INYECCIÓN DE TÉRMINOS EN LA DESCRIPCIÓN DE SERVICIOS WEB PARA MEJORAR LA EXACTITUD DE SU CLASIFICACIÓN AUTOMÁTICA

ANTONIO JOSÉ BARRIOS HOYOS

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
FACULTAD DE INGENIERÍA EN TECNOLOGÍAS
DE INFORMACIÓN Y COMUNICACIÓN
MAESTRIA EN TECNOLOGÍAS DE INFORMACIÓN
Y COMUNICACIÓN
MEDELLÍN
2019

INYECCIÓN DE TÉRMINOS EN LA DESCRIPCIÓN DE SERVICIOS WEB PARA MEJORAR LA EXACTITUD DE SU CLASIFICACIÓN AUTOMÁTICA

ANTONIO JOSÉ BARRIOS HOYOS

Trabajo de grado para optar al título de Magíster en Tecnologías de Información y Comunicación

Director
Dr. Isaac CAICEDO-CASTRO
Doctor en Ingeniería

UNIVERSIDAD PONTIFICIA BOLIVARIANA
ESCUELA DE INGENIERÍAS
FACULTAD DE INGENIERÍA EN TECNOLOGÍAS
DE INFORMACIÓN Y COMUNICACIÓN
MAESTRIA EN TECNOLOGÍAS DE INFORMACIÓN
Y COMUNICACIÓN
MEDELLÍN
2019

Nota de aceptación

_____

_____

_____

_____

_____

Firma

Nombre:

Presidente del jurado

_____

Firma

Nombre:

Jurado

_____

Firma

Nombre:

Jurado

Medellín, 12 de junio de 2019

DECLARACIÓN DE ORIGINALIDAD

"Declaro que esta tesis (o trabajo de grado) no ha sido presentada para optar a un título, ya sea en igual forma o con variaciones, en ésta o cualquier otra universidad". Art. 82 Régimen Discente de Formación Avanzada, Universidad Pontificia Bolivariana.

Firma Autor:

A Dios, a Leo, a Marelbis, a Lida Sofía, Alejandro José, María Sofía y Luciana María.

AGRADECIMIENTOS

A los ángeles que han aparecido a lo largo de mi vida, especialmente a los que constantemente están presente, me acompañan y apoyan como familia, con rostros de madre, de esta clase he sido bendecido por la presencia de dos, Leo y Membi como cariñosamente las llamamos, con rostro de esposa, Lida Sofía, que con su decidida convicción de apoyo y amor dan sentido a mi vida, con rostros de hermanos, hijos, sobrinos, abuelos, tios y primos. A los que tienen rostro de amigos que justo aparecen cuando los necesitas a veces sin buscarlos o sin estar en contacto con ellos. Y a los que Dios pone ocasionalmente, como mi director Isaac Caicedo-Castro, que con su sabiduría, paciencia y generosidad me ha guíado en este logro.

ÍNDICE

ÍNDICE DE FIGURAS

Pág.

ÍNDICE DE TABLAS

Pág.

GLOSARIO

API: *Application Programming Interface*, aplicaciones o programas que contienen reglas, funciones o protocoles para que otras aplicaciones las usen, facilitando la comunicación o integración entre software elaborados por diferentes desarrolladores.
XML: *Extensible markup language*, es un meta lenguaje, es decir lenguaje para definir otros lenguajes, permite que la información sea estructurada con partes definidas que pueden a su vez estar conformadas por sub partes también definidas, la finalidad es describir datos –auto descripción– y no es mostrarlos.
SOAP: *Simple Object Access Protocol*, protocolo basado en XML que se usa para definir el formato de los mensajes de intercambio de información entre dos objetos.
SMTP: *Simple Mail Transfer Protocol*, protocolo de transferencia de mail, más específicamente para el envío de los mismos independiente de los sitemas de transmisión.
MIME: *Multipurpose Internet Mail Extensions*, Especificaciones para permitir el intercambio de diferentes formatos de datos en los mensajes de mail de Internet
HTTP: *HiperText Transfer Protocol*, protocolo de la capa de aplicación que permite el intercambio de datos a partir de peticiones de recursos entre un cliente y un servidor.
UDDI: *Universal, Description, Discovery and Integration*, es un directorio distribuido que contiene un listado de los registros de los Servicios Web que ofrece una organización.
OWL: *Web Ontology Language*, lenguaje usado para representar el significado de los términos de un documento en vocabularios y las relaciones entre dichos términos, para expresar más significado y semántica. Diseñado para el procesamiento de documentos por aplicaciones software.
URI: *Uniform Resource Identifier*, es una cadena de caracteres que identifica los recursos de una red de tal forma que simpre tenga el mismo significado o interpretación.

RESUMEN

En esta tesis, investigamos la aplicación del aprendizaje automático para clasificar servicios, cuya funcionalidad se describe a través de un texto breve. Por lo tanto, la clasificación de servicios se ha abordado como una tarea de minería de texto, conocida como clasificación de texto.

Los algoritmos de aprendizaje supervisado como las Máquinas de vectores de apoyo –*Support Vector Machine, SVM*– y el Perceptrón Multi-capa –*Multilayer Perceptron, MLP*– fallan en la clasificación de las descripciones de los servicios porque las descripciones breves causan problemas de concordancia de términos, por ejemplo, problemas de sinonimia y polisemia, que reducen la exactitud de la clasificación. Para abordar este problema, expandimos las descripciones de los servicios con términos de un tesauro de co-ocurrencia automáticamente generado, antes de clasificar los servicios a través de los algoritmos mencionados anteriormente.

El tesauro de co-ocurrencia también se genera a través de un algoritmo de aprendizaje automático que estima la relación latente entre los términos usados en las descripciones.

A través de validación cruzada en k-pliegues –*k-fold cross-validation*– se ajustaron tanto el modelo de expansión de descripción como los algoritmos de clasificación. Una vez ampliadas las descripciones de los servicios, durante la prueba, la exactitud del Perceptrón multicapa es de 97.92 %. A nuestro entender, este enfoque supera al mejor en el estado del arte.

PALABRAS CLAVE:

Servicios Web, Aprendizaje Automático, Tesauro de Co-ocurrencia, Factorización de Matrices, Perceptron Multicapas.

ABSTRACT

In this dissertation, we investigate the application of machine learning for classifying services, whose functionality is described through brief text. Therefore, service classification has been addressed as a text mining task, namely, text classification.

Supervised learning algorithms such as Support Vector Machine and Multilayer Perceptron fail classifying service descriptions because brief descriptions cause term-mismatch-problems (e.g., synonymy and polysemy issues) reducing classification accuracy. To tackle this problem, we expand service descriptions with terms from an automatically generated co-occurrence thesaurus before classifying services through the previosly mentioned algorithms.

The co-occurence thesaurus is also generated through a machine learning algorithm which estimates the latent relationship among terms used for describing services.

We carried out k-fold cross-validation to tune the description expansion model and classification algorithms. Once services descriptions are expanded, during the test, Multilayer Perceptron accuracy is 97.92 %. To our knowledge, this approach outperforms the best one in the state-of-the-art.

KEYWORDS:

Web service, Machine Learning, Co-occurrence Thesaurus, Matrix Factorization, Multilayer Perceptron.

INTRODUCCIÓN

En esta investigación, nuestro objetivo es abordar el problema de la clasificación automática de servicios web. La clasificación del servicio se lleva a cabo sobre su descripción. De hecho, esto se plantea como un problema de clasificación de texto.

La clasificación automática de servicios es útil para programadores o ingenieros, por lo tanto, evitan la categorización o clasificación manual de los nuevos servicios web que crean. La clasificación de servicios en su forma manual es una tarea tan propensa a errores como tediosa. Este hecho se debe a la gran cantidad de servicios disponibles y temáticas que abordan.

En la web semántica, la clasificación se lleva a cabo para identificar el dominio de los servicios. Cada dominio define su terminología a través de una ontología. La terminología determina el significado semántico de los servicios, y el ingeniero usa la ontología para anotar un nuevo servicio en el mismo dominio de otros existentes para hacerlos compatibles entre sí.

En investigaciones anteriores, la clasificación de texto se usa para categorizar servicios. Esto se debe a que cada descripción del servicio suele ser un fragmento de texto breve. Los algoritmos de aprendizaje automático se adoptan para clasificar tales descripciones. Sin embargo, el texto breve en las descripciones causa problemas de concordancia de términos, por ejemplo, sinonimia y polisemia, que reducen la exactitud de los algoritmos de aprendizaje automático.

El aprendizaje automático se ha aplicado en investigaciones anteriores para clasificar los servicios web, como se puede observar en los trabajos de Patil *et al.* (2004), Zhang *et al.* (2005), Oldham *et al.* (2005), Heß y Kushmerick (2003), Corella y Castells (2006), Crasso *et al.* (2008), Mohanty *et al.* (2012), Katakis *et al.* (2009) y Sharma *et al.* (2016), entre otros.

A nuestro entender, de todos estos trabajos, el de Sharma *et al.* (2016) supera a los

demás en términos de exactitud. En este enfoque, los investigadores usan Omiotis (Tsatsaronis *et al.*, 2010) para medir la relación semántica entre los términos. Esto se hace para complementar las descripciones del servicio con información semántica, entonces el resultado es el conjunto de vectores enriquecidos, donde cada vector representa una descripción del servicio. Esos vectores son la entrada a los algoritmos de aprendizaje automático utilizados para clasificar los servicios. Este enfoque se evaluó con los siguientes algoritmos de clasificación: k-vecino más cercano –*k-Nearest Neighbour, kNN*– y Support Vector Machine,SVM. Como resultado, SVM y kNN tienen una exactitud de 97.22 % y 94.83 %, respectivamente.

El enfoque de Sharma *et al.* (2016) tiene una deficiencia a pesar de que se comporta mejor que los enfoques anteriores. El principal inconveniente es que usa *Omiotis* como mecanismo para la medición semántica y depende de *WordNet*, que es un diccionario-tesauro de sinónimos limitado al idioma inglés. Por otra parte, WordNet no incluye con frecuencia la relación semántica entre los términos de un dominio específico, por ejemplo, los términos del idioma inglés *orange* –naranja– y *machine learning* –aprendizaje automático–, que están relacionados en el dominio de la informática.

En esta investigación, enfrentamos los problemas de inconsistencia de términos mencionados anteriormente. Para este fin, proponemos expandir las descripciones de los servicios con nuevos términos. Recuperamos dichos términos de un tesauro de co-ocurrencia generado automáticamente. Según lo mejor de nuestro conocimiento, nuestra contribución es un enfoque de mejor exactitud que los propuestos en investigaciones anteriores. Obtuvimos una exactitud del 97,92 %. Considerando SVM como línea base, teniendo en cuenta que según la literatura actual de aprendizaje automático, SVM es el algoritmo de clasificación más efectivo en la práctica (Mohri *et al.*, 2018, pg. 79). La exactitud de SVM clasificando servicios del conjunto de datos usado en el presente estudio es de 94.79 %. Con nuestro enfoque contribuímos con una ganancia de exactitud del 3.3 % sobre SVM en comparación con el enfoque de Sharma *et al.* (2016) quienes contribuyeron con una ganacia de exactitud del 2.56 % con respecto a SVM. Además, nuestro enfoque automáticamente se adapta a nuevos dominios y otros idiomas debido a que es capaz de aprender relaciones ocultas entre términos a partir del corpus de descripciones de servicios.

Presentamos también, cómo se pre-procesan las descripciones de los servicios. Se describen los detalles del modelo de expansión de la descripción del servicio y su técnica de

factorización de matriz subyacente adoptada en nuestro enfoque. Luego describimos el modelo de clasificación y la configuración experimental utilizada para evaluar nuestro enfoque.

La tesis está organizada de la siguiente forma: El capítulo 1 presenta el planteamiento del problema, en el marco del aprendizaje automático como apoyo a la web semántica, se plantea el problema de investigación, como un ejercicio de clasificación de texto y la justificación, a partir de la existencia y creación de una gran cantidad de servicios web, finalizando con la contribución de la investigación. El capítulo 2 incluye los objetivos generales y específicos. El capítulo 3 muestra el marco referencial, compuesto por el contexto, los conceptos y estado del arte donde se identifican los aspectos fundamentales de Web Semántica, descripción de servicios web, el desarrollo de un ejercicio de aprendizaje automático y los trabajos relacionados con el tema, finalmente se presenta la notación matemática usada. El capítulo 4 expone el tipo de investigación experimental de los ejercicios de aprendizaje automático, la metodología de validación, el conjunto de datos, que ha sido objeto de investigaciones anteriores relacionadas con el tema y las suposiciones y limitaciones. El capítulo 5 presenta los algoritmos de clasificación y expansión de servicios, que son los aspectos fundamentales propuestos en la investigación, los algoritmos de aprendizaje usados, que son, SVM y MLP. El capítulo 6 expone los resultados y su discusión. Posterioremente se presentan las conclusiones y posibles trabajos futuros.

1. PLANTEAMIENTO DEL PROBLEMA

En este primer capítulo se presenta el planteamiento del problema como un ejercicio de clasificación de las descripciones de los servicios web, la justificación a partir de la existencia de grandes cantidades de servicios web y dominios en los que se clasifican como apoyo a los conceptos de web semántica y las contribuciones de la investigación.

1.1. PROBLEMA

Clasificar los servicios web de la manera más exacta posible en las categorías a la que se refiere la descripción de los mismos, es decir, pertenencia a un dominio, ayuda en la ontología usada para la web semántica, el reto obedece a la gran cantidad de categorías, la gran cantidad de servicios web desarrollados y la descripción de los servicios realizada con textos cortos, que ocasiona problemas en la clasificación por la dificultad de las relaciones semánticas entre las palabras de la descripción de los servicios, como la sinonimia, palabras con igual significado, por ejemplo *gafas* y *anteojos* en español y *flat* y *apartment* en inglés o como la polisemia, una palabra con varios significados, por ejemplo la palabra *banco* en español que puede denotar un objeto para sentarse o una entidad financiera o la palabra *fine* en inglés que puede denotar una multa o un estado de salud o ánimo de una persona.

Para el proyecto se usó un conjunto de datos sobre el cual se han realizado actividades de interés similar, denominado *OWLS-TC2* [*], que contiene descripciones de servicios web, fue recopilado en su momento ya que no existían conjuntos de datos estandarizados y ha sido usado desde entonces para ejercicios de clasificación, recuperación o descubrimiento

[*]Los perfiles OWL-S de la colección OWLS-TC2, http://projects.semwebcentral.org/projects/owls-tc/, recuperado en Octubre de 2017

de servicios como el realizado por Klusch *et al.* (2006).

El problema se aborda como un reto de clasificación de texto y aprendizaje automático, al ser las descripciones de los Servicios Web de texto corto, la frecuencia de los términos usados para describir los servicios es menor que en otros dominios de clasificación de texto. Esto dificulta que los modelos de aprendizaje automático estimen relaciones entre los términos, por tales razones se han realizado esfuerzos para descubrir relaciones semánticas como los hechos por Rosso *et al.* (2003), Sedding y Kazakov (2004), o Shehata (2009) a partir de diccionarios semánticos comunes como WordNet para el inglés o *HowNet* para el chino, pero al usarlos para dominios específicos se dificulta la tarea de minería de texto tal como lo afirma Jiang *et al.* (2013).

1.2. JUSTIFICACIÓN

Con la elaboración del modelo basado en aprendizaje automático, se pretende mejorar la exactitud de la clasificación de Servicios Web, teniendo en cuenta que estos proponen conceptos de interoperabilidad entre tecnologías de información y comunicación, la realización manual de esta actividad va en contra vía de dichos conceptos y ocasiona incertidumbre debido a la gran cantidad de dominios, servicios web y uso de estos, así mismo se busca disminuir la inexactitud que pueda ser ocasionada por las relaciones semánticas no identificadas debido a las descripciones cortas.

El mejoramiento de la exactitud a través de clasificaciones automáticas, pretende evitar la insatisfacción o desconfianza de los usuarios causada por clasificaciones menos exactas y por ende evitar que sean inducidos a realizar tareas de comprobación de la clasificación de forma manual, en resumen servirá para evitar la realización de acciones humanas de manipulación para que los servicios puedan ser mejor aprovechados en el uso e integración con otros servicios, aplicaciones o plataformas, mediante su manipulación, entendida esta como cualquiera de las actividades de exposición, descubrimiento, recuperación, anotación y uso.

La más exacta clasificación de los Servicios Web, la cual pretendemos sea realizada por el modelo a encontrar usando aprendizaje automático, aportará en la construcción del

concepto de Web Semántica, ya que dicha clasificación es la que determina el dominio al que se hace alusión en el servicio, que de no estar adecuadamente clasificado repercutirá en significados incoherentes en los procesamientos automatizados.

En el mejor de los esfuerzos realizado no se encontraron estadísticas de servicios web pero teniendo en cuenta la penetración y el crecimiento a nivel mundial del internet publicada por *Internet World Stats* con estimaciones a 2018 mostradas en la Tabla 1, donde la penetración alcanza la cifra de 54,4 %, los valores de referencia de la población mundial estimada en 7.634.758.428 personas, el número de usuarios de internet de 4.156.932.140 y el crecimiento de estos entre 2000 y 2018 que alcanza una cifra del 1.052 % (Internet World Stats, 2018) se puede inferir la gran cantidad de servicios web que existen y que existirán donde los usuarios accederán cada vez más a dichos servicios y los proveedores de servicios cada vez más ofrecerán dichas soluciones para cubrir las necesidades de aquellos, se hace primordial proponer clasificaciones que minimicen la participación humana para la reducción del riesgo de una indebida clasificación y también para disminuir el tiempo dedicado a dichas tareas, que luego afecten negativamente las actividades de manipulación de los Servicios Web.

Tabla 1:
Estimación de usuarios de Internet y población mundial a 2018

Regiones	Población (2018 Est.)	% Po-blac.	Usuarios Internet Dic2017	% Penetra.	Crecim. 2000 2018	% Usuarios Internet
Africa	1,287,914,329	16.9 %	453,329,534	35.2 %	9,941 %	10.9 %
Asia	4,207,588,157	55.1 %	2,023,630,194	48.1 %	1,670 %	48.7 %
Europa	827,650,849	10.8 %	704,833,752	85.2 %	570 %	17.0 %
LAyCaribe	652,047,996	8.5 %	437,001,277	67.0 %	2,318 %	10.5 %
MOriente	254,438,981	3.3 %	164,037,259	64.5 %	4,893 %	3.9 %
Nor.Amér	363,844,662	4.8 %	345,660,847	95.0 %	219 %	8.3 %
Ocean.Aus	41,273,454	0.6 %	28,439,277	68.9 %	273 %	0.7 %
TOTAL	7,634,758,428	100.0 %	4,156,932,140	54.4 %	1,052 %	100.0 %

LAyCaribe: Latino América y el Caribe. MOriente: Medio Oriente. Nor.Amér: Norte América. Ocean.Aus: Oceanía y Australia

La necesidad de identificar el dominio al que pertenece un servicio web, se puede per-

cibir si observamos plataformas software como las redes sociales, donde los usuarios realizan acciones tan diversas como una autenticación, la consulta del estado del clima, las noticias o juegos compartidos, sin percibir la interoperabilidad entre las plataformas, estas aumentan su alcance poniendo a disposición herramientas de integración por ejemplo las API como servicios web, para que terceros desarrollen aplicaciones y soluciones, plataformas como *Facebook*, por ejemplo, ofrecen un conjunto de productos y herramientas para que los desarrolladores conecten a escala global la red social, la API principal de Facebook es *Graph API*, esta es la principal forma de ingresar y extraer datos de la plataforma, está basada en HTTP y permite entre otras acciones gestionar –crear, actualizar, consultar, eliminar– los objetos de la plataforma como fotos, comentarios, páginas, historias, anuncios, grupos, datos de los usuarios, aplicaciones, juegos, comentarios, conversaciones, documentos, pagos, entre muchas otras (Facebook, 2018), lo que les ha permitido alcanzar alrededor de 1,65 mil millones de usuario de los cuales acceden alrededor de 989 millones de usuarios diariamente desde aplicaciones móviles (Dogtiev, 2016).

La exposición de servicios web se observa en diferentes dominios como por ejemplo hotelería y turismo, donde plataformas como *Booking.com* pone a disposición API's para los socios proveedores, por ejemplo la *Partner Supply API*, a través de la cual se realizan acciones para la gestión de alojamientos, habitaciones, tarifas, políticas, inventarios, restricciones de disponibilidad, reservas, entre otros (Booking, 2018). Otra plataforma del dominio de hotelería es *Trivago.com* que también pone a disposición sus herramientas de integración (Trivago, 2018). En dominios como el estado del tiempo *The Weather Channel* pone a disposición la *Weather API* mediante la cual se puede obtener información relacionada con el tiempo en imágenes, satelitales y radar, mapas, alertas, calendarios, astronomía, datos históricos, pronósticos, entre otros (Weather Underground, 2018). En general cualquier dominio puede disponer de servicios web como nos lo han dejado ver las poderosas *Amazon*, con su *Amazon Web Services, AWS*, accesibles a través de servicios web como la *Amazon API Gateway*(Amazon, 2018) y *Google* con sus servicios web en la nube, accesibles también a través de las herramientas *Google Developers* (Google Inc., 2018).

1.3. CONTRIBUCIÓN

La contribución de la investigación se evidencia en los siguientes aspectos:

1. El uso del descubrimiento de relaciones semánticas a través de la co-ocurrencia de términos en el vocabulario conformado por todas las descripciones de servicios del conjunto de entrenamiento, para ampliar las descripciones de los servicios web de las nuevas instancias de datos, mediante la inyección de los términos que se consideren similares, sin el uso de tesauros predefinidos.
2. El ejercicio experimental, de evaluación de clasificadores de texto, con el enfoque de inyección de términos donde se obtuvieron los siguientes resultados:
   - La exactitud de la clasificación de las descripciones con inyección de términos mejoró con respecto a la exactitud antes de inyectar términos y con respecto a la línea base.
   - La exactitud de los modelos inducidos con redes neuronales MLP, antes y después de inyectar términos, fue mejor con respecto a los modelos inducidos por máquinas de vectores de apoyo SVM.
3. El modelo de clasificación inducido de mejor exactitud.

En el siguiente capítulo se presentan los objetivos general y específicos.

2. OBJETIVOS

Es esta capítulo se presentan los objetivos de la investigación que plantean una mejora de la exactitud –*accuracy*–

2.1. OBJETIVO GENERAL

Aumentar la exactitud de la clasificación automática de servicios web haciendo frente a los problemas de polisemia dada su breve descripción de servicios en texto libre.

2.2. OBJETIVOS ESPECÍFICOS

1. Evaluar experimentalmente, si la inyección de términos en las descripciones de servicios web, las cuales son breves, aumenta la exactitud de la clasificación automática de dichos servicios, mediante algoritmos de aprendizaje.
2. Afinar la configuración paramétrica de los algoritmos de aprendizaje automático adoptados en el presente estudio, utilizando validación cruzada, como paso previo a la evaluación de sus respectivas exactitudes.
3. Evaluar experimentalmente la exactitud de los dos enfoques adoptados, para compararlos con la línea base del estado del arte en aprendizaje automático.

En el siguiente capítulo está el marco referencial con los conceptos de servicios web, web semántica, clasificación de texto y evaluación de clasificadores, finalizando con la presentación del estado del arte y la notación matemática usada.

3. MARCO REFERENCIAL

En este capítulo se presentan el marco referencial, conformado por el contextual, el conceptual y el estado del arte, en el primero se identifican los conceptos de servicios web, como medio para abordar los problemas de interoperabilidad entre tecnologías, se identifican los marcos de trabajo para asociar meta información a la descripción de los servicios, para su posterior uso en acciones de web semántica, también se presenta el proyecto como un ejercicio de clasificación de texto a partir de las descripciones de los servicios web, con el objetivo de mejorar la exactitud. En el estado del arte, se evidencian trabajos cuyos objetivos fueron aprovechar el descubrimiento de la información semántica a partir de las descripciones de los servicios web para el mejoramiento de la clasificación en el dominio al que pertenece, siendo el de Sharma *et al.* (2016), el que presenta mejores resultados en la exactitud. Finalmente se presenta la notación matemática usada.

3.1. MARCO CONTEXTUAL

El proyecto está enmarcado en el área de conocimiento de la Ingeniería de software, en la sublínea de ciencia de servicios, acotado al desarrollo de sistemas distribuidos débilmente acoplados, en cuyos conceptos se basan los Servicios Web, dichos conceptos proponen que las funcionalidades de los servicios pueden ser accedidas independientemente de las características computacionales del peticionario a través de una interfaz que permita la descripción, definición y descubrimiento del servicio mediante tecnologías estándares (Alonso *et al.*, 2004). Y acotado a los conceptos de web semántica, que proponen adicionar información del significado y estructura del contenido web, para afrontar la ausencia de una organización de la información de la web que permita su procesamiento por máquinas de una forma más eficiente (Berners-Lee *et al.*, 2001).

Más específicamente se enmarca en el área de la ciencia de datos, en un ejercicio de clasificación, a través del uso del aprendizaje automático, que propone que un programa de computadora se considera inmerso en un aprendizaje automático, si a partir de acciones o datos de entrenamiento, realiza una tarea y al medir su desempeño este muestra un mejoramiento (Mitchell, 1997).

3.2. MARCO CONCEPTUAL

Debido al contexto, es necesario identificar que los servicios web se plantean como la solución a los problemas de interoperabilidad que se tenían en las aplicaciones distribuidas y a la necesidad que dichas aplicaciones fueran lo más débilmente acopladas, es decir que no fueran tan dependientes las unas de las otras, que tanto necesita la una de la otra para realizar sus procedimientos. En los inicios de las aplicaciones distribuidas la solución se abordó con el concepto de capa intermedia o *middleware* de aplicaciones y máquinas que sirvieran de puente, pero se desarrollaban propietariamente, lo que no era práctico ya que obligaría a las organizaciones a tener o acceder a tantas capas intermedias -ver figura 1-, como tantas fueran sus necesidades de integración con otras organizaciones con tecnologías diferentes (Alonso *et al.*, 2004). Debido al objetivo

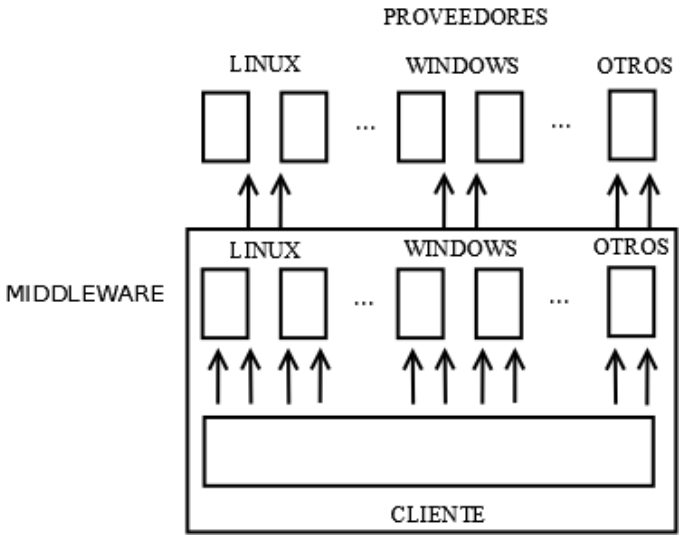

Figura 1: Arquitectura inicial de interoperabilidad. Fuente: Propia

para el cual fueron creados los Servicios Web, de intercambio y procesamiento de información entre diferentes software sin importar la forma o herramientas usadas para su construcción e implementación (Cerami, 2002), fueron entonces concebidos como una tecnología fundamentada y conformada por estándares y protocolos abiertos usados para su estructuración, comunicación, descripción y utilización.

La estructuración se fundamenta en XML, usado para la organización de la información a intercambiar, permite la definición de la gramática del lenguaje usado por los Servicios Web que tendrán sus propias estructuras orientadas al almacenamiento de información, como se observa en la figura 2.

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>
<food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>
   Two of our famous Belgian Waffles with plenty of real maple syrup
   </description>
    <calories>650</calories>
</food>
<food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>
    Light Belgian waffles covered with strawberries and whipped cream
    </description>
    <calories>900</calories>
</food>
<food>
    <name>Homestyle Breakfast</name>
    <price>$6.95</price>
    <description>
    Two eggs, bacon or sausage, toast, and our ever-popular hash browns
    </description>
    <calories>950</calories>
</food>
</breakfast_menu>
```

Figura 2: Documento XML. Fuente: Propia

La transmisión se fundamenta en SOAP, a través de mensajes, en los cuales, la información que se envía o recibe está estructurada usando XML, en otras palabras son los mensajes de peticiones y respuestas entre clientes y servidores, dichos mensajes se transmiten sobre protocolos de comunicación tales como, SMTP, MIME y el más utilizado HTTP (w3.org, 2018c).

La descripción de los servcios web se fundamenta en *Web Service Description Language,*

WSDL, lenguaje para crear el documento XML que posee la descripción del Servicio Web, con características tales como el nombre, datos a recibir, datos a enviar, la estructura de la trama de datos, protocolos que usa, los formatos de los mensajes, entre otros. Este documento debe ser obtenido por el objeto cliente para conocer la forma como va a interactuar con el objeto servidor (w3.org, 2018d).

Y la utilización, es decir que los servicios sean conocidos o ubicados, se fundamenta en conceptos como UDDI, donde se registran los Servicios Web ofrecidos, en formato XML, basados en una categorización, el objetivo es que a partir del registro se redireccione al servicio y sus componentes para su uso (uddi.org, 2018).

Es de la necesidad de registrar un servicio web en una categoría perteneciente a un dominio para el cual fue desarrollado, que se apoya el concepto de Web Semántica, el cual plantea que la información de la web debe estar acompañada de meta información que describa contenidos y sus relaciones, basados en estructuras y clasificaciones comunes, con lenguajes como *Resource Description Framework, RDF* y *Web Ontology Language, OWL*.

RDF, es un marco común o modelo de datos, actúa como lenguaje que puede estar fundamentado en XML para la especificación de metadatos cuyo objetivo es representar objetos de la realidad presentes en los contenidos de la *World Wide Web, www*, como asociaciones de los recursos URI de esta, dichos recursos pueden ser descritos en términos de unas propiedades y estas toman valores, como se observa en la figura 3, este concepto central del marco, se presenta como un triple conformado por un sujeto, un predicado y un objeto y está orientado a representar información o recursos de la Web que necesitan ser procesados e intercambiados por aplicaciones software (w3.org, 2018b). A continuación se observa un ejemplo de un documento RDF.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:contact="http://www.w3.org/2000/10/swap/pim/
                                contact#"
>
  <contact:Person
     rdf:about="http://www.w3.org/People/EM/contact#me"
  >
```

```
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>
</rdf:RDF>
```

Figura 3: Grafo RDF, describiendo a una objeto Persona. Fuente: www.w3.org/TR/rdf-primer

OWL, es un lenguaje de marcas especificado sobre RDF, permite estructurar los datos o términos de un sitio o servicio web mediante una descripción formal del significado de los mismos y sus relaciones, esta descripción será la representación del significado de la información dentro del dominio o área de conocimiento específica del sitio o servicio web, a esto se le conoce como ontología y se especificó para que las aplicaciones software puedan procesar la información de la Web de una forma automatizada (w3.org, 2018a).

RDF y OWL son los conceptos fundamentales que permiten el procesamiento por parte de agentes de software para que estos puedan realizar actividades sofisticadas y automatizadas sobre dicha información para el beneficio de los usuarios (Berners-Lee *et al.*, 2001).

La clasificación del servicio es uno de los primeros pasos en el concepto de Web Semánti-

ca y se va a realizar a partir de la descripción de los servicios web, el proyecto es un ejercicio de clasificación de texto, que autores como Witten *et al.* (2011) lo definen como la necesidad de extraer información relevante a través del descubrimiento de patrones que no están explícitamente evidentes y que dichos patrones se deben encontrar a través de procesos y herramientas que los relacionen (Hearst y A., 1999), las cuales permiten reconocer características del texto de los documentos y representarlas en datos estructurados o semiestructurados como por ejemplo vectores de frecuencia de términos, donde se identifican las veces que aparece un término en un documento (Sanger y Feldman, 2007).

En general se puede considerar el proceso de minería de texto, donde se incluye la clasificación, compuesto por dos etapas, una de preprocesamiento, representación estructurada o semiestructurada de los documentos y la otra de descubrimiento, en la que se pretende encontrar los patrones ocultos, como se observa en la figura 4 (Tan, 1999).

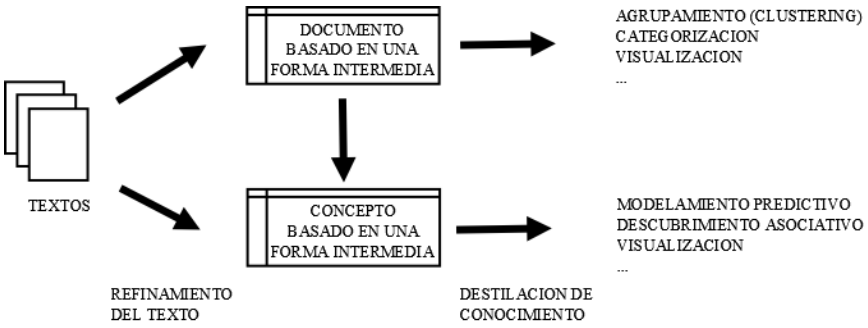

Figura 4: Marco de trabajo de la minería de texto. Fuente: (Tan, 1999)

Como la propuesta es mejorar la exactitud de la clasificación inyectando términos, es necesario tener en cuenta las relaciones semánticas entre los términos para poder integrar términos que aparentemente no tienen que ver unos con otros pero que adquieren un sentido y transmiten un significado según el contexto, por ejemplo las palabras *inteligencia* y *teléfono*, cuando se habla en el contexto de la comunicación móvil, esto se logra mediante el uso de la co-ocurrencia, que plantea que dos términos son co-ocurrentes si son usados en contextos similares, o son co-ocurrentes si los términos con que co-ocurre uno de ellos es similar a los términos con que co-ocurre el otro, o son co-ocurrentes si los dos términos se pueden reemplazar entre ellos en sus contextos y la verosimilitud del contexto no cambia (Budanitsky y Hirst, 2006), entre las relaciones semánticas se

Tabla 2:
Tabla de contingencia para la categoría *Ci*. Fuente:(Sebastiani, 2002)

Categoría *Ci*		Criterio de los expertos	
		SI	NO
Criterio del clasificador	SI	*Verdaderos Positivos (VPi)*	*Falsos Positivos (FPi)*
	NO	*Falsos Negativos (FNi)*	*Verdaderos Negativos (VNi)*

encuentran la hiponimia; el significado de una palabra se encuentra incluido en otro de mayor magnitud, sinonimia; varias palabras tienen el mismo significado, polisemia; una palabra tiene varios significados, homonimia; palabras con la misma escritura o pronunciación con significados distintos (Paradigms Lynne Murphy, 2003).

Para la clasificación se adoptaron técnicas supervisadas del aprendizaje automático, por lo cual abordamos el problema de aprendizaje según la definición que manifiesta que un programa de computador aprende si mejora el desempeño en la realización de una tarea a través de una experiencia de entrenamiento (Mitchell, 1997), según la cual nuestra tarea es la predicción de la clasificación de nuevos servicios web en la categoría del dominio a la que pertenece, la experiencia proviene del conjunto de datos clasificados, datos históricos, a partir del cual se nos inducirá un modelo partiendo del entrenamiento de este con dichos datos y la evaluación del desempeño se realiza con base en los resultados de las instancias que el modelo clasificó correctamente y las que no (ver tabla 2).

La medida de desempeño seleccionada en este proyecto es la exactitud -*accuracy*-, que se plantea como la estimación del porcentaje, a través de la razón de los documentos correctamente clasificados sobre el total de documentos; más detalladamente la exactitud es la razón entre la suma de documentos que son clasificados dentro de una clase a la que realmente pertenecen -*Verdaderos Positivos, VP*- más los documentos que correctamente no fueron clasificados en dicha clase -*Verdaderos Negativos, VN*- con relación al total de documentos, el total incluye los VP, los VN, los que erróneamente no se clasificaron en dicha clase -*Falsos Negativos, FN*- más los que erróneamente fueron clasificados en dicha clase -*Falsos Positivos, FP*- (Sebastiani, 2002).

$$\hat{A}(accuracy) = \frac{VP + VN}{VP + VN + FP + FN}$$

Dentro de las técnicas usadas en ejercicios de clasificación supervisada está SVM, que

en aplicación con el truco del kernel propone aplicar técnicas lineales a problemas no lineales, el objetivo es encontrar un hiperplano en espacios n-dimensionales que permita la separación de los documentos. Por ejemplo, en un plano bidimensional los datos están mezclados de tal forma que una solución lineal para clasificarlos no es posible, con el truco del kernel, de manera implícita se agrega una dimensión adicional en cuyo espacio son separables mediante un plano (Joachims, 1998).Y la técnica de MLP, fundamentada en redes neuronales artificiales muy interconectadas que imitan neuronas humanas, con nodos o unidades de procesamiento y vínculos que representan conexiones sinápticas, para la separación de los documentos en regiones de decisión complejas, es decir, regiones cerradas, convexas o de complejidad arbitraria (Witten *et al.*, 2011), para la clasificación de texto, la capa de neuronas de entrada representa los términos de los documentos, las capas ocultas procesan y la capa de salida representa las categorías de la clasificación (Sanger y Feldman, 2007).

3.3. ESTADO DEL ARTE

Desde hace varios años la clasificación de servicios web con base semántica a partir de sus descripciones, usando conceptos de aprendizaje automático es un área de interés, se puede observar en enfoques como el de Heß *et al.* (2004) en el cual se presentó una herramienta denominada *Automated Semantic Service Annotation with Machine Learning, ASSAM*, cuyo objetivo es generar información semántica para los Servicios Web de una forma semiautomática de manera similar a como los ambientes de desarrollo generan información descriptiva de los servicios, para esto propusieron dos componentes, el anotador WSDL y *OATS* el algoritmo de agregación de datos, el primer componente usa algoritmos de aprendizaje automático para hacer sugerencias de anotaciones semánticas a los elementos del WSDL a partir de una clasificación ontológica, partiendo de la idea que existe una relación semántica entre el Servicio, sus operaciones y variables, para la clasificación de las anotaciones semánticas se usa un algoritmo de clasificación iterativa, donde se caracterizan los servicios con un conjunto de variables intrínsecas que denotan los elementos del servicios, tales como nombre y descripción textual y las extrínsecas que denotan las relaciones entre dichos elementos, es decir se usa como variable la clase semántica que denota la relación, al inicio del aprendizaje las características extrínsecas

son desconocidas y son establecidas a partir de las características intrínsecas, se usan dos clasificadores uno para las intrínsecas y otro para las extrínsecas lo que mejora la efectividad y la predicción se hace por un sistema de votación, se usa incluso un tercer clasificador para incorporar las características extrínsecas a partir de las variables de las operaciones que son intrínsecas, haciendo más robusta la predicción ya que se combinan resultados de los clasificadores de las intrínsecas en cada iteración.

El segundo algoritmo de coincidencia de esquemas se usa para agregar las salidas de datos heterogéneos en esquemas comunes, por ejemplo, dos servicios web que arrojan la siguiente información sobre el estado del tiempo:

```
<weather>
<hi>87</hi>
<lo>56</lo>
<gusts>NE, 11 mph</gusts>
</weather>


<fcast>
<tmax>87</tmax>
<tmin>57</tmin>
<wndspd>10 mph (N)</wndspd>
</fcast>
```

Como principal algoritmo de aprendizaje usaron *Naïve Bayes*, fundamentado en probabilidad de posibles resultados. Se obtuvieron valores bajos en la exactitud, pero los autores planteaban que para la época los resultados representaban mejoras hasta de una tercera parte con respecto a otros enfoques.

En la misma época que el enfoque anterior se presentó *METEOR-S Web Service Annotation Framework, MWSAF* por Patil *et al.* (2004), consistió en un marco de trabajo para anotar de forma semiautomática las descripciones de servicios web con ontologías, mediante algoritmos para unir y anotar archivos WSDL con ontologías relevantes, usando un modelo común para representar las descripciones WSDL de los servicios y las ontologías, luego comparar dichas representaciones y si existían coincidencias se anota-

ba la descripción con la ontología del dominio al que pertenecía la descripción. No se presenta como un ejercicio de aprendizaje automático por lo tanto no se presentaron resultados de evaluación del enfoque en términos de exactitud. En la misma dirección Zhang *et al.* (2005) presentaron un marco de trabajo similar pero diferenciándose en el uso de OWL para la representación de las descripciones de los servicios y la ontología a diferencia de Patil *et al.* (2004) que introdujeron un modelo de representación semántico particular, el enfoque no es presentado como un ejercicio de aprendizaje autómatico pero incluyeron resultados de evaluación mediante la precisión donde alcanzaron valores del 96.4 %. Los dos anteriores enfoques usaron un conjunto de servicios pertenecientes a dos dominio unicamente, geografía-estado-del-tiempo y viajes-estado-del-tiempo respectivamente. Estos enfoques sirvieron para que Oldham *et al.* (2005) presentara una mejora de MWSAF a partir de los conceptos de aprendizaje automático, lo planteado específicamente consistió en reemplazar la técnica de mapear las descripciones con las ontologías, ante la dificultad de hacer coincidir las descripciones en WSDL y las ontologías principalmente en OWL, por una clasificación realizada con el algoritmo de Naïve Bayes, la cual resultó siendo más rápida que la técnica inicial, evaluando el modelo en términos de exactitud donde obtuvieron valores de 68 % en algunos casos.

Más recientemente, Yang y Zhou (2014) propusieron la exploración de clasificadores de Servicios Web para sentar bases que sirvan en la identificación de dominios para los mismos y su descubrimiento, conformación y anotación semántica, para esto realizan actividades de aprendizaje automático sobre los documentos de descripción de los servicios del conjunto de datos OWLS-TC4. En detalle se clasifican los servicios con base a funcionalidades similares, en categorías basadas en taxonomías. Para la investigación los autores tomaron elementos estáticos de la descripción, nombre de los servicios y nombre de las operaciones de los servicios, ya que representan significados funcionales, posteriormente prepararon los términos realizando acciones como la división de términos, separación de verbos y sustantivos, eliminación de palabras de parada, preposiciones, artículos, etc., derivación, poner las palabras en su raíz o en infinitivo, eliminación de términos que no tienen significados relevantes. Finalmente se hacen vectores de características a partir del nombre del servicio, las operaciones, las entradas y las salidas, para aplicarles varios algoritmos de aprendizaje automático entre los cuales se encuentran SVM, Naïve Bayes, *C4.5*, árboles de decisión y *Back Propagation Neural Network, BPNN*, redes neuronales con retro propagación, en sus conclusiones muestran que al tomar varios atributos de la descripción se obtiene mejoras en la exactitud y que

el mejor resultado de los clasificadores fue C4.5, los autores no identifican el método de validación del modelo, ni los parámetros de los algoritmos.

También se puede observar que la clasificación de los servicios web es de tal interés que autores como Nisa y Qamar (2015), exploran el tema con técnicas novedosas para el mismo, en su trabajo proponen una clasificación basada en minería mediante el algoritmo de *Máxima Entropía*, que busca la generación de modelos más uniformes por la inclusión del concepto de restricción, el cual ayuda a definir las características de la información, sin suponer lo desconocido. Los autores presentan el enfoque como un ejercicio de aprendizaje automático y realizan una conformación y categorización del conjunto de datos de forma particular. Otros autores con enfoques novedosos son Liu *et al.* (2016), estos proponen una clasificación de servicios basado en conceptos de aprendizaje activo mediante el uso de las técnicas *Latent Dirichlet Allocation, LDA* y SVM, presentando un enfoque diferente al de la representación de los documentos mediante vectores de términos, ya que la base del enfoque propuesto es la representación del contenido de los documentos, en este caso de las descripciones de los servicios web, como un conjunto o grupo de tópicos o temas alrededor de los cuales se identifican las palabras usadas en dichos tópicos. Los autores utilizan el conjunto de datos *WS-DREAM* desarrollado para resultados de calidad del servicio -*QoS*-.

Finalmente en relación directa con el enfoque que se pretende encontramos, que (Sharma *et al.*, 2016) proponen un enfoque semántico para clasificación de Servicios Web usando aprendizaje automático y medidas de relaciones semánticas, para estos autores el problema es la complejidad y la propensión al error que se presenta en la clasificación al momento de registrar los Servicios Web en los repositorios que permiten su publicación, búsqueda y hallazgo, ocasionado por ser una tarea realizada manualmente por varias personas, por presentarse un gran número de categorías de las taxonomías que permiten la clasificación, la existencia de varios repositorios de registro y la falta de conocimiento de las taxonomías, sus categorías, el dominio y perfil de las aplicaciones. Así mismo para los usuarios que pretenden reutilizar los Servicios Web que ya han sido publicados, se presenta el problema de escoger manualmente la categoría adecuada con el agravante que el servicio registrado pudo no haber quedado en la categoría más adecuada por lo tanto las consultas de los Servicios Web registrados arrojaran resultados no deseados. Las problemáticas anteriores pueden ser reducidas mediante el uso de clasificaciones automáticas que minimicen la intervención humana.

Como solución los autores plantean un enfoque de clasificación automatizada híbrida que integra la semántica del perfil de los Servicios Web con su información estadística y generar así vectores semánticos para los servicios, estos serán usados para las técnicas de clasificación del aprendizaje automático. Para la transformación del perfil y la información estadística de los documentos en vectores semánticos, los autores se apoyaron en un enfoque propuesto por Tsatsaronis *et al.* (2010) en el cual las técnicas para detección de las relaciones semánticas entre palabras fueron extendidas a textos, el enfoque incluye la documentación de los servicios hecha en lenguaje natural, para la clasificación usan procesamiento de lenguaje natural, aprendizaje automático, razonamiento enmarcado, medición de la relación semántica y minería de texto, el otro aspecto propio del enfoque es la inclusión de la desambiguación de los términos.

El enfoque consiste en cuatro pasos, creación de los vectores de los servicios, cálculo de la matriz de relaciones semánticas, enriquecimiento de los vectores con la información semántica y la clasificación de los servicios usando SVM y kNN. Obteniendo resultados de exactitud de 97.22 %, en el mejor de nuestro conocimiento el más alto identificado en investigaciones con objetivos similares al nuestro.

Esta propuesta al estar fundamentada en el trabajo de Tsatsaronis *et al.* (2010) usa el tesauro WordNet del idioma inglés para el reconocimiento de relaciones semánticas, el cual con frecuencia no contiene relaciones para dominios específicos.

Como vemos en la tabla 3, las técnicas usadas para la clasificación son variadas, pero se observa por ejemplo que SVM es una de las más usadas, por ende adoptamos este enfoque como referente para comparar nuestros resultados y el estado del arte. El resultado obtenido con SVM antes de expandir las descripciones es seleccionado como línea base, ya que se considera uno de los algoritmos de clasificación mejor fundamentado teóricamente y actualmente de los más efectivos en la práctica del aprendizaje automático (Mohri *et al.*, 2018). También proponemos usar MLP en nuestro estudio, debido a su amplia adopción en la tendencia actual de redes neuronales de aprendizaje profundo -*Deep neural network*- y en el contexto actual de volúmenes de datos a gran escala -*Big Data*-

Tabla 3:
Uso de algoritmos de minería del estado del arte. Fuente:Propia

Autores	Algoritmos
Heß *et al.* (2004)	Naïve Bayes
Oldham *et al.* (2005)	Naïve Bayes
Yang y Zhou (2014)	SVM, Naïve Bayes,C4.5, BPNN
Nisa y Qamar (2015)	Maxima Entropia
Liu *et al.* (2016)	LDA, SVM
Sharma *et al.* (2016)	SVM, KNN

3.4. NOTACIÓN MATEMÁTICA

La notación matemática usada en el presente documento es la siguiente. Las letras en minúscula n representan un escalar, el símbolo $\in$ indica pertenencia a conjunto, las letras mayúsculas en negrita $\mathbf{Y}$ indican una matriz, los vectores se representan con letras minúsculas en negrita $\mathbf{x}$, se usa la notación $\mathbb{R}^{m \times n}$ para indicar un espacio vectorial de números reales de m filas por n columnas, el superíndice T denotará un arreglo traspuesto, por ejemplo $\mathbf{Y}^T$ es una matriz traspuesta, los subíndices i y j representan un iésimo o jésimo elemento, por ejemplo $\mathbf{x}_i$ es el iésimo elemento de un vector, la expresión $\|\mathbf{x}\|$ indica la norma de un vector.

El contenido del siguiente capítulo es la metodología, conformada por el tipo de investigación, el método de validación y evaluación, la descripción del conjunto de datos y las suposiciones y limitaciones. Con estos aspectos se expone, la naturaleza experimental de los ejercicios de aprendizaje automático.

4. METODOLOGÍA

Este capítulo contiene, el tipo de investigación, donde se argumenta el ejercicio de aprendizaje automático como experimental, al ser abordado como un conjunto de factores y entradas que influyen en las salidas y mediante experimentos se pueden determinar las relaciones de causa y efecto entre dichos elementos. Contiene, la metodología de validación cruzada, para el mejor entrenamiento, la descripción del conjunto de datos y las suposiciones y limitaciones que enmarcan el ejercicio.

4.1. TIPO DE INVESTIGACIÓN

Para entender un proceso o sistema, este se puede plantear como una combinación de elementos de entrada, donde factores controlables e incontrolables, transforman las entradas en salidas o respuestas, como se muestra en la figura 5, entonces si deseamos comprender el proceso es necesario cambiar adrede las variables y factores de entrada, observar los cambios ocurridos en las variables de respuesta y tratar de determinar las relaciones de causa y efecto entre los cambios, esto se conoce como experimentar (Montgomery, 2017). En aprendizaje automático se realiza investigación experimental, debido a que el objetivo es obtener un algoritmo de software o modelo con patrones, generalidades o relaciones, a partir de observaciones o datos de un proceso o sistema, que pueda ayudar a entender el proceso o predecir situaciones, en especial para realidades donde el proceso no puede ser explicado por una serie de pasos o cálculos como por ejemplo el reconocimiento del rostro de una persona.

El modelo es formalizado mediante conceptos matemáticos y estadísticos denominados algoritmos de aprendizaje los cuales tienen ciertos parámetros que, junto con otros aspectos como la selección del algoritmo, el conjunto de datos usado, la representación de este como entrada, entre otros, se consideran factores controlables, también están

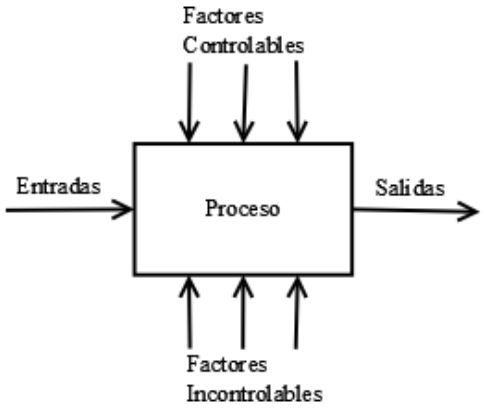

Figura 5: Modelo general de un proceso o sistema. Fuente: (Montgomery, 2017)

presentes factores incontrolables como el ruido en el conjunto de datos, la conformación de los subconjuntos de datos cuando se usa remuestreo, la aleatoriedad en el proceso de optimización, entre otros, dichas entradas y factores son ajustados en una serie de ejecuciones de los algoritmos sobre el conjunto de datos para encontrar la mejor configuración y así obtener la mejor respuesta o determinar el efecto de las configuraciones sobre la respuesta (Alpaydin, 2010).

La clasificación de documentos de texto, como ejercicio de aprendizaje automático, presentan metodológicamente tres grandes etapas, el indexamiento, el aprendizaje y la evaluación, como se observa en la figura 6. En el indexamiento la intención es obtener una representación del contenido del documento que pueda ser tratado por los algoritmos de clasificación. El aprendizaje, es el momento en que propiamente se induce el modelo a partir de documentos preclasificados, donde se identifican conjuntos para entrenamiento, validación y pruebas. Finalmente, la evaluación donde principalmente se mide el desempeño del modelo por su eficacia.

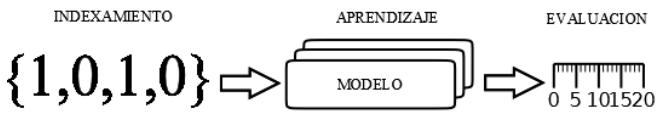

Figura 6: Método general del aprendizaje automático. Fuente: Propia

Como metodología experimental, usamos como guía, los siguientes pasos o etapas propuestos por Alpaydin (2010):

1. Seleccionamos como objetivo del ejercicio de aprendizaje, el error esperado y la medición del error de clasificación como variable de respuesta, en este proyecto particularmente se evalúa la exactitud para medir el desempeño de los modelos. La exactitud es complementaria al error, ya que este se puede expresar como $error = 1 - accuracy$, entre mayor es la exactitud menor es el error.
2. Establecemos los factores controlables, a los cuales, deseamos encontrar la mejor configuración que optimice la respuesta, en este proyecto, entre los factores elegidos tenemos.
   - La representación de la entrada con inyección de términos.
   - El umbral para determinar la similitud entre términos.
   - El número de factores latentes que expliquen la relación entre términos.
   - Los parámetros de regularización.
   - La tasa de aprendizaje.
   - La cantidad de unidades ocultas de MLP.
   - Los parámetros asociados a cada función kernel para el aprendizaje con SVM.
3. En cuanto a estrategia de experimentación, se eligió el diseño factorial, para que el ajuste de todos los factores evidencie las dependencias entre los factores y la influencia de los factores incontrolables, como por ejemplo la aleatoriedad del remuestreo o de la factorización de matrices. Esta estrategia determina que el conjunto de datos lo dividamos en tres partes, donde dos terceras partes son para entrenamiento y validación y la tercera para prueba o evaluación. Para el entrenamiento y validación realizamos validación cruzada que permite la generación de múltiples conjuntos de entrenamiento y validación, usados para las diferentes ejecuciones y por ende se inducen múltiples modelos intermedios que permiten tener una muestra de errores de validación que nos permita identificar el mejor modelo.
4. Para la realización del experimento se usarán los algoritmos de aprendizaje de la herramienta R
5. Finalmente el análisis y los resultados se sustentarán con base en la medición de la exactitud.

4.2. MÉTODO DE VALIDACIÓN Y EVALUACIÓN

La clave de la inducción del mejor modelo es el descubrimiento de las relaciones entre las entradas y como afectan la respuesta, por lo tanto para la etapa de entrenamiento o aprendizaje se plantea la validación cruzada *k-fold Cross Validation* que aleatoriamente divide el conjunto de datos histórico en k subgrupos, utilizando k-1 subgrupos como datos de entrenamiento y el restante como datos de validación, como se observa en la figura 7, luego se repite el proceso k veces, cambiando los subgrupos de prueba y validación.

$$\begin{aligned} V_1 = X_1 \quad & T_1 = X_2 \cup X_3 \cup \cdots \cup X_k \\ V_2 = X_2 \quad & T_2 = X_1 \cup X_3 \cup \cdots \cup X_k \\ & \vdots \\ V_k = X_k \quad & T_k = X_1 \cup X_2 \cup \cdots \cup X_{k-1} \end{aligned}$$

La validación cruzada se usa para mitigar el riesgo que la correlación entre las variables del conjunto de datos histórico quede sesgada por tomar solo una parte del conjunto como datos de entrenamiento(Blum *et al.*, 1999). También, mantiene los conjuntos de entrenamiento y validación lo suficientemente grandes para que las estimaciones de error sean sólidas pero la superposición de elementos sea lo más pequeña posible, igualmente permite mantener la representación de las clases en los subconjuntos de entrenamiento y validación con proporciones similares a las que tenian las clases en el conjunto antes de dividirlo (Alpaydin, 2010).

Como lo mencionamos anteriormente, la medida de evaluación tomada en nuestro proyecto es la exactitud, la cual es la estimación del porcentaje de la suma de los documentos que son clasificados dentro de una clase a la que realmente pertenecen –Verdaderos Positivos, Vp– más los documentos que correctamente no fueron clasificados en dicha clase –Verdaderos Negativos, Vn– con relación al total de estos, sumados con los que erróneamente no se clasificaron en dicha clase –Falsos Negativos, Fn– más los que erróneamente fueron clasificados en dicha clase –Falsos Positivos, Fp–, es decir, $accuracy = (Vp + Vn)/(Vp + Vn + Fp + Fn)$ (Sebastiani, 2002).

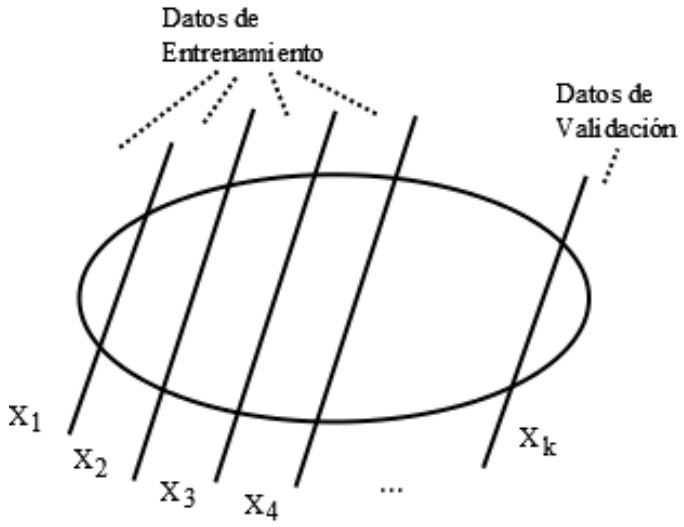

Figura 7: k-fold Cross Validation. Fuente: propia

4.3. CONJUNTO DE DATOS

Como fue descrito en la sección Planteamiento del Problema, se usó el conjunto de datos denominado OWLS-TC2, que contiene descripciones de servicios web, en el cual, después de eliminar todas las palabras de detención –*stop words*–, se tiene:

- un mínimo de 3 términos por descripción,
- 8 términos por descripción en el primer cuartil,
- una mediana de 9 términos por descripción,
- 11 términos por descripción en el tercer cuartil
- en promedio 9.99 términos por descripción,
- con una desviación estándar de 4.07 términos, y
- Máximo 50 términos por descripción.

En la figura 8 se puede observar, que la mayoría de las descripciones de los servicios web, están compuestas aproximadamente por 10 términos. Solo unos pocos servicios web se describen con más de 30 términos y estos son valores atípicos –*outliers*–. El conjunto de datos está compuesto por 576 especificaciones de servicios en OWL-S 1.1 de siete diferentes dominios entre los cuales se tiene educación, atención médica, comida, viajes, comunicación, economía y armamento, se recuerda que fue recopilado ya que no existían conjunto de datos estandarizados y ha sido usado desde entonces para ejercicios de

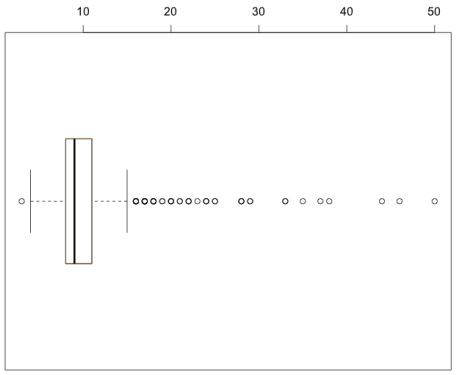

Figura 8: Cantidad de términos usados en la descripción de Servicios Web. Fuente: propia

clasificación, recuperación o descubrimiento como el realizado por Klusch *et al.* (2006). Además fue el mismo conjunto de datos usado por (Sharma *et al.*, 2016).

4.4. SUPOSICIONES Y LIMITACIONES

Para la realización del proyecto se realizaron las siguientes suposiciones:

- Los términos se extraen de etiquetas presentes en los perfiles OWL-S de los servicios web.
- Las expresiones textuales en dichas etiquetas se encuentran escritas bajo la convención de *camel case* [*] o pueden estar separadas por espacios o caracteres de subrayado.
- Las descripciones son textos cortos en comparación con otros documentos como libros o páginas web.
- No se van a inferir términos desde abreviaturas presentes en las descripciones.
- Aunque el modelo posteriormente puede ser adaptable se asume que el texto de las descripciones va a estar en un solo idioma, el cual es el inglés, debido a que el conjunto de datos fue conformado con descripciones en dicho idioma.

[*] Costumbre en el ámbito informático de escribir un nombre o frase, compuesto por varios términos sin espacio entre ellos y con la primera letra de cada término en mayúscula y el resto en minúsculas, por ejemplo, *CamelCase*

- No se van a recopilar ni construir servicios, como se manifestó anteriormente, se utiliza un conjunto de datos recopilado, estandarizado y usado en otras investigaciones.
- No se evaluará rendimiento ni escalabilidad solo exactitud.

En el capitulo siguiente se describen los conceptos principales del proyecto, el algoritmo para la expansión de las descripciones de los servicios y los algoritmos SVM y MLP de aprendizaje automático que se usaron para la clasificación.

5. ALGORITMOS DE CLASIFICACIÓN Y EXPANSIÓN DE DESCRIPCIONES DE SERVICIOS

En este capítulo se presentan los elementos principales de la investigación, la adopción de dos modelos de aprendizaje automático para clasificar los servicios, la máquina de vectores de apoyo (SVM*) propuesto por Cortes y Vapnik (1995) y perceptrón multicapa (MLP**) con propagación hacia atrás –*backpropagation*– (Werbos, 1974; Rumelhart *et al.*, 1988). Y se presenta el algoritmo de expansión de las descripciones de los servicios web, usando la co-ocurrencia y el descubrimiento de sus factores latentes ocultos, para determinar la similitud entre términos.

Dividimos el conjunto de datos en tres subconjuntos después de expandir cada descripción del servicio. Por lo tanto, utilizamos dos tercios del conjunto de datos para el ajuste de parámetros a través de validación cruzada de 10 veces y un tercio para la prueba.

5.1. MÁQUINA DE VECTORES DE APOYO

El algoritmo de máquina de vectores de apoyo –*Support Vector Machine, SVM*–, plantea que, observaciones de una realidad en un conjunto de datos, que no puedan ser separadas por una función lineal, pueden ser elevadas en dimensionalidad usando el truco del kernel y en este espacio de mayor dimensión se puede encontrar una superficie lineal o hiperplano óptimo que permita la clasificación de las observaciones transformadas (Cortes y Vapnik, 1995).

El trasfondo metodológico del algoritmo indica los siguientes pasos:

*SVM es la abreviatura de Support Vector Machine, también conocido como Support Vector Networks

**MLP es la abreviatura de Multilayer Perceptron

- Un conjunto de datos D, no separable linealmente en un espacio vectorial $\mathbb{R}^n$, en el cuerpo de los reales, es mapeado a otro espacio de mayor dimensionalidad $\mathbb{R}^m$ $m > n$ a través de una transformación ϕ obteniéndose un D' linealmente separable(Eric Kim, 2019), como se muestra en la figura 9 y figura 10

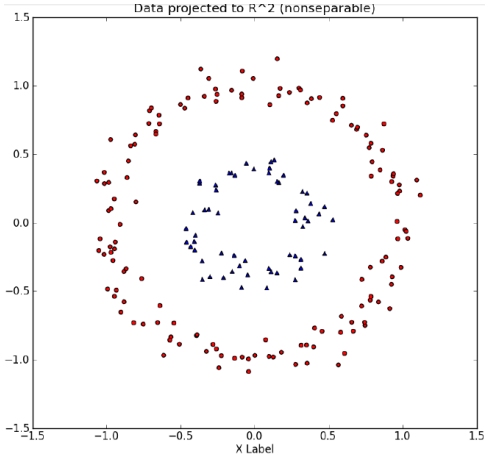

Figura 9: Datos no separables linealmente. Fuente: (Eric Kim, 2019)

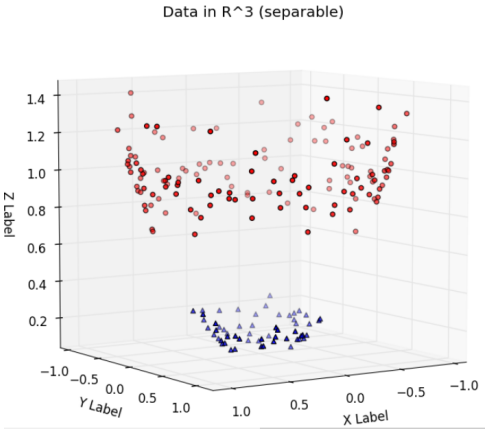

Figura 10: Datos separables linealmente en una dimensión superior. Fuente: (Eric Kim, 2019)

- Con D' se puede entrenar un modelo usando los planteamientos matemáticos lineales de los algoritmos de SVM y encontrar un límite de decisión que es un hiperplano que separe las clases, como se muestra en la figura 11.
- El conjunto de pruebas que no ha sido visto por el modelo durante el entrenamiento, también será transformado, pero usando el truco del kernel.

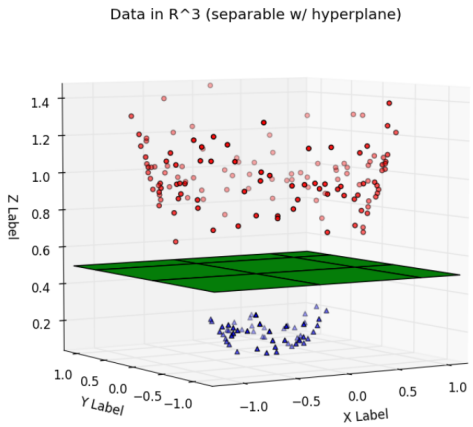

Figura 11: Hiperplano lineal de separación. Fuente: (Eric Kim, 2019)

- Al proyectar el límite de decisión o hiperplano encontrado en $\mathbb{R}^m$ al espacio original, $\mathbb{R}^n$ se producirá un límite de decisión no lineal, como se muestra en la figura 12.

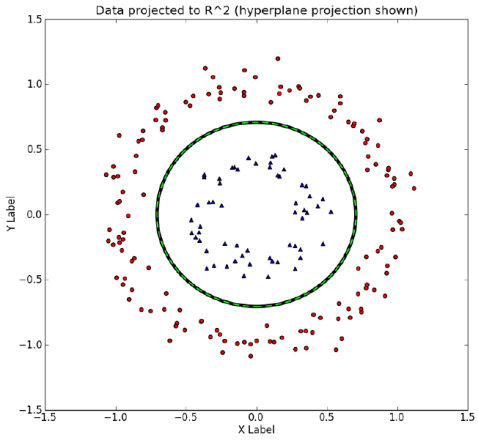

Figura 12: Proyección no lineal de separación. Fuente: (Eric Kim, 2019)

Para lograr lo planteado por el enfoque de SVM, este se fundamenta principalmente en los siguientes conceptos:

- La selección del hiperplano óptimo, que permita la separación de las observaciones en las clases a las que pertenecen, usando pocos datos del conjunto de entrenamiento denominados vectores de soporte, el trasfondo de estos, es que hay observaciones que siguen el mismo patrón del hiperplano, por ende con estos se puede realizar la búsqueda y selección del mayor margen posible entre el hiperplano y las observaciones de las clases más cercanos al hiperplano, como se muestra en la figura 13.

El mejor hiperplano, será entonces el que permita el mayor margen. Entonces si a partir de pocos datos –vectores de soporte– se puede construir el hiperplano óptimo, la capacidad de generalización del modelo será alta, incluso para espacios de altas dimensiones.

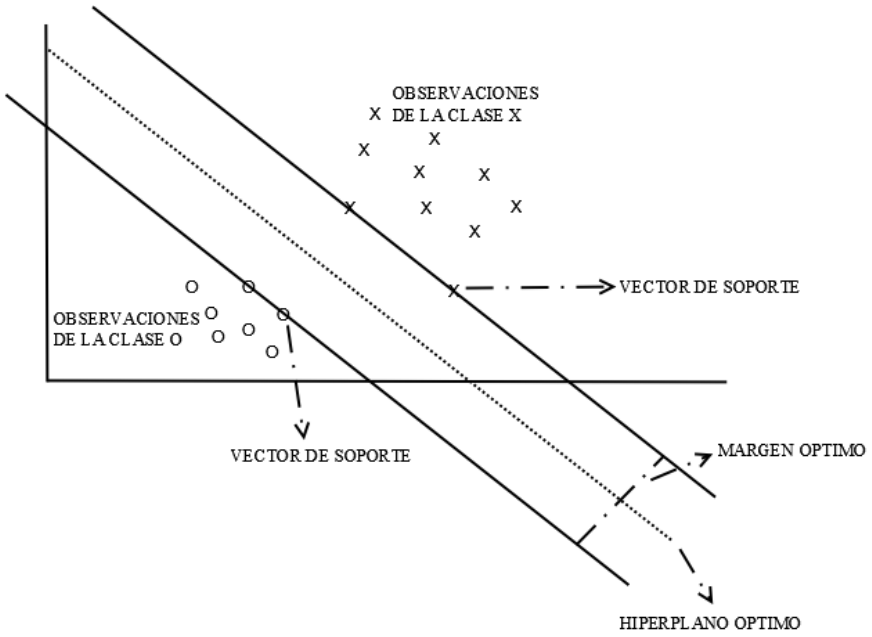

Figura 13: Hiperplano y margen óptimo. Fuente: Cortes y Vapnik (1995)

- El uso de las funciones kernel, que permite obtener el valor del producto punto de las instancias del conjunto de datos –vectores– de dimensión original sin transformar al espacio de nivel superior, posteriormente transformar el resultado del producto punto al espacio de dimensión superior, lo que permite el uso eficiente de los recursos computacionales y no incrementa de forma desmesurada el tiempo de entrenamiento.
- El margen suave, que consiste en la identificación del mayor margen posible entre los vectores de las clases, pero permitiendo algunos errores en la clasificación, este concepto es un parámetro de regularización. Para valores pequeños, el margen aumenta, los errores aumentan, podría sub entrenarse el clasificador *underfitting* y para valores grandes, el margen disminuye, los errores disminuyen, podría sobre entrenarse el clasificador *overfitting*.

5.2. REDES NEURONALES ARTIFICIALES

Los algoritmos de aprendizaje denominados Redes Neuronales Artificiales *RNA*, basan su concepto en el cerebro humano, en la forma como está compuesto por una gran cantidad de neuronas, consideradas cada una, como unidades de procesamiento conectadas unas con otras, pero operando de manera paralela, en esto se cree que radica su gran poder para realizar y aprender operaciones de diversa índole.

Para las RNA, el Perceptrón, mostrado en la figura 14, representa el elemento básico de procesamiento de forma similar a la neurona para el cerebro y tiene los siguientes elementos:

- Entradas, datos de entrenamiento a partir de los cuales se induce el modelo. $x_j \in \mathbb{R}$, $j = 1, ..., d$
- Pesos, que se aplican a las entradas, son los que el aprendizaje debe descubrir. $w_j \in \mathbb{R}$
- La función que representa el hiperplano de separación de las clases y la función de activación que permite expresar la salida como una respuesta binaria 1 o 0.
- Salidas, que pueden ir a otros perceptrones

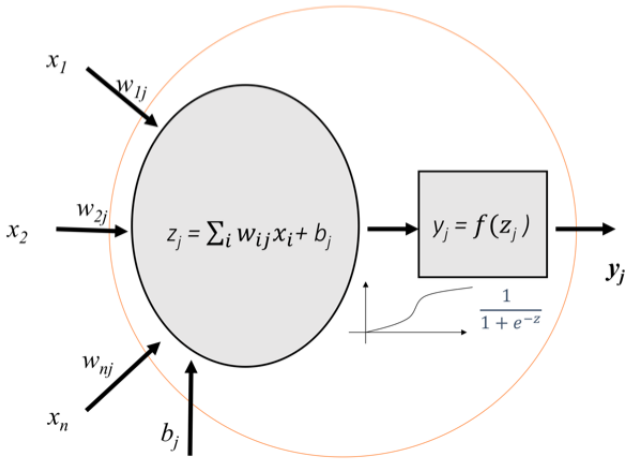

Figura 14: Perceptrón. Fuente: (Torres, Jordi, 2018)

Para el proyecto hemos usado una red neuronal perceptrón multicapa MLP* con pro-

*MLP es la abreviatura de Multilayer Perceptron

pagación hacia atrás *backpropagation*, teniendo en cuenta que un perceptrón solo puede aproximarse a funciones lineales por lo tanto no puede realizar clasificación o regresión para observaciones no separables linealmente, pero aprovechando que un modelo lineal puede ser usado para una aproximación polinomial, MLP propone una red neuronal por capas, las cuales son, una capa de entrada, unas capas ocultas y una capa de salida, como se observa en la figura 15: Donde las capas están compuestas por varios percep-

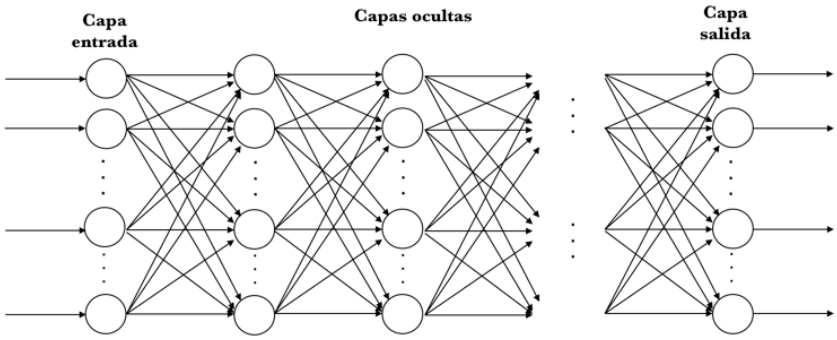

Figura 15: Multilayer Perceptrón. Fuente: (Torres, Jordi, 2018)

trones, en las capas ocultas se implementa una transformación lineal desde un espacio d-dimensional a un espacio k-dimensional superior o inferior $k > d$ o $k < d$ y en la capa de salida se implementa la clasificación o regresión en la nueva dimensionalidad mediante un hiperplano lineal. La red es entrenada teniendo en cuenta la pérdida –error– entre la predicción y el valor real, regresando a ajustar los pesos de las capas ocultas proporcionalmente según la contribución de sus neuronas a la salida. Todo lo anterior permite implementar clasificación o regresión no lineal(Alpaydin, 2010).

El trasfondo metodológico para MLP con retropropagación, indica los siguientes pasos, que se pueden observar de forma general en la figura 16:

- Los datos de entrenamiento deben cruzar toda la red neuronal, produciendo una predicción, esto se conoce como propagación hacia adelante *Forward Propagation*
- Calcular la pérdida –error– con base en una función de pérdida, mediante la comparación entre la predicción y el valor real.
- Propagar la pérdida a todas las neuronas de la capa oculta que contribuyan a la salida, pero en proporción relativa a su contribución *Back Propagation*
- Ajustar los pesos de las interconexiones entre las neuronas.

Con lo anterior se busca minimizar el error y el método adoptado en esta investigación

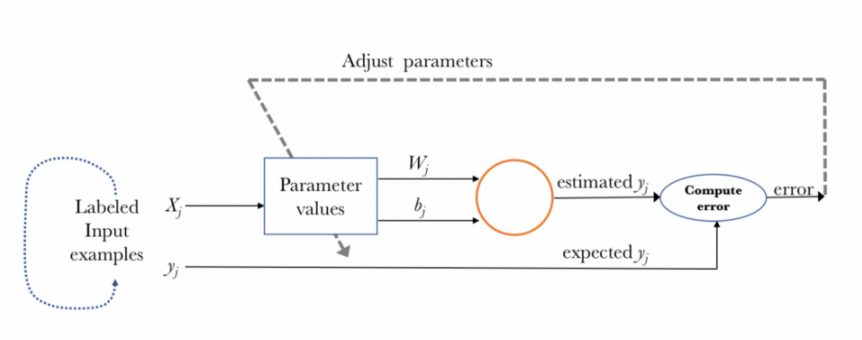

Figura 16: Aprendizaje MLP-B. Fuente: (Torres, Jordi, 2018)

para lograrlo es el gradiente descendente, el cual mediante el cálculo del gradiente –primera derivada– de la función de pérdida y el valor que le corresponde de dicho gradiente a cada neurona, los valores de los parámetros son ajustados en dirección opuesta a la indicada por el gradiente, teniendo en cuenta que la opuesta es la dirección a la que tiende a reducirse la función de pérdida, este comportamiento se puede observar en la figura 17.

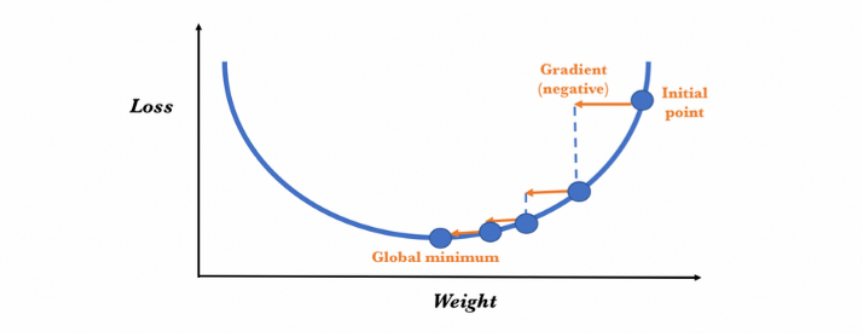

Figura 17: Gradiente descendente. Fuente: (Torres, Jordi, 2018)

5.3. EXPANSIÓN DE DESCRIPCIONES A TRAVÉS DEL TESAURO DE CO-OCURRENCIA

En primer lugar, llevamos a cabo el paso de preproceso sobre el corpus o conjunto de datos, extraemos cada término de las etiquetas:

`<profile:serviceName>` y `<profile:textDescription>`.

Se recuerda que suponemos que las etiquetas se han codificado de acuerdo con la convención *camel case*, es decir, la práctica de escribir un identificador compuesto por varios términos, donde cada uno comienza con una letra mayúscula, por ejemplo, `NonNegativeMatrixFactorization`. Además, suponemos que cada término del texto en la primera etiqueta puede estar separado por espacios o caracteres de subrayado.

Después de eso, eliminamos tanto las puntuaciones como los símbolos, cambiamos todos los términos a minúsculas, aplicamos el origen de cada término *stemming* y eliminamos todas las palabras de parada *stop words*.

Finalizado el preprocesamiento, adoptamos la frecuencia de términos –*Term Frequency*– y la frecuencia inversa de documento –*Inverse Document Frequency*– *TF-IDF* Salton *et al.* (1975) para calcular la matriz de términos de documentos $\mathbf{Y} \in \mathbb{R}^{m \times n}$, donde m y n son el número de documentos y terminos, respectivamente.

Con la matriz Y, generamos el tesauro mediante el cálculo de la Matriz de Similitud de Términos –*Terms Similarity Matrix*– $\mathbf{C} = \mathbf{Y}^T\mathbf{Y}$, donde $\mathbf{C} \in \mathbb{R}^{n \times n}$ y cada componente C_{ij} representa el grado de similitud entre los términos t_i and t_j. La similitud entre dos términos sintácticamente diferentes depende del grado en que ambos términos aparecen juntos o coinciden en varias descripciones del servicio web. En este modelo, nuestro objetivo es estimar los factores latentes ocultos en estas co-ocurrencias de términos. A continuación, calculamos los factores latentes de cada fila vector de esta matriz, factorizándola, en la siguiente sección describiremos el modelo de factorización matricial a continuación, para obtener $\mathbf{W} \in \mathbb{R}^{r \times n}$ y $\mathbf{X} \in \mathbb{R}^{n \times r}$ tal que $\mathbf{C} = \mathbf{XW}$, y r es el número de factores latentes que explican la relación entre los términos. La cantidad de factores latentes r debe ser menor que el número de términos en el vocabulario n. Sea D un conjunto de términos utilizados en la descripción del servicio, y sea V un conjunto de términos utilizados para describir todos los servicios, también conocido como un vocabulario. Agregamos cada término $t_i \in V - D$ del tesauro en la descripción D si $sim(\mathbf{x}_i, \mathbf{x}_j) > \theta$ –ver Ecuación 1–, donde $t_j \in D$, y ambos $\mathbf{x}_i$ y $\mathbf{x}_j$ corresponden al i^{th} y j^{th} filas de la matriz $\mathbf{X}$, respectivamente. El parámetro θ es el umbral. Esto indica el valor mínimo de similitud tomado en cuenta para decidir si dos términos están relacionados entre sí. Este parámetro se elige experimentalmente.

$$sim(\mathbf{x}, \mathbf{x}_i) = \frac{\mathbf{x}^T \mathbf{x}_i}{\|\mathbf{x}\| \, \|\mathbf{x}_i\|} \tag{1}$$

Algorithm 1 Algoritmo para expandir la descripción de los servicios a través del tesauro de co-occurrencia

Input: $\mathbf{X}$, D, V, θ

Output: D_e

1. Initialize the expanded description $D_e \leftarrow D$
2. **for** $(t_j \in D \cap V)$
   a) **for** $(t_i \in V - D)$ **If** $cos(\mathbf{x}_i, \mathbf{x}_j) > \theta$ **then** $D_e \leftarrow D_e \cup \{t_i\}$
3. **Return**: D_e

Nosotros expandimos la descripción de los servicios con el algortimo 1. La entrada de este algoritmo incluye la matriz de factores $\mathbf{X}$, los términos de la descripción del servicio D y el vocabulario V, y el parámetro θ. El algoritmo devuelve la descripción expandida D_e.

5.4. MODELO DE FACTORIZACIÓN MATRICIAL

Adoptamos un modelo de aprendizaje automático para hacer frente al problema de factorización matricial antes mencionado. Para este fin, estimamos ambas matrices de factores $\mathbf{W}$ y $\mathbf{X}$ minimizando la función de error, $E(W, X)$, definida como la norma al cuadrado de Frobenius de la matriz de errores de aproximación, $\|\mathbf{XW} - \mathbf{C}\|_F^2$, como sigue:

$$\min_{\mathbf{X},\mathbf{W}} E(W, X) = \frac{1}{2}\|\mathbf{XW} - \mathbf{C}\|_F^2 + \frac{\lambda}{2}(\|\mathbf{X}\|_F^2 + \|\mathbf{W}\|_F^2) \tag{2}$$

donde λ es el parámetro de regularización utilizado para evitar que la norma Frobenius de cada matriz de factores alcance grandes magnitudes. Todos los términos son al cuadrado para tener un mínimo global óptimo, es decir, para tener una función de error convexa. Esto es útil porque el gradiente descendente puede usarse más allá del mínimo global en lugar de heurísticas más costosa, tal como el recocido simulado –*simulated annealing*– o los algoritmos genéticos. Por lo tanto, para minimizar la función de error,

calculamos el gradiente descendente estableciendo la derivada de la función de error con respecto a $\mathbf{W}$ como sigue:

$$\mathbf{X}^T(\mathbf{XW} - \mathbf{C}) + \lambda\mathbf{W} \tag{3}$$

Como resultado, el gradiente descendente es:

$$\mathbf{W} \leftarrow \mathbf{W} - \eta(\mathbf{X}^T(\mathbf{XW} - \mathbf{C}) + \lambda\mathbf{W}) \tag{4}$$

donde η la tasa de aprendizaje. Usamos esta regla de actualización para estimar $\mathbf{W}$. Por otro lado, para obtener $\mathbf{X}$, tomamos la derivada de la Equación 2 con respecto a $\mathbf{X}$ y la establecemos igual a cero de la siguiente manera:

$$\mathbf{X} = \mathbf{CW}^T(\mathbf{WW}^T + \lambda\mathbf{I})^{-1} \tag{5}$$

Usamos esta ecuación para calcular $\mathbf{X}$, mientras que actualizamos $\mathbf{W}$ con la regla en la Ecuación 4.

Algorithm 2 Algortimo de factorización de matrices basado en la minimizar la función de error

Input: $\mathbf{C}$, r, η_0, λ, maxIter

Output: $\mathbf{W}$ and $\mathbf{X}$

1. Initialize the $r \times n$ matrix $\mathbf{W}$ to small random values
2. **for** (i **in** 1:maxIter)
   - *a*) $\mathbf{X} \leftarrow \mathbf{CW}^T(\mathbf{WW}^T + \lambda\mathbf{I})^{-1}$
   - *b*) $\eta \leftarrow \eta_0/(1 + \eta_0\lambda i)$
   - *c*) $\mathbf{W} \leftarrow \mathbf{W} - \eta(\mathbf{X}^T(\mathbf{XW} - \mathbf{C}) + \lambda\mathbf{W})$
3. **Return**: $\mathbf{W}$ and $\mathbf{X}$

El algoritmo 2 estima ambas matrices de factores. La entrada del algoritmo está compuesta por la matriz objetivo $\mathbf{C}$, el número de factores latentes r a estimar, el número de iteraciones $maxIter$ utilizado para encontrar el mínimo global, la tasa de aprendizaje inicial η_0, y el parámetro de regularización λ. El algoritmo comienza a inicializar $\mathbf{W}$ a valores aleatorios –ver paso 1 –. En el paso 2.a dentro del ciclo, la representación del

factor latente para los documentos ($\mathbf{X}$) se calcula aplicando la solución analítica de la Ecuación 5, con $\mathbf{W}$ fija en la versión conocida en la iteración actual dada por la variable llamada i. La tasa de aprendizaje η se actualiza en el paso 2.b. para ir más rápido hacia la dirección del gradiente al principio, pero es más pequeño con cada iteración para evitar oscilaciones o divergencias. La tasa de aprendizaje que se utiliza en este enfoque es una tasa decreciente que depende del número de iteraciones, el parámetro de regularización y el factor de aprendizaje inicial (Bottou, 2010). En el paso final de este ciclo –ver paso 2.c–, $\mathbf{W}$ se actualiza de acuerdo con la regla presentada en la Ecuación 4, con $\mathbf{X}$ fijo en la última versión conocida. Finalmente, la salida de este algoritmo está compuesta por las matrices de factores como se muestra en el paso 3.

Para lo anterior, ajustamos los siguientes parámetros:

- El umbral θ en el Algoritmo 1.
- El número de factores latentes r, el parámetro de regularización λ, y la tasa de aprendizaje inicial η_0 en el Algoritmo 2.
- La cantidad de unidades ocultas y el parámetro de regularización de MLP.
- El parámetro de regularización en la formulación de Lagrange de SVM. Además, los parámetros asociados a cada función kernel, por ejemplo, el radio γ en el kernel Gaussiano.

Finalmente, programamos cada modelo probado en R. En la siguiente capítulo presentamos los resultados de la evaluación y su discusión o análisis.

6. RESULTADOS Y DISCUSIÓN

6.1. RESULTADOS

Como se mencionó anteriormente, ajustamos los parámetros a través de validación cruzada de 10 veces, así que usamos la mejor configuración de parámetros para probar los siguientes modelos:

1. Modelo con el algoritmo de Máquina de Vectores de Apoyo, SVM con varias funciones de kernel, es decir, lineales, gaussianas y sigmoideas, aplicadas en descripciones de servicios no expandidos. En la Tabla 4 mostramos que SVM con kernel sigmoide funciona mejor que SVM con otros kernels.
2. Modelo con algoritmo de Redes Neuronales Artificiales MLP aplicado en las descripciones de servicios no expandidos. En la Tabla 4 mostramos que MLP supera a SVM.
3. Los clasificadores SVM y MLP aplicados sobre descripciones de servicios expandidas. Tabla 5 mostramos que MLP supera a los otros modelos.

Tabla 4:
Prueba de Exactitud de los modelos de aprendizaje automático aplicados en la colección de descripción de servicio original

Modelo	Prueba de exactitud
SVM con kernel lineal	93.23 %
SVM con kernel Gaussiano	93.23 % ($\gamma = 10^{-4}$)
SVM con kernel sigmoide	94.79 %
MLP	95.83 % (30 unidades ocultas y parámetro de regularización igual a 10^{-2})

En la Tabla 6 resumimos los resultados de la prueba. Nuestros hallazgos son:

1. El MLP supera a SVM en ambos casos, es decir, cuando se aplican en descripciones ampliadas o en las originales,
2. La expansión de la descripción del servicio aumenta la prueba de exactitud de los modelos de aprendizaje automático, y
3. Hasta donde sabemos, el MLP aplicado en las descripciones ampliadas supera el mejor enfoque en el estado de la técnica propuesto por Sharma et al. Sharma *et al.* (2016).

Tabla 5:
Prueba de Exactitud de los modelos de aprendizaje automático aplicados en la descripción del servicio ampliado

Función Kernel	Umbral (θ)	Factores Latentes	Prueba de exactitud
SVM con kernel lineal	95 %	120	94.79 %
SVM con kernel sigmoide	95 %	80	95.83 %
SVM con kernel Gaussiano	97 %	30	96.35 % ($\gamma = 10^{-4}$)
MLP	97 %	120	97.92 % (90 unidades ocultas y parámetro de regularización igual a 1)

Tabla 6:
Resumen del resultado de la investigación

Modelo	Prueba de exactitud	Ganancia
SVM (línea base)	94.79 %	-
MLP	95.83 %	1.09 %
SVM aplicado sobre descripciones de servicios expandidas	96.3 %	1.6 %
Sharma *et al.* (2016)	97.22 %	2.56 %
MLP aplicado sobre descripciones de servicios expandidas	**97.92** %	**3.3** %

6.2. DISCUSIÓN

El objetivo general del proyecto fue el mejoramiento de la exactitud en la clasificación de servicios web a partir de la inyección de términos en las descripciones de los mismos, teniendo en cuenta las relaciones semánticas entre los términos, debido a que los pocos términos usados y la omisión de las relaciones semánticas, inciden en el resultado de la clasificación apoyada por métodos de aprendizaje automático, que otros autores como Rosso *et al.* (2003), Sedding y Kazakov (2004), o Shehata (2009), han planteado y propuesto superar.

Respecto al mejoramiento del desempeño por inyección de términos, los mejores valores de exactitud antes de expandir las descripciones para el clasificador basado en SVM fueron, con kernel lineal 93.23 %, con kernel Gaussiano 93.23 % y con kernel sigmoide 94.79 %. Mientras que después de expandir las descripciones, los valores de la exactitud mejoraron, con kernel lineal 94.79 %, con kernel sigmoide 95.83 % y con kernel Gaussiano 96.35 %. Para el clasificador basado en MLP, antes de expandir las descripciones el valor de la exactitud fue 95.83 % y después de expandir las descripciones el valor fue 97.92 %, con una ganancia de exactitud de 3.3 % sobre la línea base, y superando la ganancia de exactitud de 2.56 % del estado del arte (Sharma *et al.*, 2016) con respecto a la línea base, cuya exactitud fue 97.22 %. Lo anterior corrobora que el descubrimiento de las relaciones semánticas y a partir de estas, inyectar términos en las descripciones mejora el ejercicio de clasificación ya que ambos métodos de aprendizaje SVM y MLP muestran valores de exactitud superiores después de expandir las descripciones.

Los resultados evidencian que el aumento de términos en las descripciones, lo cual es un aumento de la dimensionalidad del conjunto de datos antes del proceso de clasificación, permite una mejor inducción de las relaciones entre las entradas y las salidas, lo que indica que el espacio original no es el mas conveniente, ya que, aún cuando a través de los métodos de kernel y SVM se aumenta la dimensionalidad (Cortes y Vapnik, 1995) o en el método MLP con propagación hacia atrás se disminuye (Werbos, 1974; Rumelhart *et al.*, 1988), se obtienen mejoras en la exactitud, indicando que el conjunto de datos original contiene ruido.

La mejor exactitud obtenida de 97.92 % mediante MLP con propagación hacia atrás, manifiesta que en esta investigación el enfoque de reducir dimensionalidad después de

la inyección de términos, es más conveniente para la clasificación, que el enfoque de aumentarla con el truco del Kernel sobre SVM.

La construcción de un tesauro basado en la similitud de sus términos para identificar sus relaciones semánticas (Budanitsky y Hirst, 2006), es un método válido para el mejoramiento de la clasificación y adaptable a cualquier idioma o al uso de tesauros donde no estén explícitas las relaciones semánticas, lo que es una ventaja con respecto al enfoque propuesto por Sharma *et al.* (2016) con el uso del diccionario de sinónimos del idioma inglés denominado WordNet, con relaciones semánticas de caracter general para dicho idioma y puede no contener relaciones para dominios específicos, que si pueden ser detectadas en enfoques como el nuestro, por ejemplo para el dominio del emprendimiento el término *startup* y el término *company* son sinónimos, teniendo en cuenta que el primer término frecuentemente hace referencia a una compañia con un modelo de negocio escalabe que se encuentra en sus primeras etapas, como nos lo deja apreciar Ries (2011), mientras que en WordNet, el término hace referencia al acto de poner en funcionamiento algo o al acto de iniciar una nueva operación (princeton.edu, 2019). Otra ventaja de nuestro enfoque, es la capacidad de aprender las relaciones semánticas a partir de la similitud de términos según el grado en que aparecen juntos en las descripciones y sus factores latentes ocultos, lo anterior se logra mediante el algoritmo de factorización de la matriz de similitudes.

Los fundamentos de las redes neuronales artificiales, proponen la optimización mediante la minimización de la función de pérdida, a través de la búsqueda del mejor mínimo local –ver figura 18– mediante el algoritmo de gradiente descendente (Werbos, 1974; Rumelhart *et al.*, 1988), aunque otros algoritmos de optimización pueden ser usados, lo anterior causa que no siempre se obtenga la solución más óptima, por eso la búsqueda de la mejor configuración de parámetros implica muchas validaciones costosas en términos de tiempo y cómputo. A diferencia de las máquinas de vectores de apoyo, que plantean el problema de optimización por minimización de una función objetivo convexa (Cortes y Vapnik, 1995) y al encontrar la mejor configuración de parámetros los resultados de la validación y prueba siempre son los mismos en términs de exactitud –ver figura 19–.

Lo anterior presenta una aparente contradicción, el modelo basado en MLP arroja mejores resultados en la exactitud que el modelo basdo en SVM, a pesar que MLP puede converger en un mínimo local que no necesariamente es el más óptimo, debido a que la función de pérdida tiene varios mínimos locales como muestra la figura 18.

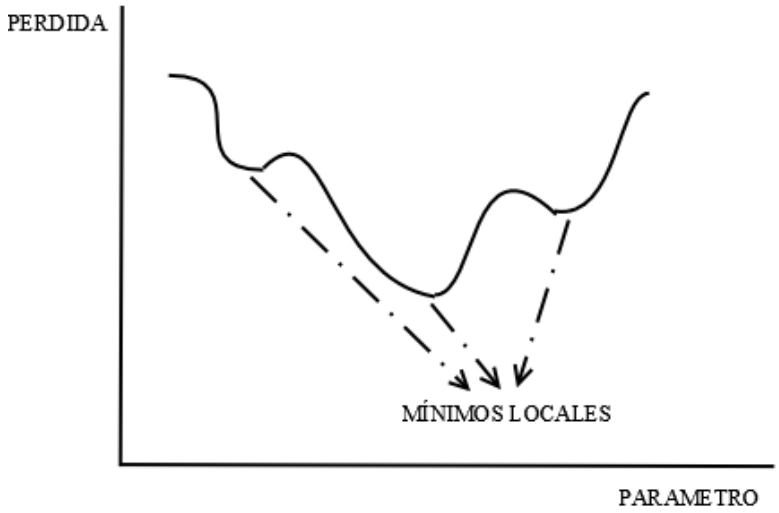

Figura 18: Función de pérdida para MLP. Fuente: Propia

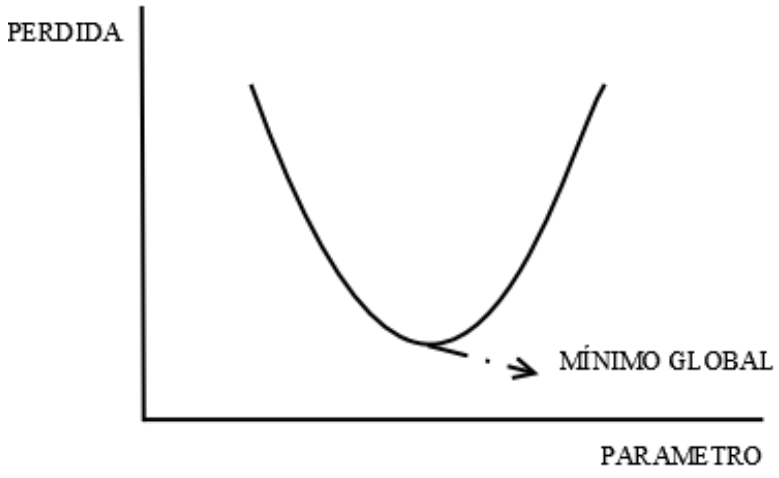

Figura 19: Función de pérdida para SVM. Fuente: Propia

Presentamos que el mejor comportamiento de las redes neuronales, se sustenta en la sinergia; el todo es más que las partes, en este caso de las unidades neuronales, o neuronas artificiales –perceptrones–, interconectadas a través de la red, emerge un comportamiento inteligente capaz de clasificar servicios con mayor exactitud, que una sola neurona artificial e incluso mejor que las máquinas de vectores de apoyo, que se sustentan en el prinicipio de *La Navaja de Ockham*, el cual plantea, que la explicación más sencilla es probablemente la explicación correcta, cuando Vapnik principal exponente de SVM, plantea como principio, nunca resolver un problema más complejo, como primer paso para resolver un problema actual, con este enfoque podríamos estar dejando de lado comportamientos del sistema que se presentan por sinergia. Una evidencia en nuestro enfoque es que en la capa oculta se descubre un espacio semántico entre las descripciones de los servicios que el modelo basado en las máquina de soporte vectorial no alcanza a descubrir.

CONCLUSIONES

La clasificación en el dominio al que pertenecen las descripciones de los servicios web del conjunto de datos OWLS-TC2 a partir de la inyección de términos similares presentes en las etiquetas `-serviceName-` y `-textDescription-`, donde esta similitud representa la relación semántica entre ellos, identificadas a partir del descubrimiento de factores latentes entre los términos y posteriormente con las descripciones enriquecidas, obtener una representación en forma de vectores de cada descripción para usarlos como conjunto de entrenamiento y validación del clasificador mediante el método de validación cruzada e inducir modelos a partir de los algoritmos de aprendizaje automáticos SVM y MLP, evaluados con la exactitud como indicador de desempeño, nos permitió obtener resultados que confirman el mejoramiento del indicador al expandir las descripciones de los servicios.

Nuestro enfoque contribuye a los ejercicios de clasificación de textos cortos, principalmente por proponer la inclusión de términos en las descripciones de los servicios a partir del descubrimiento de las relaciones semánticas de los términos del vocabulario conformado por los términos de todas las descripciones. Para lo cual se descubre la similitud entre términos y sus factores latentes ocultos, que evidencien la relación. Los aspectos anteriores permiten proponer ejercicios de clasificación adaptables a vocabularios libres y de diferentes idiomas a diferencia de otros enfoques que usan un vocabulario controlado o tesauro predefinido de un idioma específico.

La realización de la investigación nos permitió identificar los siguientes hallazgos:

1. La ganancia de exactitud de nuestro método, obtenida por el modelo inducido con MLP, con respecto a SVM como línea base, el cual es considerado como uno de los mejores métodos de clasificación en tiempos recientes (Mohri *et al.*, 2018, pg. 79), es de 3,3 %; mientras que la ganancia de exactitud del estado del arte (Sharma *et al.*, 2016) es de 2,56 % sobre la línea base.
2. Las redes neuronales tienen mayor exactitud que SVM en esta investigación, a pesar

que presentan una reducción de dimensionalidad ya que la cantidad de salidas de la capa de entrada es menor a la cantidad de entradas de la misma capa.

3. Los algoritmos de clasificación usados tienen mayor exactitud sobre las descripciones expandidas que sobre las originales.
4. En SVM con el kernel Gaussiano o radial se tiene mejor exactitud clasificando servicios que SVM con el kernel sigmoidal y lineal -solo SVM-. Lo cual refleja que aunque este es un problema de clasificación de texto, las descripciones de servicios son muy breves y por ende el espacio original de características no es conveniente para clasificación.
5. El hecho que la mejor configuración paramétrica para el modelo inducido con MLP, donde las salidas de la capa oculta es menor que el número de dimensiones del espacio de características de entrada, implica que las características que describen a cada servicio tienen presencia de ruido.

Es notable que MLP es más exacto que SVM, a pesar que este modelo se presenta como un problema de optimización convexa con una única solución óptima, es decir un mínimo global, mientras que MLP tiene varias soluciones subóptimas, donde solo una es la más óptima. Esto se debe a que la función de error definida en MLP tiene un espacio de búsqueda más amplio presentándose varios mínimos locales. Una posible explicación de estos resultados podría ser que la capa oculta del MLP reduce la dimensionalidad del espacio de entrada original, por lo tanto, se elimina el posible ruido en el conjunto de datos aunque las descripciones de los servicios sean breves.

Finalmente, los resultados revelan que nuestro enfoque es más exacto que el mejor en investigaciones previas, según lo mejor de nuestro conocimiento, propuesto por Sharma *et al.* (2016). Este enfoque, como lo discutimos en la sección Introducción se basa en Wordnet. Por lo tanto, no puede adaptarse a otro dominio o idioma, mientras que el tesauro de sinónimos de co-ocurrencia generado automáticamente sí puede. Este último aprende relaciones específicas entre términos en dominios particulares. Como consecuencia, nuestro enfoque se adapta automáticamente a un contexto específico, incluido otro idioma. Por ende, nuestro enfoque no se limita a las relaciones entre los términos en inglés.

TRABAJOS FUTUROS

A partir de este proyecto presentamos varias posibilidades de investigaciones futuras que podemos abordar, tales como:

1. Teniendo en cuenta que tanto antes de expandir las descripciones como después de expandirlas con ambos enfoques, SVM y MLP, se obtiene una mejora en la clasificación, se podría explorar si esta situación se mantiene en ejercicios con otros conjuntos de datos, donde se tengan presente las relaciones semánticas descubiertas por co-ocurrencia entre términos de documentos breves.
2. Realizar ejercicios de clasificación con expansión de términos mediante el descubrimiento de relaciones semánticas, para diferentes idiomas.
3. Estudiar el impacto de la exactitud en la clasificación, cuando el tesauro de co-ocurrencia se genera a partir de un corpus más amplio, por ejemplo, Wikipedia. Finalmente,
4. Debido al posible ruido en el conjunto de datos, se puede estudiar el impacto en la exactitud por adoptar la indexación semántica latente para descubrir relaciones semánticas entre documentos y luego realizar la clasificación con SVM, para determinar si se obtienen mejores resultados que MLP.
5. Experimentar el uso de otras arquitecturas de redes neuronales profundas, con más capas ocultas, usando redes neuronales recurrentes y autoencoder.

REFERENCIAS

Alonso, G., Casati, F., Kuno, H., y Machiraju, V. (2004). Web Services. En *Web Services*, pp. 123–149. Springer Berlin Heidelberg, Berlin, Heidelberg.

Alpaydin, E. (2010). *Introduction to Machine Learning Second Edition*. MIT Press, 2nd edición.

Amazon (2018). Amazon API Gateway: Developer Guide. Recuperado de https://aws.amazon.com/es/api-gateway/.

Berners-Lee, T., Hendler, J., Lassila, O., y Others (2001). The semantic web. *Scientific american*, 284(5):28–37.

Blum, A., Kalai, A., y Langford, J. (1999). Beating the Hold-out: Bounds for K-fold and Progressive Cross-Validation. *Proceedings of the 12th Annual Conference on Computational Learning Theory*, (c):203–208.

Booking (2018). Booking.com Connectivity Portal. Recuperado de https://connect.booking.com/user_guide/site/en-US/user_guide.html?lang=en.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. En Lechevallier, Y. y Saporta, G., editores, *Proc. of the 19th International Conference on Computational Statistics*, pp. 177–187, Vienna. Physica-Verlag HD.

Budanitsky, A. y Hirst, G. (2006). Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*.

Cerami, E. (2002). *Web services essentials*. O'Reilly.

Corella, M. Á. y Castells, P. (2006). *Semi-automatic Semantic-Based Web Service Classification*, pp. 459–470. Springer Berlin Heidelberg, Berlin, Heidelberg.

Cortes, C. y Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3):273–297.

Crasso, M., Zunino, A., y Campo, M. (2008). AWSC: an approach to web service classification based on machine learning techniques. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 12(37):25–36.

Dogtiev, A. (2016). Facebook App Statistics. Recuperado de http://www.businessofapps.com/facebook-app-statistics/.

Eric Kim (2019). The Kernel Trick. Recuperado de http://www.eric-kim.net/

eric-kim-net/posts/1/kernel_trick.html.
Facebook (2018). Graph API Reference. Recuperado de https://developers.facebook.com/docs/graph-api/reference.
Google Inc. (2018). Google Developers. Recuperado de https://developers.google.com/.
Hearst, M. A. y A., M. (1999). Untangling text data mining. En *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* -, pp. 3–10, Morristown, NJ, USA. Association for Computational Linguistics.
Heß, A., Johnston, E., y Kushmerick, N. (2004). ASSAM: A Tool for Semi-automatically Annotating Semantic Web Services.
Heß, A. y Kushmerick, N. (2003). *Learning to Attach Semantic Metadata to Web Services*, pp. 258–273. Springer Berlin Heidelberg, Berlin, Heidelberg.
Internet World Stats (2018). World Internet Users Statistics and 2018 World Population Stats. Recuperado de https://www.internetworldstats.com/stats.htm.
Jiang, L., Zhang, H.-b., Yang, X., y Xie, N. (2013). Research on Semantic Text Mining Based on Domain Ontology. *Computer and Computing Technologies in Agriculture VI*, 392:336–343.
Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. pp. 137–142. Springer, Berlin, Heidelberg.
Katakis, I., Meditskos, G., Tsoumakas, G., Bassiliades, N., y Vlahavas (2009). *On the Combination of Textual and Semantic Descriptions for Automated Semantic Web Service Classification*, pp. 95–104. Springer US, Boston, MA.
Klusch, M., Fries, B., y Sycara, K. (2006). Automated Semantic Web Service Discovery with OWLS-MX. *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems. International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006), May 8-12, Hakodate, Japan*, pp. 915–922.
Liu, X., Agarwal, S., Ding, C., y Yu, Q. (2016). An LDA-SVM Active Learning Framework for Web Service Classification. *2016 IEEE International Conference on Web Services (ICWS)*, pp. 49–56.
Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
Mohanty, R., Ravi, V., y Patra, M. R. (2012). Classification of web services using bayesian network. *Journal of Software Engineering and Applications*, 5(4):291–296.
Mohri, M., Rostamizadeh, A., y Talwalkar, A. (2018). *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2 edición.

Montgomery, D. C. (2017). *Design and analysis of experiments*. Wiley, 9th edición.

Nisa, R. y Qamar, U. (2015). A text mining based approach for web service classification. *Information Systems and e-Business Management*, 13(4):751–768.

Oldham, N., Thomas, C., Sheth, A., y Verma, K. (2005). *METEOR-S Web Service Annotation Framework with Machine Learning Classification*, pp. 137–146. Springer Berlin Heidelberg, Berlin, Heidelberg.

Paradigms Lynne Murphy, O. M. (2003). Semantic Relations and the Lexicon. pp. 43–21.

Patil, A. A., Oundhakar, S. A., Sheth, A. P., y Verma, K. (2004). Meteor-s Web Service Annotation Framework. En *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pp. 553–562, New York, NY, USA. ACM.

princeton.edu (2019). WordNet - A Lexical Database for English. Recuperado de https://wordnet.princeton.edu/.

Ries, E. (2011). *The Lean Startup: How Today's Entrepeneurs Use Continuous Innovation to Create Rdically Sucesful Businesses*.

Rosso, P., Ferretti, E., Jiménez, D., y Vidal, V. (2003). Text Categorization and Information Retrieval Using WordNet Senses. *The Second Global Wordnet Conference*, pp. 299–304.

Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1988). Neurocomputing: Foundations of Research. capítulo Learning Representations by Back-propagating Errors, pp. 696–699. MIT Press, Cambridge, MA, USA.

Salton, G., Wong, A., y Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Sanger, J. y Feldman, R. (2007). *The text mining handbook: Advanced approaches in analyzing unstructured data*. Cambridge University Press.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Sedding, J. y Kazakov, D. (2004). WordNet-based Text Document Clustering.

Sharma, S., Lather, J. S., y Dave, M. (2016). Semantic approach for Web service classification using machine learning and measures of semantic relatedness. *Service Oriented Computing and Applications*, 10(3):221–231.

Shehata, S. (2009). A WordNet-based semantic model for enhancing text clustering. En *ICDM Workshops 2009 - IEEE International Conference on Data Mining*, pp. 477–482. IEEE.

Tan, A. (1999). Text Mining: The state of the art and the challenges. En *Proceedings of*

the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases, pp. Vol. 8, pp. 65–70.

Torres, Jordi (2018). Neural Networks - Towards Data Science. Recuperado de https://towardsdatascience.com/basic-concepts-of-neural-networks-1a18a7aa2bd2.

Trivago (2018). trivago Express Booking. Recuperado de http://developer.trivago.com/expressbooking/api-workflow.html.

Tsatsaronis, G., Gr, V., Vazirgiannis, M., y Gr, M. (2010). Text Relatedness Based on a Word Thesaurus. *Journal of Artificial Intelligence Research*, 37:1–39.

uddi.org (2018). UDDI Version 3.0.1. Recuperado de http://www.uddi.org/pubs/uddi-v3.0.1-20031014.htm.

w3.org (2018a). OWL Web Ontology Language Overview. Recuperado de https://www.w3.org/TR/owl-features/.

w3.org (2018b). RDF 1.1 Concepts and Abstract Syntax. Recuperado de https://www.w3.org/TR/rdf11-concepts/.

w3.org (2018c). SOAP Version 1.2 Part 0: Primer (Second Edition). Recuperado de https://www.w3.org/TR/2007/REC-soap12-part0-20070427/.

w3.org (2018d). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. Recuperado de https://www.w3.org/TR/wsdl/.

Weather Underground (2018). Weather API. Recuperado de https://www.wunderground.com/weather/api.

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Tesis doctoral, Harvard University.

Witten, H., Frank, E., y Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*.

Yang, J. y Zhou, X. (2014). Semi-automatic Algorithem Based on Web Service Classification. *Advanced Science and Technology Letters*, 53:88–91.

Zhang, D., Li, J.-Z., y Xu, B. (2005). Web service annotation using ontology mapping. En *2005 IEEE International Workshop on Service-Oriented System Engineering (SOSE 2005), 20-21 October 2005, Beijing, China*, pp. 235–242.