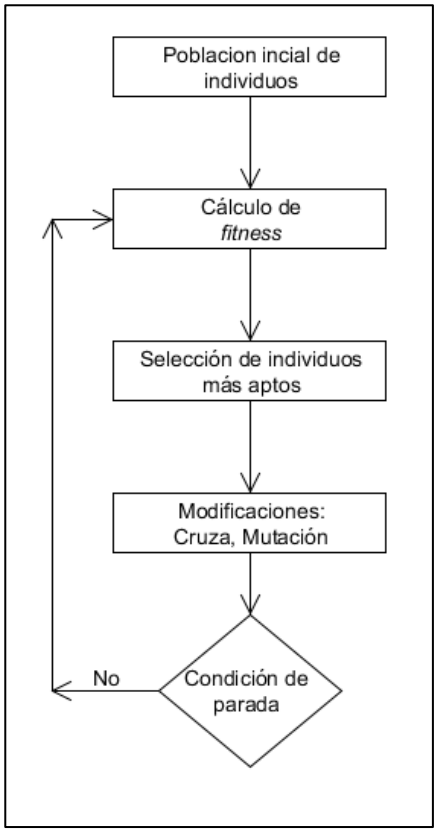
A pesar de que existen diversos tipos de algoritmos evolutivos, estos cumplen con características en común (Back *et al.*, 1997):

- Utilizan el aprendizaje colectivo de una población de individuos, donde cada individuo representa un punto dentro del espacio de búsqueda de posibles soluciones.
- Los descendientes de los individuos se generan mediante procesos no determinísticos que tratan de simular los procesos de mutación y cruza de la teoría de la selección natural. La mutación provoca cambios aleatorios en los genes del individuo y la cruza permite la creación de nuevos individuos a partir del intercambio de genes de dos individuos (Banzhaf *et al.*, 1998) (Ríos *et al.*, 2008).
- Un valor de calidad (*fitness*) es asignado a cada individuo de la población. Este valor *fitness* indica lo "bien" de la solución en el problema. El *fitness* permite además decidir qué individuos sobreviven (individuos más aptos), cuáles serán mutados, y cuales son seleccionados para generar los descendientes de la población. Cuando un individuo o la población alcanzan un valor *fitness* adecuado, el proceso de evolución se detiene.

Cualquier algoritmo evolutivo puede ser descrito de la siguiente forma: a partir de una población de posibles soluciones para el problema, se realizan modificaciones sobre las mismas y, se seleccionan aquellos individuos de acuerdo a su adaptación al entorno del problema (Berlanga, 2010). Las modificaciones que se realizan sobre la población permiten mezclar información de los padres que debe pasar a los descendientes o introducir innovación dentro de la población. El diagrama que indica el funcionamiento de este algoritmo se puede observar en la Figura 2.2.

Estos algoritmos trabajan bajo un proceso de búsqueda global, utilizando reglas probabilísticas (Goldberg, 1989), lo cual le permite no converger un punto local del proceso, logrando así un acercamiento máximo al objetivo deseado (Sette *et al.*, 2001). Aunque existen diversas clases de algoritmos evolutivos, los más utilizados son los Algoritmos Genéticos y la programación genética (Berlanga, 2010).

Figura 2.2 Pasos generales de los Algoritmos Evolutivos



2.6.3 Programación Genética

Es un área de los algoritmos genéticos tradicionales, que mantiene el mismo principio de selección natural. La programación genética pretende resolver problemas mediante la inducción de algoritmos y programas que los resuelvan (Poli *et al.*, 2008). Estos algoritmos están diseñados para encontrar de forma automática programas de computadora (donde un programa puede ser un conjunto de pasos (algoritmo), reglas gramaticales o analíticas, modelos o funciones matemáticas) que, a partir de un conjunto de valores de entrada, define un modelo de datos que describe su comportamiento (Do *et al.*, 2008).

Generalmente, la población de programas está representada por árboles sintácticos marcados (Banzhaf *et al.*, 1998) (Do *et al.*, 2008) (Barmpalexis *et al.*, 2011), donde los nodos interiores del árbol representan las funciones, pasos o métodos que son empleados, mientras que los nodos exteriores (hoja del árbol) representan los argumentos de las funciones (Brameier *et al.*, 2006). Esto permite que cualquier programa, algoritmo o expresión matemática pueda ser representado bajo un árbol sintáctico (Banzhaf *et al.*, 1998). La Figura 2.3 muestra un ejemplo de una expresión matemática representaba en un árbol sintáctico, y la Figura 2.4

representa un pequeño programa de computadora representado en un árbol sintáctico. Algo importante y sobresaliente es que, la programación genética es una técnica de cómputo evolutivo que resuelve automáticamente problemas, sin requerir del conocimiento del usuario o conocer la estructura de la solución (Koza, 1992a).

Los árboles sintácticos utilizados en la programación genética están conformados por un conjunto de datos primitivos:

- Conjunto de Terminales T: Son los nodos situados en las hojas de los árboles. Este conjunto se compone de las entradas para el programa, constantes, variables o funciones sin argumentos (Banzhaf *et al.,* 1998) (Gestal, 2010) (Koza, 1992b).
- Conjunto de funciones F: Son los nodos interiores del árbol. Son los elementos de los cuales dispondrá el programa para efectuar los cálculos sobre sus argumentos (terminales) (Ríos *et al.*, 2008). El conjunto de funciones puede incluir (Koza, 1992b):
    - Operaciones Aritméticas $(+, -, *, /)$
    - Funciones matemáticas $(Sen, Cos, Tan, Exp, Log)$
    - Operaciones booleanas (and, or, not)
    - Operaciones condicionales $(If - then - else)$
    - Funciones de iteración $(Do - For)$

Figura 2.3 Expresión matemática representada como árbol sintáctico

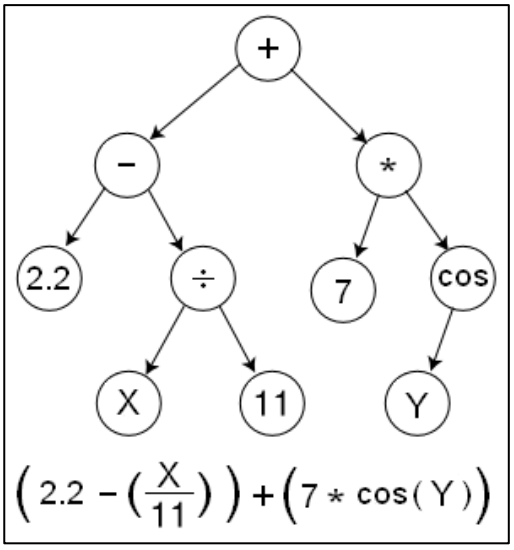
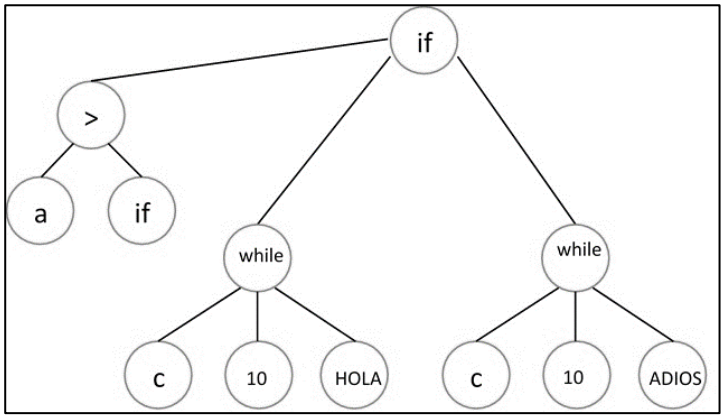


Figura 2.4 Programa de computadora representado bajo un árbol sintáctico



Un algoritmo de programación necesita de un conjunto de etapas, la cuales corresponden a las etapas utilizadas por los algoritmos genéticos, incluyendo otras específicas para la programación genética. Estas etapas son las siguientes:

1) Inicialización: La población inicial en la programación genética consiste en crear los individuos de la población, en forma de árboles sintácticos, como posibles soluciones al problema. Un parámetro importante en el proceso de inicialización de la población es indicar el tamaño máximo con el que los árboles deben ser creados.
2) Evaluación de los individuos: La evaluación permite saber qué tan aptos son los individuos en la solución problema. La aptitud (fitness) es la medida utilizada en programación genética durante la evolución, la cual indica lo bien que un programa ha aprendido a predecir las salidas a partir de las entradas (Banzhaf et al., 1998) (Gray et al., 1998), es decir, la aptitud calcula el error en los datos producidos por el programa bajo los datos del conjunto de entrenamiento. Existen métodos de evaluación muy utilizados (Do *et al.*, 2008), como el Error Absoluto, o el Error Cuadrático Medio, pero este puede ser reemplazado por cualquier método de evaluación que se considere necesario para resolver el problema.
3) Selección de los mejores individuos: Después de calcular la aptitud de todos los individuos de la población, es momento de decidir a qué individuos serán aplicados los operadores genéticos.
4) Aplicación de operadores genéticos: A los individuos seleccionados anteriormente se les aplica los operadores genéticos (con base en valores probabilísticos). Los principales operadores en la programación genética son la cruza y la mutación, pero a diferencia de los algoritmos genéticos tradicionales, estos operadores deben estar diseñados para funcionar con la población de programas, es decir, trabajar con árboles sintácticos.

5) Criterio de parada: Regularmente el criterio de parada se especifica con un número de generaciones G realizadas en el proceso de evolución; cuando se ha alcanzado el máximo ajuste de la aptitud o cuando la población llega a una convergencia (Koza, 1992b).
6) Diseño de salida: Una vez terminada la ejecución del programa genético, la solución es generada. Generalmente, esta solución es presentada como el mejor individuo (máximo ajuste al momento de terminar la ejecución del programa). Como el proceso de resolución al problema fue tratado con árboles sintácticos, es conveniente transformar esta representación a la forma correspondiente. Por ejemplo, transformar el árbol a un modelo matemático si se está trabajando con funciones matemáticas, o transforma el árbol a un programa de computadora si lo que se está esperando como solución lo es.

Existen diversas aplicaciones para la programación genética, pero sin duda la más utilizada es la regresión simbólica (Parra, 2007).

2.6.4 Regresión simbólica

La regresión simbólica es una aplicación de la programación genética que tiene el mismo objetivo que los métodos de regresión tradicionales, pero a diferencia de estos, trabaja en un espacio mayor de búsqueda y con menos limitaciones (Kommenda *et al.*, 2014). La regresión simbólica es conocida como la técnica de *Identificación de la Función*, ya que consiste en encontrar una expresión matemática, en forma simbólica, que describa la relación entre una variable dependiente y variables independientes con la mayor precisión posible (Koza, 1992b) (Murari *et al.*, 2015).

Al trabajar bajo la programación genética, la regresión simbólica se encarga de evolucionar poblaciones de individuos, los cuales representan ecuaciones, funciones o modelos matemáticos, con el fin de estimar el comportamiento de un conjunto de datos (Can *et al.*, 2011).

Las técnicas de regresión tradiciones, primero deciden el modelo de la estructura, y después resuelven los parámetros del modelo por métodos de aproximaciones (como el método de cuadrados mínimos) (Can et al., 2011). Encontrar la estructura y los coeficientes adecuados para un modelo al mismo tiempo es un reto para el cual no existe un procedimiento matemático eficiente, y es difícil definir la estructura del modelo de manera manual y *a priori*. Por otra parte, las técnicas de análisis de regresión tradicionales no son adecuadas para los problemas de modelados empíricos debido a su no linealidad y su multimodalidad. Por eso, es necesario un experto artificial el cual pueda crear un modelo a partir de los datos disponibles (Dabhi et al., 2011). Por esta razón, la regresión simbólica destaca como un enfoque viable

frente al problema de modelado de datos, ya que no asume la respuesta de una estructura, pero la descubre conforme evoluciona (Can *et al.*, 2011).

Las técnicas de análisis de regresión son útiles cuando la cantidad de datos a inducir es relativamente pequeña, o cuando la distribución de estos recaé en alguno de sus métodos (lineal, no lineal o logístico). Sin embargo, cuando la cantidad de datos es enorme y la distribución de los mismo no permite discernir qué tipo de método de análisis de regresión utilizar, es necesario contar con herramientas computacionales que puedan hacerlo, y la regresión simbólica es la opción adecuada (Can *et al.*, 2011) (Soto, 2009). Además, el proceso de descubrimiento de la solución a un problema mediante análisis de regresión y mediante regresión simbólica es el mismo, solo que el proceso es completamente manual en las técnicas de análisis de regresión (Milton, 2007).

Un proceso de regresión simbólica para la solución a un problema específico está compuesto por cinco pasos principales:

1) Declaración del problema
2) Recopilación de datos
3) Aplicación y ejecución del programa de regresión simbólica
4) Validación del modelo (Análisis)
5) Empleo del modelo creado para dar solución al problema

Declaración del problema: Es el primer paso, y quizás el más importante, donde la formulación del problema y el objetivo son definidos. Debido a que la incorrecta formulación del problema puede conducir a la generación errónea de un modelo se debe poner especial énfasis en este primer paso.

Recopilación de datos: Los datos recopilados consisten en observaciones sobre n elementos del problema a resolver, donde cada uno de estos elementos contiene los valores para cada una de las variables que serán tomadas en cuenta. Generalmente los datos son obtenidos de entornos reales.

Ejecución del programa de regresión simbólica: Se ejecuta el programa informático de regresión simbólica, y debido a que está no requiere de una definicón especifica de parámetros, puede ser ejecutado de manera sencilla. Los únicos valores a definir en este paso son aquellos que tienen que ver con el tiempo de ejecución y complejidad de la solución.

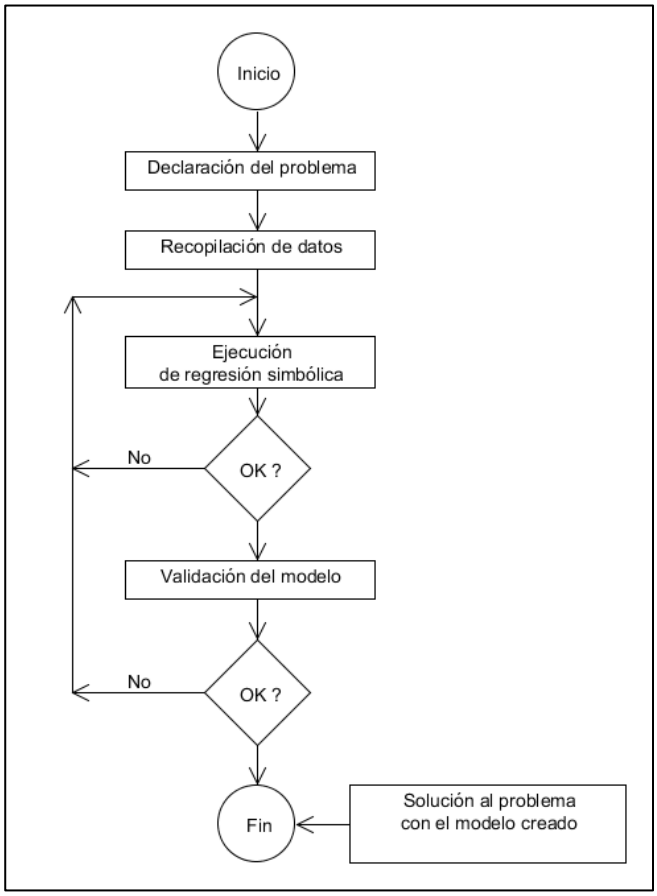
Validación del modelo: El proceso de regresión simbólica es considerado como un proceso iterativo, en el cual, las salidas generadas por el modelo se utilizan para validar la resolución del problema. Por lo cual, se considera que el modelo obtenido produce una salida

satisfactoria, es decir, se ajusta de manera razonable a los datos. Despues el modelo obtenido es pasado a la última fase.

Empleo del modelo creado para solucionar el problema: Una vez que un modelo obtiene cierto nivel de validación, éste es seleccionado, y el proceso iterativo de regresión simbólica es terminado, entonces este modelo es empleado para dar solución al problema.

En la Figura 2.5 se muestra el proceso general que sigue la regresión simbólica. Si se compara la Figura 2.1 (Proceso del análisis de regresión) con la Figura 2.5, se puede notar que, como se mencionó anteriormente, los procesos del análisis de regresión y la regresión simbólica con prácticamente lo mismo. Sin embargo, mientras que el análisis de regresión es un proceso manual y hace uso de un experto, la regresión simbólica es un proceso computacional, lo cual reemplaza el conocimiento del experto por un proceso de aprendizaje supervisado basado en un algoritmo evolutivo.

Figura 2.5 Proceso de regresión simbólica



2.7 Resumen

Es este capítulo se han introducido los conceptos básicos que se utilizan a lo largo de este trabajo. Se comenzó hablando de lenguaje natural, el cual puede ser definido como el medio por el cual las personas logran la comunicación además de la transmisión de información. Se describieron las áreas del conocimiento encargadas del estudio de los lenguajes naturales, principalmente, el procesamiento de lenguaje natural.

El procesamiento de lenguaje natural puede ser entendido como la habilidad que tiene una computadora, método o algoritmo de realizar la correcta comprensión de los lenguajes natural. Para lograr una correcta interpretación del lenguaje natural por medio de sistemas de procesamiento de lenguaje natural, es necesario la aplicación de los diversos niveles de lenguaje, los cuales abarcan desde el entendimiento morfológico de las palabras, hasta el nivel en que estas son utilizadas para mantener y realizar conversaciones.

También se describieron las dos principales aproximaciones para el desarrollo de sistemas de procesamiento de lenguaje natural, las cuales son aquellos basados en reglas, y aquellos basados procesos y modelos estadísticos. Más adelante fueron descritos los tres componentes principales de los sistemas de procesamiento de lenguaje natural: representación de la información del lenguaje, función de similitud, y paradigma de evaluación. Estos tres componentes fueron descritos, así como las diversas formas de aplicación de cada uno dentro de los sistemas de procesamiento de lenguaje natural.

Una de las principales áreas del conocimiento que se encarga del estudio del procesamiento del lenguaje natural es el *machine learning*. El *machine learning* hace referencia a la detección automática de patrones en conjuntos de datos, y dado que el lenguaje puede ser entendido como un conjunto de datos, es uno de los caminos principales para el desarrollo de sistemas de procesamiento de lenguaje natural.

Dado que el aprendizaje automático tiene sus bases en la estadística clásica, fueron descritos los principales conceptos útiles para entender el funcionamiento básico de herramientas de aprendizaje automático. Una de las herramientas del aprendizaje automático, son los algoritmos evolutivos, los cuales son técnicas de aprendizaje basadas en el proceso evolutivo y la supervivencia de individuos aptos.

Existen diversos tipos de algoritmo evolutivos, pero uno de los principales es la programación genética, la cual es una técnica de resolución de problemas mediante la inducción de programas, modelos o funciones que permitan resolverlos. Una de las principales aplicaciones de la programación genética, es la regresión simbólica.

La Regresión Simbólica tiene el mismo objetivo que los métodos de regresión tradicionales, es decir, encontrar un modelo que se ajuste a la relación que existe en un conjunto de datos; pero la regresión simbólica trabaja con menos limitaciones. Dado que la regresión simbólica

trabaja mediante programación genética, es posible inducir modelos matemáticos que permitan dar solución a problemas donde se hacen uso de grandes cantidades de datos.

CAPÍTULO 3.

Funciones de similitud

En este capítulo se describen de manera más específica los términos y elementos relacionados a las funciones de similitud, sus características, tipos, y su uso en las tareas de procesamiento de lenguaje natural.

Existen diferentes definiciones para el término *función de similitud* los cuales varían de acuerdo con el área de aplicación donde se utilice (matemáticas, física, psicología, lenguaje, etc.), en este caso, nos enfocaremos en las definiciones que son útiles para el procesamiento de lenguaje natural.

3.1 Funciones de similitud y distancia

De manera general, una medida de similitud es una expresión matemática que permite expresar mediante un dato numérico el grado de relación entre dos elementos, su semejanza o su desigualdad con base en sus cualidades o cantidades de atributos de estos dos elementos (Moreno, 2000). El concepto de similitud es básico para las experiencias humanas, usualmente es necesario saber que tan similares son dos cosas, objetos, ideas, personas, o cualquier otro tipo de comparación cotidiana, que bien puede interpretarse como similitud o diferencia (disimilitud), lo cual nos introduce otro concepto, disimilitud. El término disimilitud está más relacionado a la interpretación real de la palabra distancia, la cual, expresa el grado de cercanía que tiene un elemento de respecto a otro, siendo medido en longitud, separación tiempo o diferencia (Deza et al., 2009).

Ambos términos, funciones de similitud y funciones de distancia son parte esencial en diversas áreas de investigación, las cuales abarcan desde Matemáticas, Geometría, Estadística, Teoría de Grafos, Análisis de datos, Reconocimiento de Patrones, Procesamiento de Lenguaje, Astronomía, Biología, por mencionar algunas (Deza et al., 2009) (Chen et al., 2009) (Bellet et al., 2015). Es por eso que es uso de las medidas de similitud y distancia son muy importantes en muchos de los procesos cognitivos y desarrollo humanos (Tversky, 1977).

El uso de funciones de similitud o funciones de distancia depende del contexto del área donde se van a aplicar. Así, por ejemplo, algunas medidas de similitud están diseñadas para trabajar con datos biológicos (como secuencias de ADN), éstas no pueden ser aplicadas directamente para su uso con datos de númericos, es por eso, que se necesita saber las características de cada tipo, para saber cuándo aplicar una función de similitud o distancia (Nosofsky, 1986) (Nowak et al., 2007).

A continuación, se detallan las características que tienen las funciones de similitud y funciones de distancia.

3.1.1 Características de las funciones de similitud

Debido a que las funciones de similitud son utilizadas en varias áreas de investigación, sus propiedades son variadas su definición es confusa (Stojmirović et al., 2007). A continuación, se muestra una definición formal para el término *función de similitud*.

Dado un conjunto X, una función de similitud $S(x, y)$ sobre el producto cartesiano $X \times X$ es una métrica de similitud si, para toda $x, y, z \in X$ satisface las siguientes condiciones (Chen et al., 2009) (Goshtasby, 2012):

1) $S(x, y) = S(y, x)$
2) $S(x, x) \geq 0$
3) $S(x, x) \geq S(x, y)$

4) $S(x,y) + S(y,z) \leq S(x,z) + S(y,y)$
5) $S(x,x) = S(y,y) = S(x,y)\ si\ y\ solo\ si\ x = y$

Éstas son algunas de las propiedades formales de las funciones de similitud, en las cuales, la propiedad 1 indica la propiedad simétrica, la cual establece que el valor obtenido por una medida de similitud cuando se compara x contra y y viceversa, debe ser la misma. La condición indica que al compararse un valor consigo mismo, el resultado debe ser mayor o igual que cero, además indica, que el valor mínimo obtenido por una función de similitud es cero. La tercera condición indica que para cualquier x la similitud de sí mismo no es menor que la similitud entre x y y. La cuarta condición indica que la similitud entre x y z a través de y no es mayor que la similitud directa entre x y z más la similitud de y consigo mismo. La condición cinco, indica una equivalencia de resultados cuando los elementos x y y son los mismos (Chen et al., 2009).

Cuando una medida de similitud es usada en alguna aplicación, ésta es tomada como un requerimiento no formal (Chen et al., 2009).

3.1.2 Características de las funciones de distancia

Una función de distancia puede ser descrita como (Chen et al., 2009) (Ming et al., 2004): D es una función de distancia con valores reales no negativos, definida sobre el producto cartesiano de $X \times X$ de un conjunto X. Es llamada una métrica sobre X si para cada $x, y, z \in X$ se cumple que:

1) $D(x,y) \geq 0\ (nunca\ un\ valor\ negativo)$
2) $D(x,y) = D(y,x)\ (Axioma\ de\ simetria)$
3) $D(x,z) \leq D(x,y) + D(y,z)\ (desigualdad\ del\ triángulo)$
4) $D(x,y) = 0, si\ y\ solo\ si\ x = y\ (identidad\ de\ lo\ imperseptible)$

Donde la primera condición indica que cualquier función de distancia que sea implementada entre dos elementos, siempre debe regresar un valor mayor o igual a cero, puesto que no existen distancias negativas. La segunda condición establece la simetría, tal que, cuando dos elementos x y y sean comparados bajo una función de distancia, el valor devuelto será el mismo que comprar y y x. La tercera condición indica la desigualdad del triángulo, e indica que la comparación entre dos elementos x y z debe ser menor o igual que comparar estos elementos a través de y. Y la utima condición indica que cuando dos elementos que son iguales son compardos, el valor de su distancia es cero.

Para evitar discusiones respecto al tema de funciones de similitud y distancia, en este trabajo se utiliza el término f*unciones de similitud* para hacer referencia a las métricas empleadas (ya sean de similitud o distancia).

3.1.3 Funciones de similitud entre textos

Medir la similitud entre textos, palabras, oraciones o párrafos es una de las cuestiones más estudiadas en las tareas de procesamiento de lenguaje natural (Metzler *et al.*, 2007) (Bär *et al.*, 2015b). Generalmente, la similitud entre dos textos puede ser medida a través de la similitud que exista entre la representación de sus palabras, oraciones o párrafos. Estas representaciones pueden ser similares ya sea de manera léxica o semántica (Gomaa *et al.*, 2013).

La representación de la información de los textos es parte fundamental para saber su similitud, y la representación. Como se mencionó anteriormente, la representación de la información está basada en las características que, se extraen del texto (Grace *et al.*, 2016).

3.2 Clasificación de las funciones de similitud

La elección de una buena medida de similitud es igual de importante que la elección de la representación del texto (Hartigan, 1975). Existen diversas clasificaciones de funciones que permiten calcular la similitud entre dos textos.

Una clasificación es mostrada en (Gomaa *et al.*, 2013), quien solo indica tres tipos:

1) Basadas en secuencias de cadenas de texto: Éstas operan sobre secuencias de cadenas y caracteres. Estas funciones indican que dos textos son similares si comparten varias secuencias de caracteres iguales o aproximadas.
2) Basadas en corpus: Son funciones de similitud que toman en cuenta la información semántica de los documentos. Éstas obtienen la similitud con base en la información relacionada obtenida de grandes conjuntos de datos (corpus).
3) Basadas en conocimiento: Éstas funciones identifican el grafo de similitud entre dos textos mediante la información obtenida de redes semánticas previamente desarrolladas por expertos. Éstas identifican la relación que pueda existir entre los conceptos de dos palabras. Una de las redes semánticas más populares es Wordnet (WordNet, 2016).

Otra clasificación de las funciones de similitud es mostrada en (Yih *et al.*, 2007), quien de igual forma propone cuatro tipos:

1) *Surface matching:* Son funciones de similitud, las cuales indican que dos textos son similares de acuerdo con la cantidad de términos (palabras, caracteres) que ocurren en ambos textos. Generalmente, las operaciones realizadas por estas funciones de similitud están basadas cuando las dos cadenas de texto están representadas bajo vectores binarios, donde cada elemento del vector indica si el término está en el texto o no.

2) Basadas en corpus: Éstas funciones toman en cuenta la información de si los términos de los dos textos co-ocurren en el mismo documento de un corpus.
3) *Query-log*: Éstas funciones miden la similitud de dos textos con base en su registro de consultas almacenadas por los grandes Motores de Búsqueda (Google[2], Yahoo![3]). Estos motores de búsqueda reciben miles de consultas diariamente, por lo cual, son indicadores de que tan parecidos pueden ser dos textos. Básicamente, miden la similitud con base en la cantidad de palabras que comparten los dos textos, y que aparecen en la misma sesión de consultas de un usuario en el motor de búsqueda.
4) *Web-relevance:* Ésta se basa en la frecuencia de aparición de los términos de ambos textos que aparecen en los documentos devueltos por un motor de búsqueda. Básicamente, construye una representación vectorial, basándose en las características obtenidas, a partir de los *n* documentos devueltos por una consulta en un motor de búsqueda. Los *n* documentos se obtienen a partir de ingresar los términos de los textos como consulta en un motor de búsqueda.

Otra clasificación es presentada por (Bär *et al.*, 2015b):

1) Basadas en el contenido: las cuales se subdividen en dos: a) Medidas composicionales, las cuales *tokenizan* los textos de entrada, calculan la semejanza ente todas las palabras entre ambos textos, y al final agregan los valores resultantes a una puntuación de similitud general; b) No composicionales, las cuales, primero representan los textos bajo un modelo específico (ejemplo, modelo vectorial), y después comparan los valores de estas representaciones para obtener el valor de similitud. Estas últimas, tienen dos pasos principales, el modelo de representación de los textos, y el cálculo de similitud entre ellos.
2) Basadas en estructura: Estas funciones se basan en cómo está constituido un texto, el orden de los términos que utiliza, la estructura de su composición, el orden de sus secciones. También pueden ser representadas bajo un modelo de representación de textos. Ejemplo de estas medidas son aquellas cuyas características de los textos están representadas por *n-gramas*.
3) Basadas en el estilo: Estas funciones se basan en el estilo de redacción de los textos. Por ejemplo, pueden tomar en cuenta la riqueza del vocabulario, las variaciones entre las longitudes de las oraciones, la posición de los términos dentro de los textos, la cantidad

[2] google.com

[3] Yahoo.com

de oraciones en los textos y la cantidad de términos por oraciones, la frecuencia de las palabras, entre otras.

Como se describió, existen diversas clasificaciones para las funciones de similitud, donde cada autor indica qué características deben de contener estas medidas para ser utilizadas. Algunas son utilizadas directamente sobre el texto, otras necesitan de un modelo de representación para poder ser aplicadas.

Esta investigación está basada en las funciones de similitud que hacen uso de modelos de representación de textos para calcular la similitud. A continuación, se presenta una clasificación, resultado de esta investigación, de funciones de similitud.

3.2.1 Basadas en datos booleanos

La manera más sencilla de representar las características de un texto es mediante valores binarios $\{0,1\}$. Las características de los textos son almacenas en vectores, donde la presencia de la característica en el texto se representa con 1, mientras que la ausencia de una característica con 0. Esta representación conocida generalmente como Modelo Booleano (Stahl, 2004) (Avello, 2005).

La primera representación booleana de las características de elementos fue propuesta por Jaccard (Jaccard, 1901), y desde entonces se han propuesto diversas funciones de similitud basadas en representación booleana para distintos campos (Choi *et al.*, 2010).

Las funciones de similitud para datos binarios son expresadas de la siguiente forma: suponga que dos elementos (textos) i y j que están representados por valores booleanos en forma de vectores, entonces n es el numero de características (o atributos) diferentes de ambos vectores, y los elementos que son utilizados por las funciones de similitud están definidos por a, b, c, d, tal como se muestra en la Tabla 3.1. Esta tabla es conocida como *OTUs* (*Operational Taxonomic Units*) (Dunn *et al.*, 2004)

Tabla 3.1 Expresión de los valores para los elementos i y j

	Elemento i			
Elemento j		$1\ (Presencia)$	$0\ (Ausencia)$	$Suma$
	$1\ (Presencia)$	a	b	$a+b$
	$0\ (Ausencia)$	c	d	$c+d$
	$Suma$	$a+c$	$b+d$	$n=a+b+c+d$

Donde:

a = Es el número de características que están presentes tanto en el elemento i como en j.

b = Es el número de características presentes en j pero ausentes en i.

c = Es el número de características presentes en i pero ausentes en j.

d = Es el número de características que están ausentes tanto en el elemento i como en j.

Un gran número de medidas de similitud han sido descritas en la literatura, en el trabajo de (Choi *et al.*, 2010) se presentan 73 medidas de similitud diferentes. A continuación, se enlistan las funciones de similitud para datos booleanos que son utilizadas en este trabajo.

3.2.1.1 Distancia de Jaccard

El índice de Jaccard (Jaccard, 1901) (Leydesdorff, 2008) (Hamers *et al.*, 1989) (Cha, 2007), también conocido como el coeficiente de similitud de Jaccard es una estadística utilizada para comparar la similitud y la diversidad de los datos.

$$ja\ (x,y) = \frac{A}{A+B+C}$$

Esta medida calcula el número de términos compartidos sobre el número de todos los términos únicos que existen en ambos elementos (Gomaa *et al.*, 2013) (Jaccard, 1901).

3.2.1.2 Distancia de Hamming

En teoría de la información, la distancia de Hamming (Hamming, 1950) entre dos cadenas de igual longitud es el número de posiciones en las que los símbolos correspondientes son diferentes (Hamady *et al.*, 2008) (Georgiou *et al.*, 2008) (Mukherjee, 2014).

$b + c$	Completo
$\frac{(a+d)-(b+c)}{a+b+c+d}$	Normalizado

3.2.1.3 Similitud Russell-Rao

Este índice es simplemente la proporción de casos en los que las observaciones tienen el rasgo de interés (Russell *et al.*, 1940) (Decker *et al.*, 2016).

$$rr(x,y) = \frac{1}{a+b+c+d}$$

3.2.1.4 Similitud Sokal-Sneath

Es un índice en el que el doble de peso se les da a los elementos que no coinciden, y las ausencias conjuntas se excluyen (Mukherjee, 2014). Existen seis variantes derivadas en esta medida, a continuación, se muestran.

1. $\frac{2*(b+c)}{a+(2*(b+c))}$ (Wolfram, 2016)

2. $\frac{2*(a+d)}{(2*(a+d))+b+c}$ (Sokal *et al.*, 1963)
3. $\frac{a}{a+(2*(b+c))}$ (Sokal *et al.*, 1963)
4. $\frac{a+d}{b+c}$ (Sokal *et al.*, 1963)
5. $\frac{\left(\frac{a}{a+b}\right)+\left(\frac{a}{a+c}\right)+\left(\frac{d}{b+d}\right)+\left(\frac{d}{c+d}\right)}{4}$ (Sokal *et al.*, 1963)
6. $\frac{a*d}{sqrt(q)}$ $q = (a+b)*(a+c)*(b+d)*(c+d)$ (Sokal *et al.*, 1963)

3.2.1.5 Similitud Dice

También conocido por otros nombres tales como el índice de Sørensen (Sørenson, 1948), Coeficiente de Dice (Dice, 1945). Es un estadístico utilizado para comparar la similitud de dos muestras (Pinto *et al.*, 2007) (Gragera *et al.*, 2016). Se define como el doble de los términos comunes en ambos elementos, dividido por el número de total de términos que existen en ambos elementos (Gomaa *et al.*, 2013).

$$dice(x,y) = \frac{2c}{A+B} = \frac{2|A \cap B|}{|A|+|B|}$$

$$dice(x,y) = \frac{2*A}{2*a+b+c}$$

3.2.1.6 Similitud Kulczynski

(Kulczynski, 1927) siguiere un coeficiente basado en los datos de presencia-ausencia, es decir, divide las coocurrencias entre las diferencias, y su valor oscila entre cero y el infinito.

$$k\,(x,y) = \frac{a}{b+c}$$

3.2.1.6 Similitud Roger-Tanimoto

A las diferencias se les da más peso que a las semejanzas, esta función produce resultados con valores entre cero y uno (Rogers *et al.*, 1960) (Balestre *et al.*, 2008):

$$rt\,(x,y) = \frac{a+d}{a+d+2(b+c)}$$

3.2.1.4 Similitud en Matching

Es una medida sencilla bien conocida usada para medir los datos categóricos (Yang *et al.*, 2005) (Gan *et al.*, 2007) (Kaur *et al.*, 2014). Sean x y y dos conjuntos de valores categóricos, entonces la distancia correspondiente entre x y y está dada por:

$$sim = \frac{b+c}{a+b+c+d}$$

$$sim(x,y) = \begin{cases} 0 \; if \; x = y \\ 1 \; if \; x \neq y \end{cases}$$

3.2.1.9 Similitud Yule

Se utiliza para comparar los valores de dos elementos (Yule, 1900), y produce resultados en el rango de $[-1,1]$, donde un valor negativo indica que existen una mayor cantidad de diferencias que de semejanzas (Salazar, 2000).

$$sim(x,y) = \frac{(a*d)-(b*c)}{(a*d)+(b*c)}$$

Históricamente, la funciones de similitud para datos booleanos han tenido buen desempeño en los campos en los que han sido aplicadas, pero la elección de la mejor función de similitud booleana depende de las necesidades de los investigadores (Choi *et al.*, 2010).

3.2.2 Basadas en datos numéricos

Existen diversas maneras de representar las características de los textos en un vector, en el apartado anterior se indicó que una de las representaciones más utilizadas ha sido el enfoque booleano, el cual solamente indica la ausencia o presencia de la característica dentro del elemento (Stahl, 2004). Pero otra de las representaciones más utilizadas es aquella en que las características de los elementos están representadas por valores numéricos.

Las funciones que utilizan valores numéricos para calcular la similitud entre elementos son variadas, y de igual manera, han sido generadas a partir del análisis de estudio de otras aplicaciones.

A continuación, se presentan algunas de las funciones de similitud basadas en datos numéricos más destacadas.

3.2.2.1 Coeficiente de Pearson

El coeficiente de correlación de Pearson entre dos puntos de datos se define como la covarianza de los dos puntos divididos por el producto de sus desviaciones estándar (Benesty *et al.*, 2009) (Hauke *et al.*, 2011). La correlación de Pearson puede considerarse como la línea de mejor ajuste entre los puntos de un conjunto dado (Gadge *et al.*, 2015).

$$r = \frac{N \sum xy - (\sum x)(\sum y)}{\sqrt{((N\sum x^2) - \sum x^2)((N\sum y^2) - \sum y^2)}}$$

Donde:

$$N \; es \; el \; N\acute{u}mero \; de \; pares$$

$$\sum xy \; es\ la\ suma\ de\ los\ productos\ de\ los\ pares$$

$$\sum x \;\ es\ la\ suma\ de\ los\ valores\ de\ x$$

$$\sum y \;\ es\ la\ suma\ de\ los\ valores\ de\ y$$

3.2.2.2 Similitud Coseno

Coseno similitud es una medida de similitud entre dos vectores en un espacio que mide el coseno del ángulo entre ellos. El coseno del ángulo 0 ° es 1, y es inferior a 1 para cualquier otro ángulo. Por lo tanto, es un juicio de orientación y no magnitud: dos vectores con la misma orientación tienen una similitud coseno de 1, dos vectores a 90 ° tener una similitud de 0, y dos vectores diametralmente opuestas tienen una similitud de -1, independientemente de su magnitud. La similitud Coseno se utiliza sobre todo en el espacio positivo, donde el resultado está perfectamente delimitado en [0,1] (de Assis *et al.*, 2007) (Ravichandran *et al.*, 2005) (Sidorov *et al.*, 2014).

$$similarity = \cos(\emptyset) = \frac{x * y}{|x||y|} = \frac{\sum_{I=1}^{N} x_i * y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \;\ * \;\ \sqrt{\sum_{i=1}^{n}(y_i)^2}}$$

Donde:

x_i y y_i son componentes del vector x y y respectivamente.

3.2.2.3 Distancia Euclidiana

Con esta distancia, el espacio euclidiano se convierte en un espacio métrico. La norma asociada se denomina norma euclidiana (Black, 2004) (Saito *et al.*, 1994) (Dattorro, 2010).

$$d(x, y) = \sqrt{(x_1 - y_1)^2 \;\ + \;\ (x - y_2)^2 \; + \cdots + (x_n - y_n)^2}$$

Donde:

$$p\ y\ q\ son\ los\ vectores\ euclidianas$$

3.2.2.4 Distancia Manhattan

La distancia Manhattan es la simple suma de las componentes horizontal y vertical, mientras que la distancia diagonal puede ser calculado mediante la aplicación del teorema de Pitágoras. El nombre de la distancia de Manhattan se basa en el diseño de la red de calles en la isla de Manhattan (Chang *et al.*, 2009) (Perlibakas, 2004).

$$d = \sum_{i=1}^{n} |xi - yi|$$

Donde:

$$n\ es\ el\ número\ de\ variables$$

$$xi, yi\ son\ los\ valores\ de\ i\ en\ los\ puntos\ X\ y\ Y$$

3.2.2.5 Distancia Minkowski

La distancia de Minkowski (Merigó *et al.*, 2008) (Merigó *et al.*, 2011) (Arroyo *et al.*, 2015) es una métrica en un espacio vectorial que puede considerarse como una generalización tanto de la distancia euclidiana y la distancia Manhattan.

$$d(x-c) = \sqrt[p]{\sum_{j=1}^{n} |x_{sj} - x_{tj}|^p}$$

Donde:

$$p\ es\ el\ valor\ escalar\ positivo\ del\ exponente$$

s y t = Son los índices de los campos de los vectores x y j de la columna con el mismo valor en x.

3.2.2.6 Distancia Chebychev

La métrica de Chebychev (métrica reticular, métrica de tablero de ajedrez, métrica del movimiento del rey o 8-metric), en donde se considera el valor máximo del valor absoluto de las diferencias de los valores de las muestras x y y (Deza *et al.*, 2009) :

$$||x_n - y_n|| = max\{|x_1 - y_1|, |x_2 - y_2|\}$$

3.2.2.7 Distancia Canberra

Como una métrica de permutaciones en grupos, la distancia de Canberra es una medida del desorden de listas clasificadas, donde las diferencias de rango en las primeras posiciones tienen que pagar "multas" superiores a los de la parte inferior de las listas. La distancia Canberra es la suma de los valores absolutos de las diferencias entre las filas dividido por su suma (Lance *et al.*, 1967) (Jurman *et al.*, 2009) (Emran *et al.*, 2001).

$$Ca(x,y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

3.2.2.8 Distancia Spearman-Footrule

Spearman Footrule puede ser pensada como una medida de la desorganización de una permutación (Diaconis *et al.*, 1977). Dada una permutación σ sobre n elementos, Spearman

Footrule Distance $F(\sigma)$ es la suma de las diferencias absolutas entre i y $\sigma(i)$ sobre todos los valores de i:

$$F(\sigma) = \sum_{i=1}^{n} |\sigma(i) - i|$$

3.2.2.9 Correlación Spearman-Rank

Esta es una técnica utilizada para medir el grado de asociación (correlación) entre dos conjuntos de variables (Lyerly, 1952). Por lo tanto, se supone que existe una relación entre una variable x y y (Chen *et al.*, 2002).

$$r_s = 1 - [\frac{6 \sum d^2}{n^3 - n}]$$

Donde:

$$d \text{ es la diferencia entre los valores}$$

$$d^2 \text{ es el cuadrado de la diferencia}$$

$$\sum d^2 \text{ es el total de las diferencias cuadradas}$$

$$n \text{ es el número de pares de variables}$$

3.2.2.10 Similitud Dynamic Time Warping

Reduce al mínimo los efectos del desplazamiento y distorsión en el tiempo, permitiendo la transformación "elástica" de la serie del tiempo con el fin de detectar formas similares con diferentes fases. Dadas dos series de tiempo $x = (x_1, x_2, \dots x_N), N \in \mathbb{N}$ y $y = (y_1, y_2, \dots x_M), M \in \mathbb{N}$ que representan las series de valores, DTW obtiene uno solución óptima en tiempo $O(MN)$ que podrían mejorarse más con técnicas (Senin, 2008) (Li *et al.*, 2010).

$$DTW(X,Y) = \min W \left\{ \sum_{K=1}^{K} d_k, W = \langle w_1, w_2, \dots, w_k \rangle \right\}$$

Donde:

$$d_k = d(x_i, y_j) \text{Indica la distancia representada como } w_k = (i,j) \text{ sobre la trayectoria.}$$

$$DTW(\langle\,\rangle, \langle\,\rangle) = 0,$$

$$DTW(X, \langle\,\rangle) = DTW(\langle\,\rangle, Y) = \infty,$$

$$DTW(X,Y) = d(x_i, y_j) + \min \begin{cases} DTW\ (X, Y[2:-]), \\ DTW\ (X[2:-], y), \\ DTW\ (X[2:-], Y[2:-]) \end{cases}$$

Donde:

$\langle\ \rangle$ *Indica una serie de tiempo vacia*

$[2:-]$ $Indica\ un\ sub\ array\ el\ cual\ incluye\ los\ elementos\ del\ segundo\ elemento\ en\ un\ vector$
$d(x_i, y_j)$ $Indica\ la\ distancia\ entre\ los\ puntos\ x_i$ y y_j

el cual puede ser representado con diferentes medidas de distancia (Euclidiana, Manhatan).

3.2.2.11 Índice de Consistencia

Está definido por (Kuncheva, 2007) (Cavallo *et al.*, 2010):

$$I_c(A,B) = \frac{r - \frac{k^2}{n}}{k - \frac{k^2}{n}} = \frac{rn - k^2}{k(n-k)}$$

Donde:

$$n\ es\ el\ número\ de\ Características$$

$$k\ es\ el\ número\ de\ carcaterísticas\ en\ A\ y\ B$$

$$r\ es\ el\ número\ de\ características\ en\ común\ entre\ A\ y\ B$$

3.2.3 Basadas en cadenas o secuencias de caracteres

Estas funciones pueden ser divididas en funciones de edición, y funciones de alineación de secuencias.

Las funciones de edición han sido ampliamente utilizadas para calcular la similitud entre dos objetos, estructuras o textos. Las funciones de similitud se encargan de elegir, entre un conjunto de operaciones válidas, el menor número de cambios necesarios para transformar uno de los elementos en el otro. Las operaciones más comunes son *eliminación, inserción y sustitución* (Zesch *et al.*, 2010). Ejemplo de estas funciones son la distancia de la subsecuencia común más larga (LCS), Levenshtein, y Damerau-Levenshtein.

Las funciones de alineación de secuencias originalmente fueron diseñadas para alinear secuencias biológicas (ADN, ARN, estructuras de proteinas), pero debido a su diseño, es fácilmente adaptarlas a la alineación de caracteres de textos. En estas funciones, las secuencias se representan en vectores, que comienzan cada uno con el tamaño original del elemento, entonces las funciones tratan de alinear sus caracteres para que las zonas similares queden en las mismas posiciones de los vectores (Carmona, 2014). Para hacer esto, pueden ser insertados espacios en blanco. Ejemplo de estas funciones son Needlema-Wunch y Smith-Waterman.

3.2.3.1 Distancia Longest Common Subsequence

Sean A y B dos cadenas sobre Σ un alfabeto finito de tamaño σ, con longitudes $m = |A|$ y $n = |B|$ (Sin pérdida de generalidad, y $m \leq n$). Una subsecuencia común más larga (LCS) de A y B es una subsecuencia de A y B de tal manera que ninguna otra subsecuencia común tiene mayor longitud (Lin *et al.*, 2004) (Gondree *et al.*, 2009).

$$L[i,j] = \begin{cases} 0 \; if \; i = 0 \; o \; j = 0 \\ L[i-1, j-1] + 1 \; if \; A[i] = B[j] \\ max(L[i-1,j], L[i,j-1]) \; de \; otra \; forma \end{cases}$$

3.2.3.2 Distancia de Levenshtein (Edición)

La distancia de edición (o Levenshtein) entre dos palabras es el menor número de sustituciones, inserciones, y eliminaciones de símbolos que se pueden usar para transformar una de las palabras en la otra (Levenshtein, 1966) (Konstantinidis, 2007).

$$dist(u,v) = min\{weight(h) \mid h \in E*, inp(h) = u, out(h) = v\}$$

3.2.3.3 Distancia Damerau-Levenshtein

Frederick J. Damerau (Damerau, 1964) mejoró el algoritmo de Levenshtein con una operación adicional para comprobar la distancia entre las cadenas, eso es todo, una transposición de dos caracteres adyacentes (Hyyrö, 2003).

$$D[i,0] = 1, D[0,j] = j,$$

$$d[i,j] = \begin{cases} D[i-1,j-1], if \; A[i] = B[j] \\ D[i-1,j-1], if \; A[i-a..i] = B^R[j-1..j] \; and \; D[i-1,j-1] > d[i-2,j-2] \\ 1 + \min(D[i-1,j-1], D[i-1,j], D[i,j-1]), de \; otra \; forma \end{cases}$$

3.2.3.4 Similitud Needleman-Wunsch

Básicamente, el concepto detrás del algoritmo Needleman-Wunsch (Needleman *et al.*, 1970) se deriva de la observación de que cualquier sub-ruta parcial que tiende a un cierto punto de la ruta óptima, debe ser ella misma el camino óptimo que conduce hasta ese momento. Por lo tanto, la ruta óptima se puede determinar por extensión incremental de la sub-rutas óptimas. En una alineación Needleman-Wunsch, el camino óptimo debe "estirar" de principio a fin en ambas secuencias (de ahí el término "alineación global") (Blanco, 2009) (Muhamad *et al.*, 2015).

$$NW_{a,b}(i,j) = max \begin{cases} NW_{a,b}(i-1,j) + dx \; , Hueco \; en \; la \; secuencia \; a \\ NW_{a,b}(i,j-1) + dx \; , Hueco \; en \; la \; secuencia \; b \\ NW_{a,b}(i-1,j-1) + S(a(i), b(j))x, Alinear \; a_i, b_i \end{cases}$$

3.2.4 Basadas en información semántica

Dos textos son semánticamente semejantes si significan lo mismo (sinónimos), si son opuestos ente sí (antónimos), si se utilizan en el mismo contexto o si se puede deducir un texto del otro. Entonces, las funciones de similitud semántica son calculadas a partir de la información que está relacionada en redes semánticas, con case base en la información obtenida de grandes conjuntos de datos (Gomaa *et al.*, 2013). Se supone que, la similitud semántica entre dos textos está relacionada con la información que estos comparten, por lo cual, entre más información compartan los textos, o existan más documentos dentro una colección en los que ambos textos aparezcan, mayor será la similitud (Meng *et al.*, 2014).

A continuación, se presentan algunas funciones de similitud basadas en información semántica que son útiles en la investigación presentada en este trabajo.

3.2.4.1 DISCO

La similitud distributiva, supone que las palabras con significado similar ocurren en un contexto similar, por lo cual, las grandes colecciones de texto se analizan estadísticamente para obtener una similitud distribucional (Gomaa *et al.*, 2013). DISCO (*Extracting DIStributionally similar words using COoccurrences*) es un método que calcula la similitud distributiva entre textos usando una ventana de contexto de palabras de tamaño ± 3 para contar las co-ocurrencias.

Cuando dos textos son sometidos a la similitud DISCO, ésta recupera los vectores de términos de los textos a partir de los datos indexados y calcula la similitud de acuerdo a la medida Lin (Lin, 1998). EL índice que utiliza DISCO es una base de datos de similitudes de términos. Esta base de datos es comúnmente llamada *word space*, la cual contiene por cada término indexado un vector de término (*word embedding*) con sus términos más similares, es decir, aquellos términos cuyo vector de términos es muy similar al vector de términos del término objetivo.

Esta herramienta de similitud es una aplicación basada en JAVA que permite calcular la similitud semántica entre palabras o frases, donde las similitudes son calculadas con base en el análisis estadístico de largas colecciones de texto (Kolb *et al.*, 2017). DISCO tiene dos medidas de similitud principales:

- *First order*: que calcula la similitud semántica entre dos textos de entrada con base en sus vectores de términos, donde la similitud entre vectores puede ser calculada mediante la función Coseno o la función Kolb (Kolb, 2009).
- *Second order:* calcula la similitud entre los textos de entrada con base en los conjuntos distribuidos de las palabras que los conforman (Kolb *et al.*, 2017).

DISCO ha demostrado tener una correlación con las funciones de similitud semánticas derivadas de Wordnet (Kolb, 2008) (Kolb, 2009).

3.2.4.2 RETINA – Similarity explorer

Es una herramienta en línea que comparar el significado de dos textos cualquiera mediante la superposición de sus huellas dactilares semánticas. *Similarity explorer* es la esencia de RETINA cortical.io[4].

Cortical.io es un conjunto de tecnologías innovadoras para el procesamiento de lenguaje natural, el cual explora detalles de la teoría *Semantic Folding* (Webber, 2015), una teoría que indica una nueva perspectiva totalmente nueva para el manejo de grandes cantidades de datos textuales.

Básicamente, RETINA convierte el lenguaje a huellas dactilares semánticas, las cuales son las huellas de identidad de un solo concepto, que se caracterizan de un modo único y descriptivo de los significados asociados a ese concepto. Estas huellas dactilares semánticas son una representación numérica que captura el significado de los conceptos. Entonces, RETINA obtiene la relación semántica de dos textos mediante la superposición de las huellas dactilares semánticas que los representan.

Similarity explorer funciona realizando llamadas a la API de RETINA, de esta forma se obtienen las huellas dactilares de los textos de entrada, después, las huellas son comparadas y se devuelve un valor numérico como medida de similitud. Actualmente, el método para calcular la similitud es la función Coseno. Para la versión gratuita de la API de RETINA, la creación de las huellas se hace mediante las *"retinas"* de propósito general entrenada con Wikipedia[5] de diversos idiomas (cortical.io, 2017).

Similarity explorer es una aplicación de la API de RETINA de cortical.io de funcionamiento *online*, y para poder acceder a los servicios de Cortical.io es necesario contar con una clave de API.

3.2.4.3 SemSim

Es un método para calcular la similitud semántica entre dos textos que está basada en el algoritmo propuesto por (Lintean *et al.*, 2012), donde el valor de similitud es calculado con base en la similitud cualitativa y cuantitativa de los textos, donde la similitud cualitativa se obtiene con base en la similitud semántica de las palabras individuales de los textos calculadas

[4] http://www.cortical.io/

[5] Wikipedia.com

a partir de conjuntos de datos (corpus), mientras que la similitud cuantitativa se basa en gran medida en la frecuencia de aparición de los términos, normalizando la frecuencia de aparición local y su importancia dentro de todo el contexto de conjunto de datos (pesado IDF, *Inverse Document Frequency*).

La similitud semántica está definida por:

$$SemSim(x, y) = \frac{Sim(x \rightarrow y) + Sim(y \rightarrow x)}{2 * Max(norm_x + norm_y)}$$

Donde

$$Sim(x \rightarrow y) = \sum_{p(w_x, w_y) \in S(x \rightarrow y)} WordSim(w_x, w_y) * Max\left(idf(w_x), idf(w_y)\right)$$

$Sim(x \rightarrow y)$ Es la similitud unidireccional del texto x al texto y. Y $p(w_x, w_y)$ es un par del conjunto S de palabras parecidas, donde se emplea un umbral para definir las palabras más cercanas de x a y.

$WordSim(w_x, w_y)$ es la función de similitd empleada para encontrar las palabras más parecidas.

SemSim hace uso de conjuntos de datos para encontrar la similitud semántica entre las palabras de los textos, y está diseñado para trabajar con los corpus proporcionados por DISCO (Kolb *et al.*, 2017). Como función de similitud en $WordSim(w_x, w_y)$ hace uso de la función *Second order* indicada en DISCO (Kolb, 2008).

SemSim es un proyecto compartido que se puede encontrar en la plataforma GitHub[6].

3.2.4.4 WMD (Word Moves Distance)

Calcula la dis-similitud entre dos documentos de texto como la suma acumulada de la distancia mínima que cada palabra de un texto debe mover en el espacio vectorial a la palabra más cercana en el otro texto (Kusner *et al.*, 2015). De manera más formal, la definición de WMD es la siguiente:

$$D = \min_{T \geq 0} \sum_{i,j=1}^{n} T_{ij} \left\| x_i - x_j \right\|_2$$

Que está sujeto a:

$$\sum_{j=1}^{n} T_{ij} = d_i^a \quad , \quad \sum_{j=1}^{n} T_{ij} = d_i^a \quad \forall\, i, j$$

[6] https://github.com/mateuszzawislak/semantic-similarity

Partiendo de una matriz generada por *word2vec* $X \in \mathbb{R}^{d \times n}$ para un vocabulario de n palabras. La columna i^{th}, tal que $x_i \in \mathcal{R}^d$, representa la palabra "embebida". Adicionalmente, dados d^a y d^b que son los vectores normalizados de bolsa de palabras de dos documentos, donde d_i^a es el número de veces que la palabra i ocurre en d^a. WMD introduce una matriz de "transporte" auxiliar $T \in \mathcal{R}^{n \times n}$, tal que T_{ij} describe cuánto cuesta trasladar d_i^a a d_j^b (Kusner *et al.*, 2015) (Huang *et al.*, 2016).

Word2vec(Mikolov *et al.*, 2013) es un grupo de modelos que son usados para producir *Word embeddings* (palabras embebidas), y toman como entrada un corpus enorme de datos textuales y produce un espacio vectorial (generalmente de cientos de dimensiones), donde cada palabra única del corpus de entrada se le asigna un vector correspondiente del espacio vectorial, de tal manera que, las palabras que comparten contextos comunes en el corpus se localizan muy cerca unas de otras en el espacio.

3.3 Transformación de una función de similitud a distancia y viceversa

Ante ciertas aplicaciones es necesario usar diversas funciones de similitud y distancia, pero la integración directa de ésta no siempre es posible directamente debido a sus propiedades y características. Por ejemplo, las funciones de distancia indican que un elemento es más parecido a otro cuando el valor generado es cercano a cero; por el contrario, una función de similitud indica que dos elementos son parecidos cuando el valor es cercano a 1 (Chen et al., 2009).

Es por eso que existen formas de transformar una función de similitud a una de distancia o viceversa. Aunque, algunas funciones de similitud y distancia indican de manera precisa como transformar sus resultados, son exclusivas para esas funciones. A continuación, se muestran algunas formas de trasformar las funciones.

$$s_{jk}\ Similitud$$

$$d_{jk}\ Distancia$$

$s_{jk} = 1 - d_{jk}$ cuando d está en un rango de $[0,1]$

$d_{jk} = \sqrt{(1 - s_{jk})}$ cuando s está en el rango de $[0,1]$ (Podani, 2000)

$s_{jk} = \frac{1}{d_{jk}}$

$s_{jk} = \frac{1}{0.5 + d_{jk}}$ (Dekhtyar, 2012)

$s_{jk} = \frac{1}{1+d_{jk}}$ (Boriah *et al.*) (Kiela *et al.*, 2014)

$s_{jk} = \frac{1-d_{jk}}{1+d_{jk}}$

$s_{jk} = \frac{1-d_{jk}}{max}$ cuando d está en el rango de $[0, inf)$ (Stahl, 2002)

Como se describió anteriormente, existen diversas funciones que nos permiten calcular la similitud entre dos elementos, específicamente, dos textos. Cada una de las funciones descritas anteriormente han sido utilizadas y probadas en diversos trabajos, en los cuales han dado buenos resultados.

Una característica importante de las funciones mostradas anteriormente es que fueron definidas por humanos expertos, y aunque algunas no fueron creadas estrictamente para trabajar con documentos de textos, han sido adaptadas para tal propósito. Debido a esto, la dificultad de crear nuevas funciones de similitud está limitada por (Stahl, 2004):

- El conocimiento de un experto humano en procesamiento de leguaje natural.
- El lento proceso de la definición de éstas.
- Los cambios de requerimientos, es decir, la definición de una función de similitud textual funcionará correctamente en la tarea para la cual ha sido descrita.

Todas la funciones tienen sus fortalezas y debilidades inherentes, y en lugar de intentar crear más y más medidas para las tareas del procesamiento de lenguaje natural, una investigación prometedora es, aprovechar el potencial que tienen las medidas de similitud existentes para resolver los problemas, y tratar de combinar (de alguna forma) en un solo modelo las diversas medidas de similitud o sus características (Bär *et al.*, 2015b). Un enfoque viable es la inducción de funciones de similitud mediante el aprendizaje automático (*Machine learning*) (Maggini *et al.*, 2008).

3.4 Inducción de funciones de similitud

Debido a que la mayoría de las tareas de procesamiento de lenguaje natural hacen uso de funciones para calcular la similitud entre dos documentos de textos, como paso central de sus procesos, es necesario contar con "buena" función de similitud que nos permita mejorar los resultados en las tareas de procesamiento de lenguaje natural.

Existe una gran cantidad de funciones de similitud en la literatura, y aunque todas las funciones tienen sus fortalezas y debilidades (Bär *et al.*, 2015b) al ser aplicadas en diversas tareas, no

existe la "mejor" función de similitud que genere buenos resultados en todas las tareas de procesamiento de lenguaje natural en las que se aplique (Thada *et al.*, 2013).

Por esta razón, en la última década, el estudio del cálculo de la similitud entre dos textos ha llamado mucho la atención a los grupos de procesamiento de lenguaje natural (Stahl, 2004) (Nowak *et al.*, 2007) (Maggini et al., 2008) (Chen *et al.*, 2014) (Behzadi, 2015), debido a dos razones principales (Stahl, 2004):

1) La definición de funciones de similitud para textos depende del dominio de aplicación, por lo cual, requiere cierto conocimiento del dominio al que será aplicada.
2) Generalmente, sólo los expertos de dominio son capaces de proporcionar los conocimientos necesarios para crear nuevas funciones de similitud, donde el conocimiento adquirido se ha de formalizar mediante el uso de representaciones matemáticas complejas. Por lo tanto, la definición de medidas de similitud exacta sigue siendo un complicado y largo proceso (Stahl, 2003) (Stahl, 2004).

Una de las alternativas visualizadas por los grupos de procesamiento de lenguaje natural, no es crear directamente nuevas funciones de similitud, sino tratar de combinar las ya existentes, y de esta manera aprovechar su potencial y crear nuevos modelos de similitud. Un enfoque viable a la combinación de las medidas de similitud existentes, es a través del aprendizaje automático (Maggini *et al.*, 2008).

Como se mencionó anteriormente, el aprendizaje automático hace referencia a la detección automática de patrones significativos dentro de un conjunto de datos por medio de programas de computadora, cuyo objetivo principal es tratar de inducir de manera automática conocimiento (reglas, modelos, etc.) a partir de los datos de entrada con los que se está entrenando (Gabel, 2003). Existen varios algoritmos de aprendizaje automático, que de igual forma cuentan con sus fortalezas y debilidades, pero los más utilizados son las redes neuronales (Nettleton, 2012), las máquinas de soporte vectorial (Chang *et al.*, 2011) y los algoritmos evolutivos (Thada *et al.*, 2013).

Varios algoritmos que realizan la combinación o inducción de funciones de similitud han sido propuestos, y la mayoría de ellos están relacionados al aprendizaje semi-supervisado (Maggini *et al.*, 2008). Dentro de la literatura, al proceso de inducción y aprendizaje de funciones de similitud se le conoce como *Learn Similarity Measure* (Aprendizaje de medidas de similitud).

Learn similarity measure tiene por objetivo generar un modelo (función) de similitud a través del aprendizaje con datos de entrenamiento sujeto a las limitaciones de la información con la que se entrena, de manera que el modelo aprendido pueda reflejar las características específicas de los datos o sus relaciones (Hillel *et al.*, 2007) (Liu *et al.*, 2015a). De manera simple, el aprendizaje de medidas de similitud busca un modelo, a través del aprendizaje, que satisfaga las restricciones de similitud para un conjunto de datos (Chopra *et al.*, 2005).

Los algoritmos de *learn similarity measure* tienen dos características importantes (Stahl, 2004):

1) Deben de reducir el esfuerzo para generar las funciones de similitud.
2) El aprendizaje debe asegurar una alta calidad de la función obtenida.

Debido a que esta técnica involucra la incorporación de diversos datos (que pueden estar en distintas formas), y es necesario inferir sobre ellos. Esta técnica es conocida como un problema de optimización (Gabel, 2003) (Maggini *et al.*, 2008) (Qiong *et al.*, 2013) (Chen *et al.*, 2014) que puede resolverse a través de un proceso iterativo (Xing *et al.*, 2002) (Kumar *et al.*, 2008, Jin *et al.*, 2009) (Yin *et al.*, 2010).

Optimización es la maximización o minimización de una función objetivo sujeta a las restricciones de sus variables para darle solución a un problema (Nocedal *et al.*, 2006). Existen diferentes métodos de resolución de problemas mediante técnicas de optimización, pero aquellas técnicas basadas en algoritmos evolutivos son los más utilizados (Gestal, 2010) (Goldberg, 1989), tales como la regresión simbólica.

Existen trabajos que se han encargado de demostrar la relación que existe entre la definición y optimización de funciones de similitud y la regresión simbólica, pues como se explicó anteriormente, estas dos tareas trabajan bajo la optimización de valores que permitan la resolución a un problema en específico (Gabel, 2003) (Sette *et al.*, 2001). En esas investigaciones se trabaja sobre la definición automática de funciones de similitud mediante inducción y, como resultado, han sido beneficiadas algunas técnicas de procesamiento de lenguaje natural.

Una de las vías más prometedoras para la inducción de funciones de similitud en tareas de procesamiento de lenguaje natural, es aquella en la que, la combinación de los valores de similitud calculados previamente (a través de diversas funciones de similitud entre los documentos de texto de cierta tarea) mediante un algoritmo de regresión, como pueden ser los algoritmos genéticos y la regresión simbólica, pueda generar una combinación adecuada de los valores, y de esta forma generar un modelo basado en funciones de similitud y así mejorar los resultados en las tareas de procesamiento de lenguaje natural (Huang, 2011) (Bär *et al.*, 2012) (Carmona, 2014).

En el siguiente capítulo se dan a conocer algunos trabajos del estado del arte donde la definición u optimización de funciones de similitud es el principal objetivo.

3.5 Resumen

En este capítulo se describieron las características de las funciones de similitud y distancia, y se aclaró que, para lo relacionado a la presente investigación, se hace uso del término función de similitud para referirse a ambos terminos (similitud y distancia).

Así mismo se presentaron diversas clasificaciones de las funciones de similitud descritas en la literatura, la propuesta por (Gomaa *et al.*, 2013) quien indica tres tipos de funciones, las basadas en secuencias de cadenas textuales, aquellas basadas en corpus, y aquellas basadas en el conocimiento de redes semánticas. También se mostró la clasificación propuesta por (Yih *et al.*, 2007) quien las clasifica en *Surface matching* (basadas en el traslape de palabras), aquellas basadas en corpus, aquellas basadas en el registro de consultas generadas en grande motores de búsqueda, y aquellas basadas en la coincidencia de términos encontrados dentro los documentos devueltos por motores de búsqueda. Y la clasificación propuesta por (Bär *et al.*, 2015b): funciones basadas en el contenido del texto, basadas en la estructura del texto, y aquellas basadas en el estilo del texto.

Adicionalmente se describe una clasificación de las funciones de similitud que es útil para este trabajo, donde las funciones de similitud se clasifican de acuerdo al tipo de datos que representan a los textos dentro del modelo de espacio vectorial, por lo cual, la clasificación mostrada es: funciones cuya representación de documentos son vectores con datos booleanos, funciones para vectores con datos numéricos, funciones cuyos vectores son los términos de los textos, y funciones semánticas, donde los vectores representan al texto completo.

Debido a la amplia variedad de funciones de similitud, valor generado por cada una de ellas, puede estar en un rango diferente, es por eso que también se mostraron diversos métodos para transformar las funciones de similitud a distancia y viceversa.

Finalmente, se indicó el problema actual del uso de las funciones de similitud en las tareas de procesamiento de lenguaje natural: no existe una "mejor" función de similitud que otorgue lo mejores resultados, y la definición de nuevas funciones de similitud es un proceso lento y de alto coste, pues necesita del conocimiento de un experto en el dominio de cada tarea de procesamiento de lenguaje natural. Donde la hipótesis para dar solución a este problema, es la inducción de las funciones de similitud existentes mediante la combinación de los valores generados por ellas, haciendo uso de herramientas de análisis de grandes cantidades de datos, tal como lo es la regresión simbólica.

CAPÍTULO 4.

Estado del Arte

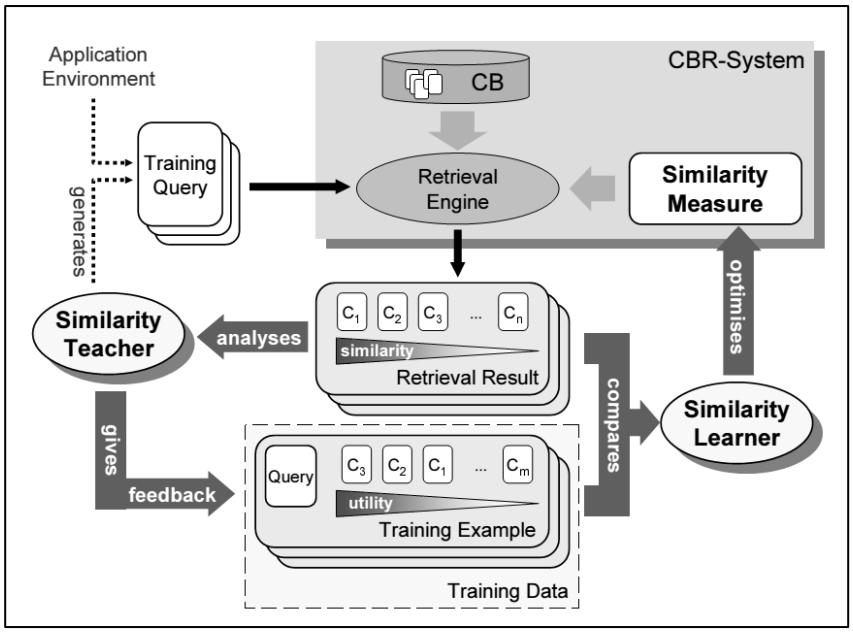
En este capítulo se presentan los trabajos relacionados al tema tratado en esta tesis.

4.1 Learning similarity seasures in case-based reasoning

El trabajo de (Stahl, 2004) presenta una aproximación para aprender medidas de similitud de alto conocimiento (*knowledge-intensive similarity measure*) para la tarea de Razonamiento Basado en Casos. Los sistemas de razonamiento basados en casos se han convertido en una herramienta popular para el desarrollo de sistemas basados en conocimiento. Estos sistemas recopilan registros de datos (llamados casos), que representan información acerca de problemas que fueron resueltos en el pasado, y la idea es la de volver a utilizar ese conocimiento en la resolución de problemas nuevos (Riesbeck *et al.*, 1989) (Stahl, 2004) (Cheng

et al., 2008). El problema radica en identificar cuándo dos problemas son considerados similares. El rendimiento de un sistema de este tipo depende a menudo de la correcta definición de una medida de similitud, pero la definición de una medida de similitud a mano puede ser muy difícil debido a diversos factores, entre ellos: la accesibilidad al conocimiento útil del dominio (por falta de entendimiento), la dificultad de transformar el conocimiento a representaciones formales o por los rápidos cambios en los requerimientos de las aplicaciones. En su trabajo, Stahl desarrolla un marco de trabajo con un enfoque metodológico para la formalización del conocimiento del dominio necesario para la definición de medidas de similitud de alto conocimiento, el cual, mediante la aplicación de estrategias de aprendizaje automático, extrae conocimiento de algún conjunto de datos de entrenamiento para después ser aplicado en entornos reales. En general, el sistema propuesto por Stahl puede ser resumido en la Figura 4.1.

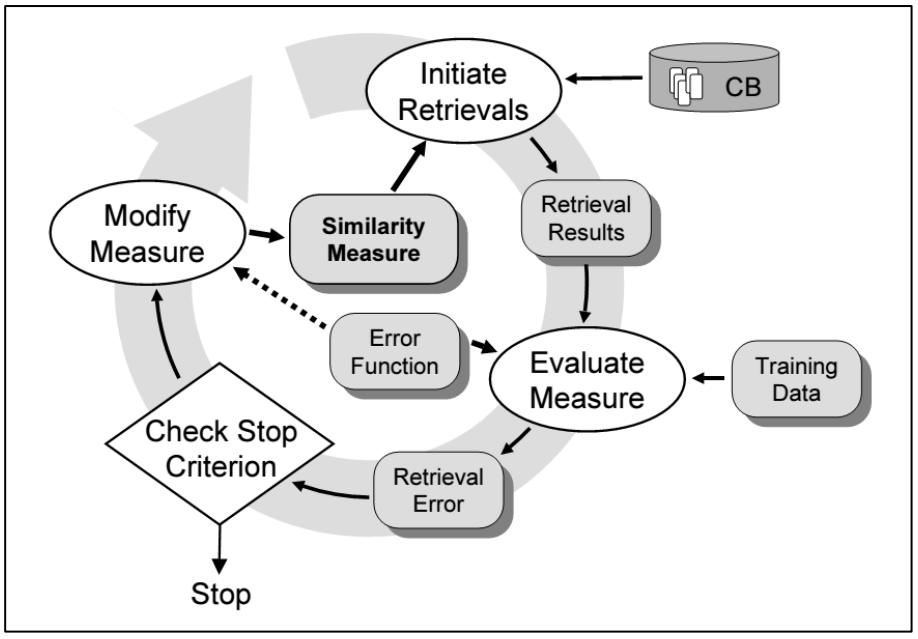
Figura 4.1 Aprendizaje de medidas de similitud para Razonamiento Basado en Casos (Stahl, 2004)



Esta arquitectura cuenta con sistema de razonamiento basado en casos el cual está compuesto por el sistema de razonamiento, una medida de similitud y un sistema de recuperación. También cuenta con el sistema de aprendizaje, el cual está compuesto por un *Similarity teacher* el cual se encarga de analizar los resultados recuperados y otorga una retroalimentación acerca de los casos útiles. El resultado del *maestro de aprendizaje* es un conjunto de datos de entrenamiento. La otra parte importante de la arquitectura es *Similarity learner*, el cual tiene como tarea modificar la medida de similitud del sistema de razonamiento

principal con el fin de incrementar la calidad de los resultados mediante una función que compara los resultados de recuperación obtenidos contra los datos de entrenamiento disponibles, donde la modificación de las medidas de similitud puede ser vista como un proceso de optimización cíclico. Este proceso de optimización se indica en la Figura 4.2. Este proceso comienza con los resultados recuperados, a los cuales se les aplica una medida de similitud. En el siguiente paso, la medida es evaluado con los datos de entrenamiento contra los resultados recuperados, de este segundo sub-proceso se obtiene un *error* de la función, el cual se utiliza en la fase que revisa la condición de parada, si el error mejora la calidad estimada del sistema, el proceso iterativo se detiene, del contrario continuo la fase de modificación de la medida. En esta última fase, la medida evaluada que no cumplió con el requerimiento de parada es modificada con la esperanza de incrementar los resultados. Después de modificar la medida de similitud, una nueva iteración en el proceso cíclico es iniciada.

Figura 4.2 Proceso de optimización (Stahl, 2004)



El proceso de optimización utilizado por Stahl está basado en los algoritmos genéticos. Su método propuesto fue evaluado con conjuntos de datos sobre Recomendación de computadoras personales, y Recomendaciones de carros usados, aunque el autor indica que su enfoque puede ser aplicado a diversos ámbitos más. Su método propuesto fue comparado frente a la técnica de *gradient descent algorithm*. Como conclusiones el autor indica que el algoritmo *gradient descent algorithm* es más rápido, debido a que explota el conocimiento

alrededor la función de error, mientras que el *algoritmo genético* resulto ser un algoritmo muy flexible en sus pruebas, además de que logro mejoras a la calidad devuelta por *gradient descent algorithm.* Finalmente indica que el algoritmo genético logro no solo aprender el peso de los atributos, también logro aprender la similitud local y general de ello.

4.2 Improving similarity measures for short segments of text

Detectar la similitud entre dos fragmentos de texto muy cortos es una de las principales tareas en problemáticas basadas en la Web, tales como la similitud entre dos consultas ingresadas en un buscador y la recomendación de productos basado en palabras clave. En el trabajo de (Yih *et al.*, 2007) se aborda el primer problema, es decir, el cálculo de similitud entre consultas y sugerencias, donde dado un segmento de texto corto q y una lista de sugerencias $\{s_1, s_2, s_3, \dots, s_n\}$ se espera obtener un rango de sugerencias con base en su similitud con q o un subconjunto de sugerencias que son similares a q. Inicialmente el autor propone una medida de similitud (*Web-relevance function*) basada en la relevancia Web el cual es una mejora al método presentado en (Sahami *et al.*, 2006) a través de un nuevo método de pesado de términos, el cual se realiza mediante el valor devuelto por la "relevancia de la palabras dentro de un documento" (Yih *et al.*, 2006). El autor indica el uso de otras medidas de similitud para tratar el problema, donde las clasifica de acuerdo con su funcionamiento.

Dentro de las medidas basadas en la coincidencia superficial están el coeficiente de Dice, Jaccard, de Traslape, de Coseno y de Coincidencia. El otro conjunto son las medidas basadas en Corpus están KL-divergence (Metzler *et al.*, 2007) y Web-based kernel (Sahami *et al.*, 2006). Debido a que las medidas entes mencionadas son "medidas estáticas" (Yih *et al.*, 2007), no razón alguna para creer que estas medidas son ideales para el problema tratado en ese trabajo, además, debido a que las medidas tienen diferente cobertura, es probable que estas cubran diferentes oraciones del problema.

Ante esa situación, (Yih *et al.*, 2007) propone el uso de una máquina de aprendizaje automático que ayude a mejorar las medidas de similitud. Hace uso de dos modelos de aprendizaje automático, en el primero trata de aprender las medidas de similitud "directamente" para cada par de datos, tal que la medida aprendida $f_m(q_i, s_i) > f_m(q_j, s_j)$ inicial. Este modelo de aprendizaje exige las etiquetas para cada par de textos, entonces, cuando se obtiene las puntuaciones específicas de otras similitudes, una función de regresión puede ser aprendida bajo este primer contexto. El segundo modelo hace uso del aprendizaje por orden de preferencia (Burges *et al.*, 2005), el cual indica que se deben de obtener un conjunto de datos bajo un ranking debido a que es más fiable tener el orden de preferencia que las etiquetas categóricas para cada caso. Los datos de prueba fueron obtenidos de manera aleatoria, donde a partir del millón de consultas más relevantes generadas en 2005 se

tomaron las 363 más realizadas. El orden de relevancia para el segundo método fue establecido en cuatro criterios "*Excellent, Good, Fair, Bad*", mientras que para el primer método se utilizó el algoritmo de regresión logística mostrado en (Goodman, 2002). Sus resultados indican que las medidas de similitud de coincidencia superficial obtienen en el mejor de los casos un 64.7% de cobertura, donde la mejor de ellas es el coeficiente de Dice. Mientras que las medidas *KL-divergence* y *Web-based kernel* obtienen un 80.5% y 82.4% respectivamente. Por otro lado, la medida propuesta por el autor obtiene un 83.3%.

De los métodos de aprendizaje utilizados, reporta un 94.4% de cobertura para ambos métodos, pero en comparación el método basado en el orden de preferencia fue mejor. El autor concluye diciendo que sus métodos mejoran los resultados mostrados en el estado del arte, además de demostrar cómo combinar diversas medidas de similitud mediante máquinas de aprendizaje automático.

4.3 Evolving kernels for SVM classification

Una de las técnicas más prometedoras para solucionar problemas de clasificación supervisada es máquinas de soporte vectorial. Estas máquinas de aprendizaje se basan en un kernel el cual se encarga de transmitir la transformación de los datos y así permitir una clasificación adecuada. El detalle de esta técnica recae precisamente en el kernel, pues la definición de este no se logra de manera trivial. La mayoría de las investigaciones se basan en un conocimiento a priori para seleccionar el kernel correcto, y después ajustar los parámetros del mismo a través de ensayo y error. A pesar de que existen métodos para su definición, cada problema requiere un kernel diferente, lo cual limita su definición a expertos del dominio específico.

En el trabajo de (Sullivan *et al.*, 2007) se hace uso de la programación genética con el fin de evolucionar el kernel y sus parámetros asociados para mejorar los resultados en la tarea de clasificación. Su método llamado *KGP*, utiliza como datos de prueba conjuntos de datos obtenidos del repositorio UCI (UCI, 2015), entre los cuales están *Ionosphere, Iris, Wine*, *Wisconsin Breast Cancer, Heart Disease* y *Vowel.* Su método funciona de la siguiente manera: empieza con la definición estándar de las funciones de kernel, (en este caso serán definidos bajo arboles sintácticos puesto que la programación genética hace uso de ellos), donde los nodos terminales son dos vectores de características x y y, y los nodos inmediatamente anteriores son las funciones de kernel básicas (polinomial, Gaussiana o Sigmoidal) cada una con sus parámetros. Los operadores del programa genético se encargan de evolucionar las funciones durante el proceso iterativo del mismo.

La evaluación de su método se realiza mediante el uso de validación cruzada. Para realizar las pruebas, el autor dividie cada conjunto de datos en dos, donde el 60% fueron datos de

entrenamiento y 40% de prueba. Su método fue comparado con otro alternativo, *SVM-Grid* encuentra los parámetros óptimos del kernel cuando se usa un kernel Gaussiano. Sus resultados indican que ambos métodos obtienen una exactitud arriba del 95%, donde KGP fue mejor en *Ionosphere*, *Iris* y *Wine*, mientras que *SVM-Grid* es superior en *Wisconsin Breast Cancer, Heart Disease* y *Vowel*.

Como conclusión el autor indica que su método KGP cual reduce la carga del conocimiento del investigador y de esta manera la dificultad de definir de funciones de kernel.

4.4 Learning similarity measures with neural networks

Una técnica que también es utilizadas en tareas de aprendizaje supervisado son las Redes Neuronales. Maggini (Maggini *et al.*, 2008) describe una red neuronal que tiene por objetivo el aprendizaje de medidas de similitud para pares de patrones enfocándose en la clasificación binaria. Su método es llamado SNN (Similarity Neural Networks), y consiste en un perceptrón multicapa de alimentación directa, el cual es entrenado para aprender una medida de similitud entre pares de patrones representados por vectores de características en el rango de números reales. Como salida, su método devuelve una función del tipo $f_{SNN}(x, y, \theta)$, la cual indica la función calculada entre los pares (x, y) y θ indica la salida dentro del intervalo $[0,1]$. Su arquitectura garantiza aprender medidas de similitud con base en sus restricciones básicas (Chen *et al.*, 2009) tales como: no negativa y simetría.

Para evaluar el trabajo de su método SNN, Maggini (Maggini *et al.*, 2008) utiliza 5 conjuntos de datos obtenidos del repositorio UCI (UCI, 2015), estos son: Balance, Boston, Ionosphere, Iris y Wine. La calidad de su método fue comparado con medidas de similitud clásicas (Euclidiana y Mahalanobis) y otras técnicas del estado del arte como la métrica de Xing (Xing *et al.*, 2002) y RCA (Hillel *et al.*, 2005), al igual que la métrica aprendida por el algoritmo MPCK-Means (Bilenko *et al.*, 2004) y la métrica aprendida por DistBoost (Hertz *et al.*, 2004). Todos los experimentos se llevaron a cabo bajo las mismas condiciones entre las diferentes técnicas, utilizando el software proporcionado por cada uno de los autores. Los resultados indican que el SNN mejora todas las demás medidas en los conjuntos de datos Balance, Ionosphere y Boston, mientras muestra resultados comparables con DistBoost en Wine e Iris. Concluye diciendo que su método SNN tiende a mejorar los resultados cuando el tamaño de datos incrementa, mientras las demás técnicas no muestran esa capacidad.

4.5 Boosting technique to genetic programming for learning to rank

Una parte muy importante dentro de la tarea de recuperación de información es determinar el orden (ranking) de acuerdo con su relevancia de los documentos recuperados dada una consulta. Ante esta situación, (Feng *et al.*, 2010) propone un método de aprendizaje el cual

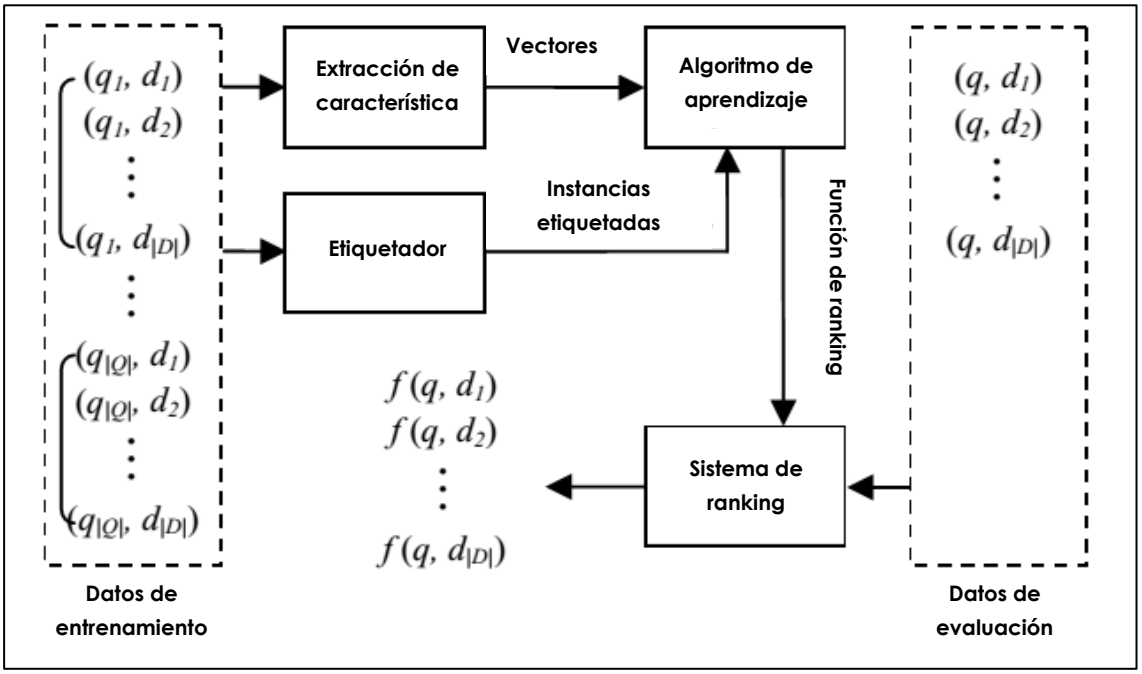
permite evolucionar funciones de ranking, las cuales permitan obtener mejores resultados en la tarea de recuperación de información. Ese desafío es nombrado como *Learning to Rank*, y su objetivo es crear un modelo de clasificación que permita ordenar los documentos según su grado de relevancia entre cada par de documentos y la consulta ingresada (generalmente por un usuario), de tal forma, que cuando se ingrese una consulta nueva, los documentos recuperados correspondiente se ordenen utilizando el modelo de ranking construido. Cabe destacar que su método está basado en dos técnicas descritas en su estado del arte, la primera es RankGP (Yeh *et al.*, 2007) y la segunda es Boosting (Xu *et al.*, 2007).

RankGP (Yeh *et al.*, 2007)está basada en programación genética, donde esta última tiene por objetivo generar nuevas funciones de ranking (mediante evolución) de tal forma que mejoren los resultados. La segunda técnica es (Xu *et al.*, 2007) la cual hace uso de la técnica de boosting (elevación) con el fin de mejorar los resultados de la optimización, donde esta última técnica se basa en la idea de crear un camino de aprendizaje mediante el entrenamiento de diferentes datos, centrándose en evolucionar funciones de ranking para las consultas más difíciles.

En la Figura 4.3, se muestra el paradigma general (Yeh *et al.*, 2007) de Learning to Rank, el cual consta de dos fases: Entrenamiento y Prueba. En la fase de entrenamiento se cuenta con un conjunto de documentos $D = \{d_1, \dots, d_{|D|}\}$ y un conjunto de consultas $Q = \{q_1, \dots, q_{|Q|}\}$, por lo tanto, el conjunto de entrenamiento consta de pares $(q_i, d_j) \in Q \times D$, donde cada par contiene un juicio de relevancia que indica la relación entre q_i y d_j, la cual es asignada por una etiqueta (un valor de relevancia booleano o un numérico). Por cada instancia (q_i, d_j) se obtiene un vector de características las cuales describen la relación entre q_i y d_j. Las entradas al algoritmo de aprendizaje son los vectores de características y las etiquetas de relevancia. La salida de este algoritmo es un modelo de ranking que recupera y ordena los documentos para cada par (q_i, d_j), de tal forma que cuando ingresen nuevas consultas pueden ser clasificadas correctamente.

Como se mencionó anteriormente, el algoritmo de aprendizaje de Feng (Feng *et al.*, 2010) está basado en programación genética y Boosting. Como conjunto de entrenamiento utiliza LETOR 3.0 (Qin *et al.*, 2010) el cual es proporcionado por Microsoft Research Asia. Este conjunto contiene los resultados obtenidos por varios algoritmos de aprendizaje los cuales son tomados como línea base. Como resultados indica que su método, llamado AdaGP-Rank, es bueno en la ordenación de documentos, devolviendo los más pertinentes en las primeras posiciones, superado a los métodos de la línea base.

Figura 4.3 Paradigma de Learning to Rank (Feng *et al.*, 2010)



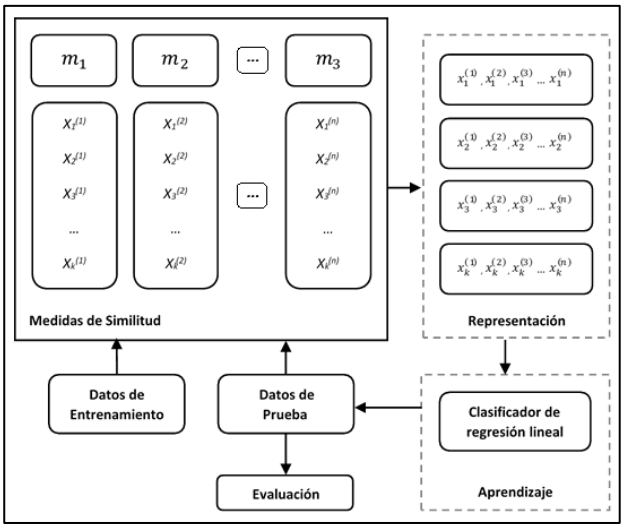
4.6 Semantic textual similarity by combining multiple similarity measures

En (Bär *et al.*, 2012) se presenta el sistema UKP, el cual es utilizado para la tarea de detección de Similitud Semántica Textual dentro de la tarea de SemEval-2012. En ese trabajo se utiliza un modelo de regresión log-lineal para combinar múltiples medidas de similitud textual de diferente complejidades, entre estas están Jaro-Winkler (Winkler, 1990), Monge and Elkan (Monge *et al.*, 1997), Levenshtein (Levenshtein, 1966), las cuales fueron aplicadas en diferentes rangos desde caracteres simples hasta n-gramas. Además, utiliza medidas basadas en similitud semántica, tales como Pairwise Word Similarity (Resnik, 1995) (Oram, 2001), Explicit Semantic Analysis (Gabrilovich *et al.*, 2007), Textual Entailment (Stern *et al.*, 2011).

Su metodología es descrita de la siguiente manera: Primero se ejecutan cada una de las medidas de similitud de manera separada al conjunto de datos de entrenamiento, a continuación, se utilizan los valores resultantes en un clasificador de aprendizaje automático, el cual combina sus valores transformados logarítmicamente. Utiliza un clasificador de regresión lineal incluido en la herramienta WEKA (Hall *et al.*, 2009) para realizar la combinación de las características (medidas). Finalmente, los resultados son evaluados contra los datos de prueba. El proceso de su método puede observarse en la Figura 4.4. Sus conjuntos de experimentación fueron MSRpar, MSRvid, y SMTeuroparl y fueron evaluados bajo la correlación

de Pearson. Los resultados de su método mejoran frente a los obtenidos por las medidas utilizadas por si solas, superando así las tres evaluaciones básicas de la tarea de Similitud Semántica Textual en SemEval-2012.

Figura 4.4 Descripción del Sistema UKP (Bär *et al.*, 2012)



4.7 Detección de similitud semántica en textos cortos

Un problema particular de la detección de similitud en textos, es cuando se trabaja con textos pequeños, esto es debido a la poca información con la que se cuenta para llevar a cabo esa tarea (Liu *et al.*, 2007). En el trabajo de (Carmona, 2014) se aborda este tema, principalmente en la detección paráfrasis en documentos cortos, donde por paráfrasis se entiende como una actividad intelectual que consiste en trasladar con palabras propias las ideas que se han expresado anteriormente, cuyo fin es el de sustituir la información recabada en un lenguaje personalizado y de mejor comprensión (Rus *et al.*, 2014).

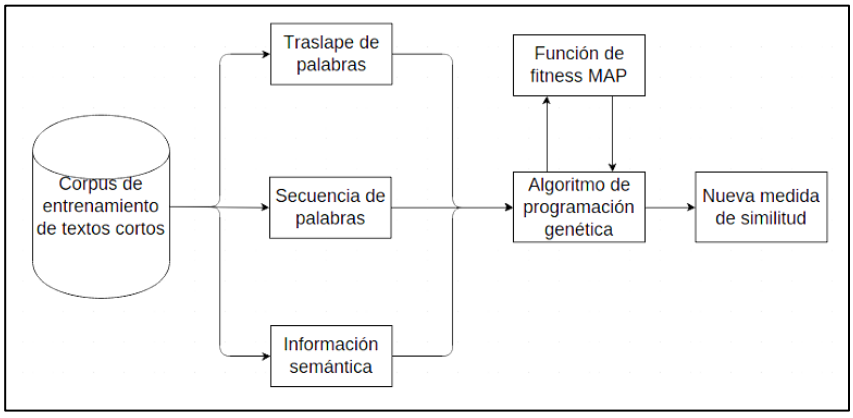
Carmona (Carmona, 2014) utiliza diversas formas para detectar la similitud entre textos cortos, aquellas basadas en el Traslape de palabras, tales como el Coeficiente de Dice, Jaccard, y Coseno. Aquellas basadas en la alineación de secuencias de palabras, como el algoritmo de la sub-secuencia común más larga o algoritmo de Levenshtein. Y aquellas basadas en información semántica, en esta se encuentra la distancia de Levenshtein con información semántica, el algoritmo de Neddleman-Wunsch con información semántica y el algoritmo de Smith-Waterman con información semántica.

El objetivo del trabajo de (Carmona, 2014) es que, a partir de los resultados de las medidas base antes mencionadas, obtener una nueva medida de similitud (mediante un algoritmo de optimización) que mejore la detección de similitud textual. El algoritmo de optimización

utilizando por el autor es la programación genética, con la cual se propuso combinar las medidas base para obtener los mejores resultados. El proceso del método propuesto por Carmona puede ser visto en la Figura 4.5. Los datos de entrenamiento utilizados fueron el corpus MSRP (Microsoft Research Paraphrase Corpus) y el corpus METER (Measuring text reuse). La tesis realizada por Carmona aborda la tarea de similitud textual desde dos enfoques. En el primero se utilizan algoritmos de alineamiento de texto con agregación de información semántica. El segundo enfoque se basa en la programación genética. En el primer enfoque se utiliza como *baseline*, los algoritmos de alineamiento de cadenas sin información semántica, y como resultados de esta primera parte, se realiza la comparación de esos algoritmos frente al algoritmo de alineamiento de texto con información semántica, dando como resultado que los algoritmos que incluyen información semántica mejoran hasta un 50% frente a los algoritmos de alineamiento de texto en ambos corpus de entrenamiento. En el segundo enfoque, se utiliza como baseline los coeficientes de Dice, Jaccard, traslape y Coseno.

Los resultados de aplicar el programa genético son superiores a los resultados del baseline en ambos conjuntos de datos. Además, los resultados no varían mucho tras la ejecución de su método, lo cual indica que es un método estable. El autor concluye diciendo que sus enfoques propuestos mantienen la competitividad en la tarea de detección de paráfrasis y detección de reúso.

Figura 4.5 Método con programación genética (Carmona, 2014)



4.8 Detección de nombres de medicamentos confusos mediante un algoritmo genético

Una aplicación reciente sobre la combinación de diversas medidas de similitud para mejorar los resultados es la mostrada en (Millán, 2016), donde se hace uso de un algoritmo genético

como optimizador de funciones para optimizar la participación de diversas medidas de similitud para la tarea de Detección de nombre de medicamentos confusos. Para esa tarea, se sabe que generalmente la confusión de nombres de medicamentos se da como consecuencia por su parecido fonético u ortográfico, por lo cual, existen medidas de similitud específicas para este campo, tales como ALINE, Bi-SIM, NED y PREFIX (Kondrak *et al.*, 2006). Estas medidas han sido aplicadas a la tarea de detección de nombres de medicamentos confusos generando buenos resultados. Además se realiza una combinación mediante el promedio aritmético de las mismas (Kondrak *et al.*, 2006). Sin embargo, en el trabajo de (Millán, 2016), se demuestra que la combinación de las medidas utilizadas para resolver esa tarea mediante el promedio aritmético de las mismas no es la mejor opción, y para demostrarlo, hace uso de un algoritmo genético en el cual asigna un valor de participación (importancia) a las medidas utilizadas, generando así un método combinatorio lineal en el cual, la participación de las medidas ya no es igual, sino que cada una tiene diferente importancia a la hora de resolver el problema. En el trabajo de Millán(Millán, 2016), además de probar que las cuatro medidas utilizadas por (Kondrak *et al.*, 2006) generan mejores resultados al optimizar su valores mediante un algoritmo genético. También se prueban con doce medidas de similitud más, demostrando que la optimización de sus valores mediante un algoritmo genético, mejora los resultados obtenidos por cada una de ellas de manera individual. De igual manera mejora, (Millán, 2016) el resultado al realizar una combinación basada en el promedio aritmético de las mismas.

Resumen

En este capítulo se han descrito los trabajos del estado del arte relacionados al tema principal tratado en esta investigación: uso de funciones de similitud en tareas de procesamiento de lenguaje natural y cómo combinar estás; o el resultado generado por estás mediante un método combinatorio o inductivo.

De manera precisa, se hace uso de Algoritmos Genéticos (Stahl, 2004) (Millán, 2016), Máquinas de Soporte Vectorial (Sullivan *et al.*, 2007), Redes Neuronales (Maggini *et al.*, 2008), programación genética (Feng *et al.*, 2010) (Carmona, 2014) como método combinatorios o inductivos de las funciones de similitud utilizadas en cada uno de los trabajos del estado del arte. Cada uno de los método utilizados y reportados en el estado del arte reporta mejoras frente a otros trabajos.

Algunos de los métodos combinatorios o inductivos utilizados en los trabajos del estado del arte tienen sus desventajas: los algoritmos genéticos tienen su principal desventaja en la linealidad de espacio de búsqueda, es decir, sus resultados generados están basados en combinaciones lineales de las funciones de similitud. Las redes neuronales, a pesar de que exploran soluciones lineales y no lineales, su principal desventaja radica en modelo final obtenido, el cual, debido

a la forma de procesamiento de las redes neuronales, los modelos obtenidos por ellas no son "entendibles" por usuarios finales (debido a que los modelos contienen demasiada información). Lo mismo ocurre con las máquinas de soporte vectorial, el modelo generado solo es entendible por la misma máquina. La programación genética, tiene como principal desventaja, el tiempo que tarda en generar resultados, debido a la exploración que realiza, su procesamiento es lento, y al hacer uso de grandes cantidades de datos, es inverosímil su uso.

A pesar de su desventaja principal, la programación genética, cuenta con una ventaja que la hace sobresalir sobre los otros métodos: su exploración no lineal y el entendimiento de los resultados generados. La programación genética permite una exploración no restringida, y sus resultados generados pueden ser delimitados de acuerdo con la complejidad del modelo que se desea obtener.

Los trabajos en el estado del arte mostrados en esta sección permitieron conocer los métodos combinatorios e inductivos que se han utilizado en las tareas de procesamiento de lenguaje natural, así como sus ventajas y desventajas. Por tal situación, el método de inducción seleccionado para llevar a cabo la presente investigación está basado en programación genética.

CAPÍTULO 5.

Método Propuesto

Recordando el problema:

¿Cómo mejorar los resultados de las tareas de procesamiento de lenguaje natural, específicamente para desambiguación del sentido de las palabras, detección de nombres de medicamentos confusos, mediante la inducción de los resultados generados por diversas medidas de similitud existentes, mediante la aplicación de regresión simbólica?

En el presente capítulo se describe el método propuesto para dar solución al problema planteado en este trabajo. El método se describe paso a paso. Así mismo, se muestra un diagrama general en el cual son representadas las etapas del mismo.

5.1 Descripción del método propuesto

Como se describió anteriormente, la mayoría de las tareas de procesamiento de lenguaje natural tienen tres elementos en común: 1) Necesitan una forma de representación de la información, 2) Uso de una función de similitud, y 3) Paradigma de evaluación para medir la calidad de los resultados. La forma de representación más utilizadas en el modelo de espacio vectorial (Salton *et al.*, 1975) (Yih *et al.*, 2010). La función de similitud es una propiedad que mide la semejanza entre dos elementos (en este caso, textos) y es expresada mediante un valor numérico. El paradigma de evaluación permite la comparación (bajo métodos estandarizados) de los resultados obtenidos por un sistema de procesamiento de lenguaje natural frente a otro.

En los últimos años, el estudio de la importancia que tienen las funciones de similitud ha aumentado (Stahl, 2004) (Nowak *et al.*, 2007) (Maggini et al., 2008) (Chen *et al.*, 2014) (Behzadi, 2015) generando nuevas investigaciones al respecto. Debido a que la definición manual de nuevas funciones de similitud para las tareas de procesamiento de lenguaje natural, y a que este proceso necesita de un experto en el área de estudio (además de gastar demasiados recursos) (Stahl, 2003) (Stahl, 2004), una nueva vertiente ha ido en aumento.

El aprendizaje de medidas de similitud (*Learn similarity measure*) es uno de los caminos más prometedores para la definición de funciones de similitud en el procesamiento de lenguaje natural (Stahl, 2004) (Hillel *et al.*, 2007). *Learn similarity measure* es una técnica que tiene por objetivo generar un modelo (función) de similitud a través del aprendizaje con datos de entrenamiento sujeto a las limitaciones de la información con la que se entrena (Hillel *et al.*, 2007) (Liu *et al.*, 2015a). Debido a que esta técnica incorpora la integración de diversos datos, y debe inducir sobre ello, es tratada como una técnica de optimización.

En los trabajos del estado del arte mostrados en el capítulo anterior, se mostraron algunos trabajos cuyo objetivo es encontrar modelos de similitud a partir de análisis de datos, donde en la mayoría de ellos, primero se calcula el valor de diversas medidas de similitud sobre los datos con los que trabaja, para después, aplicar un proceso de aprendizaje, basado en optimización, para definir un modelo (o función) de similitud que mejore los resultados de la tarea en la que se está aplicando.

El método por utilizar para resolver el problema descrito anteriormente está basado en los trabajos descritos en el estado del arte, y debido a que en los trabajos antes mencionados hacen uso de una metodología similar, se puede afirmar que ésta es una metodología robusta. Por esta razón, la metodología propuesta para la resolución del problema planteado es una generalización de los métodos descritos anteriormente.

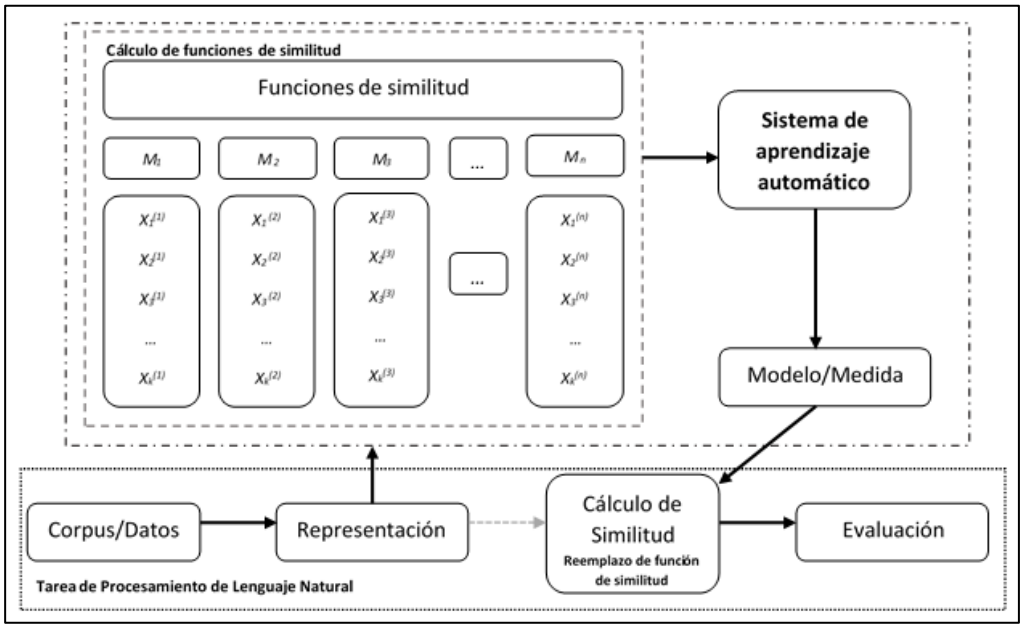
5.2 Etapas del método propuesto

La metodología propuesta se puede observar en la Figura 5.1, la cual consta de tres fases importantes: la primera es como tal el sistema de procesamiento de lenguaje natural sobre el cual se puede aplicar la metodología. La segunda, es la parte donde se calculan las medidas de similitud a partir de los datos de entrada. La tercera, es el sistema de aprendizaje el cual permitirá inducir las medidas de similitud para encontrar la mejor relación entre ellas.

Adicionalmente, y para mostrar la mejora producida por la aplicación del método propuesto a la tarea de procesamiento de lenguaje a la que se está aplicando respecto a otros métodos reportados en la literatura, se hace uso de la ecuación mostrada en (Mendoza *et al.*, 2014) para calcular la mejora de rendimientos entre los resultados de dos métodos.

A continuación, se describen cada una de las etapas que conforman al método propuesto.

Figura 5.1 Método propuesto



5.2.1 Tarea de procesamiento de lenguaje natural

Como se mencionó anteriormente, existen diversas tareas dentro del procesamiento de lenguaje natural como lo son el Manejo de conocimiento, Traducción automática, Minería de Texto, Identificación de autoría, Identificación de plagio (Gelbukh, 2010), Recuperación de información, Extracción de información, Generación de resúmenes (Vásquez *et al.*, 2009), Reconocimiento óptico de caracteres, clasificación de documentos, entre otros. Pero la mayoría de ellos comparten tres características: 1) Necesitan una forma de representación de

la información, 2) Uso de una función de similitud, y 3) Paradigma de evaluación para medir la calidad de los resultados, lo cual permite que la integración de estos sistemas se pueda generalizar.

El método propuesto en este trabajo está diseñado para poder ser aplicado a cualquier tarea de procesamiento de lenguaje natural que contenga las tres características antes mencionadas y corpus de datos.

5.2.1.1 Corpus de datos

Toda tarea de procesamiento de lenguaje natural hace uso de un corpus de datos, ya sea un corpus estándar para ese sistema o uno creado para el mismo, por lo cual, el sistema de procesamiento de lenguaje natural al que será aplicado debe contar con un corpus específico para la tarea que se pretende resolver.

5.2.1.2 Representación de la información

El modelo de representación de la información textual almacenada en el documento del corpus con el que trabaja el método propuesto, es el modelo espacio-vectorial (Salton *et al.*, 1975), en el cual, las características de los textos son almacenadas en vectores, donde cada elemento del vector representa el valor que le es otorgado a la característica que aparece en el texto.

Los valores almacenados en los vectores pueden ser booleanos $[1,0]$, númericos, o cadenas de caracteres. La elección del valor para indicar cada elemento del vector depende de las características que se desean extraer del texto, así como de la función de similitud a emplear.

5.2.1.3 Función de similitud

Debido a que la aplicación de las medidas de similitud depende directamente de la representación de la información de los textos, las funciones de similitud utilizadas en este trabajo son aquellas que trabajan con el modelo de espacio vectorial.

En general, se cuenta con las siguientes funciones:

1) Booleanas: Jaccard, Hamming, Russell-Rao, Sokal-Sneath, Dice, Kulczynski, Roger-Tanimoto, Simple Matching.
2) Numéricas: Pearson, Coseno, Euclidiana, Minkowski, Chebychev, Canberra, Spearman Footrule, Spearman Rank, Dynamic Time Warping, Índice de consistencia.
3) Cadenas o secuencias de caracteres: Longest common subsequence, Levenshtein, Damerau-Levenshtein, Needleman-Wunch.

4) Basadas en información semántica: DISCO 1, RETINA – Similarity explores, SemSim, Word moves distance.

5.2.1.4 Paradigma de evaluación

Como se mencionó anteriormente, es necesario contar con un paradigma de evaluación que permita saber qu tan bueno es un sistema de procesamiento de lenguaje natural al ser comparado con otros sistemas afines, y la mejor forma de saber cuál obtiene mejores resultados, es necesario que esta evaluación se realice mediante métodos estandarizados. Por lo cual, el mejor camino, la mejor forma de saber qué tan "bueno" es un sistema de procesamiento de lenguaje natural frente a otros es mediante la aplicación de evaluaciones intrínsecas, las cuales se realizan mediante un *gold estándar* (archivo de respuestas definido por expertos).

Por esta razón, y para tener una comparativa de los resultados generados por el método propuesto contra los resultados mostrados por otros trabajos del estado del arte, se hace uso de archivos *gold standard* para evaluar los resultados de método propuesto en las tareas del procesamiento del lenguaje natural donde es aplicado.

Cabe destacar, que el archivo *gold standar* es único para cada tarea y dominio de aplicación del procesamiento del lenguaje natural, por ejemplo, un archivo *gold standard* útil en la clasificación de documentos, no es aplicable para otras aplicaciones, es únicamente para la clasificación de documentos y más aún si pertenece a un dominio en particular.

Cada tarea donde es aplicado el método propuesto cuenta con un archivo *gold standard* específico, y cada archivo *gold standard* es comparado con los resultados obtenidos. Esta comparación es una evaluación especifica en cada aplicación de procesamiento de lenguaje natural. De manera general, un paradigma de evaluación que utiliza un archivo *gold standard* está basado en la *F-measure*.

La medida F (*F-measure*) está definida como la media harmónica del recuerdo y la precisión. La precisión se calcula dividiendo la cantidad de elementos correctos obtenidos de todos los elementos extraídos, mientras que el recuerdo calcula cuantos elementos correctos no son extraídos. La *F-measure* tiene la ventaja de resumir la eficacia de una aplicación a un solo número. La fórmula de la *F-measure* es:

$$F = \frac{1}{\frac{1}{2}\left(\frac{1}{R} + \frac{1}{P}\right)} = \frac{(2RP)}{(R+P)}$$

donde P es la precisión y R es el recuerdo.

5.2.2 Cálculo de funciones de similitud

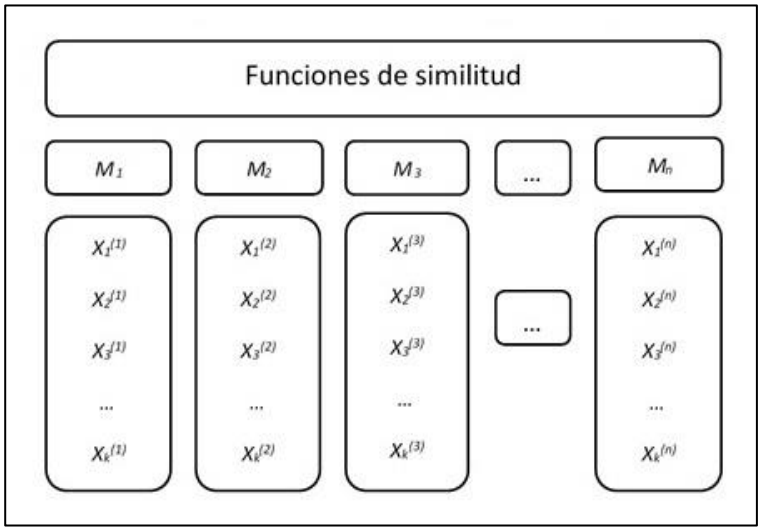
En esta etapa, se realiza el cálculo de los valores de similitud correspondientes a los pares de texto que son utilizados en una tarea de procesamiento de lenguaje natural especifica. Así, por ejemplo, si la tarea a la que es aplicada el método propuesto indica que el $textoA$ debe ser comparado con el $textoB$ mediante una función de similitud entonces, el $textoA$ y el $textoB$ deben ser sometidos al cálculo de similitud mediante la función M de similitud, por lo cual, es necesario obtener el valor x de similitud de todos los pares k de texto que existen en el corpus C de la tarea de procesamiento de lenguaje natural.

$$x_k = simM(t_a, t_b) \;\; \forall \;\; (t_a, t_b)_k \in C$$

Donde $simM$ puede ser una cualquier función de similitud aplicable a la tarea de procesamiento de lenguaje en cuestión. Así, al final se tendrá una lista de k valores que representan los valores de similitud calculados para todos los pares de textos comparables dentro del corpus utilizado.

Como parte del método propuesto, en esta fase, el cálculo de similitud entre los pares de texto es realizado por diversas medidas de similitud. Es decir, a partir de un conjunto n de funciones de similitud seleccionadas se realiza el cálculo de similitud entre los pares de texto a través de estas n funciones de similitud. De esta manera, al final se obtendrá un conjunto $k \times n$ de datos, donde cada elemento de este conjunto representa el valor de similitud x del par de texto k calculado por la función de similitud n. La Figura 5.2 muestra gráficamente el resultado esperado del cálculo de n medidas de similitud para k pares de datos.

Figura 5.2 Cálculo de n medidas de similitud



5.2.3 Sistema de aprendizaje automático

Como se describió anteriormente, el aprendizaje automático hace referencia a la detección automática de patrones significativos, y su objetivo principal inducir de manera automática reglas o modelos a partir de un conjunto de datos de entrada con los que se está entrenando (Gabel, 2003).

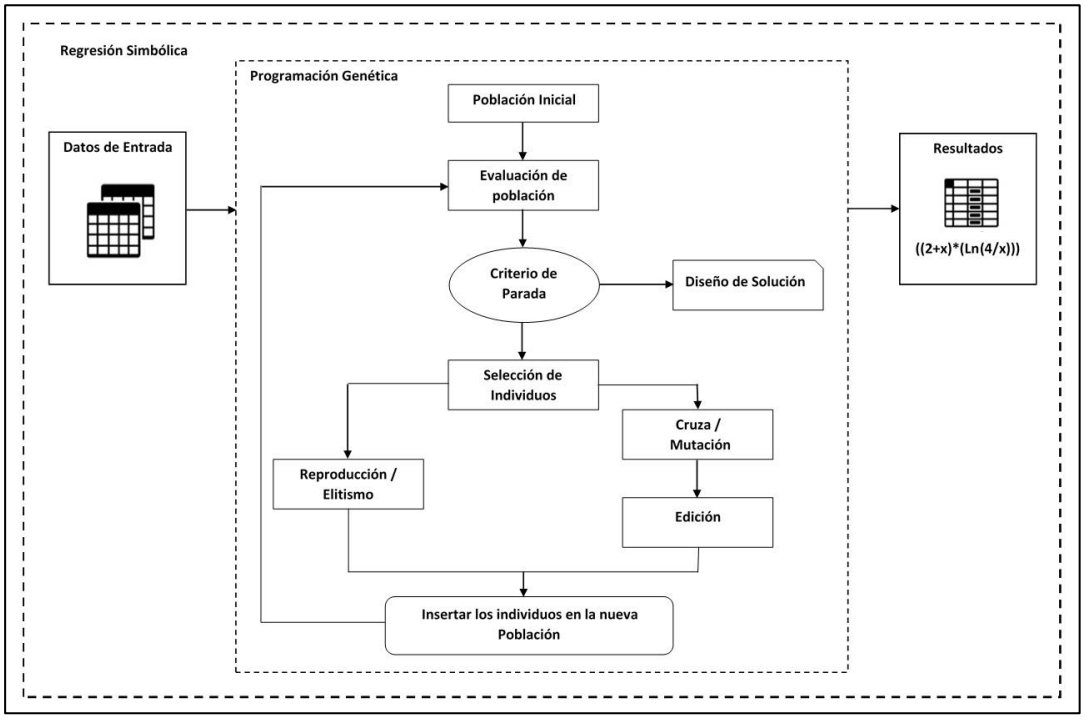
Dado el conjunto $k \times n$ de datos generado por la aplicación de n funciones de similitud a k pares de datos, y dado que el propósito de *learn similarity measure* es la creación de funciones o modelos de similitud a través del aprendizaje automático (más precisamente, inducción), y basándonos en los trabajos del estado del arte (Stahl, 2004) (Bär *et al.*, 2012) (Carmona, 2014), el modelo de aprendizaje está basado la programación genética, que una de las principales ramas de los algoritmo evolutivos.

La programación genética pretende resolver problemas mediante la inducción de algoritmos y programas que puedan resolverlos (Poli *et al.*, 2008). Estos algoritmos están diseñados para encontrar de forma automática modelos o funciones matemáticas a partir de un conjunto de valores de entrada. Una de las principales aplicaciones de la programación genética es la regresión simbólica, y ésta es conocida como la técnica de *identificación de la función*, ya que consiste en encontrar una expresión matemática, en forma simbólica, que describa la relación entre una variable dependiente y una o más variables independientes (Koza, 1992b) (Murari *et al.*, 2015).

Debido el enfoque de *learn similarity measure*, el cual pretende descubrir nuevas funciones o modelos de similitud mediante el aprendizaje automático, y dado que la regresión simbólica permite la creación de funciones matemáticas mediante la inducción de grandes cantidades de datos, tales como el conjunto $k \times n$ de datos generados por la aplicación de n funciones de similitud a k pares de datos, es viable la aplicación de regresión simbolica para dar solución al descubrimiento de funciones o modelos de similitud que prevé *learn similarity measure*.

El sistema de aprendizaje está basado entonces en la regresión simbólica, que es una aplicación de la programación genética, el diagrama de sistema se puede ver en la Figura 5.3.

Figura 5.3 Sistema de aprendizaje automático basado en regresión simbólica



Como salida del sistema de aprendizaje, se obtiene una función o modelo matemático, el cual está conformado por la inducción de los datos de entrada y, debido a que los datos en entrada, en este caso, son el conjunto $k \times n$ de datos generados por la aplicación de n funciones de similitud a k pares de datos, entonces, el modelo matemático generado por la regresión simbólica es un modelo de similitud, el cual está conformado por los valores de entrada.

5.2.4 Reemplazo de función de similitud

Una vez que se ha obtenido el conjunto $k \times n$ de datos generados por la aplicación de n funciones de similitud a k pares de datos, y que este conjunto de datos ha sido ingresado al sistema de aprendizaje, el sistema de aprendizaje, basado en regresión simbólica, devuelve un modelo matemático inducido a partir de los datos de entrada. Este modelo matemático es la función de similitud aprendida a partir de los valores individuales de las múltiples medidas de similitud calculadas.

La función de similitud aprendida es aplicada (en la fase de cálculo de similitud) en el sistema de procesamiento de lenguaje natural que se está probando. Debido a que la función aprendida está basada en los valores de otras funciones de similitud, esta puede contar con varios valores de las funciones de similitud calculadas anteriormente.

Una vez aplicada la función de similitud (en la fase de cálculo de similitud), se procede a evaluar los resultados generados por esta mediante el paradigma de evaluación adecuado al sistema de procesamiento de lenguaje natural al que es aplicado el método propuesto.

5.2.5 Comparación de rendimiento entre dos métodos

Como se mencionó anteriormente, los sistemas de procesamiento de lenguaje natural, al igual que otros sistemas informáticos, necesitan ser evaluados bajo métodos estándar que permitan comparar el funcionamiento de dos o más sistemas. En el procesamiento de lenguaje natural, cada sistema es evaluado bajo métodos específicos acorde a cada tarea, tal como se indicó en la sección 5.2.1.4 Paradigma de evaluación. A pesar de que los resultados generados por el paradigma de evaluación adecuado para cada tarea de procesamiento de lenguaje natural permiten una comparación directa entre dos sistemas, también es necesario saber cuál es la mejora porcentual de un sistema respecto a otro.

En el trabajo de (Mendoza *et al.*, 2014) se hace uso de una ecuación que permite calcular el porcentaje de mejora de los resultados obtenidos por un sistema respecto a otro. La ecuación que permite obtener este valor es la siguiente:

$$\frac{Método_{Propuesto} - Método_{referencia}}{Método_{referencia}} \times 100$$

Donde $Método_{Propuesto}$ es el resultado obtenido por el paradigma de evaluación del método propuesto, y $Método_{Referencia}$ es el resultado obtenido por el paradigma de evaluación del método contra el cual se compara. Esta ecuación será la utilizada para comparar la mejora de los resultados del método propuesto frente a otros resultados de trabajos indicados en el estado del arte.

5.3 Resumen

En este capítulo se describió el método propuesto que pretende mejorar los resultados mostrados en diversas tareas del procesamiento del lenguaje natural, esto mediante la inducción de los valores generados por diversas medidas de similitud, los cuales son obtenidos en la fase de cálculo de similitud entre pares de textos (fase importante) dentro de los sistemas de procesamiento de lenguaje natural.

Se describieron cada una de las fases que componen al método propuesto, desde la definición del sistema de procesamiento de lenguaje natural (con las fases y elementos que los componen), hasta el sistema de aprendizaje, el cual nos permite inducir funciones de similitud (a partir de datos) que mejoren los resultados en el sistema de procesamiento de lenguaje natural donde se aplica.

Así, finalmente se indica en que fases dentro del sistema de procesamiento de lenguaje natural es colocada la función de similitud inducida y obtenido por el método propuesto.

CAPÍTULO 6.

Experimentación y Resultados

En este capítulo se describe la experimentación realizada bajo el método propuesto en el capítulo anterior, así como los resultados generados por los mismos.

Como comprobación de la hipótesis propuesta se presenta la aplicación del método en dos tareas del procesamiento del lenguaje natural. Para cada una, primero se describe brevemente en que consiste cada tarea, y su importancia de estudio. Después se describen algunos trabajos que han intentado dar solución al problema de cada tarea, así como los resultados obtenidos por cada uno. Se indica las funciones de similitud que fueron utilizadas en cada caso y los resultados obtenidos al ser aplicadas. Posteriormente, se indica como el método propuesto puede dar solución a esa tarea. Para finalmente, mostrar los resultados obtenidos con la aplicación del método propuesto y el modelo obtenido por el mismo.

También se da a conocer el valor de la mejorar significativa obtenida por la aplicación del método propuesto frente a otros trabajos del estado del arte para cada tarea.

6.1 Detección de nombres de medicamentos confusos

La confusión de nombres de medicamentos se da como consecuencia por su parecido ortográfico o fonético, provocando que profesionales de la salud (médicos, enfermaros y farmacéuticos), o incluso los mismos pacientes cometan errores de medicación, provocando posibles daños a la salud, incluso, la muerte (FDA, 2012) (OMS, 2007) (Medicine, 2007) (ASHP, 1993). De acuerdo con el Programa de Reportes de Errores de Medicación (MERP) del Instituto de Prácticas Seguras de Medicación (ISMP) de Estados Unidos, el 25% de los errores de medicación, son causados por la confusión de sus nombres. Cuando dos nombres de medicamentos son confundibles con base en su parecido fonético u ortográfico, se dice que son un par de nombres *LASA* (Look-Alike & Sound-Alike).

Un ejemplo de par *LASA* viene dado por el par de nombres de medicamentos (AMARYL® - REMINYL®), donde AMARYL® es un medicamento para el tratamiento de diabetes mellitus tipo 2, y REMINYL® es un inhibidor de la acetilcolinesterasa, utilizado para el tratamiento de la demencia de Alzheimer. La aplicación de REMINYL® a personas con diabetes puede provocar hipoglucemia, náuseas, vomito, y diarrea. Mientras que la aplicación de AMARYL® a personas que sufren de Alzheimer puede provocar confusión, pérdida de conciencia, como he incluso la muerte (FDA, 2014).

De acuerdo con la *FDA* (*Food and Drug Administration*), en Estado Unidos se produce una muerte al día, y se daña alrededor de 1.3 millones de personas al año debido a los errores de confusión de nombres de medicamentos, y se genera alrededor de $3.5 millones de dólares en gastos a consecuencias de estos errores de medicación (FDA, 2014).

Con base en los datos anteriormente mencionados, la cantidad de muertes y pacientes afectados, así como el gasto económico generado por los errores de confusión de nombres de medicamentos, es necesario contar con herramientas que permitan prevenir la creación de nombres de medicamentos que sean susceptibles a confundirse con otros medicamentos, o en su defecto, herramientas que permitan la identificación de pares de nombres de medicamentos confusos.

Para abarcar el problema de la definición de un nuevo medicamento, la *FDA* pone a disposición una herramienta de software llamada *POCA* (Phonetic and Orthographic Computers Analysis), la cual utiliza un conjunto de algoritmo para determinar la similitud que existe entre el nombre propuesto y los nombres de medicamentos existentes, devolviendo una lista con los nombres de medicamentos que son más similares al nombre de medicamento propuesto (FDA, 2017).

Mientras que para el segundo problema, identificar los pares de nombres de medicamentos confusos, se han llevado a cabo investigaciones que, mediante el uso de funciones de similitud tratan de identificar el parecido ortográfico y fonético (pares LASA) entre nombres de medicamentos ya existentes (Kondrak *et al.*, 2006) (Lambert *et al.*, 1999) (Lambert *et al.*, 2004) (Nagata *et al.*, 2014) (Lebanova *et al.*, 2012) (Millán, 2016). En esos trabajos, a partir de una lista de casos de pares de confusión tomadas de reportes de errores de medicación y de una lista de pares de control, se calcula la similitud en ambas listas para así determinar el grado de probabilidad que tiene un par de nombres de medicamentos a ser confusos (Millán, 2016). La similitud calculada para cada par de nombres de medicamentos es calculada mediante funciones de similitud, las cuales permiten determinar el grado de similitud fonética y ortográfica entre dos nombres de medicamentos.

Existen funciones de similitud que permiten determinar el grado de similitud entre pares de nombres de medicamentos basándose en su similitud fonética y ortográfica. Las funciones de similitud más resaltantes son:

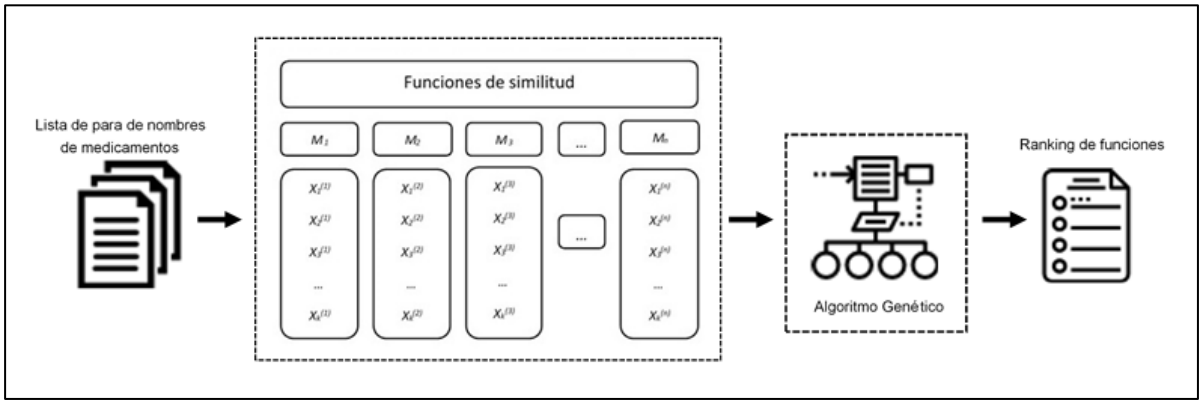
- BI-SIM, TRI-SIM, ALINE (Kondrak *et al.*, 2006)
- Bi-Gram, Tri-Gram, LCS, ED, NED, EDITEX, EDIT-SOUNDEX, EDIT-PHONIX, EDIT-OMISSION, EDIT-SKELETON (Lambert *et al.*, 2004)
- Tri-Gram, ED (Lebanova *et al.*, 2012)

En el trabajo de (Kondrak *et al.*, 2006) se propone un método combinatorio en el cual cuatro funciones de similitud son utilizadas para encontrar pares de medicamentos confusos, donde el método combinatorio está basado en el promedio aritmético de los valores obtenidos por las funciones de similitud de manera individual. Mediante el promedio aritmético de las funciones de similitud, el valor de participación (o relevancia) por el cual logra sus resultados es el mismo, es decir, debido a que utiliza cuatro medidas, el valor de participación de cada medida es el mismo (25%), y si más funciones son agregadas, el valor de participación de cada una de estas será $1/n$, donde n es el número funciones de similitud utilizadas.

Por otro lado, en el trabajo de (Millán, 2016) se propone un método basado en algoritmos genéticos para encontrar la combinación optimizada de un conjunto de funciones de similitud para mejorar la detección del parecido fonético y ortográfico entre un par de nombres de medicamentos. El método de (Millán, 2016) optimiza la participación de las funciones de similitud utilizadas, de tal forma que, los valores de participación de estas por mejorar los resultados puedes ser diferentes, es decir, algunas funciones pueden obtener una participación mayor que otras, otorgándole mayor participación a aquellas funciones que ayuden mejorar los resultados. Los resultados reportados por (Millán, 2016) superan los resultados generados por el método combinatorio de (Kondrak *et al.*, 2006), además que

indica un ranking de importancia de las "mejores" funciones de similitud para detectar pares de nombres de medicamentos confusos.

Figura 6.1 Método de optimización de funciones utilizado en (Millán, 2016)



En la Figura 6.1 Método de optimización de funciones utilizado en (Millán, 2016) se muestra el diagrama del método utilizado por (Millán, 2016), el cual, a partir de una lista de pares de nombres de medicamentos confusos, se calcula para cada par de esta lista la semejanza entre pares mediante funciones de similitud, con esto se obtiene un conjunto $k \times n$ de datos, donde k es el número de pares de nombres de medicamentos con n funciones de similitud. Este conjunto $k \times n$ de datos es la entrada al algoritmo genético, el cual, optimiza los valores de similitud. El valor de similitud es definido por el conjunto valores $X = \{x_1, x_2, x_3, \dots, x_n\}$ donde cada x_i representa el valor de las funciones de similitud utilizadas para el par de nombres de medicamentos $(a, b)_i$, y el conjunto $A = \{\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n\}$, donde cada α_i es el valor de la constante que representa la participación de la función de similitud x_i en la tarea de identificación de pares de nombres de medicamentos confusos (Millán, 2016). Por lo tanto, el cálculo de similitud para cada par de medicamentos esta dado por:

$$similitud(a, b)_i = \sum_{i=1}^{n} \alpha_i x_i (a, b)_i$$

De tal forma que se cumpla $\sum_{i=1}^{n} \alpha_i = 1$. Por lo cual, el algoritmo genético trata de encontrar la mejor combinación de valores de α_i que obtengan mejores resultados al evaluar el método. El método de evaluación utilizado para guiar la evaluación del algoritmo genético está basado en la teoría de evaluación de los sistemas de recuperación de información, utilizando como métrica principal de ponderación la medida F (*F-measure*). La evaluación se realiza, ordenando los nombres de medicamentos de acuerdo con el valor obtenido por las funciones de similitud optimizadas por el algoritmo genético, donde todos aquellos que obtuvieron la misma puntuación les corresponden la misma posición. La función de evalúa la adaptación

del individuo de acuerdo con la sumatoria de la medida F obtenida en las primeras n posiciones de la evaluación, es decir $\sum_{i=1}^{n} F\_measure_i$, donde $n = 4$.

El conjunto de datos que contiene la lista de pares de nombres de medicamentos está basado en la lista publicada por la USP Quality Review (ISMP, 2015), de la cual se seleccionaron 858 pares de nombres confusos, los cuales se consideraron como corpus USP-858, del cual se identificaron 630 nombres únicos de medicamentos, Para forma la lista de pares de nombre de medicamentos, se formaron pares a partir de los nombres únicos del corpus USP-858, por lo que se generaron 396,270 pares de nombres, obtenidos a partir del producto cartesiano de los 630 nombres de medicamentos únicos.

El conjunto de funciones de similitud utilizadas en el trabajo de (Millán, 2016) son ALINE, COSENO, EDITEXT, NED, LCSR, BI_SIM, UNIGRAM, BIGRAM, TRIGRAM, PREFIX, SOUNDEX, SUFIX. Como resultado del método de combinación optimizando la participación de las funciones de similitud, obtuvo los siguientes valores de participación para cada medida:

ALINE=0.135, COSENO=0.123, EDITEXT=0.008, NED=0.240, LCSR=0.063, BI_SIM=0.008, UNIGRAM=0.043, BIGRAM=0.013, TRIGRAM=0.046, PREFIX=0.254, SOUNDEX=0.023, SUFIX=0.044.

La optimización obtenida por la función de aptitud obtenida con los valores participación antes mostrados, obtiene un valor de 1.76804, el cual es el mejor valor de aptitud reportado. Estos resultados indican que las mejores funciones de similitud son las funciones NED y PREFIX, puesto que tienen un mayor grado de participación para mejorar los resultados de identificación de pares de nombres de medicamentos confusos.

El método propuesto en (Millán, 2016), está compuesto de manera implícita por los tres elementos de un sistema de procesamiento de lenguaje natural: representación de la información, uso de función de similitud, y paradigma de evaluación (indicados en la sección 2.5 Componentes de las tareas de procesamiento de lenguaje natural), por lo cual, el método propuesto en esta tesis (sección 5.1 Descripción del método propuesto) es aplicable al método de (Millán, 2016), además, la estructura del método de (Millán, 2016) permite fácilmente reemplazar el método de optimización basado en algoritmos genéticos, por el sistema de aprendizaje (basado en regresión simbólica) propuesto en esta tesis (sección 5.2.3 Sistema de aprendizaje).

Por lo tanto, el método propuesto en esta tesis es aplicable a la tarea de detección de nombres de medicamentos confusos, particularmente al método mostrado en (Millán, 2016).

6.1.1 Método propuesto a detección de nombres de medicamentos confusos

La aplicación del método propuesto en esta tesis, y para tener resultados comparativos precisos con el método de (Millán, 2016), fue necesario utilizar el mismo paradigma de evaluación y las mismas funciones de similitud, por lo cual, las funciones de similitud aplicadas

al método propuesto son ALINE, COSENO, EDITEXT, NED, LCSR, BI_SIM, UNIGRAM, BIGRAM, TRIGRAM, PREFIX, SOUNDEX, SUFIX. Mientras que el paradigma de evaluación es el *F-measure* acumulado a $n = 4$, tal como lo indica (Millán, 2016). A continuación, se muestran los resultados generados por el método propuesto a la detección de nombres de medicamentos confusos.

6.1.2 Experimentos y resultados

La principal desventaja del método combinatorio basado en algoritmos genéticos propuesto en (Millán, 2016), es la linealidad del mismo. La combinación y optimización de las funciones de similitud utilizadas está basada en combinación lineal de resultados. Mientras que el método propuesto en este trabajo puede ser ajustado a ser un método lineal o no lineal.

Experimento 1

En el primer experimento, se propuso la aplicación del método propuesto bajo una combinación lineal de la funciones de similitud utilizadas, con el fin de obtener una combinación de funciones similar a la mostrada en (Millán, 2016), por lo cual, el sistema de aprendizaje (basado en regresión simbólica) del método propuesto solo utilizó las funciones matemáticas de suma y resta. Como resultado de obtuvo el siguiente modelo:

$$8A + 4B + 5C + 2D + 8F + G + 3I + 12J + 2L$$

donde:

A=ALINE, B=COSENO, C=EDITEXT, D=NED, E=LCSR, F=BI_SIM, G=UNIGRAM, H=BIGRAM, I=TRIGRAM, J=PREFIX, K=SOUNDEX, L=SUFIX.

De esta manera, el modelo combinatorio de las funciones de similitud utilizadas en este primer experimento, es similar a los resultados mostrados en el trabajo de (Millán, 2016). Transformando los valores de modelo obtenido a porcentajes de participación, se obtiene:

ALINE=0.177, COSENO=0.088, EDITEXT=0.111, NED=0.044, LCSR=0, BI_SIM=0.177, UNIGRAM=0.022, BIGRAM=0, TRIGRAM=0.066, PREFIX=0.266, SOUNDEX=0, SUFIX=0.044.

Con los datos de participación obtenidos, es posible tener una comparación más precisa contra los resultados obtenidos por (Millán, 2016). En la Tabla 6.1 se muestran las funciones de similitud utilizadas en el trabajo de (Millán, 2016), así como el valor de participación obtenido por la aplicación del método de (Millán, 2016) y por el método propuesto.

Tabla 6.1 Valores de participación de las funciones de similitud obtenidos por el método de (Millán, 2016) y por el método propuesto

Función de similitud	Participación (Millán, 2016)	Participación Método propuesto
ALINE	**0.135**	**0.177**
COSENO	0.123	0.088
EDITEXT	0.008	0.111
NED	**0.240**	0.044
LCSR	0.063	0
BI_SIM	0.008	**0.177**
UNIGRAM	0.043	0.022
BIGRAM	0.013	0
TRIGRAM	0.046	0.066
PREFIX	**0.254**	**0.266**
SOUNDEX	0.023	0
SUFIX	0.044	0.044

En la Tabla 6.1 puede observarse que el método basado en algoritmos genéticos de (Millán, 2016) asigna un valor de participación a cada función de similitud utilizada, aunque sea un valor mínimo (muy pequeño, cercano a cero), por lo cual cada función tiene un valor de participación. Mientras que el método propuesto, genera valores de participación vacíos (cero) para algunas funciones de similitud, logrando que los valores que no fueron asignados de funciones sean asignados a otras funciones de similitud. También, se resaltan los valores de participación más altas obtenidos por ambos métodos: con el método de (Millán, 2016) las funciones de similitud que mayor valor de participación aportan son PREFIX=0.254, NED=0.240, ALINE=0.135; mientras que con el método propuesto en esta tesis, las medias con mayor participación son PREFIX=0.266, ALINE=.0177, BI_SIM=0.177. Como se puede observar, ambos métodos coinciden en que las funciones PREFIX y ALINE, son funciones de similitud que aportan gran participación en la identificación de pares de medicamentos confusos. El método de evaluación basado en el F-measure acumulado, con $n = 4$, fue utilizado para comparar de manera general los resultados obtenidos por el método de (Millán, 2016) y método propuesto en esta tesis. La Tabla 6.2 muestra el F-measure acumulado obtenido por ambos métodos, y

como se puede observar, el método propuesto en esta tesis (basado en regresión simbólica) obtiene un mejor F-measure que el método de (Millán, 2016) basado en algoritmos genéticos.

Tabla 6.2 F-measure acumulado obtenido por el método de (Millán, 2016) y el método propuesto

Método	F-Measure
(Millán, 2016)	1.76804
Método propuesto	1.77530

Experimento 2

Para el experimento 2, no se limitó la linealidad de combinación en el proceso evolutivo, es decir, se agregaron más funciones matemáticas con las cuales el método pedirá explorar. Las funciones matemáticas que se utilizaron para la exploración fueron (Suma, Resta, Multiplicación, División, Potencia), esto con el fin de obtener un mejor resultado en el paradigma de evaluación basado en el f-measure acumulado. A continuación, se presenta el modelo obtenido:

$$(2C + F)^{B+I} + 2J^{A} + 7A + 5B + 6C + 3D + E + 9F + H + 3I + 7J + 2L$$

donde:

A=ALINE, B=COSENO, C=EDITEXT, D=NED, E=LCSR, F=BI_SIM, G=UNIGRAM, H=BIGRAM, I=TRIGRAM, J=PREFIX, K=SOUNDEX, L=SUFIX.

Debido al tipo de modelo obtenido que representa la combinación de las funciones de similitud, no es posible establecer un valor de participación de las funciones utilizadas, tal como se hizo en el experimento 1. A pesar de que en este experimento se estableció un amplio nivel de exploración, dos funciones de similitud no fueron utilizadas para obtener los resultados: UNIGRAM y SOUNDEX. Lo cual indica que su aportación para resolver el problema es nula.

El valor de F-measure acumulado obtenido por el modelo generado en el experimento 2 es de **1.79872**, el cual, es superior al F-measure mostrado en (Millán, 2016) y al F-measure del experimento 1.

Experimento 3

Para el experimento 3, se amplió el espacio de exploración, se agregaron otras funciones matemáticas (Suma, Resta, Multiplicación, División, Potencia, Raíz cuadrada, Valor Absoluto,

Logaritmo natural) con el fin de mejorar los resultados. El modelo obtenido por el método propuesto se muestra a continuación:

$$\left(\left(AC^2GJ^{12}(F+J)(FJ^3+J)(2CJ^3(F+J)+FHL)\right)+\left(AF^3J^3L+(CJ^2)^{C^3G^2J^2}+C^2GJ\ (F+J)\right)\right)$$

donde:

A=ALINE, B=COSENO, C=EDITEXT, D=NED, E=LCSR, F=BI_SIM, G=UNIGRAM, H=BIGRAM, I=TRIGRAM, J=PREFIX, K=SOUNDEX, L=SUFIX.

En el modelo obtenido por el experimento 3, las funciones COSENO, NED, LCSR, TRIGRAM no tienen participación alguna, es decir, no ayudan a mejorar los resultados en la tarea. Al igual que en el modelo obtenido por el experimento 2, en este modelo no se puede obtener el valor de participación de cada medida, debido a la no linealidad el modelo encontrado.

El F-measure acumulado obtenido por este modelo es **1.81430**, el cual es mayor al F-measure mostrado en (Millán, 2016), y mayor a los F-measure obtenidos en los experimentos 1 y 2.

En la Tabla 6.3 se muestran los resultados obtenidos por el método propuesto, en comparación con los resultados mostrados en (Millán, 2016). Como puede observarse, los resultados obtenidos por los experimentos realizados por el método propuesto en este trabajo, superan al resultado de (Millán, 2016), además de que cada experimento mejora al anterior.

Tabla 6.3 Resultados obtenidos mediante el método de (Millán, 2016) y el método propuesto

Método		F-Measure
(Millán, 2016)		1.76804
Método propuesto	Experimento 1	1.77530
	Experimento 2	1.79872
	Experimento 3	1.81430

En la Tabla 0.1 de la sección anexos, se presentan los parámetros de configuración utilizados en los tres experimentos anteriormente mencionados. Los parámetros ahí indicados fueron utilizados para configurar el comportamiento de sistema de aprendizaje automático basado en regresión simbólica aplicado a la detección de nombres de medicamentos confusos.

6.1.3 Comparación de resultados

Como se mencionó en la sección 5.2.5 Comparación de rendimiento entre dos métodos, es necesario contar con un modelo que permita saber la mejora porcentual de la mejora de

resultados de un sistema contra otro. El modelo utilizado es el mencionado en (Mendoza *et al.*, 2014):

$$\frac{Método_{Propuesto} - Método_{referencia}}{Método_{referencia}} \times 100$$

En este caso, el $Método_{referencia}$ es el resultado generado por el método de (Millán, 2016). En la Tabla 6.4 se muestra el resultado de mejora porcentual obtenido al comparar los resultados obtenidos por los experimentos realizados con el método propuesto, frente al resultado mostrado por (Millán, 2016), para la tarea de detección de nombres de medicamentos confusos.

Tabla 6.4 Mejora porcentual de los resultados obtenidos por el método propuesto frente al resultado obtenido por (Millán, 2016)

Experimento	(Millán, 2016)	Propuesto	Mejora
Experimento 1	1.7680	1.7753	0.41 %
Experimento 2		1.7987	1.74 %
Experimento 3		1.8143	2.62 %

A partir de los resultados obtenidos por los experimentos anteriores, se pueden determinar lo siguiente:

- El algoritmo genético utilizado en (Millán, 2016) es un método de combinación lineal restringido, debido a que siempre asigna un valor de participación a cada medida de similitud utilizada, aunque el valor de participación sea muy bajo, siempre coloca algún valor. Mientras que el método propuesto (basado en regresión simbólica) puede trabajar bajo combinación lineal no restringida, es decir, puedo no asignar valor alguno de participación a las funciones de similitud utilizadas, y esto no afecta al resultado final de evaluación, al contrario, lo mejora, puesto que, al no asignar valores a ciertas funciones, ese valor puede ser repartido en las funciones que realmente ayudan a mejorar los resultados en la tarea. Tal como se ve en el modelo del experimento 1, donde las funciones LCSR, BIGRAM y SOUNDEX no tienen participación alguna, y aun así, el F-measure acumulado final (**1.77530**) es mejor que el F-measure indicado en (Millán, 2016) (**1.76804**).
- Se puede observar, que, al combinar las funciones de similitud utilizadas bajo un método no lineal, el resultado del F-measure acumulado es mejor que el obtenido combinando las funciones bajo un método lineal. La regresión simbólica es un método de combinación restrictivo, se le puede indicar el método de combinación. El

experimento 1, fue un experimento realizando una combinación lineal, y obtuvo un F-measure acumulado de **1.77530**, mientras que los experimentos 2 y 3, fueron realizados bajo una combinación no lineal, lo cual permitió obtener un f-measure acumulado superior al lineal, de **1.79872** y **1.81430** respectivamente.

- La mejora porcentual obtenida por cada uno de los experimentos realizados es notable, específicamente en los experimentos 2 y 3, que son los experimentos realizados con la aplicación directa del método propuesto (basado en regresión simbólica). El experimento 1, fue realizado con el método propuesto, pero forzado a trabajar de manera similar al método de (Millán, 2016), a pesar de esto, el resultado obtenido es superior, aunque su mejora porcentual no es tan notable.
- Finalmente, se puede decir, que la combinación no lineal de funciones de similitud permite mejorar los resultados finales, además de que reduce el número de funciones de similitud utilizadas. Esta afirmación se puede comprobar en los experimentos 2 y 3, en los cuales, el F-measure acumulado fue mejor a los experimentos anteriores, utilizando una cantidad de funciones de similitud menor. Por lo cual, es posible que, aumentando las funciones matemáticas que ayudan a la combinación no lineal de las funciones de similitud, la cantidad de funciones de similitud disminuyan, pero aumente el F-measure final.

6.2 Desambiguación del sentido de las palabras

La desambiguación del sentido de las palabras (Word Sense Disambiguation) es la capacidad de identificar el significado correcto de las palabras, a partir del contexto en el que se emplean, mediante técnicas computacionales (Navigli, 2009). Desafortunadamente, la identificación del significado que las palabras pueden asumir de acuerdo con su contexto no es sencillo. Mientras que, desambiguar palabras es una tarea fácil para los humanos, para una computadora es un proceso complejo y generalmente, para tratar de resolverlo, hace uso de técnicas de procesamiento de lenguaje natural (Yuan *et al.*, 2016).

En los últimos años, se han incrementado las investigaciones relacionadas a la desambiguación del sentido de las palabras (Navigli, 2009), ha surgido propuestas y enfoques para tratar de resolver este problema, los cuales generalmente se clasifican de acuerdo al "conocimiento" utilizado para desambiguar (Pérez, 2009). Por un lado, están los métodos que utilizan diccionarios, tesauros, bases de conocimiento léxicas, que utilizan corpus (ya sea etiquetado o no). Por el otro lado están los métodos no supervisados, que son aquellos que evitan información externa y trabajan directamente con corpus no etiquetados.

Dentro de estos últimos, un camino explorado para la desambiguación del sentido de las palabras ha sido aquel en que los textos son representados mediante grafos (Mihalcea *et al.*, 2004) en el cual, las palabras (u oraciones) están interconectadas con relaciones significativas, para de esta manera, llevar a cabo la aplicación de rankeo de grafos. El rankeo de grafos es una técnica que nos permite saber la importancia de cada vértice dentro del grafo con base en la información almacenada en el mismo grafo (Agarwal, 2006). Una idea interesante es la de aplicar el algoritmo de *PageRank* (Grin *et al.*, 1998) a grafos generados a partir de conocimientos léxicos y semánticos (tal como la información almacenada en Wordnet (Miller, 1985)), y de esta forma, ponderar los vértices basándose en el conocimiento de estos conjuntos de datos (Mihalcea *et al.*, 2004).

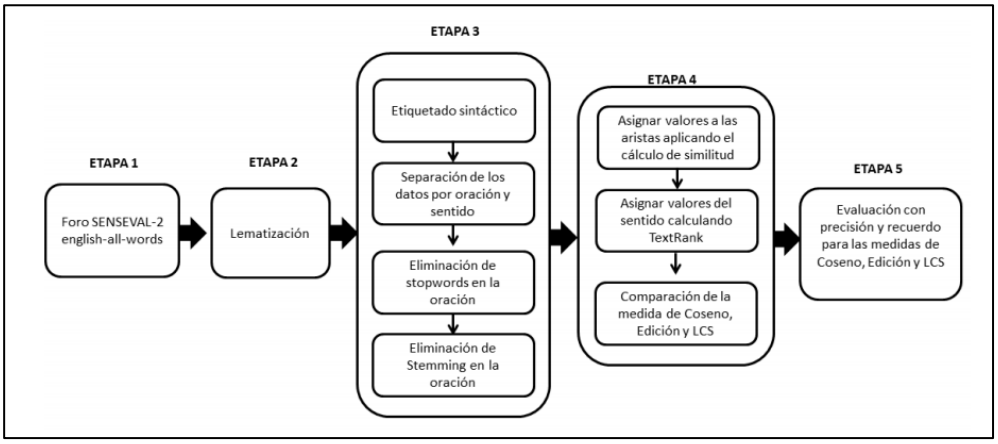
Trabajos basados en esta idea (Mihalcea *et al.*, 2004) (Vargas, 2016), han propuesto la desambiguación del sentido de todas las palabras ambiguas encontradas en un documento, representando este documento como un grafo co-ocurrente, donde cada vértice representa una palabra (y su sentido especifico extraído de Wornet), se calculan los valores (pesos) de cada vértice, y de esta manera seleccionar el sentido (vértice) correspondiente a cada palabra ambigua con base en el ranking generado. El valor de los vértices es calculado mediante el algoritmo PageRank. PageRank como valores de entrada, necesita el valor (peso) que se les otorga a las aristas del grafo, y el peso de las aristas es calculado mediante la función de similitud Coseno (u otra función de similitud).

Los trabajos de (Mihalcea *et al.*, 2004) y (Vargas, 2016), tienen en su base, los tres componentes antes descritos del procesamiento de lenguaje natural (representación de información, cálculo de similitud, evaluación del sistema), por lo cual, el aplicable el método descrito en el capítulo 5.2.

Para evaluar un sistema de desambiguación del sentido de las palabras, es necesario contar con un conjunto de datos estandarizado que permita la comparación entre sistemas, y generalmente, se hace uso de los datos proporcionados por Senseval-2 (Edmonds *et al.*, 2001), cuyo propósito es proporcionar un conjunto de tareas para evaluar diferentes sistemas de desambiguación del sentido de las palabras. El foro Senseval-2 ésta compuesto por 12 lenguajes divididos en 3 categorías:

- All-words: Czech, Duch, English, Estonian.
- Lexcal simple: Basque, English, Italian, Japanese, Korean, Spanish, Swedish.
- Translation: Japanese.

Figura 6.2 Método de WSD utilizado en (Flores, 2016)



En la tarea *all-words* los sistemas de desambiguación deben identificar el sentido de todas las palabras ambiguas contenidas en un texto. Para la tarea *lexical simple*, primero se muestrea el léxico, luego se encuentran instancias del muestreo en el contexto del documento, y la evaluación se realiza sobre las instancias encontradas. Y para la tarea *translation*, el sentido de las palabras es definido con base en la distinción de traducción.

La tarea *all-words* en idioma inglés consiste en desambiguar todas las palabras con contenido semántico (nombres, verbos, adjetivos y adverbios) que aparecen en los textos proporcionados. El número total de palabras a desambiguar supone un total de 2473 instancias (Pérez, 2009).

El trabajo de (Vargas, 2016) se basa en cinco etapas, ver Figura 6.2, y como conjunto de datos utiliza el conjunto de la tarea *all-words* en idioma inglés de foro SENSEVAL-2.

En seguida, un proceso de lematización, con el fin de obtener el representante de todas las palabras derivadas de una raíz. A estas palabras extraídas (lemas), se les aplica un etiquetado sintáctico (sustantivo (n), verbo (v), adjetivo (a), adverbio (r)). Una vez generado el etiquetado sintáctico a cada lema, y con base en esta etiqueta, se extraen los sentidos específicos para estos lemas de un conjunto de conocimientos léxicos y semánticos, en este caso WordNet. Para cada sentido extraído de WordNet, se crea un archivo en el cual se almacena la información extraída para cada sentido de cada lema, ya cada archivo generado, se le aplica un preprocesamiento (eliminación de stop-words, y stemming), con el fin de normalizar la información contenida en estos archivos. Estos archivos son los sentidos para cada lema extraído.

Cada archivo del conjunto *all-words* en inglés contiene lemas, y cada lema contiene sentidos, entonces, cada sentido generado por cada lema en el archivo que se está trabajando, es agregado al grafo de vértices, creando así un grafo co-ocurrente en el que todos los sentidos

del archivo están conectados. Entonces, se calcula la similitud entre cada sentido conectado por aristas, y el valor de similitud es colocado a esta arista. El resultado es un grafo no dirigido con pesos en las aristas. El grafo generado, es utilizado como elemento de entrada para el algoritmo TextRank, el cual, a través de su proceso, asigna un valor de importancia a cada uno de los nodos (sentidos) del nodo. Después de ejecutar el algoritmo, el grafo contiene pesos en cada sentido, y este peso indica la importancia del sentido en el grafo. Con los pesos de los sentidos del grafo, se selecciona el sentido con más peso de cada lema.

Una vez realizado el procesa de desambiguación a cada lema de cada archivo del conjunto de datos de Senseval-2, se procede a la evaluación. La evaluación se realiza mediante precisión y recuerdo. La precisión se calcula dividiendo cuantas de las oraciones de cada sentido son extraídas correctamente entre las oraciones correctas de los sentidos, mientras que el recuerdo calcula cuantas de las oraciones correctas no recupero el sistema. De manera general, Precisión es el valor estándar para evaluar un sistema de desambiguación del sentido de las palabras.

En la Tabla 6.5 se muestran los resultados obtenidos por los métodos de (Mihalcea *et al.*, 2004) (Vargas, 2016) aplicados a la tarea *all-words* en idioma inglés, del foro Senseval-2 en la tarea de desambiguación del sentido de las palabras. Los resultados muestran la Precisión obtenida por el cada método en cada una de las carpetas de la tarea seleccionada.

Tabla 6.5 Precisión obtenida por (Mihalcea *et al.*, 2004) y (Vargas, 2016) para desambiguación en la tarea all-words english de senseval-2

Archivo/Método	**(Mihalcea *et al.*, 2004)**	**(Vargas, 2016)**		
		COSENO	EDICIÓN	LCS
d00	58.17	39.18	58.47	27.04
d01	67.85	58.52	42.44	48.74
d02	63.81	73.18	28.81	41.21

Como puede observarse, algunos resultados mostrados por (Vargas, 2016) superan a los de (Mihalcea *et al.*, 2004), por ejemplo, aplicando el método de (Vargas, 2016) utilizando como medida de semejanza la función Coseno, mejora los resultados del archivo *d02* (de la tarea *all-words* idioma inglés), y utilizando la función de Edición, logra superar los resultados del archivo *d00*. El método de (Vargas, 2016) no logra superar los resultados del archivo d01 de (Mihalcea *et al.*, 2004) utilizando las funciones de similitud Coseno, Edición, y LCS.

Las aportaciones del trabajo de (Vargas, 2016) indican un método de desambiguación del sentido de las palabras basado en cinco etapas (Figura 6.2), el cual, en su etapa cuatro hace uso del algoritmo PageRank mostrado en (Mihalcea *et al.*, 2004). La aplicación del método,

sustituyendo la función de similitud (entre los nodos del grafo utilizado en la etapa cuatro), por otras funciones de similitud, permite la comparación del método bajo distintas maneras de calcular la similitud entre los diversos sentidos de las palabras. (Vargas, 2016) indica que la mejor forma de medir la similitud entre los sentidos que conforman el grafo creado en su etapa cuatro, es la función de similitud Coseno.

6.2.1 Método propuesto para la desambiguación del sentido de las palabras

Para objetivos de esta investigación, se hace uso del método propuesto en (Vargas, 2016), específicamente, la fase cuatro del mismo, en la cual se crea un grafo que contiene todos los sentidos de todas las palabras ambiguas que aparecen en un archivo, y se aplica una función de similitud que calcule la semejanza entre los mismos, para después aplicar el algoritmo PageRank, y de esta forma obtener los sentidos más relevantes para cada palabra ambigua.

Concretamente se realiza lo siguiente:

1) Generar el grafo co-ocurrente entre los sentidos de todas las palabras a desambiguar del texto de entrada.
2) Calcular el valor de las aristas del grafo a partir de las medidas de similitud seleccionadas. Con esto se obtendrá el conjunto $k \times n$ de datos, donde k es el número de comparaciones a utilizar con n funciones de similitud.
3) El conjunto $k \times n$ de datos, es el conjunto de datos de entrada al sistema de aprendizaje (Etapa 3 del método propuesto).
4) Como salida del sistema de aprendizaje, se obtendrá una función de similitud basada en los valores inducidos del conjunto $k \times n$ de datos. Esta función de similitud obtenida es entonces reemplazada en la parte correspondiente del la Etapa 4 del método de (Vargas, 2016).
5) Evaluación del resultado generado por la aplicación y sustitución de la función de similitud obtenida mediante el paradigma de evaluación para la tarea de desambiguación del sentido de las palabras.

Específicamente, el método propuesto es aplicable al trabajo de (Vargas, 2016), el cual consta de cinco etapas, las cuales se pueden observar en la Figura 6.2, y es en la Etapa 4 del mismo donde se aplica el método propuesto. La Etapa 4 del método mostrado en (Vargas, 2016), indica la "Asignación del valor a las aristas mediante el cálculo de similitud", por lo cual, en esta Etapa 4, es aplicable la Etapa 2 y 3 del método propuesto, en la cual, el valor asignado a las aristas es el valor generado por la función obtenida a partir de la inducción de los valores de otras funciones de similitud.

A continuación se muestran los experimentos y resultados de la aplicación del método propuesto en esta tesis, a la tarea de desambiguación del sentido de las palabras, tomando

como base el método de (Vargas, 2016), así mismo se describe un análisis de los resultados obtenidos.

6.2.2 Experimentos y resultados

En el trabajo de (Vargas, 2016) se hace uso de tres funciones de similitud para generar sus resultados (Coseno, Edición, LCS), y con cada función de similitud utilizada se generan resultados diferentes, lo cual supone que la aplicación otras funciones de similitud pueden generar diversos resultados. En los siguientes experimentos se muestra la inducción de diversas funciones de similitud aplicadas en la tarea de desambiguación del sentido de las palabras, así como los resultados obtenidos por ellas.

El conjunto de archivos para probar el método propuesto en esta tesis es el conjunto *all-words* en idioma inglés proporcionado por el foro Senseval-2, el cual está compuesto por tres archivos: *d00*, *d01*, *d02*. A cada archivo de esta colección se le aplico el método propuesto.

Experimento 1 (archivo d00)

En el experimento 1, el método propuesto en esta tesis para la desambiguación del sentido de las palabras, fue aplicado al archivo *d00* de la colección *all-words* de Senseval-2, donde el conjunto de funciones de similitud está delimitado por el conjunto {Coeficiente de Yule, Hamming, Rusell-Rao, Sokal-Sneath, Dice, Jaccard, Kulczynski, Roger-Tanimoto, Simple Matching, Similitud Yule, Damerau-Levenshtein, Levenstein, Needleman-Wunch, Jaro-Winkler, LCS, Smith-Waterman, Pearson, SpearmanRank, Camberra, Chebychev, Correlation, Euclidiana normalizada, Manhattan, Minkowski, SpearmanFootrule, Coseno, DTW, DISCO, Retina, SemSim, WMD} cuya descripción se encuentra en la sección 3.2 Clasificación de las funciones de similitud. Una restricción para que la combinación de los valores de las funciones de similitud sea correctamente inducida, es que estos se encuentren normalizados en un rango de [0,1].

Cabe destacar que, la exploración para inducir las funciones de similitud, como resultado puede devolver un modelo que puede o no contener todas las funciones ingresadas como entrada.

El modelo inducido para la carpeta *d00* es:

$$\left(\left(\left(-\frac{658}{A}+B\right)^{-\frac{658}{(C-\ln(AD))+B}}\right)^{\left(-\frac{658}{A}-C^{-\frac{329}{A}}\right)-\left(\left(\left(C-\ln\left(\frac{A}{B}\right)\right)+B\right)^{-\frac{329}{B}}\right)}\right)^{\ln\left(\left(E-\left((\ln(D))+AD\right)^{-\frac{663}{\frac{F}{G}}}\right)-\left(C^{-\frac{663}{B}}\right)\right)^{C}}$$

donde:

Z=Hamming, Y=Yule, X=WMD, W=Kulzcynski, V=DTW, U=SmithWaterman, T=Chebychev, S=Sperarman Footrule, R=Correlation.

La precisión obtenido por este modelo aplicado al archivo *d00* es de **61.90**, con el cual se supera la precisión reportada por (Mihalcea *et al.*, 2004) y (Vargas, 2016). El modelo generado está compuesto por nueve funciones de similitud, de las cuales tres son funciones basadas en datos booleanos: Yule, Hamming, Kulzcynski; cuatro funciones basadas en datos numéricos: Chebychev, Correlation, DTW y Spearman Footrule; una función basada en cadenas: Smith Waterman; y una función basada en información semántica: WMD.

Como puede observarse, las funciones basadas en datos numéricos aportan mayor participación para mejorar los resultados en el archivo *d00*, y aunque muy poca, las funciones basadas en cadenas y en información semántica influyen.

Experimento 2 (archivo d01)

Para el experimento 2, el método propuesto se aplicó al archivo *d01* de la colección seleccionada. Se hizo uso del conjunto de funciones de similitud {Coeficiente de Yule, Hamming, Rusell-Rao, Sokal-Sneath, Dice, Jaccard, Kulczynski, Roger-Tanimoto, Simple Matching, Similitud Yule, Damerau-Levenshtein, Levenstein, Needleman-Wunch, Jaro-Winkler, LCS, Smith-Waterman, Pearson, SpearmanRank, Camberra, Chebychev, Correlation, Euclidiana normalizada, Manhattan, Minkowski, SpearmanFootrule, Coseno, DTW, DISCO, Retina, SemSim, WMD}, descritas en la sección 3.2 Clasificación de las funciones de similitud. De igual manera, el cálculo de estas funciones tiene que estar normalizada en un rango de [0-1] para su correcta aplicación.

El modelo obtenido que genera los mejores resultados es:

$$Z^{(X+YZ^2)}\left(\left(618R(S+618)\sqrt{\left(223.558T^{9278}\right)}U^2V^2W^3YZ^3+W(X+Y^3)(Z(U^2+UY)+W)+XY\right)+(6UYZ)\right)$$

donde:

Z=Hamming, Y=Yule, X= WMD, W= Kulczynski, V=DTW, U= Smith-Waterman, T=ChebyCheb, S=Spearman Footrule, R= Correlation.

La Precisión obtenida por este modelo es de **63.40**. Con este modelo, el resultado obtenido para el archivo *d01* es superior a los resultados mostrados por (Vargas, 2016), pero no supera al resultado para este archivo indicado en (Mihalcea *et al.*, 2004). El modelo obtenido está

constituido por nueve funciones de similitud de las treinta y uno funciones ingresadas en el conjunto.

El mejor modelo encontrado está compuesto por cuatro funciones de similitud basadas en datos numéricos: Chebychev, Correlation, Spearman Footrule y DTW; tres funciones basadas en datos booleanos: Yule, Hamming, Kulczynski; una función basada en cadenas: Smith Waterman; y una función basada en información semántica: WMD. A pesar de un superar los resultados de (Mihalcea *et al.*, 2004), se puede observar que las funciones basadas en datos numéricos con las que aportan mayor participación en la mejora de resultados.

Experimento 3 (archivo d02)

Para el experimento 3, se hizo uso del conjunto de funciones de similitud {Coeficiente de Yule, Hamming, Rusell-Rao, Sokal-Sneath, Dice, Jaccard, Kulczynski, Roger-Tanimoto, Simple Matching, Similitud Yule, Damerau-Levenshtein, Levenstein, Needleman-Wunch, Jaro-Winkler, LCS, Smith-Waterman, Pearson, SpearmanRank, Camberra, Chebychev, Correlation, Euclidiana normalizada, Manhattan, Minkowski, SpearmanFootrule, Coseno, DTW, DISCO, Retina, SemSim, WMD}, descritas en la sección 3.2 Clasificación de las funciones de similitud. El modelo obtenido que mejora los resultados frente a (Mihalcea *et al.*, 2004) y (Vargas, 2016) es:

$$\left(\left(Y + \frac{1}{Z^{981}}\right)^{X*(-543*Y-19)+185}\right) + \left(\left(W - \frac{V}{544}\right)^{Y+\frac{1}{\left(\frac{1}{Z^{981}}+Y\right)^{981}}}\right)$$

dónde: Z= Retina, Y= Correlation, X= Yule, W=DTW, V= WMD, donde la precisión obtenida es de **64.00**. Con este modelo obtenido, se supera a la precisión indicada por (Mihalcea *et al.*, 2004), pero no supera a (Vargas, 2016) en la evaluación del archivo *d02*.

El modelo obtenido está compuesto por cinco funciones de similitud, de las cuales dos son funciones basadas en datos numéricos: Correlation, DTW; dos funciones basadas en información semántica: Retina, WMD; y una función basada en información booleana: Yule. Para este caso, se nota que la información semántica ayuda a mejorar los resultados, así como la información numérica, mientras que, la información de las funciones basadas en cadenas no aporta mejoras.

En la Tabla 0.2 de la sección anexos, se presentan los parámetros de configuración utilizados en los tres experimentos anteriormente mencionados. Los parámetros ahí indicados fueron utilizados para configurar el comportamiento de sistema de aprendizaje automático basado en regresión simbólica aplicado a la desambiguación del sentido de las palabras.

6.2.3 Comparación de resultados

A partir de los resultados obtenidos por la inducción de funciones de similitud a los diferentes archivos de la tarea *all-word* en idioma inglés, mediante la aplicación del método propuesto en esta tesis, se obtuvieron resultados competitivos. Los resultados obtenidos son mostrados en la Tabla 6.6 así como los resultados mostrados en (Mihalcea *et al.*, 2004) y (Vargas, 2016).

Tabla 6.6 Comparación de Precisión obtenida por los métodos (Mihalcea *et al.*, 2004), (Vargas, 2016) el método propuesto

Archivo/Método	(Mihalcea *et al.*, 2004)	(Vargas, 2016)			Método propuesto
		COSENO	EDICIÓN	LCS	
d00	58.17	39.18	58.47	27.04	61.90
d01	67.85	58.52	42.44	48.74	63.40
d02	63.81	73.18	28.81	41.21	64.00

Para obtener una comparación efectiva, basada en mejora porcentual de resultados, se hace uso del modelo mencionado en la sección 5.2.5 Comparación de rendimiento entre dos métodos:

$$\frac{Método_{Propuesto} - Método_{referencia}}{Método_{referencia}} \times 100$$

Para este caso, el $Método_{referencia}$ hace uso de los resultados mostrados por (Mihalcea *et al.*, 2004) para cada uno de los archivos (*d00, d01, d02*) de la tarea *all-words* idioma inglés de Senseval-2.

Archivo	(Mihalcea et al., 2004)	Método propuesto	Mejora
d00	58.17	61.90	6.03 %
d01	67.85	63.40	-7.00 %
d02	63.81	64.00	0.30 %

A partir de los experimentos realizados, y los resultados obtenidos por los mismos, y después de realizar la comparación entre los resultados obtenidos por la aplicación del método propuesto a la tarea de desambiguación del sentido de las palabras, se puede determinar lo siguiente:

- La tarea de desambiguación del sentido de las palabras sigue siendo un reto para el procesamiento de lenguaje natural.

- El uso de estructuras complejas para representar la información de los diferentes sentidos de las palabras, como lo son los grafos, mejora notablemente los resultados para esta tarea. Ejemplo claro de esto, es el trabajo de (Mihalcea *et al.*, 2004), en el cual se hace uso del algoritmo TextRank para encontrar el sentido correcto de las palabras. Así mismo, el trabajo de (Vargas, 2016) reporta resultados obtenidos por la aplicación de su método (basado también en el algoritmo TextRank), los cuales, en algunos casos son mejores.
- Como puede observarse, los resultados obtenidos por (Vargas, 2016) con la aplicación de diferentes funciones de similitud son variados, tanto entre las funciones como en la aplicación a cada uno de los archivos. A pesar de que en (Vargas, 2016) se prueban solo tres funciones de similitud, esto da pauta para, a partir de su método propuesto, probar otras funciones de similitud y, en este caso, utilizar un modelo de inducción (método propuesto) para combinar los resultados de diferente funciones de similitud, y de esta manera mejorar los resultados.
- La aplicación del método propuesto en esta tesis encaja perfectamente en el método de (Vargas, 2016).
- La aplicación del método propuesto genera resultados superiores a los reportados por (Mihalcea *et al.*, 2004), específicamente en los archivos *d00* y *d02*. Mientras que en comparación contra (Vargas, 2016), el método propuesto genera mejores resultados en los archivos *d00* y *d01*.
- Los resultados de mejora porcentual mostrados en la Tabla 6.6, indican que el método propuesto es superior en dos casos, en el caso en que la mejora es negativa (es decir, no supero al resultado), el valor porcentual negativo es bajo, lo cual indica que el resultado obtenido por el método propuesto es competitivo.

6.3 Resumen

En este capítulo fueron descritas las tareas de procesamiento de lenguaje natural que fueron seleccionadas para aplicar el método propuesto en esta investigación. El método propuesto fue aplicado a las tareas de Detección de nombres de medicamentos confusos y Desambiguación del sentido de las palabras.

Para cada tarea de procesamiento de lenguaje natural a la que fue aplicado el método propuesto, primero fue descrita la tarea, el objetivo de la misma, los problemas que afronta, objetivos y algunos métodos para resolverla. Después, se explica algún método del estado del arte relevante para dicha tarea de procesamiento de lenguaje natural, así mismo se muestran los resultados generados por dicho trabajo. El método explicado es la base por tomar y sobre

la cual se aplica el método propuesto en esta tesis. Después, se describe en qué fase el método del estado del arte explicado es aplicable el método propuesto en este trabajo. Así mismo, se describen los experimentos realizados, así como los resultados obtenidos. Finalmente, se muestra la comparación de los resultados obtenidos por los experimentos realizados frente a los resultados mostrado por el trabajo seleccionado del estado del arte seleccionado de la tarea de procesamiento de lenguaje natural en cuestión.

La aplicación del método propuesto a la detección de nombres de medicamentos confusos, se llevó a cabo mediante tres experimentos, los cuales generan resultados superiores a los mostrados en el trabajo de (Millán, 2016). En el primer experimento mostrado para esta tarea, se forzó al método propuesto a trabajar en modo similar al método de (Millán, 2016), de tal forma que, el resultado obtenido fue superior, además de obtener de manera similar a (Millán, 2016), una lista, basada en ranking, que indica la participación de cada función de similitud utilizada para mejorar los resultados. Los experimentos 2 y 3, trabajaron conforme al método propuesto, y de igual manera, los resultado obtenidos fueron superiores a los mostrados en (Millán, 2016).

El método propuesto aplicado a la tarea de desambiguación del sentido de las palabras se llevó a cabo de acuerdo con la división de la tarea *all-words* en idioma inglés del foro Senseval-2. La tarea *all-words* consta de tres archivos (*d00, d01, d02*), a los cuales se les aplico el método propuesto, realizando así tres experimentos. Los experimentos realizados muestran una mejora, dos de ellos, frente a los resultados de (Mihalcea *et al.*, 2004), específicamente en los archivos *d00* y *d02*. Así mismo, los experimentos realizados muestran mejora en los archivos *d00* y *d01* frente a los resultados mostrado en (Vargas, 2016). La mejora porcentual obtenida para esta tarea frente a (Mihalcea *et al.*, 2004) muestra mejoras significativas, exceptuando el caso en que el resultado no mejoró.

Los experimentos realizados, así como los resultados obtenidos indican que el método propuesto en esta tesis es efectivo aplicarlo en tareas de procesamiento de lenguaje natural, los resultados obtenidos son competitivos, y hasta superiores a otros resultados mostrados en trabajos del estado del arte utilizados en la tarea donde se aplica el método propuesto.

CAPÍTULO 7.

Conclusiones

El procesamiento de lenguaje natural es un conjunto de herramientas capaces de procesar el lenguaje oral y escrito mediante técnicas, métodos y herramientas computacionales que permitan la manipulación de lenguajes naturales (Bharati *et al.*, 1996) donde, lenguaje natural es aquel que expresa pensamientos y permite la comunicación entre personas (Cañon *et al.*, 2007).

De maneral general, se han creado diversas aplicaciones para facilitar el acceso y manipulación de la información expresada en lenguaje natural (Kao, 2007). Algunas de las tareas creadas para el procesamiento de lenguaje natural son: Recuperación de información (Baeza-Yates *et al.*, 2011), Detección de plagio (Alzahrani *et al.*, 2012), Desambiguación del sentido de las palabras (Kilgarri, 1998), Generación automática de resúmenes (Ledeneva *et*

al., 2008), Detección de nombres de medicamentos confusos (Millán, 2016), Detección de palabras clave (Yih *et al.*, 2006), Clasificación de tópicos (Sriram *et al.*, 2010), Clasificación de documentos (Aliguliyev, 2009), entre otras.

A pesar de que las técnicas, métodos y sistemas de procesamiento de lenguaje natural son variados, existen características tres que la mayoría de ellos comparten:

1) Necesitan una forma de representación de la información descrita en lenguaje natural (Hartigan, 1975) (Lee *et al.*, 2014).
2) Requieren una función de similitud que permita la comparación entre la representación del lenguaje (Huang *et al.*, 2012) (Carmona, 2014) (Niewiadomski *et al.*, 2015).
3) Necesita de un paradigma de evaluación que permita determinar la eficacia del sistema o herramienta frente al problema que pretende resolver (Clark *et al.*, 2013).

Generalmente, la representación de la información está basada en el modelo espacio vectorial (Salton *et al.*, 1975), aunque existen otras representaciones, tal como los grafos o árboles (Sonawane *et al.*, 2014) (Massung *et al.*, 2013). Dentro del modelo vectorial, algunas de las representaciones más utilizadas son: valores booleanos, y numéricos. La función de similitud mide la semejanza que existen entre dos representaciones de información. La función de similitud a utilizar depende del modelo de representación de la información, y del valor que contenga esta representación. El paradigma de evaluación depende directamente de la tarea de procesamiento de lenguaje natural que se evaluará. Esta evaluación debe ser una medida estándar que permita la comparación contra otros sistemas que resuelvan dicha tarea.

El elemento más estudiado es la función de similitud. Se han creado diversas funciones de similitud para tratar de resolver los problemas de procesamiento de lenguaje (Bär *et al.*, 2015a), y aunque estas funciones obtienen resultados competitivos en las tareas en las que se han aplicado, no son tan relevantes (en algunas situaciones) como se espera. El problema principal de las funciones de similitud es que han sido descritas, generalmente, para trabajar de manera global, es decir, la función de similitud puede ser aplicada a diversas tareas de procesamiento de lenguaje natural.

La definición de nuevas funciones de similitud, que sean específicas para una tarea o globales, es un proceso complicado, y generalmente, y solo los expertos del dominio son capaces de proporcionar los conocimientos necesarios para la definición de nuevas funciones de similitud (Stahl, 2004). Un enfoque viable para la definición de funciones de similitud específicas para cada tarea es la inducción de las funciones existentes (o de los valores generados por ellas) a través del aprendizaje automático.

Existen diversas técnicas de procesamiento automático que permiten la inducción de valores: Redes neuronales, Máquinas de soporte vectorial, algoritmos genéticos, programación genética, entre otras. Por razones de ventajas frente a otras técnicas, la programación genética fue la técnica seleccionada para la investigación realizada en esta tesis. De manera específica, la regresión simbólica es la técnica utilizada, la cual es una aplicación directa de la programación genética.

La regresión simbólica es una técnica que permite la resolución de problemas difíciles donde la búsqueda y optimización son su objetivo principal. Esta técnica permite producir modelos matemáticos (funciones) que sean capaces de definir el comportamiento de un conjunto de datos. Debido a que la regresión simbólica es un proceso basado en programación genética, y esta a su vez basada en algoritmo evolutivos, el proceso de exploración permite en cada iteración obtener posibles mejores resultados. La aplicación de regresión simbólica dentro de las tareas de procesamiento de lenguaje natural es, generalmente, utilizada para generar funciones de similitud específicas para cada tarea. Esta aplicación, es conocida como *Learn similarity metric* (aprendizaje de funciones de similitud).

El aprendizaje de funciones de similitud, implica que, a partir de un conjunto de valores numéricos (que representan el grado de similitud entre dos representaciones de texto para elementos de una tarea de procesamiento de lenguaje natural), se pueda generar un modelo matemático (en este caso, una función de similitud) inducido, que aplicado en la fase de cálculo de similitud en una tarea de procesamiento de lenguaje natural, genere resultados competitivos, o mejores, u otros trabajos del estado del arte que evalúen dicha tarea.

El objetivo de esta tesis fue: Inducir las medidas de similitud existentes (un conjunto de ellas) mediante el uso de programación genética con el fin de mejorar los resultados en las tareas de desambiguación del sentido de las palabras, Detección de nombres de medicamentos confusos, que son tareas pertenecientes al área de procesamiento de lenguaje natural. Dado el análisis de los trabajos realizados en el estado del arte, de opto por utilizar la programación genética, y más precisamente la regresión simbólica, como herramienta de inducción de datos, debido a las ventajas que tiene sobre otras herramientas de inducción.

Diseñar y desarrollar un método que permita la inducción de los diversos valores generados por funciones de similitud, es un objetivo amplio, debido a que se deben tomar diversos factores en cuenta: generalizar el método para que pueda ser aplicado a diversas tareas de procesamiento de lenguaje natural; las funciones de similitud utilizadas, específicamente los valores generados por ellas, deben estar normalizados, para que, en la aplicación del método no se comentan errores o generen resultados falsos; el paradigma de evaluación debe estar diseñado de tal forma que permita la evaluación y comparación de resultados en la tarea de procesamiento de lenguaje en la que está siendo aplicado. Afortunadamente, algunos de los

trabajos del estado del arte, ya dan una vista amplia de un sistema de inducción, y con un análisis correcto, los métodos mostrados en el estado del arte, se puede diseñar un método general que permita la inducción de funciones de similitud y, de esta manera ser aplicado a diversas tareas del procesamiento de lenguaje natural.

El método de inducción basado en regresión simbólica resultó en un diseño generalizado el cual, se demostró, puede ser aplicado a varias tareas de procesamiento de lenguaje natural. En este trabajo, su aplicación fue mostrada a dos tareas en específico: Detección de nombres de medicamentos confusos y, Desambiguación del sentido de las palabras.

El método propuesto aplicado en la tarea de Detección de nombres de medicamentos confusos generó resultados mejores a los mostrados en el estado del arte relacionado a tal tarea, así mismo, se demostró que el método propuesto basado en regresión simbólica puede trabajar como un método combinatorio similar a un algoritmo genético sin algunas restricciones de este último, lo cual permitió, en un experimento, generar resultados similares al trabajo de (Millán, 2016). En (Millán, 2016), además de generar la combinación de funciones de similitud, reporta un *Ranking* de participación de cada una de las funciones de similitud utilizadas, indicando que, todas las funciones utilizadas aportan un grado de participación para solucionar el problema.

El experimento realizado en el cual se forzó al método propuesto a trabajar de manera similar al método de (Millán, 2016), genera mejores resultados que los mostrados en el trabajo de (Millán, 2016), además se demuestra que, del conjunto de funciones de similitud utilizadas para llevar a cabo su combinación, no todas tienen una participación. El resultado obtenido indica que al menos dos funciones de similitud tienen participación Cero en la mejora de resultados, es decir, no aportan mejoría.

En los experimentos realizados aplicando el método propuesto, regresión simbólica como tal, los experimentos realizados generan resultados mejores al reportado en (Millán, 2016), además, estos resultados superan al primer resultado (donde se forzó al método a trabajar como un algoritmo genético). Esto nos indica que, en primera, al forzar al método propuesto a que trabaje como un algoritmo genético, genera mejores resultados que este último, debido a que su exploración está limitado a una combinación simple y forzada de todos los elementos. En segunda, el método propuesto (basado en regresión simbólica) demostró obtener mejores resultados que los trabajos del estado del arte en esta tarea, lo cual indica que, para la tarea de detección de nombres de medicamentos confusos, el método propuesto en esta tesis resultó todo un éxito.

Como segunda aplicación del método propuesto en este trabajo, se tomó en cuenta la tarea de Desambiguación del sentido de las palabras perteneciente al procesamiento de lenguaje natural. Para esta tarea, el método propuesto fue probado basándose en el método de desambiguación mostrado en (Vargas, 2016), el cual basa esta basado a su vez en el método de (Mihalcea *et al.*, 2004), ambos métodos hacen uso del algoritmo TextRank como principal herramienta para desambiguar palabras.

En el trabajo de (Vargas, 2016), se hace uso de tres funciones de similitud para obtener sus resultados. Estas tres funciones de similitud son aplicadas de manera separada, por lo cual reporta resultados para cada una de ellas. El conjunto de datos utilizado en el trabajo de (Vargas, 2016) consta de tres sub-conjuntos, por lo tanto, la cantidad de resultados rexportados es de nueve, tres resultados por cada sub-conjunto de datos.

La aplicación del método propuesto a la tarea de desambiguación del sentido de las palabras, utilizando el conjunto de datos indicado por (Vargas, 2016) y (Mihalcea *et al.*, 2004), indica mejora en los resultados, pero no de manera general. Comprando los resultados generados por el método propuesto contra los resultados mostrados por (Mihalcea *et al.*, 2004), el método propuesto supera a los de (Mihalcea *et al.*, 2004) en dos ocasiones, siendo tres resultados los que reporta (Mihalcea *et al.*, 2004). De manera similar, el método propuesto comparado contra los resultados mostrado por (Vargas, 2016), supera solo dos de los tres mejores resultados reportados. El método propuesto, en comparación contra los trabajos de (Vargas, 2016) y (Mihalcea *et al.*, 2004) tiene una mejora de 66 %, debido a que en solo dos de los tres resultados rexportados por ambos trabajos son superados por el método propuesto.

La aplicación del método propuesto a la tarea de desambiguación del sentido de las palabras resultó una aplicación más compleja, tanto en tiempo, como en procesamiento, pero a pesar de eso, de manera general supera los resultados indicados por trabajos del estado del arte relacionados a esta tarea. Por lo cual, se puede concluir que, el método propuesto (basado en regresión simbólica), el cual induce valores de funciones de similitud, es un método viable para mejorar los resultados en esta tarea, por lo tanto, la aplicación del método es exitosa.

En este trabajo, se propone la aplicación de un método que permite inducir los valores producidos en por diversas funciones de similitud, el cual, permita mejorar los resultados reportados por trabajos del estado del arte relacionados a tareas del procesamiento de lenguaje natural. Para demostrar que el método propuesto es efectivo, se realizó su aplicación a dos tareas: detección de nombres de medicamentos confusos y desambiguación del sentido de las palabras. En ambas tareas, el método propuesto género, de manera general, buenos resultados y superiores a los indicados en otros trabajos relacionados del estado del arte. Esto nos permite concluir que, el método propuesto es viable para mejorar los resultados en la tareas de procesamiento de lenguaje natural, siempre y cuando estas tareas cuenten

con tres elementos esenciales: Representación de información bajo un modelo, uso de funciones de similitud que permita calcular la semejanza en la representación de los elementos de información y, un paradigma de evaluación que permita evaluar el sistema de procesamiento de lenguaje natural y que permita la comparación efectiva entre diferentes sistemas.

7.1 Aportaciones

Las aportaciones generadas por esta investigación son las siguientes:

- Se describen tres elementos que comparten la mayoría de las tareas de procesamiento de lenguaje natural: Representación de información, Función de similitud, Paradigma de evaluación.
- Creación de un método de inducción de funciones de similitud basado en regresión simbólica, el cual puede ser aplicado a diversas tareas de procesamiento de lenguaje natural.
- Categorización de diversas funciones de similitud descritas en trabajos del estado del arte. Se detectaron cuatro grupos de funciones de similitud: funciones basadas en datos booleanos, funciones basadas en datos numéricos, funciones específicas para cadenas de texto, funciones que utilizan información semántica de la información.
- Se demostró la aplicación del método propuesto a dos tareas del procesamiento de lenguaje natural.
- Se demostró que el método propuesto puede trabajar con distintas funciones de similitud.
- Se reportan nuevos resultados, que son superiores a trabajos del estado del arte.

7.2 Trabajo futuro

Como trabajo futuro se espera la aplicación del método propuesto a otras tareas de procesamiento de lenguaje natural, para así demostrar que el método es general y puede ser aplicado a diversas tareas. También se espera seguir documentando funciones de similitud que existan en otros trabajos del estado del arte.

Por otro lado, se espera modificar el método propuesto para que, en lugar de trabajar en la inducción de los valores de funciones de similitud existentes, pueda inducir una función de similitud a partir de los datos correspondiente a cada una de las tareas de procesamiento de lenguaje natural.

El actual rendimiento computacional del método propuesto es factible para realizar pruebas dentro de un rango de una prueba por cada 2 días. Se pretende optimizar el procesamiento del método propuesto para que, el tiempo de espera para cada prueba sea más corto en tiempo.

Referencias Bibliográficas

Agarwal, S. Ranking on graph data. Proceedings of the 23rd international conference on Machine learning, 2006. ACM, 25-32.

Aliguliyev, R. M. 2009. Performance evaluation of density-based clustering methods. *Inf. Sci.,* 179**,** 3583-3602.

Alzahrani, S. M., Salim, N. & Abraham, A. 2012. Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* 42**,** 133-149.

Allen, J. 1994. *Natural Language Understanding*, The Benjamin/Cummings Publishing Company, Inc.

Arroyo, Á., Tricio, V., Corchado, E. & Herrero, Á. 2015. A Comparison of Clustering Techniques for Meteorological Analysis. *In:* HERRERO, Á., SEDANO, J., BARUQUE, B., QUINTIÁN, H. & CORCHADO, E. (eds.) *10th International Conference on Soft Computing Models in Industrial and Environmental Applications.* Cham: Springer International Publishing.

ASHP 1993. ASHP guidelines on preventing medication errors in hospitals. *American Journal of Health-System Pharmacy,* 50**,** 305-314.

Avello, D. G. 2005. *blindLight: Una nueva técnica para procesamiento de texto no estructurado mediante vectores de n-gramas de longitud variable con aplicación a diversas tareas de tratamiento de lenguaje natural.* Tesis Doctoral Ph., Universidad de Oviedo.

Baccour, L., Alimi, A. M. & John, R. I. 2014. Some notes on fuzzy similarity measures and application to classification of shapes recognition of arabic sentences and mosaic. *IAENG International Journal of Computer Science,* 41**,** 81-90.

Back, T., Fogel, D. B. & Michalewicz, Z. 1997. Handbook of Evolutionary Computation.

Baeza-Yates, R. & Ribeiro-Neto, B. 2011. *Modern Information Retrieval: The Concepts and Technology Behind Search*, Addison Wesley.

Balestre, M., Von Pinho, R., Souza, J. & Lima, J. 2008. Comparison of maize similarity and dissimilarity genetic coefficients based on microsatellite markers. *Genet. Mol. Res,* 7**,** 695-705.

Banzhaf, W., Francone, F. D., Keller, R. E. & Nordin, P. 1998. *Genetic programming: an introduction: on the automatic evolution of computer programs and its applications*, Morgan Kaufmann Publishers Inc.

Baquela, E. G. & Redchuk, A. 2013. *Optimización Matemática con R. Volumen I: Introducción al modelado y resolución de problemas,* Madrid, España, Bubok Publishing S.L.

Bär, D., Biemann, C., Gurevych, I. & Zesch, T. 2012. UKP: computing semantic textual similarity by combining multiple content similarity measures. *Proceedings of the First Joint Conference on Lexical and Computational Semantics***,** 435-440.

Bär, D., Zesch, T. & Gurevych, I. 2015a. Composing Measures for Computing Text Similarity. *UKP Lab.* Darmstadt, Germany: Technische Universität Darmstadt.

Bär, D., Zesch, T. & Gurevych, I. 2015b. Composing Measures for Computing Text Similarity. UKP Lab, Technische Universität Darmstadt, Darmstadt, Germany.

Barmpalexis, P., Kachrimanis, K., Tsakonas, A. & Georgarakis, E. 2011. Symbolic regression via genetic programming in the optimization of a controlled release pharmaceutical formulation. *Chemometrics and Intelligent Laboratory Systems,* 107**,** 75–82.

Behzadi, F. 2015. Natural language processing and machine learning: A review. *International Journal of Computer Science and Information Security,* 13**,** 101.

Bellet, A. & Cord, M. Similarity and distance metric learning with applications to computer vision. ECML PKDD - European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2015 Porto, Portugal. ECML/PKDD 2015.

Benesty, J., Chen, J., Huang, Y. & Cohen, I. 2009. Pearson correlation coefficient. *Noise reduction in speech processing.* Springer.

Berlanga, F. J. 2010. *Aprendizaje de Sistemas Basados en Reglas Difusas Compactos y Precisos con Programación Genética.* Masters P.h.D., University of Granada.

Bernal, A. R., Macorra, Zamora, M. & Alvarenga, J. C. L. 2015. ¿Cómo y cuándo realizar un análisis de regresión lineal simple? Aplicación e interpretación. *Dermatología Revista Mexicana,* 55**,** 395-402.

Bharati, A., Chaitanya, V. & Sangal, R. 1996. *Natural Language Processing A Paninian Perspective*, PHI Learning.

Bilenko, M., Basu, S. & Mooney, R. J. 2004. Integrating constraints and metric learning in semi-supervised clustering. *Proceedings of the twenty-first international conference on Machine learning.* Banff, Alberta, Canada: ACM.

Black, P. E. 2004. *Euclidean Distance* [Online]. Available: http://www.nist.gov/dads/HTML/euclidndstnc.html [Accessed 22/01 2016].

Blanco, J. D. 2009. *Generación de características lexicológicas para el Procesamiento del Lenguaje Natural.* Ingeniería, Universidad de Matanzas "Camilo Cienfuegos"

Bordignon, F. & Tolosa, G. 2007. Recuperación de información: un área de investigación en crecimiento. *Ciencias de la Información,* 38**,** 13-24.

Boriah, S., Chandola, V. & Kumar, V. {Similarity measures for categorical data: A comparative evaluation}. *red,* 30**,** 3.

Braga, L. P. V., L.I.O.V.S.S.R.C., Valencia, L. I. O. & Carvajal, S. S. R. 2015. *Introducción a la Minería de Datos*.

Brameier, M. F. & Banzhaf, W. 2006. *Linear Genetic Programming (Genetic and Evolutionary Computation),* Secaucus, NJ, USA., Springer-Verlag New York, Inc.

Brookshear, J. G. & Peake, E. M. 1993. *Teoría de la computación: lenguajes formales, autómatas y complejidad*, Addison-Wesley Iberoamericana Espana, S.A.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N. & Hullender, G. 2005. Learning to rank using gradient descent. *Proceedings of the 22nd international conference on Machine learning.* Bonn, Germany: ACM.

Cabarcos, E. G. H. & María Pilar, O. 2005. Psicolingüística, neurolingüística, logopedia y lingüística clínica: Juntos sí, pero no revueltos. *Revista de Filología y Lingüística de la Universidad de Costa Rica,* 31**,** 163-185.

Cambria, E. & White, B. 2014. Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]. *IEEE Computational Intelligence Magazine,* 9**,** 48-57.

Can, B. & Heavey, C. 2011. Comparison of experimental designs for simulation-based symbolic regression of manufacturing systems. *Computers and Industrial Engineering,* 61**,** 447-462.

Cañon, P. A. B. & Correa, S. R. 2007. PLN Procesamiento de Lenguaje Natural en la Recuperación de Información. Colombia: Universidad de la Salle.

Carbonell, J. El procesamiento del lenguaje natural, tecnología en transición. *In:* CERVANTES, I., ed. Congreso de la Lengua Española, 1992 Sevilla, España. Centro Virtual Cervantes.

Cárdenas, J. P., Olivares, G. & Alfaro, R. 2014. Clasificación automática de textos usando redes de palabras *Revista Signos,* 47**,** 346-364.

Carmona, M. Á. Á. 2014. *Detección de similitud semántica en textos cortos.* Ms, Instituto Nacional de Astrofísica, Optica y Electrónica, INAOE.

Cavallo, B., D'Apuzzo, L. & Marcarelli, G. 2010. Pairwise Comparison Matrices: Some Issue on Consistency and a New Consistency Index. *In:* GRECO, S., MARQUES PEREIRA, R. A., SQUILLANTE, M., YAGER, R. R. & KACPRZYK, J. (eds.) *Preferences and Decisions: Models and Applications.* Berlin, Heidelberg: Springer Berlin Heidelberg.

Clark, A., Fox, C. & Lappin, S. 2013. *The Handbook of Computational Linguistics and Natural Language Processing*, Wiley.

Codesido, A. I. 1999. Las discapacidades comunicativas en edad infantil: algunas implicaciones teóricas desde la Lingüística Clínica. *Revista de Logopedia, Foniatría, Audiología,* 18**,** 194-204.

Collobert, R. & Weston, J. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. *Proceedings of the 25th international conference on Machine learning.* Helsinki, Finland: ACM.

Copestake, A. 2003. Natural Language Processing: part 1 of lecture notes. *Lecture*. University of Cambridge, Computer Laboratory.

cortical.io. 2017. *Cortical.io - Fast, precise, intuitive NLP* [Online]. Available: http://www.cortical.io/ 2017].

Cha, S.-H. 2007. Comprehensive survey on distance/similarity measures between probability density functions. *City,* 1**,** 1.

Chang, C.-C. & Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.,* 2**,** 1-27.

Chang, D.-J., Desoky, A. H., Ouyang, M. & Rouchka, E. C. Compute pairwise manhattan distance and pearson correlation coefficient of data points with gpu. Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009. SNPD'09. 10th ACIS International Conference on, 2009. IEEE, 501-506.

Chatterjee, S. & Hadi, A. S. 2013. *Regression Analysis by Example*, Wiley.

Chen, J., Tang, Y. Y., Chen, C. L. P., Fang, B., Shang, Z. & Lin, Y. 2014. Similarity Measure Learning in Closed-Form Solution for Image Classification. *The Scientific World Journal,* 2014**,** 747105.

Chen, P. Y. & Popovich, P. M. 2002. *Correlation: parametric and nonparametric measures*, Sage Publications.

Chen, S., Ma, B. & Zhang, K. 2009. On the similarity metric and the distance metric. *Theoretical Computer Science,* 410**,** 2365-2376.

Cheng, W. & Hüllermeier, E. 2008. Learning Similarity Functions from Qualitative Feedback. *In:* ALTHOFF, K.-D., BERGMANN, R., MINOR, M. & HANFT, A. (eds.) *Advances in Case-Based Reasoning.* Springer Berlin Heidelberg.

Choi, S.-S., Cha, S.-H. & Tappert, C. C. 2010. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics,* 8**,** 43-48.

Chopra, S., Hadsell, R. & LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 20-25 June 2005 2005. 539-546 vol. 1.

Chu, H. 2003. *Information Representation and Retrieval in the Digital Age*, American Society for Information Science and Technology.

Dabhi, V. K. & Vij, S. K. 2011. Empirical modeling using symbolic regression via postfix Genetic Programming. *Image Information Processing (ICIIP), 2011 International Conference on.*

Damerau, F. J. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM,* 7**,** 171-176.

Darwin, C. 1859. *On the Origin of Species by Means of Natural Selection Or The Preservation of Favoured Races in the Struggle for Life*, J. Murray.

Dattorro, J. 2010. *Convex optimization & Euclidean distance geometry*, Lulu. com.

de Assis, G. T., Laender, A. H. F., Gonçalves, M. A. & da Silva, A. S. 2007. Exploiting Genre in Focused Crawling. *In:* ZIVIANI, N. & BAEZA-YATES, R. (eds.) *String Processing and Information Retrieval: 14th International Symposium, SPIRE 2007 Santiago, Chile, October 29-31, 2007 Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg.

Decker, C. & Baade, A. 2016. Consumer perceptions of co-branding alliances: Organizational dissimilarity signals and brand fit. *Journal of Brand Management***,** 1-18.

Dekhtyar, A. 2012. Knowledge Discovery From Data. *In:* UNIVERSITY, C. P. S. (ed.). San Luis Obispo, California.

Deza, M. M. & Deza, E. 2009. *Encyclopedia of Distances*, Springer-Verlag Berlin Heidelberg.

Diaconis, P. & Graham, R. L. 1977. Spearman's Footrule as a Measure of Disarray. *Journal of the Royal Statistical Society. Series B (Methodological),* 39**,** 262-268.

Dice, L. R. 1945. Measures of the Amount of Ecologic Association Between Species. *Ecology,* 26**,** 297-302.

Do, D. Q., Rowe, R. C. & York, P. 2008. Modelling drug dissolution from controlled release products using genetic programming. 351**,** 194–200.

Dunn, G. & Everitt, B. S. 2004. *An introduction to mathematical taxonomy*, Courier Corporation.

Edmonds, P. & Cotton, S. 2001. SENSEVAL-2: overview. *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems.* Toulouse, France: Association for Computational Linguistics.

Eifring, H. & Theil, R. 2005. Linguistics for students of Asian and African languages. *Unpublished manuscript*.

Elliott, D., Hartley, A. & Atwell, E. Rationale for a multilingual corpus for machine translation evaluation. Proceedings of CL2003: International Conference on Corpus Linguistics, 2003. Lancaster University, 191-200.

Emran, S. M. & Ye, N. Robustness of canberra metric in computer intrusion detection. Proc. IEEE Workshop on Information Assurance and Security, West Point, NY, USA, 2001. Citeseer.

Fan, W., Pathak, P. & Zhou, M. 2009. Genetic-based approaches in ranking function discovery and optimization in information retrieval - A framework. *Decis. Support Syst.,* 47**,** 398-407.

FDA, U. S. F. a. D. A. 2012. *FDA and ISMP Work to Prevent Medication Errors* [Online]. Available: http://www.fda.gov/downloads/ForConsumers/ConsumerUpdates/UCM297672.pdf 29 March 2012].

FDA, U. S. F. a. D. A. 2014. *About FDA* [Online]. Available: http://www.fda.gov/downloads/ForConsumers/ConsumerUpdates/UCM297672.pdf 11 September 2014].

FDA, U. S. F. a. D. A. 2017. *FDA POCA - Avoiding Drug Name Confusion* [Online]. Available: https://www.fda.gov/downloads/drugs/resourcesforyou/industry/ucm400150.pdf.

Feng, W. & Xinshun, X. AdaGP-Rank: Applying boosting technique to genetic programming for learning to rank. Information Computing and Telecommunications (YC-ICT), 2010 IEEE Youth Conference on, 28-30 Nov. 2010 2010. 259-262.

Fogel, D. B. 2006. Evolutionary Computation: Toward a New Philosophy of Machine Intelligence, 3rd Edition. *Evolutionary Computation: toward a New Philosophy of Machine Intelligence, 3rd Edition***,** 1-274.

Fowler, R. 1974. *Para comprender el lenguaje. Una introducción a la lingüística,* México.

Friedman, C., Rindflesch, T. C. & Corn, M. 2013. Natural language processing: State of the art and prospects for significant progress, a workshop sponsored by the National Library of Medicine. *Journal of Biomedical Informatics,* 46**,** 765-773.

Gabel, T. 2003. *Learning Similarity Measures: Strategies to Enhance the Optimisation Process.* Master Thesis, University of Kaiserslautern.

Gabrilovich, E. & Markovitch, S. 2007. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. *Proceedings of the 20th international joint conference on Artifical intelligence.* Hyderabad, India: Morgan Kaufmann Publishers Inc.

Gadge, J., Sane, S. & Kekre, H. 2015. Performance Analysis of Layered Vector Space Model in Web Information Retrieval. *International Journal of Applied Information Systems (IJAIS),* 8**,** 7-15.

Gan, G., Ma, C. & Wu, J. 2007. *Data Clustering: Theory, Algorithms, and Applications*, Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

Gantz, B. J. & Reinsel, D. 2013. The digital universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the far east. *IDC iView***,** 1-16.

Gelbukh, A. 2010. Procesamiento de lenguaje natural y sus aplicaciones. *Komputer Sapiens,* I**,** 6-11.

Georgiou, L. & Teahan, W. J. 2008. Experiments with Grammatical Evolution in Java. *In:* COTTA, C., REICH, S., SCHAEFER, R. & LIGĘZA, A. (eds.) *Knowledge-Driven Computing: Knowledge Engineering and Intelligent Computations.* Berlin, Heidelberg: Springer Berlin Heidelberg.

Gestal, M. 2010. *Introducción a los algoritmos genéticos y a la programación genética*.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc.

Gomaa, W. H. & Fahmy, A. A. 2013. A survey of text similarity approaches. *International Journal of Computer Applications,* 68**,** 13-18.

Gondree, M. & Mohassel, P. 2009. Longest common subsequence as private search. *Proceedings of the 8th ACM workshop on Privacy in the electronic society.* Chicago, Illinois, USA: ACM.

Goodman, J. 2002. Sequential conditional Generalized Iterative Scaling. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics.* Philadelphia, Pennsylvania: Association for Computational Linguistics.

Goshtasby, A. A. 2012. Similarity and Dissimilarity Measures. *Image Registration.* Springer London.

Grace, G. H. & Desikan, K. 2016. Document Clustering using a New Similarity Measure based on Energy of a Bipartite Graph. *Indian Journal of Science and Technology,* 9.

Gragera, A. & Suppakitpaisarn, V. 2016. Semimetric Properties of Sørensen-Dice and Tversky Indexes. *In:* KAYKOBAD, M. & PETRESCHI, R. (eds.) *WALCOM: Algorithms and Computation: 10th International Workshop, WALCOM 2016, Kathmandu, Nepal, March 29-31, 2016, Proceedings.* Cham: Springer International Publishing.

Grin, S. & Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems,* 30**,** 107-117.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. 2009. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.,* 11**,** 10-18.

Hamady, M., Walker, J. J., Harris, J. K., Gold, N. J. & Knight, R. 2008. Error-correcting barcoded primers allow hundreds of samples to be pyrosequenced in multiplex. *Nature methods,* 5**,** 235.

Hamers, L., Hemeryck, Y., Herweyers, G., Janssen, M., Keters, H., Rousseau, R. & Vanhoutte, A. 1989. Similarity measures in scientometric research: the Jaccard index versus Salton's cosine formula. *Information Processing & Management,* 25**,** 315-318.

Hamming, R. W. 1950. Error Detecting and Error Correcting Codes. *Bell System Technical Journal,* 29**,** 147-160.

Hartigan, J. A. 1975. *Clustering Algorithms*, John Wiley & Sons, Inc.

Hauke, J. & Kossowski, T. 2011. Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data. *Quaestiones Geographicae,* 30**,** 87-93.

Hertz, T., Bar-Hillel, A. & Weinshall, D. Learning distance functions for image retrieval. Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, 27 June-2 July 2004 2004. II-570-II-577 Vol.2.

Hillel, A. B., Hertz, T., Shental, N. & Weinshall, D. 2005. Learning a Mahalanobis Metric from Equivalence Constraints. *J. Mach. Learn. Res.,* 6**,** 937-965.

Hillel, A. B. & Weinshall, D. 2007. Learning distance function by coding similarity. *Proceedings of the 24th international conference on Machine learning*. Corvalis, Oregon, USA: ACM.

Hinze, A., Heese, R., Luczak-Rösch, M. & Paschke, A. 2012. Semantic Enrichment by Non-experts: Usability of Manual Annotation Tools. *The Semantic Web – ISWC 2012: 11th International Semantic Web Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Hogenboom, F., Frasincar, F. & Kaymak, U. 2010. An Overview of Approaches to Extract Information from Natural Language Corpora. *In:* HEIJDEN, M. V. D., HINNE, M., KRAAIJ, W., KUPPEVELD, M. V., VERBERNE, S. & WEIDE, T. V. D., eds. Tenth Dutch-Belgian Information Retrieval Workshop (DIR 2010), 2010 2010 Nijmegen, The Netherlands. 69-70.

Holland, J. H. 1975. *Adaptation in natural and artificial systems,* Ann Arbor, University of Michigan Press.

Holland, J. H. 1992. *Adaptation in natural and artificial systems*, MIT Press.

Huang, A. Similarity measures for text document clustering. Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand, 2008. 49-56.

Huang, G., Guo, C., Kusner, M. J., Sun, Y., Sha, F. & Weinberger, K. Q. Supervised Word Mover's Distance. 30th Conference on Neural Information Processing Systems, 2016. 4862-4870.

Huang, L. 2011. *Concept-based Text Clustering*. Doctor of Philosophy (PhD) Thesis, University of Waikato.

Huang, L., Milne, D., Frank, E. & Witten, I. H. 2012. Learning a concept-based document similarity measure. *Journal of the American Society for Information Science and Technology,* 63**,** 1593–1608.

Hyyrö, H. 2003. A bit-vector algorithm for computing Levenshtein and Damerau edit distances. *Nord. J. Comput.,* 10**,** 29-39.

ISMP. 2015. *ISMP's List of Confused Drug Names - confuseddrugnames.pdf* [Online]. Institute for Save Medication Practices. Available: https://www.ismp.org/tools/confuseddrugnames.pdf.

Jaccard, P. 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles,* 37**,** 547-579.

Jackson, P. & Moulinier, I. 2007. *Natural Language Processing for Online Applications: Text retrieval, extraction and categorization*, John Benjamins Publishing Company.

Jaffri, A. 2007. Knowledge Enhanced Searching on the Web. *In:* ABERER, K., CHOI, K.-S., NOY, N., ALLEMANG, D., LEE, K.-I., NIXON, L., GOLBECK, J., MIKA, P., MAYNARD, D., MIZOGUCHI, R., SCHREIBER, G. & CUDRÉ-MAUROUX, P. (eds.) *The Semantic Web*. Springer Berlin Heidelberg.

Jin, R., Shijun, W. & Zhou, Z. H. Learning a distance metric from multi-instance multi-label data. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 20-25 June 2009 2009. 896-902.

Jones, K. S. & Galliers, J. R. 1996. *Evaluating Natural Language Processing Systems: An Analysis and Review*, Springer-Verlag New York, Inc.

Jurman, G., Riccadonna, S., Visintainer, R. & Furlanello, C. Canberra distance on ranked lists. Proceedings, Advances in Ranking–NIPS 09 Workshop, 2009. 22-27.

Kao, A. 2007. *Natural Language Processing and Text Mining*, Springer-Verlag London.

Kaur, A. & Budhiraja, S. 2014. On the Dissimilarity of Orthogonal Least Squares and Orthogonal Matching Pursuit Compressive Sensing Reconstruction. *In:* KUMAR KUNDU, M., MOHAPATRA, P. D., KONAR, A. & CHAKRABORTY, A. (eds.) *Advanced Computing, Networking and Informatics- Volume 1: Advanced Computing and Informatics Proceedings of the Second International Conference on Advanced Computing, Networking and Informatics (ICACNI-2014)*. Cham: Springer International Publishing.

Kiela, D. & Clark, S. A systematic study of semantic vector space model parameters. Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) at EACL, 2014. 21-30.

Kilgarri, A. Senseval: An exercise in evaluating word sense disambiguation programs. Proceedings of the first international conference on language resources and evaluation, 1998 Liège, Belgium. 581-588.

Kolb, P. 2008. Disco: A multilingual database of distributionally similar words. *Proceedings of KONVENS-2008, Berlin*.

Kolb, P. Experiments on the difference between semantic similarity and relatedness. Proceedings of the ordic Conference on Computational Linguistics (ODALIDA), 2009. 81-88.

Kolb, P. & Prochazkova, P. 2017. *Linguatools - DISCO* [Online]. Berlin. Available: http://www.linguatools.de/disco/ [Accessed 30-05 2017].

Kommenda, M., Affenzeller, M., Burlacu, B., Kronberger, G. & Winkler, S. M. Genetic programming with data migration for symbolic regression. Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion, 07/12/2014 2014 Vancouver, BC, Canada. ACM, 1361-1366.

Kondrak, G. & Dorr, B. 2006. Automatic identification of confusable drug names. *Artificial Intelligence in Medicine,* 36**,** 29-42.

Konstantinidis, S. 2007. Computing the edit distance of a regular language. *Information and Computation,* 205**,** 1307-1316.

Koza, J. R. 1992a. A Genetic Approach to Finding a Controller to Back Up a Tractor-Trailer Truck. *American Control Conference, 1992*, 307-2311.

Koza, J. R. 1992b. *Genetic programming: on the programming of computers by means of natural selection*, MIT Press.

Koza, J. R. Introduction to genetic programming. Advances in genetic programming, 08/23/1994 1994. MIT Press, 21-42.

Kulczynski, S. 1927. Die Pflanzenassoziationen der Pieninen. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres,* Classe des Sciences Mathematiques et Naturelles B**,** 57-203.

Kumar, E. 2011. *Natural Language Processing*, I.K. International Publishing House.

Kumar, N. & Kummamuru, K. 2008. Semisupervised Clustering with Metric Learning using Relative Comparisons. *IEEE Transactions on Knowledge and Data Engineering,* 20**,** 496-503.

Kuncheva, L. I. 2007. A stability index for feature selection. *Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications.* Innsbruck, Austria: ACTA Press.

Kusner, M., Sun, Y., Kolkin, N. & Weinberger, K. Q. From Word Embeddings To Document Distances. *In:* BLEI, D. & BACH, F., eds. Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 2015 Lille, France. JMLR Workshop and Conference Proceedings, 957-966.

Lambert, B. L., Lin, S.-J., Chang, K.-Y. & Gandhi, S. K. 1999. Similarity as a risk factor in drug-name confusion errors: the look-alike (orthographic) and sound-alike (phonetic) model. *Medical care,* 37**,** 1214-1225.

Lambert, B. L., Yu, C. & Thirumalai, M. 2004. A system for multiattribute drug product comparison. *Journal of medical systems,* 28**,** 31-56.

Lance, G. N. & Williams, W. T. 1967. {Mixed-Data Classificatory Programs I - Agglomerative Systems}. *Australian Computer Journal,* 1**,** 15-20.

Langer, S. 2001. Natural languages on the Word Wide Web. *Bulletin de linguistique appliquée et générale,* 26**,** 89-100.

Le, Q. V. & Mikolov, T. 2014. Distributed Representations of Sentences and Documents. *CoRR,* abs/1405.4053.

Lebanova, H. V., Getov, I. N. & Grigorov, E. E. 2012. Descriptive study for look-alike and sound-alike medicines based on local language peculiarities. *African Journal of Pharmacy and Pharmacology,* 6**,** 2161-2165.

Ledeneva, Y., Gelbukh, A. & García-Hernández, R. A. 2008. Terms Derived from Frequent Sequences for Extractive Text Summarization. *In:* GELBUKH, A. (ed.) *Computational Linguistics and Intelligent Text Processing: 9th International Conference, CICLing 2008, Haifa, Israel, February 17-23, 2008. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg.

Ledeneva, Y. N. 2008. *Automatic language-independent detection of multiword descriptions for text summarization.* Doctorado en Ciencias de la Computación Thesis, Instituto Politécnico Nacional.

Lee, M. C., Chang, J. W. & Hsieh, T. C. 2014. A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences. *The Scientific World Journal.* Hindawi Publishing Corporation.

Levenshtein, V. I. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady,* 10**,** 707.

Leydesdorff, L. 2008. On the normalization and visualization of author co-citation data: Salton's Cosine versus the Jaccard index. *Journal of the American Society for Information Science and Technology,* 59**,** 77-85.

Li, Y., Chen, H. & Wu, Z. 2010. Dynamic Time Warping Distance Method for Similarity Test of Multipoint Ground Motion Field. *Mathematical Problems in Engineering,* 2010**,** 12.

Liddy, E. D. 1998a. Enhanced Text Retrieval Using Natural Language Processing. *Bulletin of the American Society for Information Science and Technology,* 24**,** 14-16.

Liddy, E. D. 1998b. Natural language processing for information retrieval and knowledge discovery. *Visualizing subject access for 21st century information resources*.

Liddy, E. D. 2001. Natural Language Processing. *Encyclopedia of Library and Information Science,* 2nd edition**,** 2126-2136.

Lin, C.-Y. & Och, F. J. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, 2004. Association for Computational Linguistics, 605.

Lin, D. 1998. An Information-Theoretic Definition of Similarity.

Lintean, M. & Rus, V. Measuring semantic similarity in short texts through greedy pairing and word semantics. Twenty-Fifth International FLAIRS Conference, 2012.

Liu, K., Bellet, A. & Sha, F. 2015a. Similarity Learning for High-Dimensional Sparse Data. *CoRR,* abs/1411.2374.

Liu, W., Mu, C., Ji, R., Ma, S., Smith, J. R. & Chang, S.-F. 2015b. Low-Rank Similarity Metric Learning in High Dimensions. *AAAI Conference on Artificial Intelligence; Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Liu, X., Zhou, Y. & Zheng, R. Sentence Similarity based on Dynamic Time Warping. International Conference on Semantic Computing (ICSC 2007), 17-19 Sept. 2007 2007. 250-256.

Lyerly, S. B. 1952. The average spearman rank correlation coefficient. *Psychometrika,* 17**,** 421-428.

Maggini, M., Melacci, S. & Sarti, L. 2008. Learning Similarity Measures from Pairwise Constraints with Neural Networks. *In:* KŮRKOVÁ, V., NERUDA, R. & KOUTNÍK, J. (eds.) *Artificial Neural Networks - ICANN 2008.* Springer Berlin Heidelberg.

Malhotra, S. & Dixit, A. 2013. An Effective Approach for News Article Summarization. *International Journal of Computer Applications,* 76**,** 5-10.

Marcus, A. & Wong, A. 2016. Internet for All A Framework for Accelerating Internet Access and Adoption. *World Economic Forum*.

Massung, S., Zhai, C. & Hockenmaier, J. 2013. Structural Parse Tree Features for Text Representation. *Proceedings of the 2013 IEEE Seventh International Conference on Semantic Computing.* IEEE Computer Society.

McDonald, S. & Ramscar, M. Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. Proceedings of the 23rd annual conference of the Cognitive Science Society, 2001. 611-616.

Medicine, I. o. 2007. *Preventing Medication Errors,* Washington, DC, The National Academies Press.

Mehrbod, A., Zutshi, A. & Grilo, A. A vector space model approach for searching and matching product e-catalogues. Proceedings of the eighth international conference on management science and engineering management, 2014. Springer, 833-842.

Mendoza, M., Bonilla, S., Noguera, C., Cobos, C. & León, E. 2014. Extractive single-document summarization based on genetic operators and guided local search. *Expert Systems with Applications,* 41**,** 4158-4169.

Meng, L., Huang, R. & Gu, J. 2014. Measuring semantic similarity of word pairs using path and information content. *Int. J. Futur. Gener. Commun. & Netw,* 7**,** 183-194.

Merigó, J. M. & Casanovas, M. 2011. A new Minkowski distance based on induced aggregation operators. *International Journal of Computational Intelligence Systems,* 4**,** 123-133.

Merigó, J. M. & Gil-Lafuente, A. M. 2008. Using the OWA operator in the Minkowski distance. *International Journal of Computer Science,* 3**,** 149-157.

Metzler, D., Dumais, S. & Meek, C. 2007. Similarity measures for short segments of text. *Proceedings of the 29th European conference on IR research.* Rome, Italy: Springer-Verlag.

Mihalcea, R., Tarau, P. & Figa, E. PageRank on semantic networks, with application to word sense disambiguation. Proceedings of the 20th international conference on Computational Linguistics, 2004. Association for Computational Linguistics, 1126.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Milton, J. S. 2007. *Estadística para Biología y Ciencias de la Salud*.

Millán, C. E. H. 2016. *Detección de nombres de medicamentos confusos por su parecido ortográfico o fonético mediante un algoritmo genético.* Maestro en Ciencias de la Computación, Universidad Autónoma del Estado de México.

Miller, G. A. WordNet: a dictionary browser. Proceedings of the 1st International Conference on Information in Data, 1985 University of Waterloo, Waterloo, Ontario, Canada., 25-28.

Ming, L., Xin, C., Xin, L., Bin, M. & Vitanyi, P. M. B. 2004. The similarity metric. *Information Theory, IEEE Transactions on,* 50**,** 3250-3264.

Mitchell, T. M. 2006. *The discipline of machine learning*, Carnegie Mellon University, School of Computer Science, Machine Learning Department.

Monge, A. & Elkan, C. An efficient domain-independent algorithm for detecting approximately duplicate database records. Proceedings of the SIGMOD Workshop on Data Mining and Knowledge Discovery, 1997. 23-29.

Moreno, A. H. 2000. *La clasificación numérica y su aplicación en la ecología*, Instituto Tecnológico de Santo Domingo.

Muhamad, F. N., Ahmad, R., Asi, S. M. & Murad, M. 2015. REDUCING THE SEARCH SPACE AND TIME COMPLEXITY OF NEEDLEMAN-WUNSCH ALGORITHM (GLOBAL ALIGNMENT) AND SMITH-WATERMAN ALGORITHM (LOCAL ALIGNMENT) FOR DNA SEQUENCE ALIGNMENT. *Jurnal Teknologi,* 77.

Mukherjee, S. 2014. Text Processing. *Thinking in LINQ: Harnessing the power of functional programing in .NET applications.* Berkeley, CA: Apress.

Murari, A., Peluso, E., Gelfusa, M., Lupelli, I., Lungaroni, M. & Gaudio, P. 2015. Symbolic regression via genetic programming for data driven derivation of confinement scaling laws without any assumption on their mathematical form. *Plasma Physics and Controlled Fusion,* 57.

Nagata, T., Kimura, M. & Tsuchiya, F. 2014. Similarity index for sound-alikeness of drug names with pitch accents. *Procedia Computer Science,* 35**,** 1519-1528.

Navigli, R. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR),* 41**,** 10.

Needleman, S. B. & Wunsch, C. D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology,* 48**,** 443-453.

Nettleton, D. 2012. *Técnicas para el análisis de datos clínicos*.

Niewiadomski, A. & Akinwale, A. 2015. Efficient Similarity Measures for Texts Matching. *Journal of Applied Computer Science,* 23**,** 7-28.

Nocedal, J. & Wright, S. J. 2006. *Numerical Optimization*, Springer New York.

Nosofsky, R. M. 1986. Attention, similarity, and the identification-categorization relationship. *J Exp Psychol Gen,* 115**,** 39-61.

Nowak, E. & Jurie, F. Learning Visual Similarity Measures for Comparing Never Seen Objects. Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on, 17-22 June 2007 2007. 1-8.

OMS, O. M. d. l. S. 2007. *Seguridad del paciente* [Online]. Available: http://www.who.int/patientsafety/es/ May 2007].

Oram, P. 2001. WordNet: An electronic lexical database. Christiane Fellbaum (Ed.). Cambridge, MA: MIT Press, 1998. Pp. 423. *Applied Psycholinguistics,* 22**,** 131-134.

Palacios-Cruz, Lino, Pérez, M., Rivas-Ruiz, R. & Talavera, J. O. 2013. Investigación clínica XVIII. Del juicio clínico al modelo de regresión lineal. *From clinical judgment to linear regression model,* 51**,** 656--61.

PARC 2016. Natural Language Processing at PARC - PARC, a Xerox company. 28/02/2016.

Parra, R. A. M. 2007. Programación genética: La regresión simbólica. *Entramado,* 3**,** 76-85.

Pathak, R. & Thankachan, B. 2012. Natural Language Processing Approaches, Application And Limitations. *International Journal of Engineering Research & Technology* 1.

Pérez, S. V. 2009. *Resolución de la ambigüedad semántica mediante métodos basados en conocimiento y su aportación a tareas de pln.* Tesis Doctoral, Universitat d'Alacant - Universidad de Alicante.

Perlibakas, V. 2004. Distance measures for PCA-based face recognition. *Pattern Recognition Letters,* 25**,** 711-724.

Pinker, S. 1994. *El instinto del lenguaje*.

Pinto, F. R., Carriço, J. A., Ramirez, M. & Almeida, J. S. 2007. Ranked Adjusted Rand: integrating distance and partition information in a measure of clustering agreement. *BMC bioinformatics,* 8**,** 1.

Podani, J. 2000. *Introduction to the exploration of multivariate biological data*, Backhuys Publishers.

Poibeau, T. & Messiant, C. 2008. Do we still Need Gold Standards for Evaluation? Language Resource and Evaluation Conference, 2008 Morocco.

Poli, R., Langdon, W. B. & McPhee, N. F. 2008. *A Field Guide to Genetic Programming*, Lulu Enterprises, UK Ltd.

Qin, T., Liu, T.-Y., Xu, J. & Li, H. 2010. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Inf. Retr.,* 13**,** 346-374.

Qiong, C., Yiming, Y. & Peng, L. Similarity Metric Learning for Face Recognition. Computer Vision (ICCV), 2013 IEEE International Conference on, 1-8 Dec. 2013 2013. 2408-2415.

Ragalo, A. W. & Nelishia, P. A building block conservation and extension mechanism for improved performance in Polynomial Symbolic Regression tree-based Genetic Programming. Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC 2012), 2012 Mexico City. 123--129.

Ravichandran, D., Pantel, P. & Hovy, E. Randomized algorithms and nlp: using locality sensitive hash function for high speed noun clustering. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, 2005. Association for Computational Linguistics, 622-629.

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1.* Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc.

Resnik, P., Niv, M., Nossal, M., Schnitzer, G., Stoner, J., Kapit, A. & Toren, R. Using intrinsic and extrinsic metrics to evaluate accuracy and facilitation in computer-assisted coding. Perspectives in Health Information Management Computer Assisted Coding Conference Proceedings, Fall, 2006.

Riesbeck, C. K. & Schank, R. C. 1989. *Inside Case-Based Reasoning*, L. Erlbaum Associates Inc.

Ríos, M., Renato, Paredes Merino, J. S. & Oporto Díaz, S. 2008. Captchas y Programación Genética. Lima, Peru: Universidad Nacional de Ingeniería.

Rogers, D. J. & Tanimoto, T. T. 1960. A Computer Program for Classifying Plants. *Science,* 132**,** 1115-1118.

Rus, V., Banjade, R. & Lintean, M. On Paraphrase Identification Corpora. *In:* PIPERIDIS, N. C. C. C. A. K. C. A. T. D. A. H. L. A. B. M. A. J. M. A. A. M. A. J. O. A. S., ed. Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), 2014 Reykjavik, Iceland. European Language Resources Association (ELRA).

Russell, P. F. & Rao, T. R. 1940. On Habitat and Association of Species of Anopheline Larvae in South-eastern Madras. . *Journal of the Malaria Institute of India,* 3**,** 153-178.

Sahami, M. & Heilman, T. D. 2006. A web-based kernel function for measuring the similarity of short text snippets. *Proceedings of the 15th international conference on World Wide Web.* Edinburgh, Scotland: ACM.

Saito, T. & Toriwaki, J.-I. 1994. New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications. *Pattern recognition,* 27**,** 1551-1565.

Salazar, M. E. R. 2000. *Coeficientes de Asociación*, Plaza y Valdes.

Salton, G., Wong, A. & Yang, C. S. 1975. A vector space model for automatic indexing. *Commun. ACM,* 18**,** 613-620.

Schwefel, H.-P. P. 1993. Evolution and Optimum Seeking: The Sixth Generation.

Senin, P. 2008. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 1-23.

Sette, S. & Boullart, L. 2001. Genetic programming: principles and applications. *Engineering Applications of Artificial Intelligence,* 14**,** 727-736.

Shai, S. & Shai, B. 2014. Understanding machine learning: from theory to algorithms. Cambridge University Press Cambridge.

Sidorov, G., Gelbukh, A., Gómez-Adorno, H. & Pinto, D. 2014. Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Computación y Sistemas,* 18**,** 491-504.

Sokal, R. R. & Sneath, P. H. 1963. *Principles of Numerical Taxonomy,* San Francisco: W. H. .

Sonawane, S. & Kulkarni, P. 2014. Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications,* 96.

Sørenson, T. 1948. A method of stabilizing groups of equivalent amplitude in plant sociology based on the similarity of species content and its application to analyses of the vegetation on Danish commons. *Kongelige Danske Videnskabernes Selskab Biologiske Skrifter,* 5**,** 1-34.

Soto, M. R. 2009. *Generación automática de resúmenes mediante aprendizaje no supervisado.* Tesis de Licenciatura, INSTITUTO TECNOLÓGICO DE TOLUCA.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H. & Demirbas, M. 2010. Short text classification in twitter to improve information filtering. *In:* ACM (ed.) *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval.*

Stahl, A. 2002. Defining Similarity Measures: Top-Down vs. Bottom-Up. *In:* CRAW, S. & PREECE, A. (eds.) *Advances in Case-Based Reasoning.* Springer Berlin Heidelberg.

Stahl, A. 2003. *A Framework for Learning Similarity Measures in Case-Based Reasoning*. PhD PhD, University of Kaiserslautern.

Stahl, A. 2004. *Learning of Knowledge-Intensive Similarity Measures in Case-Based Reasoning.* PHD-Thesis, Technische Universität Kaiserslautern.

Stahl, A. & Gabel, T. 2003. Using Evolution Programs to Learn Local Similarity Measures. *In:* ASHLEY, K. & BRIDGE, D. (eds.) *Case-Based Reasoning Research and Development*. Springer Berlin Heidelberg.

Stern, A. & Dagan, I. A confidence model for syntactically-motivated entailment proofs. Proceedings of the International Conference on Recent Advances in Natural Language Processing, 2011. 455-462.

Stojmirović, A. & Pestov, V. 2007. Indexing schemes for similarity search in datasets of short protein fragments. *Information Systems,* 32**,** 1145-1165.

Sullivan, K. M. & Luke, S. 2007. Evolving kernels for support vector machine classification. *Proceedings of the 9th annual conference on Genetic and evolutionary computation.* London, England: ACM.

Thada, V. & Jaglan, D. V. 2013. Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *International Journal of Innovations in Engineering and Technology (IJIET),* 2**,** 202.

Tomassini, M. 1995. A Survey of Genetic Algorithms. *Annual Reviews of Computational Physics, World Scientific,* III**,** 87--118.

Tversky, A. 1977. Features of similarity. *Psychological Review,* 84**,** 327-352.

UCI. 2015. *UCI Machine Learning Repository* [Online]. Available: http://archive.ics.uci.edu/ml/ [Accessed July 2015].

Vargas, S. I. F. 2016. *Comparación de medidas de similitud para desambiguación del sentido de las palabras utilizando rankeo de grafos.* Tesis de Maestría Tesis de Maestría, Universidad Autónoma del Estado de México.

Vásquez, A. C., Huerta, H. V., Quispe, J. P. & Huayna, A. M. 2009. Procesamiento de lenguaje natural. *Revista de investigación de Sistemas e Informática,* 6.

Vázquez, E. V. 2015. *Modelo de relevancia de la posición de las oraciones en resúmenes de texto, mediante regresión simbólica.* Ingeniero en Software Tesis de Licenciatura, Universidad Autónoma Del Estado De México.

Villena, O. S. 2005. ¿Hacia dónde va la psicolingüística? *Forma y Función,* 18**,** 229-249.

Wang, G. A., Jiao, J., Abrahams, A. S., Fan, W. & Zhang, Z. 2013. ExpertRank: A topic-aware expert finding algorithm for online knowledge communities. *Decision Support Systems,* 54**,** 1442-1451.

Webber, F. D. S. 2015. Semantic Folding Theory And its Application in Semantic Fingerprinting. *arXiv preprint arXiv:1511.08855*.

Winkler, W. E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proceedings of the Section on Survey Research, 1990. 354--359.

Wissler, L., Almashraee, M., Díaz, D. M. & Paschke, A. The Gold Standard in Corpus Annotation. 2014. IEEE GSC.

Wolfram 2016. SokalSneathDissimilarity - Wolfram Language Documentation.

Wolniewicz, R. 2015. Computer-assisted coding and natural language processing. *Natural Language Processing.* 3M Health Information Systems.

WordNet. 2016. *https://wordnet.princeton.edu/wordnet/* [Online]. [Accessed may 30 2016].

Xing, E. P., Jordan, M. I., Russell, S. & Ng, A. Y. 2002. Distance Metric Learning with Application to Clustering with Side-Information. *In:* THRUN, S. & OBERMAYER, K. (eds.) *Advances in Neural Information Processing Systems 15.* Cambridge, MA: MIT Press.

Xu, J. & Li, H. 2007. AdaRank: a boosting algorithm for information retrieval. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval.* Amsterdam, The Netherlands: ACM.

Yang, F.-J. An Overview of Natural Language Generation Systems Evaluation. Proceedings of the World Congress on Engineering and Computer Science, WCECS 2015, 2015 San Francisco, USA.

Yang, X., Miao, D., Cao, F. & Ma, Y. Study on the matching similarity measure method for image target recognition. International Conference on Fuzzy Systems and Knowledge Discovery, 2005. Springer, 289-292.

Yeh, J.-Y., Lin, J.-Y., Ke, H.-R. & Yang, W.-P. Learning to Rank for Information Retrieval Using Genetic Programming. *In:* JOACHIMS, T., LI, H., LIU, T.-Y. & ZHAI, C., eds. SIGIR 2007 workshop: Learning to Rank for Information Retrieval, 2007. Microsoft.

Yih, W.-t., Goodman, J. & Carvalho, V. R. 2006. Finding advertising keywords on web pages. *Proceedings of the 15th international conference on World Wide Web*. Edinburgh, Scotland: ACM.

Yih, W.-T. & Meek, C. 2007. Improving similarity measures for short segments of text. *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*. Vancouver, British Columbia, Canada: AAAI Press.

Yih, W.-t. & Meek, C. 2010. Learning Vector Representations for Similarity Measures. Microsoft Research.

Yin, X., Chen, S., Hu, E. & Zhang, D. 2010. Semi-supervised clustering with metric learning: An adaptive kernel method. *Pattern Recognition,* 43**,** 1320-1333.

Yuan, D., Richardson, J., Doherty, R., Evans, C. & Altendorf, E. 2016. Semi-supervised Word Sense Disambiguation with Neural Models. *arXiv preprint arXiv:1603.07012*.

Yule, G. U. 1900. On the Association of Attributes in Statistics: With Illustrations from the Material of the Childhood Society, &c. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character,* 194**,** 257-319.

Zesch, T. & Gurevych, I. 2010. Wisdom of crowds versus wisdom of linguistics; measuring the semantic relatedness of words. *Nat. Lang. Eng.,* 16**,** 25-59.

Anexos

En este apartado se muestran los anexos incluyentes a la presente tesis. Estos anexos incluyen parámetros de experimentación, términos y tecnologías utilizadas.

Tabla 0.1 Parámetros de configuración para los experimentos de Detección de nombres de medicamentos confusos

Parámetro	Descripción	E1	E2	E3
TMAX	Valor máximo de constantes	100000	1000	10000
TMIN	Valor mínimo de constantes	100000	-1000	-10000
FUNSET	Conjunto de funciones	+,-	+, -, *, /, ABS, SQRT, ^	+, -, *, /, ABS, SQRT, ^
TERMSET	Conjunto de terminales	A, B, C, D, E, F, G, H, I, J, K, L	A, B, C, D, E, F, G, H, I, J, K, L	A, B, C, D, E, F, G, H, I, J, K, L
MDP	Profundidad máxima inicial	3	4	4
METODO	Método de inicialización de individuos	FULLRAMP	FULLRAMP	FULLRAMP
NPOB	Tamaño de la población	500	500	800
DIVC	División de unidad para el valor de las constantes	1	10	10
ELIT	Cantidad de elitismo a realizar por generación	2	1	2
REP	Individuos a reproducir (clonar) por generación	1	1	1
METREPRO	Método de reproducción	RULETA	TORNEO	TORNEO
NTOR	Individuos seleccionados para el operador Torneo	2	2	2
METFAT	Método de sección de padres	TORNEO	TORNEO	TORNEO
TCRUZA	Tipo de cruza a realizar	NORM	NORM	NORM
MAXP	Profundidad máxima de los individuos en la ejecución del programa	3	6	8
MAXPS	Valor que indica el aumento de MAXP en una unidad cada cierta generación	150	0	200
METMUT	Método de mutación	RAND	RAND	RAND
PMUT	Probabilidad de mutación	19	17	21
OPEDIT	Funciones del conjunto FUNSET que se "editan" en la ejecución	+,-	+, -, *, /, ABS, SQRT, ^	+, -, *, /, ABS, SQRT, ^
PEDIT	Probabilidad de edición	11	25	25
NGEN	Número de generaciones	3000	3000	7000
RING	Reinyección genética	200	200	2000

Tabla 0.2 Parámetros de configuración para los experimentos de Word sense disambiguation

Parámetro	Descripción	E1	E2	E3
TMAX	Valor máximo de constantes	1000	100000	1000
TMIN	Valor mínimo de constantes	-1000	-100000	1000
FUNSET	Conjunto de funciones	+, -, *, /, SQRT, ^, LOGN	+, -, *, /, SQRT, ^, LOGN	+, -, *, /, SQRT, ^
TERMSET	Conjunto de terminales			
MDP	Profundidad máxima inicial	5	4	4
METODO	Método de inicialización de individuos	FULLRAMP	FULLRAMP	FULLRAMP
NPOB	Tamaño de la población	500	200	500
DIVC	División de unidad para el valor de las constantes	1	1	10
ELIT	Cantidad de elitismo a realizar por generación	1	1	1
REP	Individuos a reproducir (clonar) por generación	1	1	1
METREPRO	Método de reproducción	TORNEO	TORNEO	TORNEO
NTOR	Individuos seleccionados para el operador Torneo	2	2	2
METFAT	Método de sección de padres	TORNEO	TORNEO	TORNEO
TCRUZA	Tipo de cruza a realizar	NORM	NORM	NORM
MAXP	Profundidad máxima de los individuos en la ejecución del programa	9	6	6
MAXPS	Valor que indica el aumento de MAXP en una unidad cada cierta generación	0	800	0
METMUT	Método de mutación	RAND	RAND	RAND
PMUT	Probabilidad de mutación	18	22	18
OPEDIT	Funciones del conjunto FUNSET que se "editan" en la ejecución	+, -, *, /, SQRT, ^, LOGN	+, -, *, SQRT, ^, LOGN	+, -, *, /, SQRT, ^,
PEDIT	Probabilidad de edición	35	15	35
NGEN	Número de generaciones	5000	9000	5000
RING	Reinyección genética	100	100	120