UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Combinación de clustering, selección de atributos y métodos ontológicos para la clasificación semántica de texto

D. Alexander josé Mackenzie Rivero

2023

UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Combinación de clustering, selección de atributos y métodos ontológicos para la clasificación semántica de texto

Autor: D. Alexander José Mackenzie Rivero

Director/es: D. Rodrigo Martínez Béjar

D. Fernando Jiménez Barrionuevo

UNIVERSIDAD DE MURCIA

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD
DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

Aprobado por la Comisión General de Doctorado el 19-10-2022

D./Dña. Alexander José Mackenzie Rivero

doctorando del Programa de Doctorado en

Informática

de la Escuela Internacional de Doctorado de la Universidad Murcia, como autor/a de la tesis presentada para la obtención del título de Doctor y titulada:

Combinación de clustering, selección de atributos y métodos ontológicos para la clasificación semántica de texto

y dirigida por,

D./Dña. Rodrigo Martínez Béjar

D./Dña. Fernando Jiménez Barrionuevo

D./Dña.

DECLARO QUE:

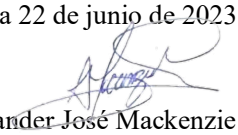
La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita, cuando se han utilizado sus resultados o publicaciones.

Si la tesis hubiera sido autorizada como tesis por compendio de publicaciones o incluyese 1 o 2 publicaciones (como prevé el artículo 29.8 del reglamento), declarar que cuenta con:

- *La aceptación por escrito de los coautores de las publicaciones de que el doctorando las presente como parte de la tesis.*
- *En su caso, la renuncia por escrito de los coautores no doctores de dichos trabajos a presentarlos como parte de otras tesis doctorales en la Universidad de Murcia o en cualquier otra universidad.*

Del mismo modo, asumo ante la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada, en caso de plagio, de conformidad con el ordenamiento jurídico vigente.

En Murcia, a 22 de junio de 2023



Fdo.: Alexander José Mackenzie Rivero

Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en la primera página de la tesis presentada para la obtención del título de Doctor.

Información básica sobre protección de sus datos personales aportados	
Responsable:	Universidad de Murcia. Avenida teniente Flomesta, 5. Edificio de la Convalecencia. 30003; Murcia. Delegado de Protección de Datos: dpd@um.es
Legitimación:	La Universidad de Murcia se encuentra legitimada para el tratamiento de sus datos por ser necesario para el cumplimiento de una obligación legal aplicable al responsable del tratamiento. art. 6.1.c) del Reglamento General de Protección de Datos
Finalidad:	Gestionar su declaración de autoría y originalidad
Destinatarios:	No se prevén comunicaciones de datos
Derechos:	Los interesados pueden ejercer sus derechos de acceso, rectificación, cancelación, oposición, limitación del tratamiento, olvido y portabilidad a través del procedimiento establecido a tal efecto en el Registro Electrónico o mediante la presentación de la correspondiente solicitud en las Oficinas de Asistencia en Materia de Registro de la Universidad de Murcia

UNIVERSIDAD DE
MURCIA

D. Rodrigo Martínez Béjar, Catedrático de Universidad del Área de Informática en el Departamento de Ingeniería de la Información y las Comunicaciones , AUTORIZA:

La presentación de la Tesis Doctoral titulada "Combinación de clustering, selección de atributos y métodos ontológicos para la clasificación semántica de texto", realizada por D. Alexander José Mackenzie Rivero, bajo mi inmediata dirección y supervisión, y que presenta para la obtención del grado de Doctor por la Universidad de Murcia.

En Murcia, a 15 de junio de 2023

Firmado por RODRIGO MARTINEZ BEJAR - NIF:***6206** el día 15/06/2023 con un certificado emitido por ACCVCA-120

UNIVERSIDAD DE
MURCIA

D. Fernando Jiménez Barrionuevo, Catedrático de Universidad del Área de Informática en el Departamento de Ingeniería de la Información y las Comunicaciones , AUTORIZA:

La presentación de la Tesis Doctoral titulada “Combinación de clustering, selección de atributos y métodos ontológicos para la clasificación semántica de texto“, realizada por D. Alexander José Mackenzie Rivero, bajo mi inmediata dirección y supervisión, y que presenta para la obtención del grado de Doctor por la Universidad de Murcia.

En Murcia, a 15 de junio de 2023

FERNANDO |JIMENEZ| BARRIONUEVO

Firmado digitalmente por FERNANDO| JIMENEZ| BARRIONUEVO
Fecha: 2023.06.15 13:32:06 +02'00'

A mi madre y abuelo llevo siempre presente sus palabras y enseñanzas.
A mi esposa e hijos por ser mi motivación.

Esta tesis ha sido escrita usando LaTeX.

Agradecimientos

Agradezco a quienes han sido mis directores de tesis, los Doctores Rodrigo Martínez Béjar y Fernando Jiménez Barrionuevo, por toda la ayuda y dedicación en la realización de este trabajo, asi como por motivarme a seguir adelante.

De igual manera debo agradecer a la Universidad Estatal del Sur de Manabí por su respaldo al inicio de mis estudios doctorales, a: Dr. Edwin Joao Merchan, Dra. Karina Mero y Dr. Alberto Rodriguez.

Agradezco a mi madre y a mi esposa por siempre creer en mi y apoyarme en la realización de este trabajo.

Resumen

Con el aumento exponencial en la cantidad de datos textuales disponibles en Internet desde fuentes diversas como redes sociales, blogs/foros, sitios web, correos electrónicos, bibliotecas en línea, etc., se ha hecho necesaria la utilización de la Inteligencia Artificial en plataformas digitales, como la aplicación de métodos de aprendizaje profundo y de reconocimiento de patrones, para que esta información pueda ser aprovechada por todo tipo de modelos de negocios, estudios de mercado, planes de marketing, campañas políticas o toma de decisiones estratégicas entre otros, con la finalidad de hacer frente a la competencia y dar respuesta de manera eficiente.

El objetivo de esta tesis doctoral fue desarrollar un modelo que combina clustering, selección de atributos y métodos ontológicos para la clasificación semántica de texto, que permita estructurar una metodología aplicable en conjuntos de datos textuales y así mejorar la clasificación automática de texto. El modelo propuesto en esta tesis doctoral se realizó siguiendo los siguientes objetivos específicos: redactar el estado del arte relacionado con la temática estudiada; conformación de un conjunto de datos textuales lo suficientemente extenso para la aplicación de las diferentes técnicas de análisis de datos; desarrollo de una metodología para la clasificación semántica de datos textuales y evaluación de los resultados obtenidos.

Se pudo determinar que haciendo SToWVector junto con selección de atributos mediante el wrapper MOES (estrategia de búsqueda) y NaiveBayesMultinomial (evaluador) con ACC (métrica), se obtienen mejores resultados con el clasificador NaiveBayesMultinomial que con otros métodos de clasificación evaluados. Además el método de búsqueda ENORA ha sido utilizado y evaluado demostrando ser un método eficaz para la selección de atributos en datos textuales. De igual manera se pudo dar significado a los dos clústeres obtenidos, logrando identificar un concepto para cada clúster. Clúster 1: UE−G20−G77−MEC y clúster 2: Resto del mundo. Ello permitió establecer una relación directa entre los clústers.

Abstract

With the exponential increase in the amount of textual data available on the Internet from various sources such as: social networks, blogs/forums, websites , emails, online libraries, etc. It has made necessary the use of artificial intelligence in digital platforms, the application of parallel processing, deep learning and pattern recognition so that this information can be used by all kinds of models business, market research, marketing plans, political campaigns or making strategic decisions among others, in order to deal with competition and respond efficiently.

This doctoral thesis is focused on developing a model that allows combine clustering, attribute selection and ontological methods for the semantic classification of text, which allows structuring an applicable methodology in textual data sets to improve the automatic classification of text. The model proposed in this doctoral thesis is carried out following the following specific objectives: draft the status of the art related to the theme studied, conformation of a set of textual data extensive enough for the application of different data analysis techniques, development of a methodology for the semantic classification of textual data and evaluation of the results obtained.

Could determine that by doing SToWVector together with feature selection using the MOES wrapper (search strategy) and NaiveBayesMultinomial (evaluator) with ACC (metric), better results are obtained with the NaiveBayesMultinomial classifier than with other classification methods evaluated, in addition, the ENORA search method has been used and evaluated, proving to be an effective method for the selection of attributes in text data. In the same way, it was possible to give meaning to the two clusters obtained, managing to identify a concept for each cluster. Cluster 1: EUG20G77MEC and cluster 2: Rest of the world. This allowed us to establish a direct relationship between the clusters.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Motivación

1.1. Introducción

La clasificación automática de texto se emplea para organizar documentos en clases predeterminadas, generalmente usando algoritmos de aprendizaje automático, siendo uno de los métodos más importantes para organizar y aprovechar la inmensa cantidad de información que existe en formato de texto estructurado. La clasificación de textos es un área de investigación ampliamente estudiada de los procesos lingüísticos, procesamiento y minería de texto. En la clasificación de texto tradicional, un documento se representa como una bolsa de palabras donde estas, se extraen de su contexto más fino en otras palabras términos. Solo el contexto más amplio del documento se representa con algún tipo de término de información de frecuencia en el espacio vectorial. En consecuencia, la semántica de las palabras que puede ser inferida del contexto más fino de su ubicación en una oración y sus relaciones con los vecinos generalmente es ignorada. Sin embargo, el significado de las palabras, las conexiones semánticas entre palabras, los documentos e incluso las clases son obviamente importantes ya que los métodos que capturan la semántica generalmente alcanzan mejores rendimientos de clasificación. Se han publicado varias investigaciones para analizar diversos enfoques en los métodos tradicionales de clasificación de texto. La mayoría de estos trabajos logran cubrir la aplicación de diferentes métodos de relación semántica de términos en la clasificación de texto hasta un cierto grado.

Los problemas de clasificación automática de texto han sido ampliamente estudiados en muchas aplicaciones reales [7–15] durante las últimas décadas. Especialmente con los avances recientes en el procesamiento del lenguaje natural (PLN) y minería de texto, muchos investigadores se encuentran interesados en el desarrollo de aplicaciones que aprovechen los métodos de clasificación de textos. Los estudios de minería de textos son cada vez más populares en los últimos años debido a la amplia gama de fuentes que producen enormes cantidades de datos, como redes sociales, blogs/foros, sitios web, correos electrónicos y bibliotecas en línea que publican trabajos de investigación. Sin duda, el crecimiento de los datos textuales electrónicos seguirá aumentando con los nuevos desarrollos tecnológicos, como los motores de voz a texto y los asistentes digitales o asistentes personales inteligentes. Todo ello conlleva que procesar, organizar y manejar automáticamente los datos textuales

se haya convertido en un problema fundamental. La minería de texto tiene varias aplicaciones importantes, como clasificación (es decir, clasificación supervisada, no supervisada y semisupervisada), filtrado de documentos, resumen y análisis de sentimientos. Los métodos de procesamiento de lenguaje natural (PLN), aprendizaje automático (ML, por las siglas en inglés de Machine Learning) y minería de datos (DM por las siglas en inglés de Data Mining) trabajan en conjunto para detectar patrones de los diferentes tipos de documentos y clasificarlos de manera automática.

Lo anteriormente expuesto representa la principal motivación para realizar la presente tesis doctoral, cuyo objetivo fue combinar clustering, selección de atributos y métodos ontológicos para la clasificación semántica de texto que permita estructurar una metodología aplicable en conjuntos de datos textuales para mejorar la clasificación automática de texto.

En los métodos de clasificación semántica de textos, se consideran las relaciones semánticas entre palabras para medir la similitud entre documentos. El enfoque semántico se centra en el significado de los términos. Es por ello, más general y exacto que palabras y las conexiones semánticas ocultas entre los términos y en consecuencia entre documentos. Las ventajas de la clasificación semántica de texto sobre la clasificación de texto tradicional son:

- Descubrimiento de relaciones implícitas o explícitas entre palabras.
- Extraer y utilizar relaciones latentes entre palabras y documentos.
- Capacidad para generar palabras clave representativas de las clases existentes.
- Comprensión semántica del texto, que mejora la precisión de la clasificación.
- Capacidad para manejar la sinonimia y la polisemia en comparación con los algoritmos de clasificación de texto tradicionales, ya que utilizan la semántica.

1.2. Contexto del problema

En los últimos años, hemos sido testigos de un aumento en la cantidad de datos textuales digitales disponibles, generando nuevos conocimientos y, por lo tanto, abriendo oportunidades para la investigación a través de nuevos canales. En este campo de rápida evolución de las técnicas analíticas de big data, la minería de texto ha ganado una atención significativa en una amplia gama de aplicaciones. Tanto en la academia como en la industria, ha habido un cambio hacia proyectos de investigación y preguntas de investigación más complejas que exigen más que la simple recuperación de datos.

Debido a la creciente importancia de la Inteligencia Artificial (IA) y a su implementación, por ejemplo, a través de métodos de aprendizaje profundo y de reconocimiento de patrones, la información textual es crucial. Todo tipo de modelos de negocio, estudios de mercado, planes de marketing, campañas políticas o toma de

decisiones estratégicas se enfrentan a una necesidad creciente de técnicas de minería de texto para hacer frente a la competencia. Es posible recopilar grandes cantidades de datos textuales como parte de una investigación, como literatura científica, transcripciones en los sectores económicos y de marketing, discursos en el campo político, como campañas presidenciales y discursos de toma de posesión, y transcripciones de reuniones. Además, las fuentes en línea, como correos electrónicos, páginas web, blogs, publicaciones en redes sociales y comentarios, proporcionan una amplia fuente de datos textuales para la investigación. También se recopilan grandes cantidades de datos en forma semiestructurada, como archivos de registro que contienen información de servidores y redes. Como tal, el análisis de minería de textos es útil tanto para datos textuales no estructurados como semiestructurados [16].

La minería de datos y la minería de textos difieren en el tipo de datos que manejan. Mientras que la minería de datos maneja datos estructurados provenientes de sistemas, como bases de datos, hojas de cálculo, ERP, CRM y aplicaciones de contabilidad, la minería de texto trata con datos no estructurados que se encuentran en documentos, correos electrónicos, redes sociales y la web. Por lo tanto, la diferencia entre la minería de datos regular y la minería de texto es que en la minería de texto los patrones se extraen del texto en lenguaje natural en lugar de bases de datos estructuradas.

Dado que toda la información escrita o hablada se puede representar en forma de texto, debe utilizar la mayor cantidad de herramientas cuando se trata de la interpretación y análisis de oraciones, palabras, frases, discursos, reclamos, anuncios y declaraciones. Este documento lleva a cabo un análisis extenso de las aplicaciones de minería de texto tal como se utiliza en varios campos comerciales y estudios académicos. Si bien la gran mayoría de la literatura trata sobre la optimización de una técnica específica de minería de textos, esta tesis doctoral persigue sintetizar en una metodología los métodos de análisis de datos, en combinación con técnicas ontológicas, resumiendo así las prácticas y enfoques más avanzados de la minería de texto.

1.3. Objetivos de la tesis

- **Describir el estado del arte relacionado con la temática estudiada**. Para cumplir con el objetivo principal de esta tesis doctoral, se requiere del estudio de las técnicas actualmente utilizadas para la clasificación automática de texto, así como también de los trabajos relacionados con esta investigación.

- **Conformar un conjunto de datos textuales lo suficientemente extenso para la aplicación de las diferentes técnicas de análisis de datos**. Con la finalidad de cubrir todas las etapas del analisis de texto, se conformará un conjunto de datos propios extrayendo la información del repositorio de noticias de Euronews conformado por 434 instancias y 20 atributos.

- **Desarrollar una metodología para la clasificación semántica de datos textuales**. Este objetivo se basa en estructurar una metodología que permita la implementación de distintas técnicas de análisis y clasificación de datos en combinación con el análisis ontológico para realizar la clasificación semántica de datos textuales.
- **Evaluar los resultados obtenidos** con el propósito de comparar los resultados obtenidos se emplean distintas técnicas para la evaluación estadística de los resultados alcanzados.

.

1.4. Estructura de la tesis

La presente tesis está estructurada por los capítulos que a continuación se describen. Capítulo 2 (Antecedentes y estado del arte): Se describen los antecedentes de la investigación así como el estado actual de las tecnologías y métodos utilizados para el análisis de datos textuales. Iniciando por los métodos de clustering, donde se ofrece una visión de su clasificación según los tipos así como la validación del clustering. En lo referente a la selección de atributos, se describen los métodos para la selección de atributos actualmente más utilizados. Con respecto a la clasificación de texto se destacan sus aplicaciones, técnicas utilizadas así como clasificadores específicamente utilizados para texto. Con respecto a las ontologías, se realiza una descripción extensa que va desde la definición, componentes, tipos, lenguajes y aplicaciones. Finalmente se completa este capítulo con la descripción de trabajos relacionados con la presente tesis doctoral. Capítulo 3 (Metodología): En este capítulo se describen cada uno de los aspectos de la metodología, la cual está comprendida de 9 etapas: preprocesamiento, clustering, selección de atributos, clasificación, test estadísticos, validación del clúster, análisis semántico, interpretación, y finalmente la representación de relaciones. Capítulo 4 (Experimentos y resultados): en este capítulo son descritos los experimentos realizados durante esta tesis doctoral (experimento 1, experimento 2, experimento 3, así como los resultados obtenidos). Capítulo 5 (Análisis de los resultados): presenta la comparación entre los resultados obtenidos después de la realización de los experimentos, con la finalidad de observar en detalle los aspectos de interés para la investigación. Capítulo 6 (Conclusiones y trabajos futuros): En este último capítulo se redactan tanto las conclusiones como los trabajos futuros de esta tesis doctoral.

Capítulo 2

Antecedentes y estado del arte

En este capítulo se describen los fundamentos que han servido como antecedentes para la realización de esta Tesis Doctoral, así como el estado del arte y los trabajos relacionados con la temática de la misma. Entre estos fundamentos destacan los métodos de clustering, la selección de atributos en tareas de clasificación, particularmente la selección de atributos evolutiva multi-objetivo, y finalmente los relacionados con la construcción de ontologías.

2.1. Métodos de clustering

Definición

El clustering es la tarea de agrupar un conjunto de objetos, donde cada objeto se separa de acuerdo con una característica similar. Ver figura 2.1. Según [17], el clustering es una de las técnicas más utilizadas para el análisis exploratorio de datos. En todas las disciplinas, desde las ciencias sociales hasta la biología y la informática, las personas intentan tener una primera impresión sobre sus datos identificando grupos significativos entre los puntos de datos. Por ejemplo, los biólogos computacionales agrupan genes basados en similitudes en su expresión con diferentes experimentos; los comerciantes agrupan a los clientes en función de sus perfiles, para fines de marketing dirigido; y los astrónomos agrupan estrellas sobre la base de su proximidad espacial.

El clustering ha sido estudiado ampliamente en el aprendizaje automático debido a sus numerosas aplicaciones. Según [18], la agrupación en clústeres es omnipresente en ciencia e ingeniería, con numerosos y diversos dominios de aplicación, que van desde la bioinformática y la medicina hasta las ciencias sociales y la web. Tal como se estudia en [19]. El clustering es un problema de minería de datos ampliamente utilizado en los dominios de texto y aplicables a numerosas aplicaciones en la segmentación de clientes, clasificación, filtrado colaborativo, visualización, organización de documentos e indexación. En este estudio los autores proporcionan además, un análisis detallado del problema de la agrupación de texto, donde se estudian los desafíos clave del problema de la agrupación en clústeres aplicado al dominio del texto, se discuten los métodos clave utilizados para la agrupación de texto y sus ventajas relativas. De igual manera se describen los avances recientes en el área en el contexto

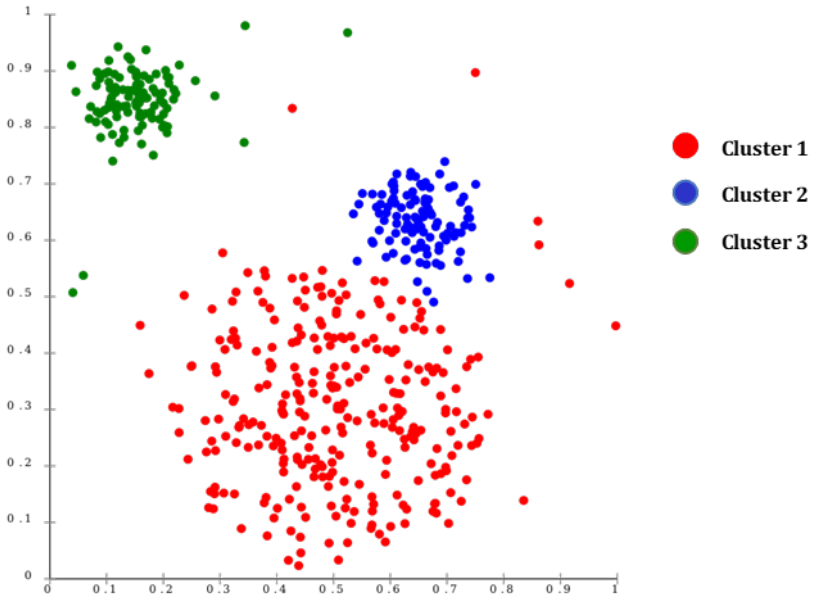


Figura 2.1: Ejemplo de clustering. Fuente: Google Image.

de las redes sociales y los datos vinculados. En [20], los autores presentan conceptos y algoritmos de selección de características, analizando los algoritmos de selección de características existentes para clasificación y clustering, comparando diferentes algoritmos con un marco de categorización basado en estrategias de búsqueda.

De acuerdo a [21], dependiendo de los datos y las características deseadas del clúster hay diferentes tipos de paradigmas de agrupación como: representativos, jerárquicos, clustering espectral, basado en densidad y basado en gráficos.

Clustering representativo

Según [21], dado un conjunto de datos con n puntos en un espacio $d-dimensional$,

$$D = \{X_i\}_{i=1}^{n} \tag{2.1}$$

y dado el número de grupos deseados k, el objetivo del clustering representativo es dividir el conjunto de datos en k grupos, que se denomina clustering y se denota como:

$$C = C_1, C_2, \ldots, C_k \tag{2.2}$$

además, para cada grupo c_i existe un punto representativo que resume al grupo, una opción común es la media (también llamada centroide) μ_i de todos los puntos en el grupo, es decir:

$$\mu = \frac{1}{n_i} \sum_{xj \epsilon C_i} X_j \tag{2.3}$$

Donde $n_i = |C_i|$ que es el número de puntos en el clúster C_i.

Algoritmo K-means
Tal como se indica en [22] el algoritmo K-means es uno de los algoritmos más utilizados para la tarea de clustering por su rapidez y versatilidad aplicable a una gran variedad de problemas. De acuerdo a [23], el algoritmo K-means es el método de agrupación por partición más utilizado. Recientemente, una gran cantidad de literatura se ha centrado en el procesamiento de flujo de datos como un método eficiente para derivar conocimiento de big data. El algoritmo de agrupamiento K-means se usa con frecuencia para agrupar datos estáticos, pero también se emplea para dividir una gran secuencia de datos en segmentos o segmentos de datos.
Suponiendo que hay N datos en S y $X_i^{'}$ i es el dato i^{th} en S. El algoritmo K-means [24] puede describirse en pseudo-código de la siguiente manera:

Algoritmo K-means(S, K)

Entrada: conjunto de datos $S = \left\{X_1^{'}, X_2^{'}, X_3^{'}, X_N^{'}\right\}$ y el número de k clústers deseado
Salida: k clústeres
1) Los datos de k se eligen aleatoriamente de S como centros de agrupación de los k clústers.
2) Asigna cada dato al clúster cuyo centro tenga la distancia más corta.
3) Recalcula el centro de cada clúster en función de los datos del clúster.
4) Cuando los nuevos centros de clúster son los mismos que los centros de clúster obtenidos en la iteración anterior, genera los resultados de clúster; de lo contrario, repita desde el paso (2).

Es un buen método de agrupamiento para clasificar una gran cantidad de datos numéricos de alta dimensión. Los datos asignados por el algoritmo K-means a un mismo grupo son muy similares. Sin embargo, el algoritmo tradicional de K-means tiene las siguientes desventajas como se señalan en: [25–29].

- El número de grupos debe estar predeterminado. Sin embargo, en muchas aplicaciones, es difícil predeterminar el número de grupos.
- Los centros de grupo iniciales dados afectarán los resultados de agrupamiento.
- Es inadecuado para agrupar datos categóricos ya que describirlos por valor es difícil.
- Los resultados de la agrupación se verán afectados por datos ruidosos y valores atípicos.
- En general, las influencias de las características dimensionales son diferentes al calcular las distancias de un dato y los centros de agrupación. Por lo tanto, es necesario aplicar pesos apropiados para las características de los datos.
- Diferentes las unidades de medida de las características de los datos y las diferentes funciones de distancia adoptadas afectarán los resultados de la agrupación. Por lo tanto, se requiere normalización de atributos.

- Su complejidad temporal O (N × K × ITER) es muy alta, donde ITER es el número de iteraciones que realiza el algoritmo.

Algoritmo Kernel K-means

Cuando los clústeres no son linealmente separables, el algoritmo estándar K-means no ofrece resultados óptimos. Para superar esta limitación, el algoritmo clásico se ha extendido al Kernel K-means [30] el cual facilita un punto de partida inicial bastante utilizado en esquemas de agrupamiento de última generación [31]. Una encuesta reciente sobre los métodos de agrupación del núcleo se puede encontrar en [32]. En este artículo los autores presentan un estudio de los métodos de agrupación de Kernel y espectrales, dos enfoques capaces de producir hipersuperficies de separación no lineales entre agrupaciones. Los métodos de agrupamiento de kernel presentados son la versión del kernel de muchos algoritmos de agrupamiento clásicos, por ejemplo, K-means, y SOM. El agrupamiento espectral surge de conceptos en la teoría de grafos espectrales y el problema del agrupamiento se configura como un problema de corte de gráfico donde se debe optimizar una función objetivo apropiada. Se reporta una prueba explícita del hecho de que estos dos paradigmas tienen el mismo objetivo, ya que se ha demostrado que estos dos enfoques aparentemente diferentes tienen el mismo fundamento matemático. Además, los métodos de agrupamiento de kernel difusos se presentan como extensiones del algoritmo de agrupamiento de kernel K-means. Mientras que en [33] se presenta un estudio comparativo que respalda la superioridad de los métodos de agrupación del kernel, sobre los enfoques de clústeres más convencionales. El enfoque básico detrás del kernel es proyectar los datos en un espacio dimensional superior o incluso infinito, valiéndose de una función de kernel para calcular implícitamente el producto escalar de los vectores en el espacio del kernel, utilizando el vector de los atributos.

Sean $a_i, i = 1, ..., n$ el conjunto de datos y $X_i \in \mathbf{R}^d$, $i = 1, ..., n$ las $d-dimensiones$ de los atributos de los vectores. Si $\emptyset\ (X_i)$, $\emptyset\ (X_j)$ son las proyecciones del vector de los atributos (X_i) y (X_j) el espacio del kernel, entonces $k(X_i, X_j) = \emptyset(X_i)^T, \emptyset(X_j)$ es una función del kernel. Diferentes funciones del kernel corresponden a diferentes proyecciones finalmente, se puede medir la distancia euclidiana en el espacio del kernel mediante el producto escalar. Recientemente se ha demostrado que los algoritmos kernel K-means funcionan mejor que los algoritmos K-means convencionales en una clasificación no supervisada.

Esperanza-Maximización

Según [34] (EM) es uno de los algoritmos más comunes utilizados para la estimación de densidad de puntos de datos en un entorno no supervisado. El algoritmo se basa en encontrar las estimaciones de máxima probabilidad de los parámetros cuando el modelo de datos depende de ciertas variables latentes. En EM, los pasos alternos de Esperanza (E) y Maximización (M) se realizan de forma iterativa hasta que los resultados convergen. El paso E calcula una expectativa de la probabilidad al incluir las variables latentes como si se observaran, y un paso de maximización (M), que calcula las estimaciones de máxima verosimilitud de los parámetros al maximizar la probabilidad esperada encontrada en el último paso E. [25]. Los parámetros encontrados en el paso M se utilizan para comenzar otro paso E, y el proceso se repite hasta la convergencia. Para un modelo mixto, el algoritmo EM [35] se usa comúnmente

para un enfoque no paramétrico, los métodos de agrupación pueden basarse en una función objetiva de medidas de similitud o disimilitud, y estos pueden dividirse en métodos jerárquicos y particionales. Un método de agrupamiento jerárquico es un procedimiento para transformar un conjunto de datos en un diagrama, conocido como dendrograma, basado en la matriz de similitud o disimilaridad del conjunto de datos. La mayoría de los métodos de partición suponen que el conjunto de datos puede ser representado por prototipos de clúster finito con sus propias funciones objetivas. Por lo tanto, la definición de la disimilitud (o distancia) entre un punto y un prototipo es esencial para los métodos de partición.

Las propiedades de convergencia del algoritmo EM han sido bien discutidas en [36]. Los autores estudian dos aspectos de convergencia del algoritmo EM: (i) ¿encuentra el algoritmo EM un valor máximo local o estacionario de la función de probabilidad (datos incompletos)? (ii) ¿converge la secuencia de estimaciones de parámetros generadas por EM?. Se obtienen varios resultados de convergencia en condiciones que son aplicables a muchas situaciones prácticas. Dos casos especiales útiles son: (a) si la especificación de datos completos no observados se puede describir mediante una curva exponencial con espacio de parámetros compacto, todos los puntos límite de cualquier secuencia EM son puntos estacionarios de la función de verosimilitud; (b) si la función de verosimilitud es unimodal y se cumple una determinada condición de diferenciabilidad, entonces cualquier secuencia EM converge a la estimación única de máxima verosimilitud. Se incluye una lista de propiedades clave del algoritmo. Después en [36], Los autores consideraron más propiedades de convergencia del algoritmo EM para mezclas gaussianas. Desarrollaron la conexión matemática entre el algoritmo EM y enfoques basados en gradientes para el aprendizaje de máxima probabilidad de mezclas gaussianas finitas. Demostraron que el paso EM en el espacio de parámetros se obtiene del gradiente a través de una matriz de proyección P y proporcionaron una expresión explícita para la matriz. Luego analizaron la convergencia de EM en términos de propiedades especiales de P obtuvieron nuevos resultados analizando el efecto que tiene P sobre la superficie de probabilidad. Con base en estos resultados matemáticos, presentaron una discusión comparativa de las ventajas y desventajas de EM.

En [37], los autores establecieron la tasa de convergencia del algoritmo EM para mezclas gaussianas además de que el algoritmo EM es bastante sensible a la inicialización, en la que los números de clústeres deben darse a priori.

Clustering jerárquicos

Según [38], el agrupamiento jerárquico construye árboles de grupos de objetos, en los que dos grupos cualesquiera son disjuntos o uno incluye al otro. El grupo de todos los objetos es la raíz del árbol.

Dados n puntos en un espacio $d-dimencional$ el objetivo del clustering jerárquico es crear una secuencia de particiones anidadas, que se pueden visualizar a través de un árbol o jerarquía de grupos, también llamada dendrograma de grupo. Los

grupos en la jerarquía van desde el grano fino hasta el grano grueso, el nivel más bajo del árbol (las hojas) cada punto es su propio grupo, mientras que el nivel más alto (la raíz) consiste en que todos los puntos están en un grupo. Ambos se pueden considerar agrupaciones triviales. En un nivel intermedio, se pueden encontrar grupos significativos. Si el usuario suministra k, el número deseado de grupos se puede seleccionar así como también, el nivel en el que hay k grupos.

Existen dos enfoques algorítmicos principales para poblar grupos jerárquicos: aglomerativo y divisivo. Las estrategias de aglomeración funcionan de manera ascendente, iniciando con cada uno de los n puntos en un grupo separado y se fusionan repetidamente el par de grupos más similar hasta que todos los puntos sean miembros del mismo grupo. Las estrategias divisivas hacen exactamente lo contrario, trabajando de arriba hacia abajo. Empezando con todos los puntos en el mismo grupo y dividen recursivamente los grupos hasta que todos los puntos estén en clusteres separados. Según [39] los métodos de agrupación jerárquica son métodos de análisis de agrupaciones que crean una descomposición jerárquica de los conjuntos de datos dados. Los métodos de agrupamiento jerárquico se clasifican en divisivos (de arriba hacia abajo) y aglomerativos (de abajo hacia arriba), dependiendo de si la descomposición jerárquica se forma de abajo hacia arriba o de arriba hacia abajo. Un agrupamiento aglomerativo comienza con un grupo singleton (un objeto) y luego fusiona sucesivamente pares de grupos hasta que todos los grupos se fusionan en un grupo grande que contiene todos los objetos. El agrupamiento divisivo es un enfoque inverso del agrupamiento aglomerativo, comienza con un grupo de datos y luego particiona el grupo apropiado. Aunque el agrupamiento jerárquico es fácil de implementar y aplicable a cualquier tipo de atributo, son muy sensibles a valores atípicos y no funcionan con datos faltantes. Además, las semillas iniciales tienen un fuerte impacto en los resultados finales (involucrando muchas decisiones arbitrarias).

Clustering jerárquico aglomerativo

Según [40], hay varias formas posibles de realizar agrupaciones jerárquicas aglomerativas. Sin embargo, generalmente siguen los siguientes pasos principales:

1. Cálculo de la matriz de proximidad para los grupos iniciales que son el resultado del proceso de K-means anterior.

2. Búsqueda de la distancia mínima en la matriz.

3. Combinado de los dos grupos con la distancia mínima.

4. Actualización de la matriz de proximidad calculando las distancias entre el nuevo grupo y los otros grupos.

5. Repetición de los tres pasos anteriores si queda más de un grupo.

En la agrupación jerárquica aglomerativa, se inicia con cada uno de los n puntos en un grupo separado y se combinan repetidamente los dos grupos más cercanos hasta que todos los puntos sean miembros del mismo clúster. Dado un conjunto de grupos $C = \{C_1, C_2, ..., C_n\}$, se encuentra el par de grupos más cercano C_i y C_j para

fusionarlos en un nuevo clúster $C_{ij} = C_i \cup C_j$, luego se actualiza el conjunto de clústeres eliminando a C_i y a C_j, para incluir a C_{ij} como sigue $C = C\{C_i, C_j\} \cup \{C_{ij}\}$. Esta operación se repite hasta que C contenga solo un clúster. Tomando en cuenta que la cantidad de los grupos disminuyen en uno en cada paso, este proceso da como resultado una secuencia de n clústeres anidados. Si se especifica, es posible detener el proceso de fusión cuando hay exactamente k grupos restantes.

Distancia entre clústeres
Según [21], el paso principal en el algoritmo es determinar el par de grupos más cercano. Las distancias entre grupos es el cálculo de la distancia entre dos puntos, que normalmente se calcula utilizando la distancia euclidiana definida como:

$$\delta(X, Y) = \|X - Y\|_2 \left(\sum_{i=1}^{d} (X_i - Y_j)^2 \right)^{1/2} \tag{2.4}$$

Método de enlace simple o vecino más cercano
En este método de acuerdo a [41], dados dos grupos C_i y C_j, la distancia entre ellos, se denota $\delta(C_i, C_j)$ y está definida como la distancia mínima entre un punto en C_i y un punto en C_j de manera tal que:

$$\delta(C_i, C_j) = \text{mín}\{\delta(X, Y) \mid X \in C_i, Y \in C_j\} \tag{2.5}$$

En enlace simple o vecino más cercano se parte de la premisa de que si es elegida la distancia mínima entre puntos en los dos grupos y son conectados, solo existiría un enlace entre esos grupos porque todos los demás pares de puntos estarían más lejos.

Método de enlace completo o vecino más lejano
Según [21], la distancia entre dos grupos se define como la distancia máxima entre un punto en C_i y un punto en C_j

$$\delta(C_i, C_j) = \text{máx}\{\delta(X, Y) \mid X \in C_i, Y \in C_j\} \tag{2.6}$$

En el método de enlace completo, la similitud entre los dos grupos está dada por los individuos de cada grupo que se parecen menos. Este método generalmente conduce a grupos compactos y discretos con valores de similitud relativamente pequeños.

Método de promedio grupal
Tal como se describe en [21]. La distancia entre dos clústeres está definida como la distancia promedio por pares entre puntos en C_i y C_j :

$$\delta(C_i, C_j) = \frac{\sum X \in C_i \sum Y \in C_j \delta(X, Y)}{n_i . n_j} \tag{2.7}$$

donde $n_i = |C_i|$ que denota el número de puntos en un clúster C_i.

Método de la media o de centroide par-grupo ponderado (WPGMC)
La distancia entre dos grupos se define como la distancia entre las medias o centroides de los dos grupos, fue propuesto por [42] para evitar la contribución desigual de los centroides de diferentes agrupaciones en la formación de un nuevo centroide (grupo) candidato. En [43] se concluye que si un grupo es considerado pequeño en términos de número de individuos, su contribución a la formación de un el grupo con un nuevo centroide no será diferente de la media (centroide) del grupo con el mayor número de individuos. La media se puede calcular utilizando la siguiente ecuación:

$$d_{(i,j)k} = \frac{d_{ik} + d_{jk}}{2} - \frac{1}{4}d_{ij} \tag{2.8}$$

Método de varianza mínima de Ward
Para [44], en la formación inicial del clúster con el método de variación mínima, los individuos proporcionan la suma más baja de cuadrados de desviaciones, se asume que cualquier etapa puede cuantificarse por la relación entre la suma de los cuadrados de las desviaciones dentro del grupo en formación y la suma total de los cuadrados de las desviaciones. La suma de los cuadrados de las desviaciones se calcula considerando solo las accesiones dentro del grupo en formación, y la suma de cuadrados de las desviaciones totales se calcula considerando todos los individuos disponibles para análisis de conglomerados. La distancia entre dos grupos se define como el aumento en la suma de los errores al cuadrado (SSE) cuando se fusionan los dos clústeres. El SSE para un clúster C_i viene dado como:

$$SSE_i = \sum_{X \in C_i} \|X - \mu_i\|^2 \tag{2.9}$$

El SSE para un clustering $C = \{C_1, ..., C_m\}$ viene dado como:

$$SSE = \sum_{i=1}^{m} SSE_i = \sum_{i=1}^{m} \sum_{X \in C_i} \|X - \mu_i\|^2 \tag{2.10}$$

La media de Ward define la distancia entre dos grupos C_i y C_j como el cambio en la red en el valor SSE cuando se fusiona C_i y C_j en C_ij, está dada como:

$$\delta(C_i, C_j) = \Delta SSE_{ij} - SSE_i - SSE_j \tag{2.11}$$

Clustering basados en densidad

Para [21], los métodos de clustering basados en densidad son utilizados en grupos no convexos, donde los otros métodos tienen problemas para encontrar los verdaderos grupos, ya que dos puntos de diferentes grupos pueden estar más cerca de dos puntos

en el mismo grupo. Los métodos basados en la densidad son capaces de encontrar estos grupos no convexos.

Algoritmo DBSCAN

Utiliza la densidad local de puntos para determinar las agrupaciones, en lugar de usar solo la distancia entre puntos como se describe en [45] es un método de agrupamiento de datos espaciales basado en la densidad, propuesto por [46] puede encontrar un grupo con cualquier forma en una condición de densidad.

DBSCAN también se puede considerar como un buscador de los componentes conectados en un gráfico, donde los vértices corresponden a los puntos centrales en el conjunto de datos, y allí existe un borde entre dos vértices si la distancia entre ellos es menor que $\in$ es decir, cada uno de ellos está en el $\in -vecindario$ del otro punto. Los componentes conectados de este gráfico corresponden a los puntos centrales de cada clúster. A continuación, cada punto central incorpora en su clúster cualquier punto que se encuentre en el borde del vecindario.

La limitación de DBSCAN radica en su sensibilidad frente a la elección de $\in$, en particular si los clústeres tienen diferentes densidades. Si $\in$ es demasiado pequeño, los clústeres más dispersos se clasificarán como ruido. Si $\in$ es demasiado grande, los clústeres más densos pueden fusionarse.

Kernel de estimación de densidad (KDE)

El objetivo de la estimación de densidad es determinar la densidad desconocida, encontrando las regiones densas de los puntos, que a su vez pueden usarse para el clustering. La estimación de la densidad del núcleo es una técnica no paramétrica que no supone modelo de probabilidad fija de los clústeres, como en el caso de K-medias o en el algoritmo EM. En cambio, trata de inferir directamente la densidad de probabilidad en cada punto del conjunto de datos. De acuerdo a [47]. KDE se basa principalmente en un conjunto de observaciones y variables aleatorias de una función de distribución desconocida para estimar su función de densidad. El método KDE no depende del conocimiento previo de la distribución de datos, ni adjunta ninguna hipótesis a la distribución de datos, por lo que es una estimación que solo utiliza los datos de la muestra en sí. Suponiendo que $X_i(i = 1, 2, ..., n)$ es una muestra tomada de una distribución continua. El KDE de la función de densidad $f(x)$ en cualquier punto x define como:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} \left(\frac{X_i - x}{h} \right) \tag{2.12}$$

donde n es el número de ejemplos.

Clustering basados en cuadrícula

De acuerdo a [48], este método de agrupación adopta un enfoque basado en el espacio de incrustación en las celdas, independientemente de la distribución de los objetos de entrada. El enfoque de agrupamiento basado en cuadrícula utiliza una estructura de datos de cuadrícula de resolución múltiple. Eso cuantifica el espacio del objeto en

un número finito de celdas que forman una estructura de cuadrícula en el que se realizan todas las operaciones para la agrupación en clústers. La principal ventaja del enfoque es su rápido tiempo de procesamiento, que generalmente es independiente de la cantidad de datos.

Cuadrícula de información estadística (STING)
Conforme a [48]. STING es una técnica de agrupación multi-resolución basada en cuadrícula en la que el área espacial de los objetos de entrada se divide en celdas rectangulares. El espacio se puede dividir tanto de forma jerárquica como de manera recursiva. Varios niveles de celdas rectangulares corresponden a diferentes niveles de resolución que forman una estructura jerárquica. Cada celda en un nivel alto se divide para formar varias celdas en el siguiente nivel inferior. Es de hacer notar que la información estadística con respecto a los atributos en cada celda de la cuadrícula, como la media, el máximo y el mínimo, se calcula previamente y se almacena como parámetros estadísticos. Estos parámetros estadísticos son útiles para el procesamiento de consultas y para otras tareas de análisis de datos.

Una propiedad interesante de STING es que se acerca al resultado de agrupamiento de DBSCAN si la granularidad se acerca a 0 (es decir, hacia datos de muy bajo nivel). Lo que implica que, utilizando la información de conteo y tamaño de celda, se pueden identificar grupos densos aproximadamente usando STING. Por lo tanto, STING también puede considerarse como un método de agrupamiento basado en densidad.

Clustering en búsqueda (CLIQUE)
Para [49], CLIQUE es un método simple basado en cuadrícula para encontrar la densidad de agrupaciones en subespacios. CLIQUE divide cada dimensión en intervalos de no superposición, dividiendo así todo el espacio de incrustación de los objetos de datos en celdas. Utiliza un umbral de densidad para identificar células densas y dispersas. Una celda es densa si el número de objetos asignados excede el umbral de densidad.

La estrategia principal detrás de CLIQUE para identificar un espacio de búsqueda candidato es utilizar la monotonicidad de células densas con respecto a la dimensionalidad. Esto se basa en la a priori, propiedad utilizada en minería frecuente de patrones y reglas de asociación.

Validación de clustering

Existen muchos métodos de clustering diferentes y dependiendo del tipo de clustering buscado y de las características de los datos. (Han et al., 2012) la validación y evaluación del clúster abarca tres principales tareas las cuales son:

- **Evaluar la tendencia del clúster:** en esta tarea, para un conjunto de datos dado, se evalúa si existe una estructura no aleatoria en los datos. Aplicar a ciegas un método de agrupamiento en un conjunto de datos devolverá grupos, sin embargo, los grupos extraídos pueden ser engañosos. El análisis de un

conjunto de datos es significativo solo cuando hay una estructura no aleatoria en ellos.

- **Determinación del número de clústeres en un conjunto de datos:** algunos algoritmos, como K-means, requieren el número de clústeres en un conjunto de datos como parámetro. Por otra parte, el número de grupos pueden considerarse como una estadística resumen interesante e importante de un conjunto de datos por lo tanto, es deseable estimar este número antes de realizar un clustering.

- **Medición de la calidad en el clúster:** después de aplicar un método de clustering en un conjunto de datos, es necesario evaluar qué tan buenos son los clústeres resultantes. Se pueden usar varias medidas. Algunos métodos miden que tan bien se ajustan los grupos al conjunto de datos, mientras que otros miden que tan bien coinciden los grupos con la verdad básica, si dicha verdad está disponible. También hay medidas que puntúan agrupaciones y, por lo tanto, pueden comparar dos clústeres en el mismo conjunto de datos.

De igual manera [21] propone una serie de medidas estadísticas para la validación de clústeres divididas en tres tipos principales:

- **Externas:** las medidas de validación externas emplean criterios que no son inherentes al conjunto de datos que puede ser tanto en forma de conocimiento previo como especificado por expertos.

- **Internas:** las medidas de validación internas emplean criterios derivados de los datos, se puede usar distancias intraclúster e interclúster para obtener medidas de densidad del clúster.

- **Relativas:** las medidas de validación relativas tienen como objetivo comparar directamente diferentes clústeres, generalmente aquellos obtenidos a través de diferentes configuraciones de parámetros para el mismo algoritmo.

2.2. Selección de atributos

La selección de atributos se define en [50] como el proceso de eliminar atributos de la base de datos que son irrelevantes para la tarea a realizar. Básicamente, el proceso de selección de atributos consta de los siguientes cuatro pasos: a) un procedimiento de generación, b) una función de evaluación, c) un criterio de parada y d) un procedimiento de validación. Entre ellos, el procedimiento de generación y la función de evaluación son los dos pasos principales. El procedimiento de generación es un proceso de búsqueda que genera subconjuntos de características para su evaluación. Varias estrategias de búsqueda en este procedimiento incluyen estrategias completas, heurísticas y aleatorias. La función de evaluación tiene como objetivo medir la capacidad de discriminación de un subconjunto de atributos para distinguir diferentes etiquetas de clase. Además, en [50] dividieron las funciones de evaluación en cinco categorías: a) funciones de distancia, b) información, c) dependencia, d) consistencia

y e) tasa de error del clasificador.

La selección de atributos, ha demostrado ser eficaz y eficiente en el pre- procesamiento de datos [51] para diversos problemas de minería de datos y aprendizaje automático, eliminando del conjunto de datos aquellos atributos irrelevantes para el análisis. Los objetivos de la selección de atributos son: construir modelos más simples y comprensibles, mejorar el rendimiento en la minería de datos, disminuir el tiempo de cálculo en los diferentes procesos, reducir el tamaño de los conjuntos de datos para optimizarlos y hacerlos más comprensibles. La reciente proliferación de big data ha presentado algunos desafíos y oportunidades sustanciales para la selección de atributos lo que hace necesario su estudio y aplicación. La selección de atributos puede clasificarse ampliamente como supervisada, métodos no supervisados y semi-supervisados dependiendo del etiquetado en el conjunto de datos de entrenamiento. La selección de atributos supervisadas generalmente está diseñada para problemas de clasificación o de regresión. Su objetivo es seleccionar un subconjunto de atributos que puedan discriminar muestras de diferentes clases (clasificación) o para aproximar los objetivos de regresión. Con la información de supervisión, la relevancia de los atributos se evalúa generalmente a través de correlación con las etiquetas de clase o el objetivo de regresión. La selección de atributos no supervisada generalmente está diseñada para problemas de clustering. Como adquirir los datos etiquetados son particularmente costosos tanto en tiempo como en esfuerzo, la selección de atributos sin supervisión de manera reciente ha ganado especial atención. Sin información de etiqueta para evaluar la importancia de los atributos, los métodos de selección de atributos no supervisados buscan criterios alternativos para definir la relevancia del atributo. es un aspecto clave del procesamiento de datos y los métodos de aprendizaje automático. Es una tarea primordial en el proceso de descubrimiento del conocimiento. Este proceso a menudo enfrenta el problema de la alta dimensionalidad de los datos, características ruidosas y no relevantes que aumentan exponencialmente los requisitos de memoria y tiempo para procesar los datos [52].

Clasificación de métodos de selección de atributos

Según [53], los métodos de selección de atributos generalmente se clasifican en: envoltura, filtro e integrado.

Métodos de envoltura

Los métodos de envoltura [54] usan un algoritmo de aprendizaje predeterminado para determinar la calidad de los atributos seleccionados según una evaluación métrica. Aunque los métodos de envoltura generalmente logran una mejor precisión de clasificación que los métodos de filtro, una desventaja importante es que requieren mucho tiempo de procesamiento. Para un conjunto de datos con N características, los métodos de selección de variables basados en evaluación de subconjuntos de características evalúan aproximadamente la calidad de los subconjuntos de características $O(N^2)$ cuando se utiliza el esquema de selección secuencial [55], e incluso los métodos de contenedor incrementales manejan un número lineal o subcuadrático

de subconjuntos de características candidatos [56, 57]. Una cantidad tan grande de evaluaciones de envoltura requeriría una gran cantidad de tiempo de CPU cuando funcionan con arreglos de datos de alta dimensión.

Los modelos de envoltura usan un clasificador para medir la calidad de un subconjunto de características, generalmente obtienen bajas tasas de error de clasificación debido a la interacción específica entre el clasificador y el conjunto de entrenamiento. Obviamente, enumerar todas las combinaciones de atributos y evaluar sus cualidades garantiza a su vez obtener la óptima globalmente, pero a costa de una alta complejidad computacional que crece exponencialmente con el número de características [54]. En la práctica, tal alta complejidad temporal a menudo es inaceptable, particularmente para los perfiles de expresión génica con alta dimensionalidad. Para acelerar este proceso, los investigadores han propuesto varias estrategias de búsqueda para generar candidatos. En la selección de atributos, los esquemas de búsqueda comúnmente usados incluyen, entre otros, selección secuencial hacia adelante (SFS), selección secuencial hacia atrás (SBS), búsqueda flotante secuencial, búsqueda bidireccional, búsqueda aleatoria y búsqueda heurística [54]. Entre estas estrategias de búsqueda, SFS logra una mejor compensación entre la calidad del subconjunto de características obtenido y la complejidad computacional. Específicamente, al inicializar el subconjunto de características seleccionado para que esté vacío, SFS selecciona la primera característica que es más relevante para la clase de destino y luego busca la siguiente característica candidata que más reduce la tasa de error de clasificación y continúa con el procedimiento hasta que no quede ninguna característica candidata o no haya más mejoras en el rendimiento de la clasificación. Si finalmente se seleccionan K características del total de N características, los métodos de envoltura con SFS evalúan aproximadamente $O(KN)$ subconjuntos de características candidatas.

Métodos de filtro

Los modelos de filtrado son independientes de cualquier algoritmo de aprendizaje, suelen ser más eficientes computacionalmente que los métodos de envoltura. Sin embargo, debido a la falta de un algoritmo de aprendizaje que guíe la fase de selección de atributos estos pueden no ser óptimos para los algoritmos de aprendizaje de destino [51]. Los métodos de filtro aplican medidas estadísticas para evaluar los atributos, como por ejemplo: correlación, consistencia, ganancia de información, etc., se basan en las características generales de los datos para evaluar y seleccionar subconjuntos de características sin involucrar ningún algoritmo de minería de datos.

Como se destaca en [58], en este trabajo los autores introducen el concepto de correlación predominante y proponen un método de filtrado rápido que puede identificar características relevantes, así como abundancia entre características relevantes sin análisis de correlación por pares. La eficiencia y la efectividad del método es verificado a través de extensas comparaciones con otros métodos que utilizan datos del mundo real de alta dimensionalidad.

Métodos integrados

Según [59]. Los métodos integrados son una combinación entre el método de filtro y

el método de envoltura que incrustan la selección de atributos en el aprendizaje de los modelos. Así heredan los méritos de envoltura y métodos de filtrado, incluyen las interacciones con el algoritmo de aprendizaje y son mucho más eficientes que los métodos de envoltura ya que no necesitan evaluar conjuntos de atributos de forma iterativa. Los métodos embebidos más utilizados son de regularización que apuntan a ajustarse a un modelo de aprendizaje minimizando los errores de ajuste y forzando los coeficientes de características a ser pequeños, posteriormente ambos, el modelo de regularización y los conjuntos de atributos seleccionados se devuelven como resultados finales.

Selección de atributos evolutiva multi-objetivo

El uso de algoritmo genéticos para la selección de atributos se propuso en [60]. El primer enfoque evolutivo basado en optimización multi-objetivo para selección de atributos fue propuesto en [61]. Los enfoques multi-objetivo se utilizan comúnmente en optimización de múltiples objetivos, buscando simultáneamente múltiples soluciones óptimas. Estos algoritmos pueden encontrar un conjunto de soluciones óptimas en su población final con una sola ejecución, para posteriormente seleccionar la más satisfactoria aplicando un criterio de preferencia. Los algoritmos evolutivos multi-objetivo [62] han demostrado ser muy eficaces en la búsqueda de soluciones óptimas para múltiples problemas. Un problema de optimización multi-objetivo se formula como una tupla de n funciones objetivo de minimización/maximización. La selección de atributos, donde se debe maximizar la precisión de un clasificador y minimizar el número de atributos, es un ejemplo de tal situación [63]. Los autores proponen el algoritmo ENORA para resolver el problema multi-objetivo mediante un método de envoltura, el cual es comparado con el conocido algoritmo NSGA-II [64]. En [63] muestra también una revisión de algoritmos evolutivos multi-objetivo para selección de atributos.

2.3. Clasificación de texto

Las tecnologías de Internet se han convertido en las formas más importantes de difusión de información relacionada con todos los temas, lo que implica que la mayor parte del conocimiento humano esté disponible en la web muy rápidamente. La clasificación de texto (CT) es una de las áreas más importantes de la minería de datos, se preocupa por el uso eficaz y eficiente del procesamiento y clasificación de textos en lenguaje natural para maximizar la utilidad de la información extraída de la web. La mayoría de los investigadores, definen CT como un proceso en el que los documentos se predicen en una de las clases predefinidas en función de su contenido. Este proceso tiene numerosos usos, como la clasificación de páginas web y la clasificación de correo electrónico, que se vuelven cada vez más vitales en la sociedad actual orientada a la información. Según [65], la creación de una CT automatizada es uno de los problemas más importantes en las áreas de minería de datos y aprendizaje automático. A diferencia de la clasificación manual, que requiere alta precisión y consumo de tiempo, la CT automatizada permite que el proceso de clasificación

sea más eficiente y rápido.

Los problemas de clasificación de textos se han estudiado y abordado de manera amplia en muchas aplicaciones reales durante las últimas décadas como se destaca en: [66], en este artículo, los autores utilizan un modelo de clasificación de texto híbrido novedoso basado en deep belief network and softmax regression. Para resolver el escaso problema de cálculo matricial de alta dimensión de los datos de los textos. Después de la extracción de características con DBN, se emplea la regresión softmax para clasificar el texto en el espacio de características aprendidas. En los procedimientos de preentrenamiento, primero se entrena la deep belief network and softmax regression, respectivamente. Luego, en la etapa de ajuste fino, se transforman en un todo coherente y los parámetros del sistema se optimizan con el algoritmo de memoria limitada de Broyden-Fletcher-Goldfarb-Shanno. Los resultados experimentales en el corpus Reuters-21.578 y 20-Newsgroup muestran que el modelo propuesto puede converger en la etapa de ajuste fino y funcionar significativamente mejor que los algoritmos clásicos, como SVM y K-NN.

El número cada vez mayor de documentos que se producen anualmente exige que se mejoren constantemente los métodos de procesamiento de la información para buscar, recuperar y organizar el texto [67]. Un elemento central de estos métodos de procesamiento de información es la clasificación de documentos, que se ha convertido en una aplicación importante para el aprendizaje supervisado. Recientemente, el rendimiento de estos clasificadores tradicionales se ha degradado a medida que ha aumentado el número de documentos. Esto se debe a que junto con este crecimiento en el número de documentos ha venido un aumento en el número de categorías en [68]. Este artículo aborda este problema de manera diferente a los métodos actuales de clasificación de documentos que ven el problema como una clasificación de clases múltiples. En su lugar, se realiza una clasificación jerárquica utilizando un enfoque denominado Hierarchical Deep Learning for Text classification (HDLTex). HDLTex emplea pilas de arquitecturas de aprendizaje profundo para proporcionar una comprensión especializada en cada nivel de la jerarquía de documentos.

En [69] se estudian los enfoques para la clasificación de texto, donde se señalan dos modelos probabilísticos de primer orden diferentes utilizados para la clasificación de texto, los cuales hacen uso de Naive Bayes: en primer lugar, se encuentra el modelo de Bernoulli multivariable, es decir, una red bayesiana sin dependencias entre palabras y características de palabras binarias. Y en segundo lugar se analiza el modelo multinomial, es decir, un modelo de lenguaje uni-grama con recuentos de palabras enteras. En este artículo los autores describen las diferencias y los detalles de estos dos modelos, al comparar empíricamente su desempeño de clasificación en cinco corpus de texto. Encontrando que el Bernoulli multivariable funciona bien con vocabulario de tamaño pequeño, pero que el multinomial funciona normalmente incluso mejor con tamaños de vocabulario más grandes, lo que proporciona una reducción promedio del 27 % en el error sobre el modelo Bernoulli multivariable en cualquier tamaño de vocabulario.

En [70], los autores aplicaron el enfoque de clasificación asociativa (AC) para clasificar artículos médicos árabes. El enfoque se logra en cuatro pasos principales: preprocesamiento de artículos, generación de reglas, clasificación e inferencia de nuevos artículos árabes no vistos. En el primer paso, se implementaron varios métodos de preprocesamiento. Estos métodos implican: tokenización, eliminación de palabras vacías, derivación y selección de características. Durante el segundo paso, descubren todas las reglas frecuentes. Después de eso, se aplican métodos de ordenación y poda de reglas para ordenar y hacer la clasificación más eficiente. El último paso consiste en adivinar el valor de clase de los artículos médicos árabes invisibles.

Aplicaciones de la clasificación de texto

En la historia más reciente del aprendizaje automático y la inteligencia artificial, las técnicas de clasificación de texto han sido utilizadas principalmente para sistemas de recuperación de información. Sin embargo, a medida que han surgido avances tecnológicos a lo largo del tiempo, la clasificación de textos y la categorización de documentos han sido utilizadas globalmente en muchos dominios como la medicina, las ciencias sociales, la salud, la psicología, el derecho, la ingeniería, etc., en esta sección, se destacan varios dominios que hacen uso de las técnicas de clasificación de textos.

Recuperación de información

La recuperación de información consiste en encontrar documentos de datos no estructurados que satisfagan una necesidad de información dentro de grandes colecciones de documentos [71]. Con el rápido crecimiento de la información en línea, especialmente en formato de texto, la clasificación de texto se ha convertido en una técnica importante para gestionar este tipo de datos [72]. Algunos de los métodos importantes utilizados en esta área son: Naive Bayes, SVM, árbol de decisión, C4.5, K-NN e IBK [73]. Una de las aplicaciones más desafiantes para el procesamiento de conjuntos de datos de documentos y texto es la aplicación de métodos de categorización de documentos para la recuperación de información [74].

Análisis de sentimientos

El análisis de sentimientos es un enfoque computacional para identificar opiniones, sentimientos y subjetividad en el texto [75]. Los métodos de clasificación de sentimientos clasifican un documento asociado con una opinión en positivo o negativo. Se asume que el documento d expresa una opinión sobre una sola entidad e las opiniones se forman a través de un único titular de opinión h. La clasificación Naive Bayes y SVM son algunos de los métodos de aprendizaje supervisado más populares que se han utilizado para la clasificación de sentimientos [76]. En las técnicas de clasificación de sentimientos se han utilizado características tales como términos y su frecuencia respectiva, parte del discurso, palabras y frases de opinión, negaciones y dependencia sintáctica.

Filtrado de Información
El filtrado de información se refiere a la selección de información relevante o al rechazo de información irrelevante de un flujo de datos entrantes. Los sistemas de filtrado de información se utilizan normalmente para medir y pronosticar los intereses a largo plazo de los usuarios [77]. Los modelos probabilísticos, como la red de inferencia bayesiana, se utilizan comúnmente en sistemas de filtrado de información. Las redes de inferencia bayesianas emplean inferencia recursiva para propagar valores a través de la red de inferencia y devolver los documentos con la clasificación más alta [74].

En [78] se utilizó un modelo de espacio vectorial con refinamiento iterativo para la tarea de filtrado, en su investigación los autores proporcionan una descripción general de los principales desarrollos y enfoques en los campos del filtrado de información y la recuperación de información, analizando las ventajas y los defectos asociados con los sistemas existentes en la actualidad. Entre sus resultados estuvo la incorporación de una función de búsqueda web, que utiliza índices web existentes junto con un mecanismo de filtrado más inteligente para localizar y recuperar información basada en web.

Sistemas de recomendaciones
Los sistemas de recomendación basados en contenido sugieren elementos a los usuarios según la descripción de un elemento y un perfil de los intereses del usuario [79].

El perfil de un usuario se puede aprender a partir de los comentarios de los usuarios (historial de las consultas de búsqueda o autoinformes) sobre los elementos, así como las características autoexplicativas (filtro o condiciones de las consultas) en el perfil del usuario. De esta manera, la entrada a dichos sistemas de recomendación puede ser semiestructurada de modo que algunos atributos se extraigan del campo de texto libre mientras que otros se especifiquen directamente [80]. Se han utilizado muchos tipos diferentes de métodos de clasificación de texto, como árboles de decisión, métodos del vecino más cercano, algoritmo de Rocchio, clasificadores lineales, métodos probabilísticos y Naive Bayes, para modelar las preferencias del usuario.

Resumen de documentos
La clasificación de texto es utilizada para resúmenes de documentos en los que el resumen de un documento puede emplear palabras o frases que no aparezcan en el documento original [81]. En múltiples documentos también es necesario un resumen debido al rápido aumento de la información en línea [82]. Por tanto, los investigadores se centran en esta tarea utilizando la clasificación de texto para extraer características importantes de uno o varios documentos.

Clasificación de texto para el apoyo

Medicina
La mayor parte de la información textual en el ámbito médico se presenta de forma no estructurada o narrativa con términos ambiguos y errores tipográficos. Esta

información debe estar disponible instantáneamente a lo largo de los encuentros médico-paciente en las diferentes etapas de diagnóstico y tratamiento [83]. La codificación médica, que consiste en asignar diagnósticos médicos a valores de clase específicos obtenidos de un gran conjunto de categorías, es un área de aplicaciones sanitarias donde las técnicas de clasificación de textos pueden ser muy valiosas. En otra investigación, [84]. Introdujeron Patient2Vec para aprender una representación profunda e interpretable de los datos de la historia clínica electrónica longitudinal (HCE) que se personaliza para cada paciente. Patient2Vec es una técnica novedosa de incrustación de características de conjuntos de datos de texto que puede aprender una representación profunda interpretable personalizada de datos EHR basada en redes neuronales recurrentes y el mecanismo de atención. La clasificación de texto también se ha aplicado en el desarrollo de encabezamientos de materias médicas (MeSH) y Ontología genética (GO) [85].

Ciencias sociales
La clasificación de textos y la categorización de documentos se ha aplicado cada vez más para comprender el comportamiento humano en las últimas décadas [86]. Los recientes esfuerzos impulsados por datos en la investigación del comportamiento humano se han centrado en el lenguaje minero contenido en notas informales y conjuntos de datos de texto, incluido el servicio de mensajes cortos (SMS), notas clínicas, redes sociales, etc. [87]. Estos estudios se han centrado principalmente en el uso de enfoques basados en la frecuencia de aparición de palabras (es decir, la frecuencia con la que aparece una palabra en un documento) o características basadas en el recuento de palabras de la investigación lingüística (LIWC) [88], un léxico de categorías de palabras bien validado con relevancia psicológica [89].

Negocios y marketing
Las empresas y organizaciones rentables utilizan progresivamente las redes sociales con fines de marketing [90]. La apertura de la minería de medios sociales como Facebook, Twitter, etc. es el principal objetivo de las empresas para aumentar rápidamente sus beneficios [91]. La clasificación de textos y documentos es una herramienta poderosa para que las empresas encuentren a sus clientes más fácilmente.

Ámbito legal
Las instituciones gubernamentales han generado enormes volúmenes de información y documentos legales. Recuperar esta información y clasificarla automáticamente no solo puede ayudar a los abogados sino también a sus clientes. En muchos países, la ley se deriva de cinco fuentes: derecho constitucional, tratados, reglamentos administrativos y derecho consuetudinario. Cada año se crean muchos documentos legales nuevos. La categorización de estos documentos es el principal desafío para la comunidad de abogados [92].

Técnicas de preprocesamiento de texto

La mayoría de los conjuntos de datos de texto y documentos contienen muchas palabras innecesarias, como palabras vacías, errores ortográficos, términos locales, etc.

En muchos algoritmos, especialmente los algoritmos de aprendizaje estadístico y probabilístico, el ruido y las características innecesarias pueden tener efectos adversos en el rendimiento del sistema [93].

De manera general los textos y los documentos son conjuntos de datos no estructurados, estas secuencias de texto no estructuradas deben convertirse en un conjunto de características estructurado cuando es usado un modelo matemático como parte de un clasificador. De manera que, en primer lugar: los datos deben limpiarse para omitir caracteres o palabras innecesarias. Una vez que se han limpiado los datos, las técnicas formales de extracción de características pueden ser aplicadas.

Según [94]. Antes de casi cualquier procesamiento de un texto en lenguaje natural, el texto debe normalizarse. El paso de normalización tiene como objetivo limpiar los datos eliminando elementos innecesarios y ruidosos. Datos, como números, símbolos, etiquetas de código y caracteres especiales, el filtrado de ruido es una tarea esencial del tokenizador. Asimismo, los resultados obtenidos, como los fragmentos contextuales proporcionados como entrada, que pueden incluir nombres de archivo; URL; caracteres que delimitan partes de documentos completos, como los puntos suspensivos (@, %, &, etc.) y otros símbolos cuyos significados no son evidentes.

Un tokenizador confiable debe ser capaz de reconocer y deshacerse de este tipo de ruido mientras crea una secuencia de fichas. Este paso es necesario para llevar a cabo la representación y un preprocesamiento aceptables de los datos [95].

En esta sección, se explican brevemente algunas de las técnicas más utilizadas para la limpieza y el preprocesamiento de conjuntos de datos de texto

Reemplazo de valores faltantes
Uno de los problemas de calidad de datos más comunes es que falten algunos valores de atributos [93] existen varios métodos de mitigación diferentes para abordar este problema. El primer paso para gestionar la falta de valores es comprender la razón detrás de por qué faltan los valores. El rastreo de datos desde la fuente de datos puede llevar a identificar problemas sistémicos en captura de datos, errores en la transformación de datos o puede haber un fenómeno que el usuario aún no comprende. Conocer la fuente de un valor perdido a menudo guiará qué metodología de mitigación utilizar. Es posible sustituir el valor faltante con una variedad de datos artificiales para gestionar el problema con un impacto marginal en los pasos posteriores de la minería de datos. Los valores se pueden reemplazar con un valor derivado del conjunto de datos (media o valor mínimo o máximo, según las características del atributo). Este método es útil si los valores perdidos ocurren de forma completamente aleatoria y la frecuencia de aparición es bastante rara. De no ser así, la distribución del atributo al que le faltan datos se distorsionará. Alternativamente, para construir el modelo, es posible ignorar todos los registros de datos con valor perdido o registros con datos de mala calidad. Este método permite reducir el tamaño del conjunto de datos.

Extracción de características
Para [96] la clasificación de texto comienza a menudo mirando los documentos y encontrando las palabras significativas en ellos de manera tal, que la primera suposición podría ser que las palabras que aparecen con más frecuencia en un documento son las más significativas. Sin embargo, esta afirmación es exactamente opuesta a la verdad. Las palabras más frecuentes serán seguramente las palabras comunes como "el" o "y", que ayudan a construir ideas, pero no tienen ningún significado en sí mismas. De hecho, los cientos de palabras más comunes en inglés (llamadas stop words) a menudo se eliminan de los documentos antes de cualquier intento de clasificarlas.

La medida formal de cuán concentradas en relativamente pocos documentos están las apariciones de una palabra dada se llama TF-IDF (Term Frequency times In-verse Document Frequency). Normalmente se calcula de la siguiente manera: Teniendo una colección de N documentos. Definida f_{ij} para ser la frecuencia (número de ocurrencias) del término (palabra) i en el documento j luego, quedaría definido el término frecuencia TF_{ij} de la manera siguiente:

$$TF_{ij} = \frac{f_{ij}}{\max_k f_{ij}} \tag{2.13}$$

Es decir, el termino de frecuencia un documento se normaliza dividiéndolo por el número máximo de apariciones de cualquier término (quizás excluyendo las palabras vacías) en el mismo documento. Por lo tanto, el término más frecuente en el documento j obtiene un TF de 1 y otros términos obtienen fracciones como su frecuencia de término para este documento.

El IDF para un término se define como sigue: Suponiendo que el término i aparece en n_i de los N documentos de la colección. Entonces:

$$IDF_i = \log_2 \frac{N}{n_i} \tag{2.14}$$

La puntuación final $TF.IDF$ para el documento se define entonces como: $TF_{ij} \times IDF_i$. De tal manera que los términos con la puntuación TF.IDF más alta suelen ser los términos que mejor caracterizan el tema del documento.

Tokenización
Se define como la tarea de dividir un flujo de texto en palabras, frases, símbolos u otros elementos significativos llamados tokens [97]. El objetivo principal de la tokenización es el análisis de las palabras en una oración y su uso principal es la identificación de palabras clave significativas. [98]. Varios estudios establecen la tokenización como uno de los primeros pasos durante la preparación del texto:

En [99], los autores abordan la importancia y la complejidad de la tokenización, el paso inicial de la PNL. Se discuten las nociones de palabra y token de igual manera se

definen desde los puntos de vista de la lexicografía y la implementación pragmática, respectivamente. La segmentación automática de palabras chinas se presenta como una ilustración de la tokenización. Se desarrollan enfoques prácticos para la identificación de tokens compuestos en inglés, como modismos, verbos y expresiones fijas.

En [100], los autores establecen que la combinación de unidades parecidas a palabras de un texto se denomina tokenización. Los resultados de esta tokenización son dos tipos de tokens: un tipo correspondiente a unidades cuya estructura de caracteres es reconocible como signos de puntuación, números, fechas, etc., el otro tipo son unidades que se someterán a un análisis morfológico. De igual manera exponen que la explotación lingüística del texto de origen natural puede verse como una progresión de transformaciones del texto original el cual está definido como una secuencia de caracteres que, antes de realizar cualquier análisis sintáctico del corpus, suelen tener lugar dos transformaciones: a) las oraciones deben estar aisladas ya que la mayoría de las gramáticas describen oraciones, b) para que las oraciones estén aisladas, las palabras deben aislarse del flujo original de caracteres. Los autores recomiendan que para mantener la mayor flexibilidad posible, el proceso de tokenización debe considerarse como una serie de filtros modulares a través de los cuales se puede pasar el texto de manera selectiva.

Lematización
Para [94] la lematización es la tarea de determinar que dos palabras tienen la misma raíz, a pesar de sus diferencias superficiales. Por ejemplo, las palabras (soy, son y es) tienen el lema compartido ser, mientras que las palabras (cena y cenas) tienen el lema cena. ¿Cómo se realiza la lematización? Los métodos más sofisticados de lemaización implican un análisis morfológico completo de la palabra. La morfología es el estudio de la forma en que las palabras se construyen a partir de unidades portadoras de significado más pequeñas llamadas morfemas. Los algoritmos de lematización pueden ser complejos, por eso a veces se emplea un método más simple, que consiste principalmente en cortar el final de la palabra. Esta versión ingenua del análisis morfológico se llama stemming. Uno de los algoritmos de más utilizados es el Porter stemmer. El algoritmo se basa en una serie de reglas de reescritura que se ejecutan en serie, como una cascada, en que la salida de cada ejecución es la entrada para la siguiente.

Stemming
El algoritmo de stemming fue introducido en [101], es comúnmente empleado por investigadores para el análisis de texto en el idioma inglés. Se refiere a una versión más simple de lematización en la que principalmente solo quita los sufijos del final de la palabra. El objetivo de stemming es obtener formas de raíz o raíz de palabras derivadas. Dado que las palabras derivadas son semánticamente similares para sus formas de raíz, las ocurrencias de palabras generalmente se computan después de aplicar la raíz en un texto dado.

Un algoritmo de stemming es un procedimiento computacional que reduce todas las palabras con la misma raíz a una forma común, generalmente eliminando cada

palabra de sus sufijos derivacionales e inflexibles. Los investigadores de muchas áreas de la lingüística computacional y la recuperación de información encuentran que este es un paso necesario.

En [102] el autor establece que stemming es un paso previo al procesamiento en las aplicaciones de minería de textos, así como un requisito muy común de las funciones de procesamiento del lenguaje natural. De hecho, es muy importante en la mayoría de los sistemas de recuperación de información. De igual manera determina que el propósito principal del stemming es determinar las diferentes formas gramaticales, adjetivo, verbo, adverbio, etc. a su forma raíz.

Diferencia entre lematización y stemming
Tal como se establece en [102] existe una diferencia muy sutil entre ambos conceptos. Al realizar stemming, la raíz se obtiene después de aplicar un conjunto de reglas, pero sin preocuparse por la parte del discurso (POS) o el contexto de la ocurrencia de la palabra. Por el contrario, la lematización trata de obtener el "lema"de una palabra, lo que implica reducir las formas de la palabra a su forma raíz después de comprender el POS y el contexto de la palabra en una oración dada.

Al realizar stemming, la conversión de las formas morfológicas de una palabra a su raíz se realiza asumiendo que cada una está relacionada semánticamente. No es necesario que la raíz sea una palabra existente en el diccionario, pero todas sus variantes deben asignarse a esta forma una vez que se haya completado la derivación. Hay dos puntos que deben tenerse en cuenta al utilizar un stemmer:

1. Las formas morfológicas de una palabra tienen el mismo significado básico y, por lo tanto, deben asignarse a la misma raíz.

2. Las palabras que no tienen el mismo significado deben mantenerse separadas.

Estas dos reglas son lo suficientemente buenas siempre que las raíces resultantes sean útiles para la aplicación de procesamiento de lenguaje o minería de texto. Para los idiomas con una morfología relativamente simple, la influencia del stemming es menor que para aquellos con una morfología más compleja. En cambio, la lematización [94] se ocupa del complejo proceso de comprender primero el contexto, luego determinar el POS de una palabra en una oración y finalmente encontrar el "lema". De hecho, un algoritmo que convierte una palabra a su raíz lingüísticamente correcta se denomina lemmatizer. Un lema en morfología es la forma canónica de un lexema. Lexema, en este contexto, se refiere al conjunto de todas las formas que tienen el mismo significado, y lema se refiere a la forma particular que se elige por convención para representar el lexema.

Data Sampling
Existe una variedad de procedimientos para muestrear instancias de un gran conjunto de datos. Los más conocidos son [103]:

- Muestreo aleatorio que selecciona un subconjunto de instancias al azar.

- Muestreo estratificado que es aplicable cuando los valores de clase no están distribuidos uniformemente en los conjuntos de entrenamiento. Las instancias de la (s) clase (s) minoritaria (s) se seleccionan con mayor frecuencia para igualar la distribución.

El muestreo [104] es bien aceptado por la comunidad estadística, que observa que un procedimiento potente computacionalmente que opera en una submuestra de datos puede, de hecho, proporcionar una precisión superior que uno menos sofisticado que utiliza toda la base de datos. Es un proceso de selección de un subconjunto como representación del conjunto de datos original para su uso en análisis o modelado de datos. Los datos de muestra sirven como representativos del conjunto de datos original con propiedades similares, como una media similar de manera tal que muestreo reduce la cantidad de datos que deben procesarse para su análisis y modelado.

Clasificadores para texto

A continuación se describen los clasificadores más utilizados por la comunidad científica en la clasificación de texto, dentro de los cuales se encuentran: C4.5, Naive Bayes, k-nearest neighbor y Support vector machines (SVMs) [105].

Naive Bayes

El clasificador de texto Naive Bayes (NB) ha sido utilizado ampliamente en tareas de categorización de documentos desde la 1950 [101, 106].El método clasificador Naive Bayes se basa teóricamente en el teorema de Bayes, que fue formulado por Thomas Bayes entre 1701-1761 [107, 108]. Naive Bayes sigue siendo uno de los 10 mejores algoritmos de extracción de datos debido a su simplicidad y eficiencia [109].

Si el número de documentos (n) pertenece a (k) categoría donde:
$(k) \in \{ c_1, c_2, ..., c_k\}$, la clase de salida predicha es: $c \in C$. El algoritmo Naive Bayes puede ser descrito de la manera siguiente:

$$P\left(c|d\right) = \frac{P\left(d|c\right) P\left(c\right)}{P\left(d\right)} \tag{2.15}$$

Los clasificadores bayesianos han ido ganando popularidad últimamente, y se ha descubierto que funcionan sorprendentemente bien. Estos enfoques probabilísticos hacen suposiciones sólidas sobre cómo se generan los datos, y postulan un modelo probabilístico que asumen estos supuestos, luego usan una colección de etiquetas ejemplos de entrenamiento para estimar los parámetros del modelo generativo. La clasificación de nuevos ejemplos es realizada con la regla de Bayes seleccionando la clase más probable.

Estudios recientes han abordado ampliamente esta técnica en la recuperación de información, como [110], en esta investigación los autores proponen un método más eficaz y preciso para la clasificación automática de información, llamado método de Bayes mejorado basado en el peso de la característica TF-IDF y la ponderación de

la característica del factor de grado (TIGFIB), que estima las probabilidades condicionales de Naive Bayes por característica TFIDF e importa la característica del factor de grado en la fórmula de Naive Bayes. Además, aplican el nuevo método Bayes mejorado a la clasificación de texto en el idioma chino. Los resultados muestran que nuestro método Bayes mejorado es superior a otros métodos Naive Bayes de ponderación de características.

En minería de datos, el algoritmo Naive Bayes es un enfoque simple para construir modelos que adivinan etiquetas de clase de instancias invisibles basadas en la aplicación del teorema de Bayes con fuertes supuestos de independencia entre los atributos [111]. El método NB se ha investigado intensamente en el reconocimiento de patrones durante cinco décadas en diferentes aplicaciones como:

Clasificación de texto [112], en este artículo los autores proponen soluciones heurísticas simples para algunos de los problemas con los clasificadores Naive Bayes, abordando tanto cuestiones sistémicas como problemas que surgen debido a que el texto no se genera realmente de acuerdo con un modelo multinomial. Descubren que realizando correcciones simples dan como resultado un algoritmo rápido que es competitivo con los algoritmos de clasificación de texto de última generación, como Support Vector Machine.

Detección de spam. El enfoque bayesiano del filtrado de spam fue uno de los primeros métodos utilizados para filtrar el spam, y sigue siendo relevante hasta el día de hoy. En [113], los autores analizan dos optimizaciones específicas de la clasificación de texto y el filtrado de spam en Naive Bayes, observando las diferencias entre ellas y cómo se han utilizado en la práctica. En este artículo se demuestra que el filtrado bayesiano se puede implementar para un clasificador de texto razonablemente preciso y que se puede modificar para tener un impacto significativo en la precisión del filtro.

Las ventajas y desventajas de Naive Bayes son: [2].

Ventajas

1. Funciona muy bien con datos de texto.
2. Fácil de implementar.
3. Rápido en comparación con otros algoritmos.

Desventajas

1. Una fuerte suposición sobre la forma de la distribución de datos.
2. Limitado por la escasez de datos para los que cualquier valor posible en el espacio de características, un valor de probabilidad debe ser estimado por un frecuentista.

Naive Bayes Multinomial
Es adecuado para la clasificación de texto ya que su probabilidad condicional se

calcula en función de los valores del vector de características que representan la frecuencia de cada palabra en el documento [114]. Los enfoques recientes para la clasificación de textos han utilizado dos diferentes modelos probabilísticos de primer orden para la clasificación, ambos hacen la suposición de Naive Bayes. Algunos utilizan un modelo de Bernoulli Multivariable, es decir, una red bayesiana sin dependencias entre palabras y características de palabras binarias como se utiliza en [115], los autores proponen un enfoque que utiliza la estructura jerárquica de temas para descomponer la tarea de clasificación en un conjunto de problemas más simples, uno en cada nodo del árbol de clasificación. Donde muestran que cada uno de estos problemas más pequeños se puede resolver con precisión centrándose solo en un conjunto muy pequeño de características más relevantes para la tarea en cuestión. Este conjunto de características relevantes varía ampliamente a lo largo de la jerarquía, de modo que, aunque el conjunto de características relevantes en general puede ser grande, cada clasificador solo examina un pequeño subconjunto. Concluyen que el uso de conjuntos de características reducidas permite utilizar modelos más complejos (probabilísticos), sin encontrar muchas de las dificultades estándar de cálculo y robustez. Otros autores utilizan un modelo multinomial, es decir, un modelo de lenguaje uni-grama con recuento de palabras enteras por ejemplo, en [116]. Los autores establecen la necesidad de poder entrenar clasificadores de texto de manera económica para su uso en la recuperación de información, el análisis de contenido, el procesamiento del lenguaje natural y otras tareas que involucran datos que son parcial o totalmente textuales. Con este propósito los autores desarrollaron y probaron un algoritmo para el muestreo secuencial de clasificadores estadísticos durante el aprendizaje automático en una tarea de categorización de texto de newswire. Este método le denominaron muestreo por incertidumbre, logrando una reducción hasta 500 veces la cantidad de datos de entrenamiento que tendrían que ser clasificados manualmente para lograr un nivel de efectividad óptimo.

Ha sido verificado el mejor desempeño de Naive Bayes Multinomial en dominios especificos tal como se analiza en [69], en este artículo los autores describen las diferencias y detalles de estos dos modelos (Bernoulli Multivariable y Naive Bayes Multinomial), y por comparación evalúan el desempeño de clasificación en cinco corpus de texto. Los autores encuentran que el modelo Bernoulli Multivariable se desempeña bien con vocabulario pequeños, pero el modelo Naive Bayes Multinomial funciona mejor en vocabularios más grandes, proporcionando en promedio una reducción del 27 % en el error sobre el modelo Bernoulli Multivariante.

De igual manera en [117], los autores establecen que Naive Bayes Multinomial tiene las siguientes ventajas y desventajas:

Ventajas

1. La longitud del documento se contabiliza de forma natural en el modelo.

Desventajas

1. El modelo asume independencia no solo entre diferentes palabras, sino entre diferentes ocurrencias múltiples de la misma palabra, una suposición muy

relevante cuando de datos reales se trata.

k-nearest neighbor (K-NN)
Es una técnica no paramétrica que se utiliza para la clasificación. Tal como se define en [17], K-NN se encuentran entre los algoritmos más simples de todo el aprendizaje automático. K-NN consiste en memorizar el conjunto de entrenamiento y luego predecir la etiqueta de cualquier nueva instancia sobre la base de las etiquetas de sus vecinos más cercanos en el conjunto de entrenamiento ver figura 2.2. El fundamento de este método se basa en el supuesto que las características que se utilizan para describir los puntos del dominio son relevantes para sus etiquetas haciendo que los puntos cercanos tengan la misma etiqueta. Además, en algunas situaciones, incluso cuando el conjunto de entrenamiento es inmenso, encontrar el vecino más cercano se puede hacer extremadamente rápido (por ejemplo, cuando el conjunto de entrenamiento es toda la Web y las distancias se basan en enlaces).

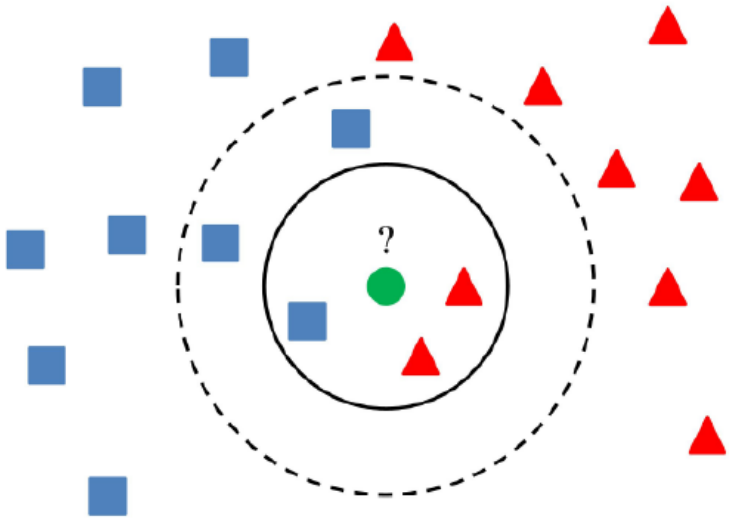


Figura 2.2: El conocimiento común del clasificador K-NN Fuente: Google Image.

En otras palabras, dado un documento de prueba x algoritmo K-NN encuentra los k vecinos más cercanos de x entre todos los documentos del conjunto de datos de entrenamiento para puntúar los candidatos de categoría según la clase de k vecinos.

Donde la similitud de x y cada documento vecino sería la puntuación de la categoría del documento vecino. Varios documentos K-NN pueden pertenecer a la misma categoría, en este caso, la suma de estas puntuaciones sería la puntuación de similitud de la clase k con respecto al documento de prueba n. Después de ordenar los valores de puntuación, el algoritmo asigna al candidato a la clase con la puntuación más alta del documento de prueba x [118].

El clasificador tradicional k-NN utiliza la fórmula de distancia euclidiana para categorizar el texto en una o más clases predefinidas según sus materias. Por tanto, el clasificador utiliza las palabras clave de los textos que se han comparado con las

palabras clave de los nuevos textos.

Este método ha sido utilizado en aplicaciones de clasificación de textos en muchos campos de investigación en las últimas décadas. En [118], los autores establecen que la categorización de texto K-NN es un método de clasificación efectivo, pero menos eficiente que NB y SVM. En este artículo, los autores proponen un algoritmo K-NN mejorado para la categorización de texto, que construye el modelo de clasificación mediante la combinación de un algoritmo de agrupación de un solo paso restringido y la categorización de texto K-NN. Los resultados empíricos en tres corpus de referencia muestran que algoritmo puede reducir sustancialmente el cálculo de similitud de texto y superar a los clasificadores de última generación K-NN, NB y SVM.

Sin embargo, en [119], los autores establecen que no es práctico para los métodos tradicionales de K-NN asignar un valor k fijo (aunque lo establezcan los expertos) a todas las muestras de prueba. Las soluciones anteriores asignan diferentes valores k a diferentes muestras de prueba mediante el método de validación cruzada, que por lo general requieren mucho tiempo. Este artículo propone un método de kTree para aprender diferentes valores de k óptimos para diferentes pruebas, al involucrar una etapa de entrenamiento en la clasificación con K-NN. Específicamente, en la etapa de entrenamiento, el método kTree primero aprende valores de k óptimos para todas las muestras mediante un nuevo modelo de reconstrucción dispersa, y luego construye un árbol de decisión (es decir, kTree) usando muestras de entrenamiento y los valores de k óptimos aprendidos. En la etapa de prueba, kTree genera rápidamente el valor k óptimo para cada muestra de prueba, y luego, la clasificación K-NN se puede realizar utilizando el valor k óptimo aprendido y todas las muestras de entrenamiento. Como resultado, el método kTree propuesto tiene un costo de funcionamiento similar pero una mayor precisión de clasificación, en comparación con los métodos K-NN tradicionales, que asignan un valor k fijo a todas las muestras de prueba.

En la Tabla 2.1, se muestran las ventajas y desventajas de las técnicas basadas en Nearest neighbor

Algoritmo	Ventajas	Desventajas
K-Nearest Neighbor (K-NN) [120]	1. El entrenamiento es muy rápido. 2. Simple y fácil de aprender. 3. Robusto ante datos difusos en el conjunto de entrenamiento. 4. Eficaz si el conjunto de entrenamiento es grande.	1. Sesgado por el valor de k. 2. Complejidad computacional. 3. Limitación de la memoria. 4. Su ejecución es lenta. 5. Fácilmente engañado por atributos irrelevantes.

Weighted k nearest neighbor (WkNN) [121]	1. Supera las limitaciones de K-NN de asignar el mismo peso a k vecinos implícitamente. 2. Utiliza todas las muestras de entrenamiento, no solo k. 3. Hace que el algoritmo sea global.	1. La complejidad de los cálculos aumenta al calcular los pesos. 2. El algoritmo funciona con lentitud.
Condensed nearest neighbor (CNN) [122]	1. Reduce el tamaño de los datos de entrenamiento. 2. Mejora el tiempo de consulta y los requisitos de memoria. 3. Reduce la tasa de reconocimiento.	1. CNN depende del pedido; es poco probable que capte puntos en el límite. 2. Complejidad en los cálculos.
Reduced Nearest Neigh (RNN) [123]	1. Reduce el tamaño de los datos de entrenamiento y elimina las plantillas. 2. Mejora el tiempo de consulta y los requisitos de memoria. 3. Reduce la tasa de reconocimiento.	1. Complejidad computacional. 2. Tiempo requerido.
Model based k nearest neighbor (MkNN) [124]	1. Más precisión de la clasificación. 2. El valor de k se selecciona automáticamente. 3. Alta eficiencia al reducir el número de puntos de datos.	1. No considera datos marginales fuera de la región.
Rank nearest neighbor (kRNN) [125]	1. Se desempeña mejor cuando hay variaciones entre las funciones. 2. Resistente según el rango 3. Menor complejidad de cálculo en comparación con K-NN	1. KRNN multivariante depende de la distribución de los datos.
Modified k nearest neighbor (MkNN) [126]	1. Supera parcialmente la baja precisión de WkNN.	1. Complejidad computacional.
Pseudo Generalized Nearest Neighbor (GNN) [127]	1. Utiliza clases n-1 que consideran todo el conjunto de datos de entrenamiento.	1. No es válido para datos pequeños. 2. Complejidad computacional.

Clúster k nearest neighbor [128]	1. Supera el defecto de distribuciones desiguales en las muestras de entrenamiento. 2. Es de naturaleza robusta.	1. La selección del parámetro de umbral es difícil antes de ejecutar el algoritmo. 2. Basado en el valor de k para la agrupación.
Ball Tree k nearest neighbor(KNS1) [129]	1. Sintoniza bien la estructura de los datos representados. 2. Trata bien con entidades de alta dimensión. 3. Fácil de implementar.	1. Costos algoritmos de inserción. 2. A medida que aumenta la distancia, KNS1 se degrada.
k-d tree nearest neighbor (kdNN) [130]	1. Genera un árbol perfectamente equilibrado. 2. Rápido y sencillo.	1. Más cálculos. 2. Requiere una búsqueda intensiva. 3. Corta ciegamente los puntos por la mitad que pueden perder la estructura de datos.
Nearest featu-re Line Neighbor (NFL) [131]	1. Mejora la precisión en la clasificación. 2. Altamente efectivo en conjuntos de datos pequeños. 3. Utiliza información ignorada en el vecino más cercano, es decir, plantillas por clase.	1. Fracasa cuando el prototipo en la NFL está lejos del punto de consulta. 2. Complejidad de los cálculos. 3. Describe puntos de características por línea recta es una tarea difícil.

Tabla 2.1: Comparación de técnicas nearest neighbor.

Support vector machines (SVMs)

La versión original de SVM fue desarrollada por [132]. En 1963 [133] adaptó esta versión a una formulación no lineal a principios de 1990. Dado un conjunto de puntos, donde cada uno de ellos forma parte de una categoría, el algoritmo SVM genera un modelo que puede predecir si un punto nuevo pertenece a una u otra categoría. En este algoritmo el conjunto de datos de entrada es visto como un vector p-dimensional.

Para [134], los métodos de entrenamiento convencionales determinan los modelos de tal manera que cada par de entrada-salida se clasifica correctamente dentro de la clase a la que pertenece. Sin embargo, si el clasificador es demasiado apto para los datos de entrenamiento, el modelo comienza a memorizar los datos de entrenamiento en lugar de aprender a generalizar, degradando la capacidad de generalización del clasificador. La principal motivación de SVM es separar varias clases en el conjunto de entrenamiento con una superficie que maximice el margen entre ellos. En otras palabras, SVM permite maximizar la capacidad de generalización de un modelo. Este es el objetivo del principio de minimización del riesgo estructural (SRM) que permite minimizar el error de generalización de un modelo, en lugar de minimizar el error cuadrático medio en el conjunto de datos de entrenamiento, que es el uso frecuente de los métodos.

SVM fue originalmente diseñado para tareas de clasificación binaria. Sin embargo, muchos investigadores utilizan esta técnica para resolver problemas [135]. A continuación, se describen implementaciones de SVM en la resolución de problemas de la vida real específicamente en la clasificación de texto.

En [136], los autores presentaron la implementación de un marco de clasificación de documentos de texto que utiliza el enfoque SVM en la fase de entrenamiento y la distancia euclidiana en la fase de clasificación. En el enfoque propuesto, los vectores de soporte para cada categoría se identifican a partir de los puntos de datos de entrenamiento durante la fase de entrenamiento usando SVM. Durante la clasificación, cuando se mapea un nuevo punto de datos en el espacio vectorial original, las distancias promedio entre el nuevo punto de datos y los vectores de soporte de diferentes categorías se miden utilizando la distancia euclidiana. La decisión de clasificación se toma en función de la categoría de vectores de soporte que tiene la distancia promedio más baja con el nuevo punto de datos, tomando la decisión de clasificación independientemente de la eficacia del hiperplano formado al aplicar la función de kernel particular y el parámetro de margen suave.

En [137], se evalúan tres métodos de aprendizaje automático, k vecino más cercano, SVM y mapa asociativo de resonancia adaptativa, se evalúan para la categorización de documentos chinos. Basados en dos corpus chinos, una serie de experimentos controlados evaluaron sus capacidades de aprendizaje y eficiencia en el conocimiento de clasificación de textos de minería. SVM es muy eficiente en el aprendizaje de muestras bien organizadas de tamaño moderado, aunque en datos relativamente grandes y ruidosos la eficiencia de SVM y mapa asociativo de resonancia adaptativa son comparables.

En [138], los autores demostraron que en el caso de la clasificación de texto, las transformaciones de frecuencia de términos tienen un impacto mayor en el rendimiento de SVM que un kernel en sí. Se discute el papel de las ponderaciones de importancia, que no se comprende totalmente dada la complejidad del modelo y el costo de cálculo. También se muestra que la lematización o la derivación, que consume mucho tiempo, puede evitarse incluso al clasificar un lenguaje con mucha inflexión. SVM es una de las técnicas utilizadas en el aprendizaje activo para reducir el esfuerzo de etiquetado de datos en diferentes campos del reconocimiento de patrones.

En [139], Los autores presentaron un aprendizaje activo en modo batch utilizando SVM para clasificación de texto, ya que la mayoría de los trabajos relacionados que aplican métodos de aprendizaje activo a la clasificación automática de texto se enfocan en solicitar la etiqueta de un documento sin etiquetar en cada iteración.

En, [140] los autores presentan SVM lineal junto con agrupamiento distributivo de palabras para darse cuenta de su potencial en el ámbito de la categorización de texto. El agrupamiento de distribución se ha presentado como una alternativa eficiente a la selección de características que se usa convencionalmente en la categori-

zación de texto. La agrupación en clústeres distribuida junto con SVM lineal reduce la dimensionalidad de los documentos de texto sin comprometer el rendimiento de la clasificación. En este estudio, se emplearon SVM lineal y su extensión SVM difusa junto con agrupamiento distribuido para la categorización de texto.

En [141] los autores proponen un enfoque para la ponderación de términos en documentos muy cortos que se utiliza con un clasificador SVM. El artículo se centra en la investigación de mercado y los documentos de las redes sociales. En ambas fuentes de datos, la extensión media de un documento es inferior a veinte palabras. Como los documentos son breves, cada palabra suele aparecer una sola vez en un documento. Por lo tanto, se propuso un enfoque para la ponderación de términos que no utiliza la frecuencia de los términos dentro de un documento, sino que la sustituye por otras estadísticas de palabras.

En datos textuales de grandes dimensiones y clases múltiples a gran escala, es común ignorar la semántica entre palabras con el método tradicional de selección de características. En [142], los autores introdujeron la información de las categorías en el algoritmo de selección de características del modelo LDA (Latent Dirichlet Allocation) existente y construyeron un clasificador de múltiples clases SVM en la matriz implícita de tema-texto.

En [143], los autores presentaron un clasificador de texto utilizando ejemplos positivos y sin etiquetar. El desafío de este problema en comparación con el problema de clasificación de texto clásico es que no hay documentos negativos etiquetados disponibles en el conjunto de ejemplos de capacitación. Muchos documentos negativos más fiables se identifican mediante un algoritmo mejorado de 1-DNF. Luego, se construye un conjunto de clasificadores aplicando iterativamente el algoritmo SVM en un conjunto de datos de entrenamiento, que se aumenta durante la iteración. Posteriormente, a diferencia de los trabajos anteriores de clasificación de textos orientados a PU, se adopta el voto ponderado de todos los clasificadores generados en los pasos de iteración para construir el clasificador final en lugar de elegir uno de los clasificadores como clasificador final. Los autores proponen un enfoque para evaluar el voto ponderado de todos los clasificadores generados en los pasos de iteración para construir el clasificador final basado en la optimización del enjambre de partículas.

En [144], los autores proponen las transformaciones de dicotomías combinadas, un sistema de categorización de texto que combina clasificadores binarios entrenados con diferentes conjuntos de dicotomías utilizando transformación de dicotomías, donde el número de ejemplos de entrenamiento aumenta exponencialmente cuando se comparan con el conjunto original. Esta propiedad es deseable porque cada clasificador puede entrenarse con diferentes datos sin reducir el número de ejemplos o características. De esta forma, es posible componer un conjunto con clasificadores diversos y fuertes. Los experimentos se realizan utilizando SVM, subespacio aleatorio, boostexter y Random Forest.

En [145], los autores promovieron un nuevo punto de referencia llamado RTA-

news, que es un conjunto de datos de artículos de noticias árabes de múltiples etiquetas para la categorización de texto. Llevaron a cabo una amplia comparación de la mayoría de los conocidos algoritmos de aprendizaje de etiquetas múltiples para la categorización de texto en árabe con el fin de tener resultados de referencia y mostrar la eficacia de estos algoritmos para la categorización de texto en árabe en RTAnew. La evaluación involucra varios algoritmos, como relevancia binaria, cadenas de clasificadores, clasificación calibrada, SVM, K-NN, Random Forest y cuatro algoritmos basados en adaptación. Los resultados demuestran que los algoritmos basados en la adaptación son más rápidos que los algoritmos basados en la transformación.

Ventajas	Desventajas
1) SVM puede modelar límites de decisión no lineal. 2) Se desempeña de manera similar a Logistic regression cuando la separación es lineal. 3) Robusto contra problemas de sobreajuste (especialmente para datos de texto establecido debido a las altas dimensiones espacio).	1) Falta de transparencia en los resultados causado por una gran cantidad de dimensiones (especialmente para texto datos). 2) Elegir un kernel eficiente la función es difícil (susceptible a problemas de sobreajuste/entrenamiento dependiendo del kernel). 3) Complejidad en el uso de la memoria.

Tabla 2.2: Ventajas y desventajas de SVM [2].

Random Forest

Fue desarrollado por [146] y es un grupo de árboles de regresión o clasificación no podados, hechos a partir de la selección aleatoria de muestras de los datos de entrenamiento. Las características aleatorias se seleccionan en el proceso de inducción. La predicción se realiza agregando (voto mayoritario para la clasificación o promediando para la regresión) las predicciones del conjunto. Random Forest es muy rápido de entrenar para conjuntos de datos de texto en comparación con otras técnicas como el aprendizaje profundo, pero bastante lentos para crear predicciones una vez entrenados [147]. Por lo tanto, para lograr una estructura más rápida, se debe reducir la cantidad de árboles en el bosque, ya que más árboles en el bosque aumenta la complejidad del tiempo en el paso de predicción.

Random Forest es apropiado para el modelado de datos de alta dimensión porque puede manejar valores perdidos y datos continuos, categóricos y binarios. El esquema de arranque y conjunto hace que Random Forest sea lo suficientemente fuerte como para superar los problemas de ajuste excesivo y, por lo tanto, no hay necesidad de podar los árboles. Además de una alta precisión de predicción, Random Forest es eficiente, interpretable y no paramétrico para varios tipos de conjuntos de datos [148]. La interpretación del modelo y la precisión de la predicción proporcionada por Random Forest es única entre los métodos populares de aprendizaje automático. Se logran predicciones precisas y mejores generalizaciones debido a la utilización de estrategias de conjunto y muestreo aleatorio.

Las tres características principales de Random Forest que ganaron atención [149] son:

1. Resultados precisos de predicciones para una variedad de aplicaciones.
2. A través del entrenamiento del modelo, se puede medir la importancia de cada característica.
3. El modelo entrenado puede medir la proximidad por pares entre las muestras.

En Random Forest, las características se seleccionan al azar en cada división de decisiones. La correlación entre árboles se reduce seleccionando aleatoriamente las características que mejoran el poder de predicción y dan como resultado una mayor eficiencia.

Ventajas	Desventajas
1) No se necesita podar arboles. 2) Precisión y variable importancia generadas automáticamente. 3) No es muy sensible a los valores atípicos en los datos de entrenamiento. 4) Fácil de configurar los parámetros	1) La regresión no puede predecir más allá del rango en los datos de entrenamiento. 2) Con la regresión, a menudo los valores extremos no se predicen con precisión: subestiman los máximos y los mínimos.

Tabla 2.3: Ventajas y desventajas de Random Forest [3].

Al resolver problemas de clasificación, la predicción de RF es la mayoría no ponderada de los votos de la clase. La figura 2.3, presenta una arquitectura general de RF, donde B es el número de árboles en RF y k_1, k_2, k_B y k son etiquetas de clase. A medida que aumenta el número de árboles en RF, las tasas de error del conjunto de prueba convergen hasta un límite, lo que significa que no hay sobreajuste en RF grandes [146]. El bajo sesgo y la baja correlación son esenciales para la precisión. Para obtener un sesgo bajo, los árboles se cultivan a la profundidad máxima. Para lograr una baja correlación, se aplica la aleatorización:

1. Cada árbol de RF se cultiva en una muestra de arranque del conjunto de entrenamiento.
2. Al hacer crecer un árbol, en cada nodo, se seleccionan al azar n variables de las N disponibles.
3. Usualmente, $n \ll N$ en este supuesto se inicia con: $n = \log_2(N)+1$ ó $n = \sqrt{N}$ y luego disminuyendo y aumentando n hasta que se obtenga el error mínimo para el conjunto de datos OOB. En cada nodo, solo una variable, que proporciona la mejor división, se usa de las n seleccionadas.

La popularidad de los clasificadores de RF se destaca por su aplicación exitosa en varios dominios a continuación, se destacan aquellas aplicadas en la clasificación de texto:

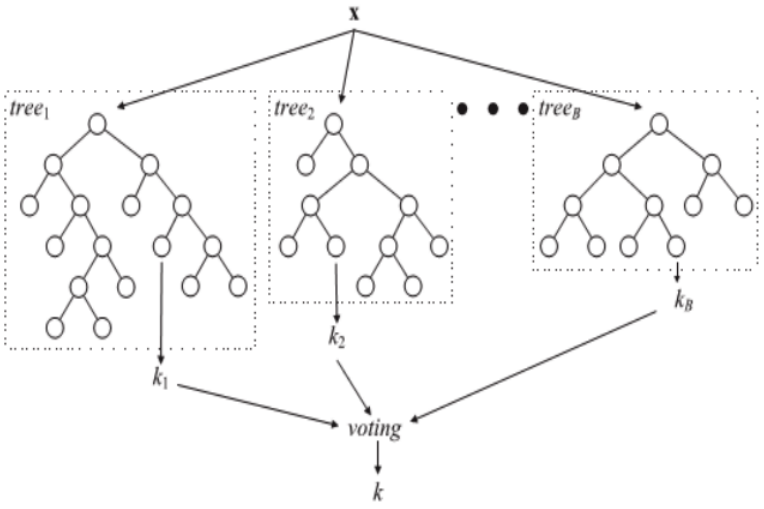


Figura 2.3: Arquitectura general de un Random forest [1].

En [150], los autores describen el uso de árboles de decisión (DT) y (RF) en el modelado de lenguaje checo. Evidencian como el enfoque de RF puede implementarse con éxito para el modelado de lenguaje en este idioma. Además, evalúan el rendimiento de DT y RF en la tarea de reconocimiento de conferencias. Concluyendo que RF ofrece mejores resultados que DT.

En [151], los autores investigan el funcionamiento de RF en la clasificación de texto y comparan su desempeño. Llevan a cabo experimentos en cuatro conjuntos de datos de texto ampliamente utilizados utilizando SMV y Naive Bayes como clasificadores. RF logra un mejor desempeño que otros algoritmos de FS en el 73 % de las instancias experimentales. También analizaron el rendimiento de RF para TC en términos de sus parámetros y sesgos de clase de los conjuntos de datos, arrojando resultados interesantes para posteriores investigaciones.

En [152]. Los autores proponen una versión del clasificador tradicional RF llamado LazyNNRF, especialmente diseñado para tareas de clasificación ruidosas de gran dimensión. La proyección de formación "localizada" de LazyNNRF está compuesta por ejemplos que se asemejan mejor a los ejemplos a clasificar, obtenidos a través de la proyección del conjunto de formación del vecindario más cercano. Dicha proyección filtra datos irrelevantes, evitando en última instancia algunos de los inconvenientes de los RF tradicionales, como el sobreajuste debido a árboles muy complejos, especialmente en conjuntos de datos ruidosos de alta dimensión. Evidencian que el enfoque es altamente efectivo y factible, siendo un fuerte candidato a considerar para resolver tareas de clasificación automática de texto en comparación con clasificadores de última generación.

C4.5

Ha habido muchas variaciones para los algoritmos de árboles de decisión. C4.5 es uno de los conocidos algoritmos de inducción de árboles de decisión [153]. En 1993,

Ross Quinlan propuso el algoritmo C4.5 que amplía el ID3 algoritmo. Usando la relación de ganancia de información para seleccionar el mejor atributo, C4.5 evita el sesgo de ID3 hacia características que ocurre con muchos valores. C4.5 tiene la capacidad para manejar atributos continuos proponiendo dos pruebas diferentes en función de cada tipo de valores de atributo. En la etapa de entrenamiento, el C4.5 utiliza la estrategia de arriba hacia abajo basada en enfoque de dividir y conquistar para construir el árbol de decisiones [154]. C4.5 mapea el conjunto de entrenamiento y usa la información obtenida relacionándola como medida para seleccionar atributos de división y genera nodos desde la raíz hasta las hojas, cada ruta ilustrativa desde el nodo raíz hasta el nodo hoja forma una regla de decisión para determinar cuál es la clase de una nueva instancia [155]. El nodo raíz contiene todo el conjunto de entrenamiento con todos los pesos de casos de entrenamiento iguales a 1.0, para tener en cuenta valores de atributos desconocidos [153]. Si todos los casos de entrenamiento del nodo actual pertenecen a una sola clase, el algoritmo termina. De lo contrario, si todos los casos de entrenamiento pertenecen a más de una clase, el algoritmo calcula la relación de ganancia de información para cada atributo A_j con la relación de ganancia de información más alta se selecciona para dividir información en el nodo [156]. Para un atributo discreto A_j la relación de ganancia de información se calcula dividiendo los casos de entrenamiento del nodo actual en función de cada valor de A_j. Si A_j es atributo continuo, debe ser encontrado un valor de umbral para realizar la división [157]. C4.5 es uno de los algoritmos de árboles de decisión más conocidos y utilizados. Su nivel de precisión es lo suficientemente alto, independientemente del volumen de datos a procesar [158].

El algoritmo C4.5 tiene la capacidad de manejar un conjunto de datos de entrenamiento incompleto y podar la decisión resultante para reducir su tamaño y optimizar la ruta de decisión C4.5 también tiene la capacidad de lidiar con atributos continuos, mediante el proceso de binarización. Estos atributos son reemplazados por valores discretos que utilizan de umbral para separar los datos en dos intervalos [159].

Varios estudios identifican al algoritmo C4.5 como uno de los principales clasificadores en minería de datos [160], los autores presentan los 10 algoritmos principales de minería de datos identificados por la Conferencia Internacional de Minería de Datos (ICDM) de IEEE en diciembre de 2006: C4.5, K-means, SVM, Apriori, EM, PageRank, AdaBoost, K-NN, Naive Bayes y CART. Estos 10 algoritmos principales se encuentran entre los algoritmos de minería de datos más influyentes en la comunidad de investigación. De cada algoritmo, proporcionan una descripción, discuten el impacto y revisan las investigaciones actuales y futuras sobre el cada uno de ellos. Estos 10 algoritmos cubren clasificación, agrupamiento, aprendizaje estadístico, análisis de asociación y minería de enlace, que se encuentran entre los temas más importantes en la investigación y el desarrollo de minería de datos.

En [161], los autores proponen un modelo de árbol de decisión utilizando el algoritmo C4.5 para clasificar la semántica (positiva, negativa, neutral) en documentos en el idioma inglés. Utilizan el algoritmo C4.5 en 70,000 oraciones positivas en inglés para generar un árbol de decisión y muchas reglas de asociación de la polaridad po-

sitiva. También utilizan el algoritmo C4.5 en las 70,000 oraciones negativas en inglés para generar un árbol de decisión y muchas reglas de asociación de la polaridad negativa fueron creadas. Los sentimientos de un documento en inglés son identificados con base en las reglas de asociación de la polaridad positiva y la polaridad negativa, con su modelo basado en el algoritmo C4.5 los autores logran una precisión del 60,3 % en la clasificación de sentimientos en el conjunto de datos de prueba.

Ventajas	Desventajas
1) Puede generar reglas a partir de un solo árbol y transformar múltiples árboles de decisión para la misma tarea en un conjunto de reglas. 2) Puede clasificar registros que tienen valores de atributos desconocidos estimando la probabilidad de los diversos resultados posibles. 3) Usa la ganancia de información, la normalización y la poda para el rendimiento del tiempo.	1) Debilidades en dominios con atributos continuos. 2) La generación de reglas de C4.5 es relativamente más lenta en comparación con la generación de árboles.

Tabla 2.4: Las ventajas y desventajas de C.4.5 [4].

ZeroR

ZeroR es el método de clasificación más simple que se basa en el objetivo e ignora todos los predictores. A pesar de que ZeroR carece de poder de previsibilidad, es útil para determinar el desempeño de una línea base como una métrica para otros métodos de clasificación. ZeroR construye una tabla vacilación para la función y selecciona su valor de vacilación más alto [162]. Al comienzo de la clasificación supervisada, se puede utilizar para establecer un rendimiento de referencia como criterio para la validez de otros métodos de clasificación [163].

En [164] los autores proponen un modelo para la detección de noticias falsas, combinando métodos de analisis de texto (primer paso) obtener datos estructurados provenientes de noticias no estructuradas para posteriormente (segundo paso) aplicar algoritmos de inteligencia artificial para su clasificación. En este estudio el problema de detección de noticias falsas se ha modelado como un problema de clasificación utilizando veintitrés algoritmos de inteligencia artificial supervisados, obteniendo como resultado que los algoritmos de árbol de decisión. ZeroR, CVPS y WIHW, parecen los mejores algoritmos en términos de métricas de recuperación.

.

En [165] se diseñaron modelos de aprendizaje supervisado utilizando ZeroR, Naive Bayes, J48 y Random Forest para la clasificación de las declaraciones de la guía de prácticas clínicas GPC. Fueron entrenados para clasificar cada enunciado de la GPC en ninguna condición (NC), condición-acción (CA), o condición consecuencia (CC). Además, se utilizó el etiquetado de parte del discurso (POS) para eliminar las restricciones de dependencia del dominio y se identificaron declaraciones de recomendación mediante el uso de modificadores y expresiones regulares. Las declaraciones de recomendación identificadas fueron transformadas en formato "si la condición

entonces las consecuencias".

Ventajas	Desventajas
1) Simple y efectivo. 2) Facil de entender e interpretar.	1) clasificador trivial, pero da un límite inferior en el rendimiento de un conjunto de datos dado que debe ser mejorado significativamente por clasificadores más complejos.

Tabla 2.5: Las ventajas y desventajas de ZeroR [5].

Regresión logística

La regresión logística [6] es una técnica de clasificación utilizada en el aprendizaje automático que implementa una función logística para modelar la variable dependiente. Esta variable es de naturaleza dicotómica, es decir, solo podría haber dos clases posibles. Como resultado, esta técnica se utiliza al tratar con datos binarios.

Aunque generalmente se usa para predecir variables objetivo binarias, la regresión logística se puede extender y clasificar en tres tipos:

- Binomial: donde la variable objetivo puede tener solo dos tipos posibles. Ej.: Predecir un correo como spam o no.
- Multinomial: donde la variable objetivo tiene tres o más tipos posibles, que pueden no tener ningún significado cuantitativo. ej.: Predicción de enfermedades.
- Ordinal: Donde las variables objetivo tienen categorías ordenadas. Ej.: Calificaciones de videos del 1 al 10.

La regresión logística se usa en varios campos, incluido el aprendizaje automático, la mayoría de los campos médicos y las ciencias sociales. En [166] los autores utilizan regresión logística para la clasificación de sentimientos de la revisión de medicamentos recopilada de el conjunto de datos de Drug Reviews. La regresión logística es utilizada para la predicción de la pertenencia a un grupo. El poder de predicción con las características dadas tiene una precisión del 80 % con el método de clasificación LSTM. En [167] el autor ha aplicado enfoques de aprendizaje automático supervisado como Naive Bayes, arbol de decisión, SVM y Regresión logística finalmente se construyó una ontología de dominio para la especificación formal del conocimiento en ese dominio. La descripción semántica se logra haciendo coincidir el concepto de la ontología que mejora el rendimiento de la clasificación de texto. En [168] desarrollan un modelo para la clasificación de notas clínicas concluyendo que el mejor rendimiento entre todos los enfoques computacionales de aprendizaje automático utilizado es la regresión logística, la cual produce mejores cifras de rendimiento en términos de precisión y recuperación. .

Ventajas	Desventajas
1. La regresión logística es más fácil de implementar, interpretar y muy eficiente de entrenar. Es muy rápido en la clasificación de registros desconocidos. 2. Funciona bien cuando el conjunto de datos es linealmente separable. 3. Puede interpretar los coeficientes del modelo como indicadores de la importancia de las características.	1. Construye límites lineales. La regresión logística necesita que las variables independientes estén linealmente relacionadas con las probabilidades logarítmicas. 2. La principal limitación de la regresión logística es la suposición de linealidad entre la variable dependiente y las variables independientes. 3. Los algoritmos más potentes y compactos, como las redes neuronales, pueden superar fácilmente a este algoritmo.

Tabla 2.6: Las ventajas y desventajas de Regresión Logística [6].

2.4. Ontologías

Definición

El concepto de ontología se ha definido de múltiples formas, si bien ha sido definida comúnmente como una disciplina del elemento puro de todo nuestro conocimiento, o que agrupa todo el conocimiento previo que se tiene sobre algo [169]. Otro de los conceptos ampliamente reconocido entre diferentes disciplinas, en particular en el área de la inteligencia artificial se puede analizar en [169] donde se define a la ontología como una especificación explícita de una conceptualización. Además, las ontologías constituyen el elemento de mayor relevancia de la Web Semántica debido a que logra una visión explícita y detallada entre agentes [170]. De igual manera, se ha podido notar un incremento en la exploración y utilización de ontologías como tecnología de intercambio e interconexión de conocimiento a través de la web, por parte de la comunidad de Ingeniería del Software [171]. De tal manera que la Web Semántica ha sido de utilidad para establecer o ejemplificar el término actual de ontología [172], haciendo la salvedad que este concepto ya había sido desarrollado previamente en el campo de la filosofía.

Componentes de una ontología

Una ontología está conformada por dos partes, conceptos y relaciones. La extracción del concepto de una fuente de datos representa el primer paso en la construcción de una ontología de dominio de manera que es requerido determinar, la fuente de datos para obtener el concepto [173]. Las ontologías y lenguajes estándar como Resource Description Framework (RDF), Protocolo simple, RDF Query Language (SPARQL) y Web Ontology Language (OWL) son consideradas herramientas que agregan y recuperan valor semántico a un conjunto de datos [174]. El conocimiento en una ontología es formalizado mediante la utilización de: clases, relaciones, propiedades,

axiomas e instancias [175]. Una clase es un conjunto de objetos (físicos, procesos, tareas), la cual generalmente es organizada en una taxonomía. Las clases se basan en la representación del conocimiento en ontologías, porque describen los conceptos del dominio. Una clase cuyos miembros son clases llamadas superclases o metaclases.

Las relaciones representan las interacciones entre los conceptos establecidos de una ontología. Se definen formalmente como cualquier subconjunto de un producto de n conjuntos: $R = C_1xC_2x...xC_n$, donde C_i representa los conjuntos con $i = 1..n$. Algunas de las relaciones más utilizadas son:

- Instancia: asocia objetos a clases.
- Relaciones temporales: implican precedencia en el tiempo.
- Relaciones topológicas: establecen conexiones espaciales entre conceptos.
- Relaciones taxonómicas: una clasificación jerárquica, en la que las elementos se organizan en grupos o tipos.

Los axiomas son elementos que se utilizan para modelar oraciones que siempre son verdaderas. Los axiomas pueden ser estructurales o no estructurales.

Los axiomas estructurales constituyen condiciones relacionadas con jerarquías de conceptos de ontología y atributos definidos, mientras que un axioma no estructural es la relación entre los atributos de un concepto y son específicos de un dominio.

Las instancias o individuos son miembros de objeto de una clase que se utilizan para representar elementos y se pueden agrupar en clases.

Propiedades o ranuras. Los objetos se describen mediante un conjunto de características o atributos que se almacenan en las ranuras. Estas ranuras almacenan diferentes tipos de valores. Las especificaciones, rangos y restricciones sobre estos valores o características se denominan facetas.

Clasificación de los tipos de ontologías

Existen diferentes tipos de ontologías según la perspectiva que se aplique. Básicamente, se siguen dos criterios para estas clasificaciones: el tipo de conocimiento que contienen y la motivación de la ontología.

El criterio donde hay mayor diversidad es en la clasificación por los conocimientos que contienen. De acuerdo a [176], las ontologías se clasifican según la cantidad y tipo de estructura de la conceptualización.

Terminología y ontologías lingüísticas:
Especifican los términos que se utilizan para representar el conocimiento en el universo del discurso. Un ejemplo de esto es la red semántica UMLS (Unified Medical Language System) [177]. Suelen utilizarse para la unificación del vocabulario en

un campo determinado. Una de las ontologías del lenguaje más utilizadas es WordNet [178,179] que es una gran base de datos léxica en la que se contemplan diferentes tipos de relaciones.

- Ontología de la información: especifica las bases de datos de la estructura de almacenamiento que proporciona un marco para almacenar información estandarizada. El esquema de la base de datos es un ejemplo.

- Conocimientos de modelado de ontologías: especifican conceptualizaciones del conocimiento. Estas ontologías contienen una rica estructura interna y son las ontologías que se adaptan más específicamente al conocimiento que describen.

Existe una clasificación alternativa que se puede encontrar en [180], donde también se sugieren tres categorías:

- Ontologías de tareas: describen el vocabulario relacionado con una tarea o actividad genérica, proporcionando un vocabulario sistemático de términos utilizados para resolver problemas relacionados con las tareas que no pertenecen al mismo dominio. Estas ontologías utilizan el conocimiento del dominio para buscar información, mientras que otra podría gestionar la asignación de bloques de memoria libres.

- Ontologías de dominio: contienen todos los conceptos asociados a un dominio en particular.

- Ontologías generales: las ontologías comunes o generales se utilizan para exponer el conocimiento reutilizable común a través del dominio. Contiene vocabulario relacionado con objetos, eventos, relaciones temporales, relaciones causales, modelos a seguir y funcionalidades.

Otros autores mencionan una categoría intermedia llamada ontologías centrales, que cubren los conceptos más importantes en un dominio determinado. Por ejemplo, la Red Semántica en UMLS 3 contiene conceptos médicos generales como enfermedad, hallazgo, síndrome, por lo que es una ontología médica central. Otros diferencian entre el dominio de la aplicación y las ontologías de la tarea de la aplicación [181]. El primero instancia el conocimiento de dominio de propósito general a restricciones de aplicación particulares, mientras que el segundo corresponde de manera similar a las ontologías de aplicación introducidas por [182], a una combinación de conocimiento declarativo y procedimental relevante para el dominio.

Una última categoría de ontologías, son las llamadas meta-ontologías u ontologías de representación del conocimiento: describen las primitivas que se utilizan para formalizar el conocimiento de conformidad con un paradigma de representación específico. The Frame Ontology [175] o las ontologías de representación de los lenguajes de Web Semántica del W3C RDFS y OWL. 4 son ejemplos bien conocidos de esta categoría

OWL

En los últimos años, se han desarrollado una variedad de lenguajes de ontología, por lo que estos se han adoptado en el contexto de la Web Semántica [183].

Según [184], el Consorcio World Wide Web ha contribuido mucho a estandarizar la especificación necesaria para la tecnología de Web Semántica al introducir el Marco de Descripción de Recursos (RDF) y el Esquema RDF [185].

OWL (Web Ontology Language) es una propuesta de estandarización del lenguaje ontológico especificada por un grupo de trabajo del W3C Web Ontology que ayudaría a resolver los impedimentos actuales para la construcción cooperativa de ontologías entre diferentes plataformas de construcción ontológica, así como potenciar la Web Semántica. OWL proporciona un lenguaje que utiliza la conexión proporcionada por RDF para agregar las siguientes capacidades a las ontologías:

- Capacidad para distribuirse en múltiples sistemas.
- Escalable a las necesidades de la Web.
- Compatible con estándares Web e internacionalización de accesibilidad.
- Abierto y extensible.

El primer borrador de la especificación del lenguaje (OWL 1.0) apareció en julio de 2002 y fue presentado formalmente por el W3C en febrero de 2004 [186]. En octubre de 2009 llegó la última versión hasta ahora, la 2.0.
En OWL 1.0, se definen los siguientes tres lenguajes de expresión subincrementales:

OWL Lite
Es el sub-lenguaje menos expresivo. Proporciona una ruta de migración rápida para tesauros y otras taxonomías. Está destinado principalmente a usuarios que requieren una clasificación jerárquica y restricciones simples.

OWL DL
Proporciona la máxima expresividad de cálculo que incluye todas las construcciones del lenguaje OWL, pero solo se puede usar con ciertas restricciones. Su nombre es bien conocido por su correspondencia con las lógicas descriptivas (DL), por el hecho de asegurar la integridad y finitud de los argumentos para la base formal de OWL.

OWL Full
Está destinado a usuarios que requieren la máxima expresividad y libertad sintáctica por RDF pero sin garantías computacionales. Permite que la ontología aumente el significado de vocabulario predefinido, RDF o OWL. En la versión 2.0 de OWL se definen tres nuevos sub-lenguajes con útiles propiedades de procesamiento. Además, la nomenclatura de los sub-lenguajes se ha modificado al concepto de perfil, orientándose a las características particulares en función principalmente de la expresividad requerida por la aplicación, la prioridad que da el razonamiento sobre clases o datos, el

tamaño de los conjuntos de datos y la importancia de escalabilidad, entre otras cosas.

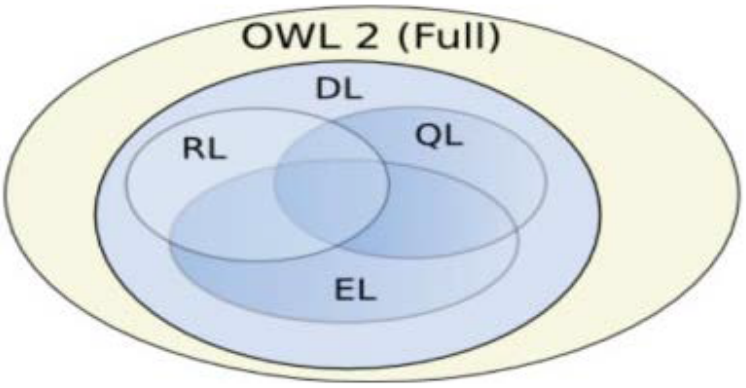


Figura 2.4: Estructura OWL 2. Fuente: Google Image.

OWL 2
Está diseñado para aplicaciones que emplean ontologías que contienen una gran cantidad de propiedades y/o clases. El acrónimo EL refleja el origen del sublenguaje en la familia EL de lógica descriptiva. Es simple de implementar y permite una gran escalabilidad para expresiones complejas; sin embargo, la expresividad que proporciona es bastante limitada. No permite la negación y disyunción de clases, cuantificación universal de propiedades ni propiedades inversas.

OWL 2QL
Este perfil se recomienda para aplicaciones con un gran volumen de datos de instancia, y donde el rendimiento de las consultas es más importante, está diseñado para aplicaciones que priorizan la interoperabilidad de OWL con bases de datos. Es una variante de OWL-Lite, que es muy común en las tareas de integración de bases de datos. Es adecuado para aplicaciones con ontologías ligeras que tienen un gran número de personas que requieren acceso a los datos. Es fácil extender los lenguajes relacionales habituales como SQL, incorporando consultas con axiomas definidos por subconjunto. El razonamiento se puede implementar de manera eficiente reescribiendo técnicas para consultas. Por último, facilita el mapeo entre UML y diagramas Entidad-Relación dando una representación esquemática inmediata de los datos. En cuanto a la expresividad, todavía es bastante limitada.

OWL 2 RL
Es adecuado para aplicaciones que requieren un razonamiento escalable sin sacrificar demasiada expresividad. Está dirigido principalmente a aplicaciones que priorizan la interoperabilidad de OWL con máquinas de reglas, definiendo un subconjunto sintáctico de reglas para su implementación a través de tecnologías basadas en reglas, facilitando el razonamiento. Dichas implementaciones permiten la operación directa en triples RDF y pueden aplicarse arbitrariamente a gráficos RDF. Los lenguajes

descritos se representan esquemáticamente en la Figura 2.4.

Lenguajes de modelado de servicios web

El WSML [185] proporciona una semántica de sintaxis formal para la ontología de modelado de servicios web de (WSMO). WSML se basa en diferentes formalismos, a saber, lógica descriptiva, lógica de primer orden y programación lógica. WSML proporciona una selección significativa de variantes capaces de adaptarse tanto a requisitos como al dominio de la aplicación de destino. Estos son: WSML-Core, WSML-DL, WSML-Flight, WSML-Rule y WSML-Full. La familia de lenguajes WSML proporciona una sintaxis legible por humanos y dos adicionales, a saber, XML, RDF, más una asignación a OWL para el intercambio entre equipos de cómputo.Ver figura 2.5

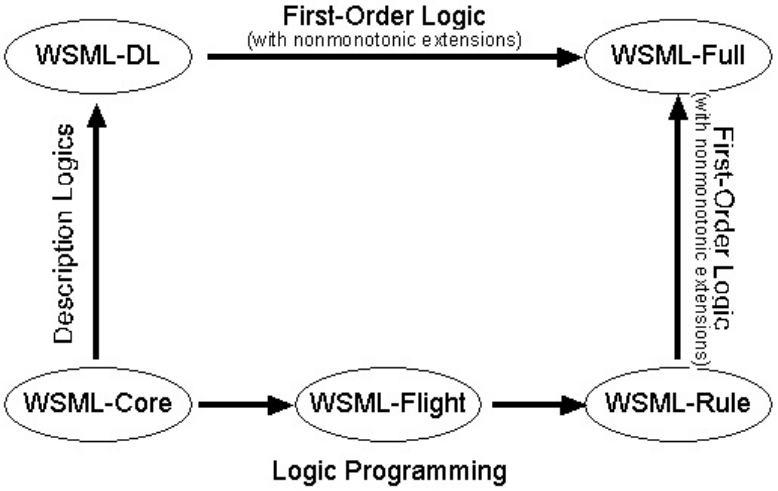


Figura 2.5: Lenguajes de modelado de servicios web. Fuente: Google Image.

WSML-core Se basa en OWL Lite, que es una versión restringida de OWL Lite, lo que lo hace totalmente compatible con este subconjunto de OWL. El formalismo utilizado, cumple precisamente con la intersección de la Lógica descriptiva y Horn Logic (sin símbolos de función y sin igualdad), extendido con soporte de tipo de datos. El lenguaje proporciona los medios para modelar clases, atributos, relaciones binarias (jerarquías de clases y jerarquías de relaciones) e instancias. Debido a que tiene el poder menos expresivo de toda la familia WSML idiomas, WSML-core ofrece las características computacionales más favorables, por lo tanto, es el idioma preferido para aplicaciones prácticas. WSML-Core se amplía, tanto en la dirección de la lógica de descripción como en la dirección de programación con WSML-DL y WSML-Flight respectivamente [184].

WSML-DL

WSML-DL [187]. Extiende la sintáctica y semántica WSML-Core a un paradigma de lógica de descripción completo, a saber, SHIQ, cubriendo así la parte de OWL que se puede implementar de manera eficiente. WSML-DL amplía WSML-Core proporcionando una sintaxis menos restrictiva. En esta dirección, la mayoría de las

limitaciones en el uso de la sintaxis WSMD-DL son las derivadas del uso de lógicas de descripción.

WSML-Flight
WSML-Flight [188] proporciona un poderoso lenguaje que extiende WSML-Core en la dirección de la programación lógica. Proporciona un amplio conjunto de primitivas de modelado para atributos, como restricciones de valor y restricciones de integridad. WSML-Flight es semánticamente equivalente a Datalog con desigualdad y negación estratificada (localmente). Incorpora un lenguaje de reglas en toda regla, al mismo tiempo que permite un razonamiento eficiente.

WSML-Rule
Según [189]. WSML-RUL, extiende WSML-Flight en la dirección de la programación lógica, proporcionando un lenguaje de programación lógica completamente desarrollado. Admite el uso de símbolos de función y reglas inseguras y no restringe el uso de variables en expresiones lógicas.

WSML-Full
WSML-Full [189]. Unifica todas las variantes de WSML-DL y WSML-Rule bajo un marco común de primer orden con una extensión no monótona de WSML-Rule.

Metodologías de ingeniería ontológica

Cuando se crea contenido semántico, ya sean ontologías o instancias de las mismas, se deben seguir una serie de pasos, en algún orden particular, para garantizar que los resultados se entreguen en un plazo razonable, a un nivel de calidad razonable y con costos razonables. Este proceso involucra varios actores, incluidos expertos en el dominio, ingenieros del conocimiento y desarrolladores de software, que trabajan en colaboración para entregar un agente semántico específico de acuerdo con un conjunto de requisitos del usuario, para asegurar la operacionalización del proceso subyacente en términos de resultados, mano de obra y duración. Se han realizado esfuerzos en la comunidad de la Web Semántica para comprender el ciclo de vida del contenido semántico y para diseñar metodologías que proporcionen descripciones del proceso mediante el cual las necesidades del usuario se traducen en agentes semánticos.

En general, una metodología se puede definir como un sistema integrado por una serie de técnicas o métodos que crean una teoría general de sistemas de cómo debe realizarse una clase de trabajo intensivo en pensamiento [186]. En particular una metodología incluye una descripción del proceso a realizar y de los participantes así como de los roles involucrados en el proceso, asigna responsabilidades a actividades y personas y da recomendaciones en forma de mejores prácticas y directrices. Puede estar relacionada con un modelo de proceso específico, que proporcione detalles adicionales sobre el orden y las relaciones entre las actividades previstas por el ciclo de vida correspondiente. En analogía con otras disciplinas, en particular Ingeniería de Software, las metodologías para la creación de contenido semántico suelen utilizar

modelos como Waterfall [190], el modelo en espiral [191] o alguna forma de desarrollo ágil.

WSML-Full
Metodologías para la creación de ontologías Según [192]. Las metodologías de ingeniería de ontología se pueden dividir en dos categorías principales, dependiendo del entorno en el que se puedan aplicar:

- Ingeniería de ontología centralizada: El equipo de ingeniería de ontología está concentrado en un solo lugar, la comunicación entre los miembros del equipo ocurre en reuniones regulares cara a cara. Esta configuración es más relevante para el desarrollo de ontologías para un propósito específico dentro de una organización.

- Ingeniería de ontología descentralizada: Esta configuración es más relevante en el contexto de la Web Semántica o en otros entornos distribuidos abiertos a gran escala. El equipo de ingeniería de ontología está compuesto por individuos dispersos en varias ubicaciones geográficas y afiliados a diferentes organizaciones. La comunicación dentro del equipo suele ser asíncrona. La ontología proporciona una lengua franca entre diferentes partes interesadas o asegura la interoperabilidad entre máquinas, humanos e incluso entre ambos.

Ejemplos de metodologías que pertenecen a la primera categoría son IDEF5 [193], METHONTOLOGY [194] o la metodología OTK [195]. IDEF5 y METHONTOLOGY describen el proceso de diseño de ontología genérica. Comparado con el modelo de proceso introducido en IDEF5 y METHONTOLOGY, OTK no prevé tareas o actividades adicionales, sino que integra el proceso de ingeniería ontológica en un marco más completo para la creación de aplicaciones de gestión del conocimiento basadas en ontologías.

Metodología IDEF5
Es una metodología de ingeniería ontológica que apoya la creación de ontologías para entornos centralizados [193]. Está bien documentada, se divide en cinco actividades principales: organizar y definir proyectos, recopilar datos, analizar datos, desarrollar la ontología inicial, perfeccionar y validar la ontología. La organización y la actividad de definición definen los aspectos administrativos de la ontología. Durante la actividad de recopilación de datos, el análisis del dominio se realiza para determinar y explotar las fuentes de conocimiento. El resultado de la actividad de analizar datos es una primera conceptualización del dominio. En las siguientes actividades, los ingenieros de ontología comienzan a definir los llamados Proto-Conceptos, que son conceptos de alto nivel que caracterizan al dominio y que se refinan con relaciones y axiomas hasta que la validación dé como resultado una ontología que cumpla con los requisitos definidos al principio.

METHONTOLOGY
METHONTOLOGY pertenece a la ingeniería ontológica y es la metodología más completa, ya que se puede utilizar para construir ontologías desde cero o reutilizando

otras ontologías tal como están, o mediante un proceso de reingeniería [194]. El marco permite la construcción de ontologías tanto en el nivel de conocimiento como en el nivel conceptual, en contraposición al nivel de implementación. El marco consiste en: identificación del proceso de desarrollo de la ontología con la identificación de las principales actividades, tales como, evaluación, configuración, gestión, conceptualización, integración o implementación; un ciclo de vida basado en prototipos en evolución; y la metodología en sí misma especifica los pasos para realizar las actividades, las técnicas utilizadas, los resultados y su evaluación. El proceso para construir una ontología centralizada se describe en detalle. Sin embargo, METHONTOLOGY no proporciona orientación para la ontología descentralizada y no se centra en los procesos posteriores al desarrollo.

Metodología OTK
La metodología OTK divide el proceso de ingeniería ontológica en cinco pasos principales y cada uno de ellos tiene numerosos subpasos, que requieren una toma de decisión principal al final y que dan como salida un resultado especial [196]. Las fases son:

1. Estudio de viabilidad.
2. Kickoff.
3. Refinamiento.
4. Evaluación.
5. Aplicación y evolución.

Los subpasos de refinamiento son: Refinar la descripción de la ontología semiformal, formalizar en la ontología de destino, crear prototipo.

Los documentos resultantes de cada fase son: p. ej., para la fase inicial, un documento de especificación de requisitos de ontología (ORSD) y la descripción de ontología semiformal. Los documentos son la base de las principales decisiones que deben tomarse al final para pasar a la siguiente fase, por ejemplo, ya sea en la fase Kickoff que se han captado suficientes requisitos. La metodología OTK describe completamente todos los pasos que son necesarios para construir ontologías de conocimiento centralizado. Los requisitos de la configuración descentralizada fueron los principales impulsores de la definición de las metodologías de ingeniería de ontologías HCOME y DILIGENT.

HCOME
En [197], los autores presentan un enfoque muy reciente para el desarrollo de ontologías. HCOME, que significa metodología de ingeniería de ontología centrada en el ser humano, apoya el desarrollo de ontologías de forma descentralizada. HCOME presenta tres espacios diferentes en donde las ontologías pueden ser almacenadas. El primero es el espacio personal. En este espacio los usuarios pueden crear y fusionar

ontologías, controlar versiones de ontologías, asignar términos y palabras así como sentidos a conceptos y consultar la ontología superior. En el espacio personal las ontologías se pueden compartir en el espacio, lo que permite que puedan ser accedidas por todos los participantes. En el espacio compartido los usuarios pueden discutir decisiones ontológicas basadas en el modelo IBIS [198]. Después de una discusión y un acuerdo la ontología se mueve al espacio acordado. HCOME no proporciona descripción detallada de los pasos del proceso a seguir para llegar a un acuerdo entre los participantes.

DILIGENT
Es una metodología de ingeniería ontológica que aborda los requisitos de un escenario de gestión del conocimiento distribuido [199]. DILIGENT distingue cinco etapas principales que se repiten interactivamente, a saber, construcción central, adaptación local, análisis central, revisión central y actualización local. Un equipo compuesto por ingenieros de ontología comienza el proceso construyendo una pequeña ontología inicial que se distribuye a los usuarios. A los usuarios les es permitido adaptar localmente la ontología compartida para cumplir requisitos. Los cambios realizados por los usuarios sirven como entrada para una próxima versión de la ontología compartida. Después un equipo compuesto por ingenieros ontológicos y los usuarios, actualizan la ontología compartida en la etapa central de análisis y revisión. Finalmente, los usuarios actualizan localmente su ontología compartida a la nueva versión. De este modo, la ontología compartida responde a los requisitos emergentes, mientras que el proceso permite reducción de costos a través de pequeños costos de instalación en comparación con una central.

Un problema importante en DILIGENT se relaciona con el proceso de construcción de consenso, debido a que los modelos de conocimiento heterogéneos creados por los usuarios deben estar parcialmente integrados en la ontología compartida. DILIGENT apoya el proceso de construcción ampliando un modelo de argumentación existente y adaptando a los requisitos de las discusiones de ingeniería ontológica. Sugiere un conjunto restringido de tipos de argumentos y, por lo tanto, ofrece una guía sistemática para las discusiones. Como resultado, el proceso de acuerdo se vuelve más estructurado y rastreable.

Ingeniería de ontologías basada en Wikis
Tiene como objetivo la separación de los desarrollos de ontologías y desarrollos de árboles de conceptos y similares. Los motores Wiki semánticamente mejorados apoyan la fase conceptual del proceso de ingeniería de ontología [200]. Esta fase la realizan todos los interesados utilizando el Wiki como herramienta colaborativa. Los conceptos se definen y describen no sólo por un equipo experto, sino que además es abierto a todos los interesados en los resultados. Posteriormente, se extendió formalmente. Este enfoque es especialmente adecuado para empresas que pretenden establecer y definir un glosario para toda la empresa y, al mismo tiempo, definir un modelo de datos común para la integración de aplicaciones.

Ingeniería de ontología de juegos
Otro enfoque interesante para la obtención de conocimientos se presenta en [201]. Allí, se desarrollan juegos para encontrar conceptualizaciones compartidas de un dominio. Durante el juego, los jugadores describen imágenes, texto o videos. Los usuarios obtienen puntos si describen el contenido de la misma manera. Su aportación está formalizada y traducida a OWL.

Ingeniería de ontologías de etiquetado
El etiquetado es un enfoque muy exitoso para organizar todo tipo de contenido en la web. Los usuarios etiquetan, es decir, agregan descripciones breves a sus marcadores, fotos, artículos, weblogs y otros tipos de contenido. Las etiquetas a menudo describen el significado del contenido etiquetado en un término. Establecido sobre esta observación, [201] introdujo una ingeniería de ontología basada en una metodología de etiquetado. La metodología introduce un proceso de cuatro pasos que comienza con aparición de ideas, consolidación en comunidades, formalización y axiomatización. Se utiliza una herramienta de etiquetado para apoyar este proceso.

Aplicaciones de ontologías

A continuación, se describen algunas de las aplicaciones de ontologías de mayor impacto:

Aplicaciones en biomedicina
Las ontologías han ganado mucha importancia en las últimas dos décadas, especialmente en el ámbito biomédico. Varias ontologías biomédicas como la Gene Ontology (GO), el National Cancer Institute theasurus (NCIt) en Estados Unidos, el Foundational Model of Anatomy (FMA), y la Standarized Nomenclature of Medicine (SNOMED-CT) han surgido y mantenido con el pasar del tiempo además de ser utilizadas ampliamente en la anotación de registros, la estandarización de formatos, la representación e integración de conocimientos y la toma de decisiones.

- **Gene Ontology (GO)**
  El proyecto Gene Ontology (GO) es un esfuerzo colaborativo para abordar dos aspectos de la integración de la información: proporcionar descriptores consistentes para productos genéticos, en diferentes bases de datos; y la estandarización de clasificaciones para secuencias y características de secuencia. El proyecto comenzó en 1998 como una colaboración entre tres bases de datos: FlyBase (Drosophila), Saccharomyces Genome Database (SGD) y Mouse Genome Informatics (MGI). Desde entonces, el Consorcio GO ha crecido para incluir muchas bases de datos, incluidos varios de los principales repositorios de genomas de plantas, animales y microbios del mundo [202].

- **National Cancer Institute theasurus (NCIt)**
  El National Cancer Institute theasurus (NCIt) es una terminología de referencia y una ontología biomédica utilizada por el National Cancer Institute (NCI)

de Estados Unidos y un número creciente de otros sistemas. NCIt se publicó por primera vez en el año 2000 y tenía la intención de facilitar la interoperabilidad y el intercambio de datos entre los diversos componentes de NCI. Es el recurso terminológico central publicado como parte de Enterprise Vocabulary Services (EVS), un conjunto de servicios y recursos que proporcionan terminología controlada al NCI y sus colaboradores, es actualizado mensualmente [203].

- **Foundational Model of Anatomy (FMA)**
  El Foundational Model of Anatomy, [204] es una ontología de dominio de los conceptos y relaciones que pertenecen a la organización estructural del cuerpo humano. Abarca los objetos materiales desde los niveles moleculares hasta los macroscópicos que constituyen el cuerpo humano y asocia con ellos entidades no materiales (espacios, superficies, líneas y puntos) necesarias para describir las relaciones estructurales. El enfoque de modelado disciplinado empleado para el desarrollo de FMA se basa en un conjunto de principios declarados, esquemas de alto nivel, definiciones aristotélicas y un entorno de autor basado en marcos.

- **Standarized Nomenclature of Medicine (SNOMED-CT)**
  Se basa [205] en una taxonomía de más de 390 000 conceptos vinculados a términos y sinónimos multilingües. Además de este componente terminológico, SNOMED CT cuenta con una capa ontológica independiente del lenguaje, compuesta por una gran columna vertebral taxonómica poli-jerárquica enriquecida por axiomas formales que conecta conceptos a través de las jerarquías y proporciona criterios necesarios, SNOMED CT proporciona un mecanismo para crear vocabularios específicos definidos por los usuarios, denominados subconjuntos.

Aplicaciones en la Geología

- **GeoCore Ontology**
  Contiene definiciones bien fundamentadas de un conjunto limitado de conceptos generales dentro del campo de la geología que actualmente son considerados por todos los geólogos, independientemente de su habilidad [206]. Permite a los modeladores considerar por separado un objeto geológico, la sustancia que lo constituye, los límites que lo limitan y la disposición interna de la materia dentro de él. La ontología central también permite la descripción de las cualidades existencialmente dependientes asociadas a un objeto geológico y el proceso geológico que lo generó en una edad geológica particular.

Aplicaciones en servicios web

- **Webulous Google Add-On**
  Webulous [207] proporciona una infraestructura para especificar plantillas con la finalidad de poblar patrones de diseño de ontologías que se transforman en afirmaciones de OWL en una ontología de destino. Webulous proporciona

acceso programático al servidor de plantillas y a un cliente. Además se ha desarrollado una aplicación para hojas de cálculo de Google que permite cargar, rellenar y volver a enviar plantillas a el servidor Webulous de manera que pueda ser procesada.

2.5. Trabajos relacionados

A continuación, se describen investigaciones que combinan clustering, selección de atributos y métodos ontológicos para la clasificación de texto, las cuales tienen estrecha relación con la presente investigación.

En [208], los autores proponen un nuevo enfoque para el preprocesamiento de los datos con el fin de mejorar los resultados de agrupación. Preprocesaron los datos de entrada aplicando heurística basada en ontología para la selección y agregación de características. Basándose en estas representaciones, calcularon múltiples resultados de agrupamiento usando el algoritmo K-means. Los resultados obtenidos se pueden explicar mediante la correspondiente selección de conceptos en la ontología.

En [209], se presenta un sistema de agrupamiento jerárquico modificado para indexación ontológica. Sus hallazgos mostraron que las ontologías pueden imponer una interpretación o agrupamiento subjetivo en un documento conjunto que sea al menos tan bueno como la búsqueda de metapalabras.

En [210], los autores realizaron una investigación cuyo objetivo fue identificar grupos de trastornos del desarrollo y representarlos en una ontología. La ontología se utiliza para interpretar y mejorar los resultados de la agrupación y los resultados de la agrupación se utilizan para validar la ontología y sugerir direcciones para su desarrollo. La metodología utilizada combina una ontología con un método de agrupamiento para apoyar la identificación y representación sistemática de grupos de trastornos del desarrollo.

En [211], los autores presentan dos métricas de evaluación de características para el clasificador Naive Bayes Multi-class Odds Ratio (MOR) y Class Discriminating Measure (CDM), aplicado en conjuntos de datos de texto de clases múltiples. Como indican los resultados, CDM y MOR obtienen mejores resultados que otros enfoques de selección de características.

En [212], los autores proponen un modelo transformado de indexación semántica latente (LSI) que puede capturar adecuadamente la similitud semántica asociada a una ontología basada en corpus. Para analizar los métodos de ontología más eficientes en la agrupación de texto, implementan dos estrategias híbridas que utilizan varias medidas de similitud. Los resultados de los experimentos muestran que el método de algoritmo genético junto con la estrategia de ontología en combinación de la medida basada en LSI transformada con la medida basada en el tesauro, supera al de las medidas tradicionales de similitud. El algoritmo de agrupamiento propuesto mejora

el rendimiento en comparación con K-means en los mismos entornos de similitud.

En [213], los autores desarrollan una ontología de dominio y construyen un nuevo modelo de espacio vectorial conceptual en la etapa de preprocesamiento de la agrupación de texto, sustituyendo la matriz inicial (matriz léxico-texto) en el análisis semántico latente con una matriz de concepto-texto. En la etapa de análisis de clustering, este modelo adopta la similitud semántica, superando parcialmente la dificultad de evaluar de manera precisa y efectiva el grado de similitud del texto. Los resultados experimentales indican que este método es útil para mejorar el resultado de la agrupación de texto.

En [214], los autores discuten las ventajas y desventajas del algoritmo K-means, para posteriormente combinarlo con un conjunto de datos basado en ontologías, estableciendo un modelo de web semántica el cual mejora el algoritmo de agrupamiento existente demostrando que el algoritmo mejorado tiene una mejor eficiencia y precisión en web semántica.

En [184], los autores proponen un esquema de recuperación semántica basado en ontologías de dominio para la búsqueda de conocimiento y la recuperación de información tipo texto. En el esquema desarrollado, la ontología de dominio se construye utilizando el enfoque basado en gráficos para automatizar la construcción de la ontología de dominio, y luego se adopta la extensión y recuperación semántica de consultas para la recuperación de conocimiento basada en la semántica. Para la extensión semántica de la consulta, se adopta el análisis semántico latente para descubrir las relaciones semánticas entre las consultas y las características semánticas de la ontología. Para la recuperación semántica, se propone un método de K-means basado en gráficos para realizar clustering, empleando una estrategia de búsqueda jerárquica para la recuperación de documentos. Finalmente, los resultados experimentales evidencian los beneficios del modelo propuesto.

En [215], los autores realizan un estudio empírico sobre el agrupamiento automatizado de texto, aplicado a artículos científicos y textos de periódicos en portugués brasileño. El objetivo fue encontrar el método computacional más efectivo capaz de agrupar la entrada de textos en sus grupos originales. Teniendo en cuenta los experimentos llevados a cabo, los resultados de la clasificación de textos humanos y la agrupación automatizada son lejanos, de igual manera encontraron que los resultados de corrección de agrupamiento varían según el número de textos de entrada y sus temas.

En [216], se propone un enfoque novedoso basado en ontologías para la agrupación en clústeres de servicios web. Se centran en la similitud y especificidad de los términos para la generación de ontologías. La cantidad de información específica del dominio incluida en un término se usa para definir la especificidad de ese término. Los términos específicos son más relevantes que los términos generales para describir una gran cantidad de información de dominio. Aprovechando esto, generan una nueva ontología, que luego se utiliza para calcular la similitud mediante la definición

de nuevos filtros basados en la lógica. Basado en una evaluación integral que realizan para evaluar el rendimiento del método, demostró ser más efectivo en términos de precisión, recuperación, pureza y entropía que otros enfoques de agrupamiento existentes.

En [217], los autores describen los métodos de construcción de ontologías, proponiendo un método de extensión de ontologías automático basado en el aprendizaje supervisado y la agrupación de textos. Este método utiliza el algoritmo de agrupación de K-means para separar el conocimiento del dominio y guiar la creación del conjunto de entrenamiento para el clasificador Naive Bayes.

En [218], los autores presentan un enfoque para mejorar los conjuntos de características de entrada del método DC y mejorar la precisión. Proponen un enfoque de dos etapas. En la primera etapa, se emplea un diccionario específico de dominio, a saber, el Unified Medical Language System (UMLS), para extraer las características clave pertenecientes a los conceptos más relevantes como enfermedades o síntomas. En la segunda etapa, se aplica PSO para seleccionar más características relacionadas de las características extraídas en la primera etapa. El rendimiento del enfoque propuesto se evalúa en el conjunto de datos de Informática para la integración de la biología (i2b2) en 2010, conjunto de datos de texto médico ampliamente utilizado. Los resultados experimentales muestran una mejora sustancial por el método propuesto en la precisión de la clasificación.

En [219], los autores proponen un nuevo sistema de recuperación de información semántica que utiliza la selección y clasificación de características para mejorar la puntuación de relevancia. Este sistema está compuesto por un algoritmo de selección de características, una ontología inteligente y un algoritmo de recuperación de información semántica basado en la asignación de Dirichlet latente. Las principales ventajas de los algoritmos propuestos son el aumento de la relevancia, la capacidad de manejar big data y la rápida recuperación de información.

En [220], este estudio propone una metodología para realizar clustering de documentos utilizando ontología de dominio y la ontología WordNet. El principal objetivo de este trabajo es aumentar la calidad de la salida del clúster. De igual manera, se examinan y analizan métodos de selección de atributos. Los documentos se agrupan utilizando el algoritmo K-Means convencional con el proceso de selección de atributos de reducción de dimensiones y clustering basado en densidad. El enfoque de ontología propuesto emplea el conjunto de atributos reducido para agrupar los documentos de texto. Los resultados se comparan con dos enfoques tradicionales en dos conjuntos de datos. Demuestran que el enfoque de agrupamiento basado en ontología es efectivo para agrupar los documentos con alta precisión, recuperación y exactitud.

Capítulo 3

Metodología

En esta investigación se han aprovechado las técnicas de Machine Learning disponibles en combinancion con la Ingeniería Ontológica, con el propósito de desarrollar una metodología que permita la inferencia y representación del conocimiento a partir de bases de datos textuales. Las aplicaciones de ambas temáticas son incalculables en todos los sectores productivos y aplicaciones imaginables. Sin embargo, a pesar de aplicarse ambos tipos de técnicas en dominios de creciente complejidad, como la medicina, la socio-política, etc., son pocos los intentos de integrar dichos tipos de técnicas en problemas o dominios concretos para aprovechar sinérgicamente el potencial que la combinación de dichas temáticas ofrece.

En este capítulo se describe la metodología propuesta en esta Tesis Doctoral. La figura 3.1 muestra la metodología propuesta partiendo de un conjunto de datos original la cual consta de 9 etapas, las 5 primeras (preprocesamiento, clustering, selección de atributos, clasificación y test estadístico). El preprocesamiento y el análisis de datos será realizado mediante el uso de la herramienta Weka versión 3.9. Weka (Waikato Environment for Knowledge Analysis) el cual es software de Machine Learning escrito en Java y desarrollado en la Universidad de Waikato, Nueva Zelanda (Universidad de Waikato, 2020), la utilización de este sistema permite obtener un conjunto de datos que servirá de insumo al análisis ontológico. Este análisis inicia con la validación del clustering dando lugar a las 4 etapas adicionales (validación del clúster, análisis semántico, interpretación y representación de relaciones).

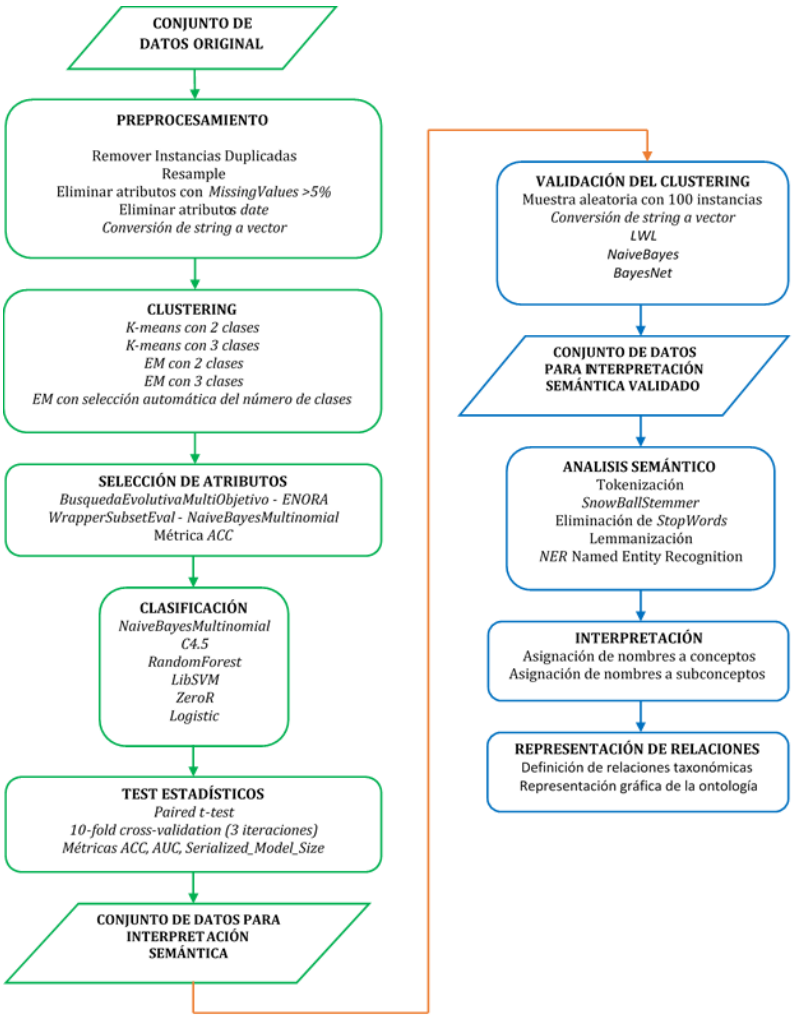


Figura 3.1: Metodología de la investigación.

3.1. Preprocesamiento

Los métodos de preprocesamiento son el primer paso en un proceso de minería de texto y tienen como objetivo transformar la información no estructurada de los archivos de texto en una forma estructurada y ordenada, que luego puede ser interpretada por los algoritmos de aprendizaje automático [212]. Algunas de estas técnicas también pueden reducir el eventual ruido presente en una colección y el espacio necesario para almacenarla. En realidad, se ha comprobado que entre el 20 % y el 30 % del total de palabras de un documento son palabras vacías, es decir, términos que pueden eliminarse por ser repetitivos y no tener valor semántico. En esta sección se presenta un resumen de las prácticas más comunes utilizadas en esta área.

La importancia del preprocesamiento se debe a que los datos en un entorno real contienen impurezas, están incompletos o tienen información difusa. Se encuentran incompletos cuando carecen de valores, de atributos o cuando contienen valores atípicos e inconsistentes que posean discrepancias en los nombres o códigos. Los valores

atípicos y las anomalías en los datos pueden plantear problemas especiales para el analista de datos durante el proceso de depuración de datos. Los datos brutos pueden estar disponibles, pero es necesario que sean útiles. La recopilación de datos es un aspecto importante en todos los campos, pero si la información es irrelevante, será el gran problema (valores perdidos, combinación de datos imposible, valores fuera de rango). Tal problema puede producir resultados engañosos que pueden dar lugar a datos de mala calidad. Si hay mucha información redundante e irrelevante o datos ruidosos y poco fiables, el descubrimiento de conocimientos durante la fase de entrenamiento es más difícil. Los pasos de preparación y filtrado de datos pueden requerir una cantidad considerable de tiempo de procesamiento. El preprocesamiento de datos incluye limpieza, normalización, transformación, extracción y selección de características, etc. El producto del preprocesamiento de datos es el conjunto de entrenamiento final.

El propósito del preprocesamiento es eliminar fragmentos de texto irrelevantes, ya que pueden alterar su significado, obteniendo un rendimiento deficiente del clasificador y redundancia en el análisis [93].

Varios estudios analizaron la influencia de los métodos de preprocesamiento en la CT. A continuación, se presenta un resumen de ellos:

En [212], los autores examinaron 32 combinaciones de cinco métodos de preprocesamiento: eliminación de palabras vacías, derivación de palabras, indexación con frecuencia de términos (TF), ponderación con frecuencia inversa de documentos (IDF) y normalización de cada documento vector de características a la unidad de longitud. Estas combinaciones se aplicaron a dos conjuntos de datos de referencia: Reuters-21578 y 20 Newsgroups. Utilizando una SVM lineal y diferentes longitudes de una representación de bag-of-words que asocia el texto con un vector que indica el número de ocurrencias de cada palabra en la representación del conjunto de datos de entrenamiento. Sus resultados experimentales demostraron que la normalización de cada documento en vectores que representan la información de ocurrencia de palabras pudieron mejorar significativamente la eficacia de los clasificadores de texto.

En [221], los autores implementaron la lematización, tokenización, la derivación y la eliminación de palabras irrelevantes utilizando el clasificador Naive Bayes multinomial en dos conjuntos de datos: 8000 documentos en inglés seleccionados del conjunto de datos “Reuters Corpus Volumen 1” dividido en seis categorías y 8000 documentos Checos proporcionados por Agencia Checa de Noticias, dividida en cinco categorías. Concluyeron que el mejor enfoque de preprocesamiento para la CT es aplicar la eliminación de palabras vacías. Sus experimentos indicaron que la eliminación de las palabras clave mejoró la precisión de la clasificación en la mayoría de los casos.

En [222] fue analizado el uso de eliminación de palabras vacías, derivación y diferentes esquemas de tokenización para la clasificación en correos electrónicos no deseados. Utilizaron tres métodos Machine Learning: Naive Bayes, boosting trees y

SVM. Su principal conclusión fue que el desempeño de SVM es sorprendentemente eficaz cuando no se utilizan derivaciones y eliminación de palabras vacías. Una de las razones es que algunas palabras vacías son poco frecuentes en los mensajes de spam y no deben eliminarse para mejorar el rendimiento del filtrado de spam.

En [93] los autores estudiaron el impacto del preprocesamiento en CT utilizando cuatro métodos de preprocesamiento: tokenización, eliminación de palabras irrelevantes, conversión en minúsculas y derivación. Su estudio se realizó utilizando todas las combinaciones posibles de los métodos de preprocesamiento en cuatro conjuntos de datos: correos electrónicos en turco, correos electrónicos en inglés, noticias en turco y noticias en inglés. Aplicaron el método de Machine Learning SVM usando tamaños de características de 10, 20, 50, 100, 200, 500, 1.000, y unigramas de 2000 palabras. Su principal conclusión fue que las combinaciones apropiadas de las tareas de preprocesamiento, según el dominio y el idioma, pueden proporcionar una mejora significativa en la precisión de la clasificación, mientras que las combinaciones inapropiadas también pueden degradar la exactitud. Otro hallazgo es la importancia de palabras vacías en contraste con muchos estudios de CT, que asumen que las palabras vacías son irrelevantes.

En [223] los autores investigaron el efecto de tres métodos de preprocesamiento (eliminación de palabras vacías, derivación de palabras y normalización de ciertas letras árabes que tienen diferentes formas en la misma palabra en un formulario) en la CT para un corpus interno que contiene 32.620 documentos de noticias dividido en diez categorías descargadas de diferentes sitios web de noticias árabes. En este estudio, Se aplicaron tres métodos de Machine Learning: Naive Bayes, K-NN y SVM. El análisis experimental reveló que el preprocesamiento tiene un impacto significativo en la precisión de la clasificación, especialmente con la estructura morfológica de la lengua árabe. Elegir combinaciones apropiadas de las tareas de preprocesamiento proporcionan una mejora significativa en la precisión de CT dependiendo del tamaño de las características y los métodos de Machine Learning.

El procesamiento descrito en la metodología propuesta consta de las siguientes 6 etapas:

1. Eliminar instancias duplicadas.
2. Resample.
3. Eliminar atributos con valores faltantes $> 5\,\%$.
4. Eliminar atributos tipo date.
5. Conversión de string a vector.

Eliminar instancias duplicadas
Un conjunto de datos puede contener elementos repetidos entre sí en la mayoría de los casos. De manera tal que como parte de la metodología propuesta se realiza este paso con la finalidad de eliminar aquellas instancias repetidas para evitar que se

dé a ese objeto de datos en particular una ventaja o sesgo, cuando se ejecuten los algoritmos de aprendizaje automático.

Resample
Es un método empleado para la selección de un subconjunto extraído de la base de datos en análisis. A menudo se utiliza tanto en la investigación preliminar de los datos como para su análisis final. Usar una muestra funciona casi tan bien como usar el conjunto de datos completo si la muestra es representativa, lo que implica que tiene aproximadamente la misma propiedad (de interés) que el conjunto de datos original. En esta Tesis Doctoral es utilizado el Resample porque trabajar con el conjunto de datos completo resulta costoso computacionalmente, teniendo en cuenta las limitaciones de tiempo y memoria. El uso de un algoritmo de muestreo permite reducir el tamaño del conjunto de datos a un punto en el que se pueden utilizar algoritmos de aprendizaje automático de manera eficiente. En el caso del data set de Euronews luego de aplicar esta técnica se obtiene una muestra representativa con el 10 % del conjunto de datos original.

Eliminar atributos con valores faltantes $> 5\,\%$
Este paso consiste en identificar aquellos atributos que tienen más de 5 % de datos faltantes para eliminarlos del conjunto de datos, esto con la finalidad de obtener un conjunto de datos óptimo para la investigación, se estableció como aceptable un máximo de 5 % de datos faltantes para ser tratados posteriormente.

Eliminar atributos tipo date
Estos valores son significativos cuando se requieren datos temporales. Mientras que en la presente investigación no es de interes conocer cuando fue originada la información, de tal manera que son eliminados con la finalidad de proporcionar a los algoritmos de aprendizaje un data set con información relevante para el análisis.

Conversión de string a vector
Se hace necesaria la utilización de esta técnica debido a que el concepto de una palabra no es entendido por un ordenador, dado que este solo puede comprender valores binarios o numéricos, de tal manera que deben ser convertidos nuestros datos en un lenguaje que pueda ser procesado por el ordenador. Para lograr este proposito es empleada la técnica de word embedding word2Vector propuesta en [224, 225]. Para la utilización de cadenas de texto que puedan ser entendidas por el ordenador, donde son incluidos tanto el significado como el contexto de las palabras lo que excede a otras técnicas de word embedding. Para lograrlo incluye dos algoritmos de aprendizaje, a saber, bag-of-word (CBOW) y skip-gram. La similitud entre palabras es calculada a través de la similitud del coseno de los vectores de las palabras.

Word2Vector comprende y vectoriza el significado de las palabras basándose en la hipótesis de que las palabras con significados similares en un contexto dado exhiben distancias cercanas [226], ver figura 3.2. Ambos algoritmos de aprendizaje exhiben capas de entrada, proyección y salida, aunque sus procesos de derivación de salida son diferentes. La capa de entrada recibe $W_n = \{W_{(t-2)}, W_{(t-1)}, ..., W_{(t+1)}, W_{(t+2)}\}$ como

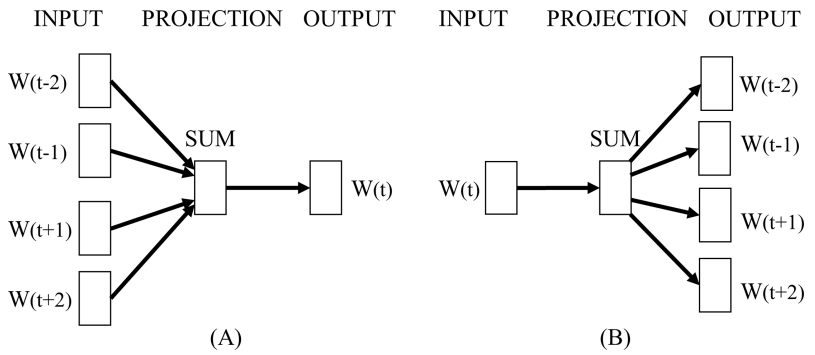


Figura 3.2: Model architecture of (A) CBOW and (B) Skip-gram.

argumentos, donde W_n representa a las palabras. La capa de proyección corresponde a una matriz de vectores multidimensionales y almacena la suma de varios vectores. La capa de salida corresponde a la capa que genera los resultados de los vectores de la capa de proyección. Específicamente, CBOW es similar al modelo de lenguaje de red neuronal (NNLM) [227] y predice la palabra de salida de otros vectores de palabra cercana. El principio básico de CBOW consiste en predecir cuando aparece una determinada palabra mediante el análisis de palabras vecinas. La capa de proyección de CBOW proyecta todas las palabras en la misma posición y, por lo tanto, los vectores de todas las palabras mantienen un promedio y comparten las posiciones de todas las palabras. La estructura de CBOW presenta la ventaja de organizar uniformemente la información distribuida en el conjunto de datos. Por el contrario, el Skip-gram exhibe una estructura para predecir vectores de otras palabras a partir de una palabra. El principio básico de Skip-gram consiste en predecir otras palabras que aparecen alrededor de una determinada palabra. La capa de proyección del Skip-gram predice palabras vecinas alrededor de la palabra insertada en la capa de entrada. La estructura del Skip-gram presenta la ventaja de vectorizar cuando aparecen nuevas palabras. Según el estudio de Mikolov, CBOW es más rápido y más adecuado en comparación con Skip-gram cuando el tamaño de los datos es grande, y Skip-gram exhibe un mejor rendimiento en comparación con CBOW mientras aprende nuevas palabras. Word2Vector nos proporciona una forma de agrupar los datos similares y reducir su dimensión [228].

3.2. Clustering

El procesamiento del leguaje natural, puede lograr un mejor rendimiento mediante la agrupación de palabras similares [229]. Para lograrlo, debemos buscar los centros de cada palabra en el clúster. Como ha sido discutido en el Capítulo I de esta investigación, en la actualidad existen varios algoritmos de agrupación en clústeres que pueden realizar este trabajo.

K-means con 2 y 3 clases
Teniendo en cuenta su simplicidad, que es buen método de agrupamiento para clasificar una gran cantidad de datos numéricos y que los datos asignados a un mismo

grupo son muy similares, implementamos el algoritmo de agrupamiento K-means con $k = 2$ y $k = 3$. K-means nos ha permitido agrupar palabras similares. Con valores dados a k, construimos k conglomerados.

Esperanza-Maximización
El algoritmo general de EM consta de dos pasos, el paso de esperanza y el paso de maximización. Este algoritmo primero entrena a un clasificador usando los datos de etiquetado disponibles y etiqueta los datos sin etiqueta por clasificación estricta (paso de esperanza). Luego entrena un nuevo clasificador usando las etiquetas de todos los documentos (paso de Maximización (M)) e itera hasta la convergencia. EM utiliza el clasificador Naive Bayes en sus dos pasos para encontrar parámetros de máxima verosimilitud (local) de un modelo estadístico en los casos en que las ecuaciones no se pueden resolver directamente. Por lo general, estos modelos involucran variables latentes además de parámetros desconocidos y observaciones de datos conocidos. Es decir, o existen valores faltantes entre los datos o el modelo se puede formular de manera más sencilla asumiendo la existencia de más puntos de datos no observados. Encontrar una solución de máxima verosimilitud normalmente requiere tomar las derivadas de la función de verosimilitud con respecto a todos los valores desconocidos, los parámetros y las variables latentes para resolver simultáneamente las ecuaciones resultantes. En modelos estadísticos con variables latentes, esto suele ser imposible. En cambio, el resultado es típicamente un conjunto de ecuaciones entrelazadas en las que la solución de los parámetros requiere los valores de las variables latentes y viceversa. Como fue definido en el capítulo anterior EM es uno de los algoritmos comúnmente utilizado para la estimación de densidad de puntos de datos en un entorno no supervisado. En este punto de la metodología se utiliza este algoritmo y se inicializa a 2 y 3 clústeres. De igual manera se utiliza el algoritmo EM combinando validación cruzada para seleccionar de manera automática el número de clústeres a utilizar.
Después de la aplicación del clustering fueron generadas 5 bases de datos:

Utilizando el algoritmo K-means:

1. euronewspreprocessed-StringToWordVector-Kmeans2.arff
2. euronewspreprocessed-StringToWordVector-Kmeans3.arff

Utilizando el algoritmo EM:

1. euronews_preprocessed-StringToWordVector-EM2.arff
2. euronews_preprocessed-StringToWordVector-EM3.arff
3. euronews_preprocessed-StringToWordVector-EM9.arff

3.3. Selección de atributos

La cantidad de aplicaciones diferentes con miles de atributos está en aumento, creando la necesidad de utilizar técnicas capaces de manejar un número mucho mayor de

atributos. La selección de atributos reduce el coste computacional al disminuir la dimensionalidad de los datos mediante la eliminación de redundancia y características extrañas. El enfoque wrapper es un tipo popular de evaluador que permite evalúar la relevancia de las características mediante el uso de un clasificador, seleccionando solo el subconjunto más relevante de características. Por tanto, los resultados obtenidos por un wrapper son diferentes a los de un enfoque de filter, porque selecciona un subconjunto de las características más relevantes en lugar de enumerar todas las características en orden de relevancia [230], este modelo requiere un algoritmo de minería predeterminado y utiliza su desempeño como criterio de evaluación, buscando características que se adapten mejor al algoritmo de minería con el objetivo de mejorar el rendimiento, pero tiende a ser más costoso computacionalmente que el modelo de filtro, el cual se basa en las características generales de los datos para evaluar y seleccionar subconjuntos de características sin que involucre ningún algoritmo de aprendizaje. Hemos decidido utilizar el enfoque wrapper que aún siendo más costoso computacionalmente hablando, nos ofrece mayor precisión en la clasificación que un enfoque filter. Además los algoritmos evolutivos son bien conocidos por buscar una solución óptima para el problema de la selección de características [231–234]. En este sentido seleccionamos búsqueda evolutiva multi-objetivo.

La estrategia de búsqueda evolutiva multi-objetivo

La Computación Evolutiva Multi-objetivo (CEM) [62,235] es uno de los pilares sobre los que se sustenta esta tesis doctoral. A continuación se describe el modelo de optimización que usa la estrategia de búsqueda evolutiva multi-objetivo adoptada por los métodos de selección de atributos empleados en la metodología propuesta, junto con el algoritmo *ENORA*, el cual ha servido como algoritmo de optimización para el modelo de optimización multi-objetivo propuesto. También se describe el algoritmo *NSGA-II* ya que sirve de referencia para poder describir el algoritmo *ENORA* y ambos poseen elementos comunes.

El modelo de optimización multi-objetivo

La estrategia de búsqueda evolutiva multi-objetivo resuelve el siguiente problema de optimización booleana de dos objetivos:

$$\begin{array}{l} Maximizar\ F_D^{\Phi}(x) \\ Minimizar\ C(x) \end{array} \tag{3.1}$$

donde $x = \{x_1, x_2, ..., x_w\}$ es un conjunto de variables de decisión booleanas, es decir, $x_t \in \{true, false\}$, $t = 1, ..., w$, siendo w el número de atributos de la base de datos. La función $F_D^{\Phi}(x)$ es una medida de rendimiento de un algoritmo de aprendizaje Φ entrenado con los atributos seleccionados $x_t = true, t = 1, ..., w$ y evaluado con el conjunto de datos D con p-fold cross-validation. En esta tesis se ha usado $p = 5$. p-fold se repite (5 veces como máximo) si la desviación estándar de la media excede un valor umbral, por defecto 0.01. La función $C(x)$ mide el número de atributos

seleccionados, es decir:

$$C(x) = \sum_{t=1}^{w} N(x_t) \tag{3.2}$$

donde N es una función que transforma un valor booleano en numérico ($true = 1$ y $false = 0$). El problema (1) es, por lo tanto, un problema de optimización booleana multi-objetivo donde $x_t = 1$ representa que el atributo x_t está seleccionado y $x_t = 0$ representa que el atributo x_t no está seleccionado, para todo $t = 1, ..., w$. Esto es un problema de optimización NP-hard donde existen 2_w subconjuntos de atributos candidatos.

NSGA-II

El algoritmo *NSGA-II* [64] fue creado por K. Deb en 2002 como una mejora del algoritmo *NSGA* [236] .*NSGA-II* es un AEMO que usa una estrategia elitista $(\mu+\lambda)$ (Algoritmo 1) con $\mu = \lambda = N$, donde μ corresponde al número de padres, λ se refiere al número de hijos y N es el tamaño de la población. *NSGA-II* utiliza *selección por torneo binario* (Algoritmo 2) y una función de ranking basada en *frentes de Pareto y crowding* (Algoritmo 3 y 4) El ranking de un individuo en una población es el nivel de no dominación (o número de frente) del individuo en toda la población. A su vez, los individuos en cada frente son ordenados de acuerdo al nivel de amontonamiento (crowding) de éstos en su frente. *RankCrowding* (Algoritmo 3) es una relación de orden *rank(S,I)* y *rank(S,J)* devuelven el número de frente del individuo I y J respectivamente en un conjunto de individuos S. *NSGA-II* usa *ordenación rápida no dominada* [64] para asignar un número de frente a cada individuo. Compara cada solución con el resto de las soluciones y almacena los resultados para evitar comparaciones duplicadas entre cada par de soluciones. El Algoritmo 1 muestra una implementación de *NSGA-II* en pseudo-código. *NSGA-II* ha demostrado ser un algoritmo muy potente y rápido en contextos de optimización multi-objetivo de todo tipo. La mayoría de los investigadores en CEM usan *NSGA-II* como algoritmo base para comparar el rendimiento de sus propios algoritmos. Aunque *NSGA-II* se desarrolló en 2002, actualmente sigue siendo un desafío superarlo. Existe una versión reciente actualizada para *problemas de optimización many-objective* llamada *NSGA-III* [237].

ENORA

ENORA (Evolutionary NOn-dominated Radial slots based Algorithm) es el algoritmo evolutivo multi-objetivo desarrollado por F. Jiménez y G. Sánchez, en el que se está trabajando intensamente durante la última década. Se ha aplicado *ENORA* a la optimización restringida de parámetros reales [238], optimización difusa [239], clasificación difusa [240], selección de atributos para la clasificación [241, 242], selección de atributos para regresión [63], selección de atributos para clasificación fuzzy [243] y clasificación basada en reglas para datos categóricos [244].

La diferencia entre *NSGA-II* y *ENORA* está en cómo se realiza el ranking de los individuos en la población. *ENORA*, el espacio de los objetivos se divide en slots, y cada individuo pertenece a un slot. El número de slot de un individuo I se calcula de acuerdo con la ecuación 3.3 donde $d = \lfloor \sqrt[l-1]{N} \rfloor$ y h_i^I es f_i (I) normalizado en

Algoritmo 1 *NSGA-II / ENORA*

Require: $T > 1$ {Número de generaciones}
Require: $N > 1$ {Número de individuos en la población}
1: $P \leftarrow InitializePopulation(N)$
2: **for** $I \in P$ **do**
3: $Evaluate(I)$
4: **end for**
5: $t \leftarrow 0$
6: **while** $t < T$ **do**
7: $Q \leftarrow \emptyset$
8: $i \leftarrow 0$
9: **while** $i < N$ **do**
10: $Parent1 \leftarrow BinaryTournamentSelectionMOEA(P)$
11: $Parent2 \leftarrow BinaryTournamentSelectionMOEA(P)$
12: $(Offspring1, Offspring1) \leftarrow EAVariation(Parent1, Parent2)$
13: $Evaluate(Offspring1)$
14: $Evaluate(Offspring2)$
15: $Q \leftarrow Q \cup \{Offspring1; Offspring2\}$
16: $i \leftarrow i + 2$
17: **end while**
18: $R \leftarrow P \cup Q$
19: $P \leftarrow N$ mejores individuos en R de acuerdo a la relación de orden $RankCrowding(SlotRankCrowdingenENORA)$
20: $t \leftarrow t + 1$
21: **end while**
22: **return** Individuos no dominados en la población P

Algoritmo 2 *BinaryTournamentSelectionMOA*

Require: S {Conjunto de individuos}
1: $I \leftarrow$ Individuo aleatorio de S
2: $J \leftarrow$ Individuo aleatorio de S
3: **if** I es mejor que J in S de acuerdo a la relación de orden *RankCrowding*(*SlotRankCrowding* en *ENORA*) **then**
4: **return** I
5: **else**
6: **return** J
7: **end if**

[0,1]:

$$slot(I) = \sum_{i=1}^{l-1} d^{i-1} \lfloor d \frac{\alpha_i^I}{\pi/2} \rfloor \tag{3.3}$$

Algoritmo 3 *RankCrowding*

Require: S {Conjunto de individuos}
Require: $I \in S, J \in S$ {Individuos a comparar en S}
if $rank(S, I) < rank(S, J)$ **then**
  return $True$
end if
if $rank(S, I) > rank(S, J)$ **then**
  return $False$
end if
$F_I \leftarrow \{K \in S | rank(P, K) = rank(P, I)\}$
$F_J \leftarrow \{K \in S | rank(S, K) = rank(S, J)\}$
return $CrowdingDistance(F_I, I) > CrowdingDistance(F_J, J)$

Algoritmo 4 *CrowdingDistance*

Require: S {Conjunto de individuos}
Require: $I \in S$ {Individuos en S}
Require: l {Número de objetivos}
for j=1 to l **do**
  $max_j \leftarrow max_{I \in S}\{f_i(I)\}$
  $min_j \leftarrow min_{I \in S}\{f_i(I)\}$
  $suc_j \leftarrow$ Individuo adyacente superior al individuo I en el objetivo J
  $pre_j \leftarrow$ Individuo adyacente inferior al individuo I en el objetivo J
end for
for j=1 to l **do**
  if $f_j(I) = max_j$ *or* $f_j(I) = min_j$ **then**
    return ∞
  end if
end for
$CD \leftarrow 0.0$
for j=1 to l **do**
  $CD \leftarrow CD + \frac{f_j(suc_j) - f_j(pre_j)}{max_j - min_j}$
end for
return CD

donde :

$$\alpha_i^I = \begin{cases} \pi/2 & \text{if} \quad h_i^I = 0 \\ \arctan\left(\frac{h_i^I + 1}{h_i^I}\right) & \text{otherwise} \end{cases}$$

El ranking de un individuo de la población es el nivel de no dominación del individuo en su slot, es decir, un individuo solo se compara con los individuos que pertenecen a su slot para calcular su ranking. *ENORA* también usa los Algoritmos 1 y 2, excepto que se aplica la relación de orden *SlotRankCrowding* (Algoritmo 5) en lugar de la relación de orden *RankCrowding*.

La razón principal por la que *ENORA* y *NSGA-II* se comportan de manera diferente es la siguiente: *NSGA-II* nunca selecciona a un individuo dominado por

Algoritmo 5 *SlotRankCrowding*

Require: S {Conjunto de individuos}
Require: $I \in S, J \in S$ {Individuos a comparar en S}
1: $S_I \leftarrow \{K \in S \mid slot(S, K) = slot(S, I)\}$
2: $S_J \leftarrow \{K \in S \mid slot(S, K) = slot(S, J)\}$
3: **if** $rank(S_I, I) < rank(S_J, J)$ **then**
4: **return** $True$
5: **end if**
6: **if** $rank(S_I, I) > rank(S_J, J)$ **then**
7: **return** $False$
8: **end if**
9: $F_I \leftarrow \{K \in S \mid rank(S, K) = rank(S, I)\}$
10: $F_J \leftarrow \{K \in S \mid rank(S, K) = rank(S, J)\}$
11: **return** $CrowdingDistance(F_I, I) > CrowdingDistance(F_J, J)$

otro en el torneo binario, mientras que en *ENORA*, un individuo dominado por otro puede ser el ganador del torneo. A título de ilustración, consideremos, sin pérdida de generalidad, un problema ficticio de optimización multiobjetivo con $l = 2$ objetivos $f1$ y $f2$ para la minimización. Suponemos que el tamaño de la población $N = 4$. Entonces, el número de slots en *ENORA* es de $d = \lfloor \sqrt[2-1]{4} \rfloor = 4$. La Figura 3.3 muestra la asignación de ranking de los individuos con *NSGA-II* y *ENORA*. En la Figura 3.3, las líneas discontinuas delimitan el área dominada por cada individuo, y las líneas continuas delimitan los slots en *ENORA*. *NSGA-II* asigna el rank 1 a los individuos A y D ya que son individuos no dominados en la población, asigna el rank 2 al individuo B ya que B está dominado por el individuo A, y asigna el rank 3 al individuo C, ya que C está dominado por el individuo B. Sin embargo, *ENORA* asigna el rank 1 al individuo C, ya que éste es el único individuo en su slot y, por lo tanto, no hay ningún individuo en su slot que lo domine. Entonces, si los individuos B y C son seleccionados para el torneo binario con *NSGA-II*, el individuo B gana al C porque B domina a C. Por el contrario, el individuo C gana B con *ENORA* porque el individuo C tiene un mejor rank en su slot que el individuo B. De esta manera, *ENORA* permite que los individuos en cada slot evolucionen hacia el frente de Pareto incluso si estos individuos están muy alejados, mejorando así la diversidad. Este enfoque genera un mejor hipervolumen que *NSGA-II* durante el proceso evolutivo.

El resto de características de ENORA para selección de atributos son las siguientes:

- Representación binaria de longitud fíja.
- Inicialización aleatoria uniforme.
- Selección por torneo binario.
- Operadores de variación auto-adaptativos: cruce uniforme y mutación flip de un bit.

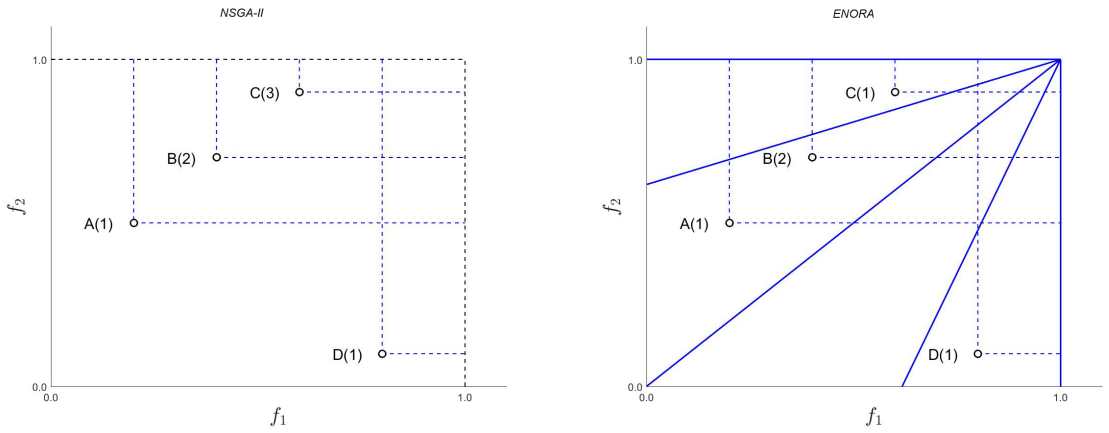


Figura 3.3: Asignación de ranking de individuos en *NSGA-II* vs. *ENORA*.

Métrica ACC
Se utiliza ACC como método para evaluar la precisión del modelo, debido a que nos permite determinar la relación binaria entre las magnitudes de predicciones correctas contra el número total de predicciones. La mátrica ACC se calcula conforme a la ecuación 3.4:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.4}$$

donde TP, TN, FP, y FN representan: true positive, true negative, false positive y false negative, respectivamente.

Con la aplicación de la selección de atributos son generadas 5 bases de datos adicionales para su análisis:

1. euronews_preprocessed-StringToWordVector-Kmeans2-MOES-NBM-ACC.arff
2. euronews_preprocessed-StringToWordVector-Kmeans3-MOES-NBM-ACC.arff
3. euronews_preprocessed-StringToWordVector-EM2-MOES-NBM-ACC.arff
4. euronews_preprocessed-StringToWordVector-EM3-MOES-NBM-ACC.arff
5. euronews_preprocessed-StringToWordVector-EM9-MOES-NBM-ACC.arff

3.4. Clasificación

Una de las aplicaciones más comunes del aprendizaje automático es la clasificación de datos. En esencia, la clasificación de datos investiga las relaciones entre variables de características. Los métodos de clasificación se han utilizado en una amplia gama de aplicaciones como fue descrito en el capítulo anterior de esta tesis Doctoral. Usando métodos de clasificación, los datos son clasificados según sus características

específicas. A partir de este modelo se pueden comprender los datos y predecir su comportamiento. En los métodos de clasificación, los datos dentro de cada clase tienen características similares y comunes, donde las características se encuentran con mayor frecuencia entre las categorías [245].

NaiveBayesMultinomial
Como ha sido descrito en el capítulo anterior, esta técnica es utilizada para evaluar un conjunto de atributos a través de un esquema de aprendizaje el cual emplea validación cruzada para estimar la precisión del esquema de aprendizaje en un conjunto de atributos. Naive Bayes Multinomial se basa en la aplicación del teorema de Bayes para predecir la probabilidad condicional de que un atributo pertenezca a una instancia determinada.

Hemos decidido utilizar este algoritmo por ser un método de aprendizaje supervisado enfocado en casos de clasificación de texto. Este método sigue el principio de distribución multinomial en probabilidad condicional [246]. Aunque se utilizan distribuciones multinomiales, este algoritmo se puede aplicar a casos de texto convirtiéndolo a una forma nominal que se puede calcular con un valor entero. El cálculo de probabilidad se describe en la ecuación 3.5:

$$P\left(c|d\right) \propto P\left(c\right) \prod_{1 \leq k \leq n_d} P\left(t_k|c\right) \tag{3.5}$$

donde: $P\left(t_k|c\right)$ es la probabilidad condicional de que la palabra tk aparezca en el documento con clase c. En la ecuación $P\left(t_k|c\right)$ es la probabiliad de verosimilitud tk en la clase c. Mientras que $P\left(c\right)$ es la probabilidad a priori del documento apareciendo en la clase c. Para la determinación de la clase se comparan los resultados de la probabilidad obtenidos entonces la clase con la probabilidad más grande es la clase elegida como resultado predicho. La fórmula de probabilidad a priori se puede ver en la ecuación 3.6:

$$P\left(c\right) = \frac{N_c}{N} \tag{3.6}$$

N_c es la suma de la categoría c, mientras que N es la suma de todas las categorías. La formula de la probabiliad de verosimilitud puede verse en 3.7:

$$P\left(t_k|c\right) = \frac{T_{tc}}{\sum_{t\prime \in V} T_{ct\prime}\prime} \tag{3.7}$$

T_{tc} es el número de ocurrencias de la palabra t en el documento siendo de la clase c, y $\sum_{t\prime \in V} T_{ct\prime}\prime$ es el número total de ocurrencias de todas las palabras en la clase c.

C4.5
Se ha decidido utilizar el algoritmo de árbol de decisión debido a que es uno de los

métodos de aprendizaje más exitosos, a consecuencia de su simplicidad, comprensibilidad, ausencia de parámetros y la capacidad de poder manejar datos de tipo mixto. El árbol de decisiones se induce a partir de un conjunto de instancias de entrenamiento etiquetadas, representado por una tupla de valores de atributo y una etiqueta de clase.

Debido al amplio espacio de búsqueda, el aprendizaje del árbol de decisiones es un proceso de arriba hacia abajo y recursivo, que comienza con todos los datos de entrenamiento y un árbol vacío. Donde el atributo que mejor divida los datos de entrenamiento es elegido como tributo raíz y los datos de entrenamiento se dividen en subconjuntos disjuntos que satisfacen los valores del atributo de división. Para cada subconjunto, el algoritmo procede de forma recursiva hasta que todas las instancias de un subconjunto pertenecen a la misma clase. Por lo general, los árboles producidos por C4.5 [247] son pequeños y precisos, lo que resulta en clasificadores fiables. Estas propiedades hacen de los árboles de decisión una herramienta valiosa y popular para clasificación. La utilidad de los árboles de decisión está ampliamente aceptada por los investigadores, debido a que el aprendizaje del árbol de decisiones funciona. Por ejemplo, el aprendizaje del árbol de decisiones supera a Naive Bayes en conjuntos de datos más grandes, mientras que Bayes funciona mejor en conjuntos de datos más pequeños. Un problema muy interesante es el de overfitting.

Un árbol de decisiones que clasifique correctamente todos los ejemplos en un conjunto de entrenamiento podría no ser un clasificador tan bueno como un árbol más pequeño que no se ajusta a todos los datos de entrenamiento. Con el fin de evitar este problema, la mayoría de los algoritmos de árboles de decisión emplean un método de “parada”, que significa que hacen crecer un árbol grande y luego eliminan una parte de él. El método es detener el crecimiento del árbol una vez que el conjunto de entrenamiento se ha subdividido lo suficiente. Utilizando un criterio de “parada” [247] ha experimentado con criterios de detención en el pasado y, de hecho, algunas versiones de ID3 utilizaron este enfoque para evitar el sobreajuste. Pero él explica aquí que los resultados fueron desiguales, por lo que ha adoptado el enfoque de poda para C4.5. utilizando un criterio de ”parada”. El método de poda de C4.5 se basa en estimar la tasa de error de cada subárbol, y reemplazar el subárbol con un nodo hoja si el error estimado de la hoja es menor. La idea es la siguiente: suponiendo que se puede estimar la tasa de error de cualquier nodo, incluidos los nodos de las hojas. Comenzando en la parte inferior del árbol, si las estimaciones indican que el árbol será más preciso cuando los hijos del nodo n se eliminen y n sea un nodo hoja, entonces C4.5 eliminará los hijos de n. Si las estimaciones fueran perfectas, esto siempre conduciría a un mejor árbol de decisiones. En la práctica, aunque estas estimaciones son muy bastas, el método a menudo funciona bastante bien.

Random Forest

Es seleccionado este algoritmo debido a que utiliza múltiples combinaciones de árboles de decisión para predecir con precisión datos de prueba. El clasificador de Random Forest supera el problema de sobreajuste de los árboles de decisión mediante la construcción de múltiples árboles de decisión. Los árboles resultantes varían porque

cada árbol está construido con datos aleatorios y características aleatorias, además de que este algoritmo es ampliamente recomendado por su eficiencia en conjuntos de datos grandes.

SVM
SVM es ampliamente utilizada en problemas de clasificación, ya que produce una corrección notable con menos potencia de cálculo.
Dados los datos de entrenamiento etiquetados, el algoritmo genera el mejor hiperplano que clasificó nuevos ejemplos. En el espacio bidimensional, este hiperplano es una línea que divide un plano en dos partes donde cada clase se encuentra a cada lado. La intención del algoritmo SVM es encontrar un hiperplano en un espacio N dimensional que clasifique por separado los puntos de datos. Es un método muy útil si no tenemos mucha idea sobre los datos. Se puede usar para datos que no se distribuyen regularmente y tienen una distribución desconocida. Algo muy importante en tener en cuenta referente a SVM es que puede manejar datos de alta dimensión y esto demuestra ser de gran ayuda teniendo en cuenta su uso y aplicación en el campo del aprendizaje automático. En comparación con otros clasificadores tiene una mayor complejidad computacional e incluso si la cantidad de ejemplos positivos y negativos no es la misma, se puede usar SVM ya que tiene la capacidad de normalizar los datos o proyectarlos en el espacio de la frontera de decisión que separa las dos clases.

ZeroR
Es el procedimiento de referencia para los algoritmos de clasificación cuyo resultado es simplemente la clasificación que ocurre con mayor frecuencia en un conjunto de datos. Si el 70 % de los elementos de datos tienen esa clasificación, ZeroR supondría que todos los elementos de datos la tienen y tendría razón el 70 % de las veces. Representa un punto de referencia simple y efectivo: si un algoritmo predice correctamente las clasificaciones con menos frecuencia que ZeroR, obviamente no tiene valor para el dominio en cuestión.

Para los problemas de clasificación, la única regla es predecir el valor de clase que es más común en el conjunto de datos de entrenamiento. Esto significa que, si un conjunto de datos de entrenamiento tiene 90 instancias de clase “1” además 10 instancias de clase “2”, predecirá “1”, con lo cual logrará una precisión de referencia de 90/100 o 90 %.

Logistic regression
El clasificador de regresión logística es una técnica de clasificación muy popular y ampliamente utilizada. Esto debido a lo simple y fácil de implementar, además de proporcionar un buen rendimiento en una amplia variedad de problemas, como la predicción de correos no deseados. La regresión logística también es mejor para predecir la probabilidad discreta donde la salida de la probabilidad ya sea “si”, “no”, ganar o perder. La regresión logística es fácil de ejecutar y brinda una excelente ejecución en una amplia variedad de temas [248].

Capítulo 4

Experimentos y resultados

4.1. Introducción

En esta tesis doctoral son utilizados dos conjuntos de datos textuales diferentes para su análisis, el primero es smsSpamCollection del Repositorio UCI de Machine Learning (aprendizaje supervisado), smsSpamCollection es un conjunto de datos público conformado de mensajes SMS que han sido recopilados para la investigación de correos no deseados en teléfonos móviles, compuesto de 5574 instancias y 2 atributos [249]. El segundo conjunto de datos utilizado es conformado para esta investigación y tiene como finalidad obtener un conjunto de datos textual lo suficientemente extenso para la utilización de técnicas de Big Data y de Machine Learning, el cual es extraído de la página web de noticias Euronews. La estructura del repositorio de noticias puede verse en la figura 4.1. El conjunto de datos generado está compuesto por 104.434 instancias y 20 atributos.

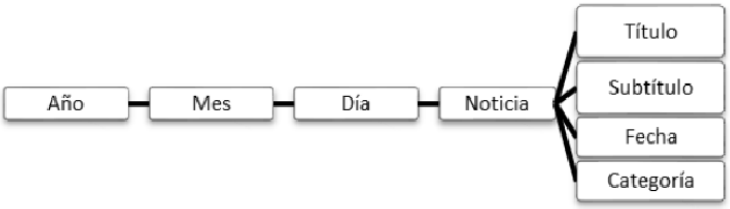


Figura 4.1: Estructura del repositorio de noticias Euronews.

La descarga de los datos se realiza utilizando la herramienta de software libre GNU Wget que permite descargar archivos mediante HTTP, HTTPS, FTP y FTPS, los protocolos de internet más utilizados. Es una herramienta de línea de comandos no interactiva, que puede ser ejecutada fácilmente desde scripts.

Para localizar y extraer cadenas de texto específicas, que permitan la depuración del conjunto de datos, es utilizado GNU Grep, el cual es una herramienta de software libre que permite buscar en uno o más archivos aquellas cadenas de texto que coincidan con un patrón específico de búsqueda.

Con el objetivo de reformatear los datos extraídos se utiliza la herramienta de software libre GNU Sed, aplicada comúnmente para filtrar texto, es decir, toma una entrada de texto, realiza alguna operación (o un conjunto de operaciones) en él y genera el texto modificado, sirve además para extraer partes de un archivo usando patrones de coincidencia o sustituyendo múltiples ocurrencias de una cadena de texto en un archivo (GNU Operating System, 2020).

En resumen el proceso de generar el conjunto de datos euronews consistió de los siguientes pasos:

1. Descarga del directorio de noticias de cada año.
2. Extracción de los enlaces que hacen referencia a los meses.
3. Descarga de las noticias correspondiente a cada mes.
4. Consolidación de las noticias por año.
5. Integración de todas las noticias en un solo conjunto de datos.
6. Formateo de data set.
7. Finalmente se genera el archivo euronews.arff compuesto por 104.434 instancias y 20 atributos.

El código principal del script puede verse en el anexo A.

Preprocesamiento: Uno de los aspectos a tener en cuenta es que si un atributo tiene el mismo valor para todas las instancias en el conjunto de datos no proporciona ninguna información adicional y, por lo tanto, se puede considerar inútil, de manera que en este paso del preprocesamiento serán removidos de nuestro conjunto de datos (Remove useless). Cuantos más datos de entrenamiento, mayor será la tasa de éxito de los clasificadores.

4.2. Experimento 1

Este experimento consiste en:

1. Clasificación con la base de datos resultante (smsSpamCollectionRemoveUseless) utilizando los clasificadores:
   - Stochastic gradient descent Text (SGDtext).
   - NaiveBayesMultinomial para datos textuales.
2. Test estadísticos: en 10−fold cross-validation (30 iteraciones) en las métricas:
   - ACC.
   - AUC.

- Tiempo de entrenamiento.
- Tamaño del modelo.

Data set	SGDText	NBMText
msSpamCollectionRe− moveUseless	98.36	98.43

Tabla 4.1: Percent correct.

4.3. Experimento 2

Este experimento consiste en:

1. Tokenizar, mediante este proceso se corta el texto de entrada (palabras/tokens) manteniendo la secuencia y descartando simultáneamente caracteres específicos.

2. Selección de atributos:

   - Search: BestFirst
     Evaluador: CfsSubsetEval (BF−CFS).
     Data set resultante:
     smsSpamCollection−SToWVector−BF−CGS.

   - Search: MultiObjectiveEvolutionarySearch (ENORA, generations 10000, population size 100, reportFrecuency 10000).
     Evaluador: WrapperSubsetEval.
     Clasificador: trees/RandomForest.
     EvaluationMeasure: Default accuracy.
     Data set resultante:
     smsSpamCollection−SToWVector−MOES−RF−ACC.

   - Search: PSOSearch (generations 10000, population size 100, reportFrecuency 10000).
     Evaluador: WrapperSubsetEval.
     Clasificador trees/J48.
     EvaluationMeasure: Default accuracy..
     Data set resultante:
     smsSpamCollection−SToWVector−PSO−J48−ACC.

   - Search: Ranker (numToSelect 10).
     Evaluador: InfoGainAttributeEval.

Data set resultante:
smsSpamCollection−SToWVector−RANKER−INFOGAIN.

- Search: MultiObjectiveEvolutionarySearch (ENORA, generations 10000, population size 100, reportFrecuency 10000).
  Evaluador: WrapperSubsetEval.
  Clasificador:NaiveBayesMUltinomial.
  EvaluationMeasure: Default accuracy.
  Data set resultante:
  smsSpamCollection−SToWVector−MOES−NBM−ACC.
- Test estadísticos: en 10−fold cross-validation (30 iteraciones) en las métricas:
  - ACC
  - AUC
  - Tamaño del modelo.

Para los conjuntos de datos:
smsSpamCollection−SToWVector
smsSpamCollection−SToWVector−BF−CGS
smsSpamCollection−SToWVector−MOES−RF−ACC
smsSpamCollection−SToWVector−RANKER−INFOGAIN
smsSpamCollection−SToWVector−PSO−J48−ACC
smsSpamCollection−SToWVector−MOES−NBM−ACC

Algoritmos de aprendizaje utilizados:

- J48.
- Random Forest.
- Logistic.
- LibSVM.
- MLPClassifier.
- ZeroR.
- NaiveBayesMultinomial.

Los resultados obtenidos de este experimento pueden verse en la Tabla 4.2

Algoritmo	(4)SmsSpam	(1)SmsS	(2)SmsS	(3)SmsS
trees.J48 −C 0.25 −M 2 (30)	96.15	93.95*	95.94	96.06
trees.RandomForest −P 10 (30)	97.92	93.91*	97.22*	97.36*
functions.Logistic −R 1 (30)	98.39	93.89*	97.54*	95.48*
functions.LibSVM −S 0 −K (30)	93.58	93.70	95.50v	86.60*
functions.MLPClassifier (30)	98.38	94.04*	97.27*	98.27
rules.ZeroR 4805554146 (30)	86.60	86.60	86.60	86.60
bayes.NaiveBayesMultinomi (30)	99.32	92.90*	97.20*	98.61*

Tabla 4.2: Percent correct.

4.4. Experimento 3

Con el conjunto de datos generado en la fase de Preprocesamiento de Euronews (euronews_Preprocessed_StringToWordVector).

- Missing values
- Remove use less

1. Clustering con K-means con 2 y 3 clases:
   data set resultantes:
   - euronews_preprocessedStringToWordVectorNew−Kmeans2
   - euronews_preprocessed_StringToWordVectorNew−Kmeans3
2. Clustering con EM con 2 y 3 clases, y otra con búsqueda automática del número de clústeres
   - euronews_preprocessedStringToWordVectorNew−EM2
   - euronews_preprocessedStringToWordVectorNew−EM3
   - euronews_preprocessedStringToWordVectorNew−EM−1
3. Para cada conjunto de datos anterior:
   Feature selection con:
   - Search: MultiObjectiveEvolutionarySearch (ENORA, generations 10000, population size 100, reportFrecuency 10000).
     Evaluador: WrapperSubsetEval.
     Clasificador:NaiveBayesMUltinomial.
     EvaluationMeasure: Default accuracy.

   Data sets resultantes:
   - euronews_preprocessed−StringToWordVector−Kmeans2−MOES−NBM−ACC

- euronews_preprocessed−StringToWordVector−Kmeans3− MOES−NBM−ACC
- euronews_preprocessed−StringToWordVector−EM2−MOES− NBM−ACC
- euronews_preprocessed−StringToWordVector−EM3−MOES− NBM−ACC
- euronews_preprocessed−StringToWordVector−EM9−MOES− NBM−ACC

Resultados del experimento 3
Mediante la realización del experimento 3 se completan 10 data sets producto de la aplicación de los algoritmos anteriormente descritos ver Tabla 4.3.
Para realizar la evaluación de los conjuntos de datos generados son aplicados los siguientes métodos de evaluación:

- Perfom test utilizando como base al data set número (6) euronews_preprocessed−Strin− gToWordVector−Kmeans2− MOES−NBM−ACC ver Tabla 4.4.
- Perform test utilizando Weighted_avg_area_under_ROC ver Tabla 4.5.
- Perform test using Comparison field: SerializedModelSizever Tabla 4.6.
- Finalmente se realiza Ranking test en los 10 conjuntos de datos ver Tabla 4.7.

.

Número	Data set
1	euronews_preprocessedStringToWordVectorNew−Kmeans2
2	euronews_preprocessed_StringToWordVectorNew−Kmeans3
3	euronews_preprocessedStringToWordVectorNew−EM2
4	euronews_preprocessedStringToWordVectorNew−EM3
5	euronews_preprocessedStringToWordVectorNew−EM−1
6	euronews_preprocessed−StringToWordVector−Kmeans2−MOES−NBM−ACC
7	euronews_preprocessed−StringToWordVector−Kmeans3−MOES−NBM−ACC
8	euronews_preprocessed−StringToWordVector−EM2−MOES− NBM−ACC
9	euronews_preprocessed−StringToWordVector−EM3−MOES − NBM−ACC
10	euronews_preprocessed−StringToWordVector−EM9−MOES− NBM−ACC

Tabla 4.3: Conjuntos de datos resultantes.

Data set/Algoritmo	NBM	J48	RF	LibSVM	MLP Classifier	ZeroR	Logistic
6	1.00	99.99	99.99	99.99	99.95	84.27	99.99
1	99.63*	99.99	1.00*	1.00	99.83*	84.27	99.78
2	1.00*	1.00	1.00*	0.99*	0.99	0.50	1.00*
3	88.80*	83.56*	85.12*	75.10*	93.88*	53.82*	93.55*
4	83.08*	68.04*	74.08*	63.83*	82.54*	35.79*	88.13*
5	65.38*	35.31*	40.25*	22.80*	24.98*	11.52*	51.21*
7	99.23*	99.56*	99.44*	99.21*	87.70*	49.76*	99.18*
8	93.61*	84.61*	89.47*	75.82*	92.19*	53.82*	93.50*
9	88.16*	69.27*	76.29*	64.93*	79.38*	35.79*	88.39*
10	64.10*	36.02*	40.62*	28.80*	24.84*	11.52*	59.75*

Tabla 4.4: Perfom test utilizando al data set número 6 como base.

Data set/Algoritmo	NBM	J48	RF	LibSVM	MLP Classifier	ZeroR	Logistic
6	1.00	1.00	1.00	1.00	1.00	0.50	1.00
1	1.00	1.00	1.00	1.00	1.00	0.50	1.00
2	1.00*	1.00	1.00*	0.99*	0.99	0.50	1.00*
3	0.96*	0.84*	0.94*	0.74*	0.98*	0.50	0.98
4	0.95*	0.80*	0.90*	0.73*	0.93*	0.50	0.97*
5	0.95*	0.71*	0.85*	0.57*	0.73*	0.50	0.91*
7	1.00*	1.00*	1.00*	0.99*	0.98	0.50	1.00*
8	0.98*	0.87*	0.96*	0.74*	0.97*	0.50	0.98*
9	0.97*	0.83*	0.92*	0.73*	0.91*	0.50	0.97*
10	0.95*	0.70*	0.85*	0.60*	0.73*	0.50	0.94*

Tabla 4.5: Perfom test utilizando Weighted_avg_area_under_ROC.

Data set/Algoritmo	NBM	J48	RF	LibSVM	MLP Classifier	ZeroR	Logistic
6	2734.00	4431.00	48249.73	10204.67	15464.00	896.00	12448.00
1	66409.00v	52058.00v	6389810.43v	563087.27v	210164.00v	896.00	172919.00v
2	74525.00v	57809.47v	21377858.10v	1323785.80v	210222.00v	930.00v	181017.00v
3	66325.00v	204465.73v	35887543.13v	2785115.67v	209660.00v	896.00	172499.00v
4	74441.00v	468808.93v	53428101.20v	3261549.80v	209718.00v	930.00v	180597.00v
5	141620.00v	1882852.60v	154446165.33v	3447007.93v	169473.00v	1379.00v	227454.00v
7	12241.00v	14794.13v	5675189.93v	200434.00v	41117.00v	930.00v	34673.00v
8	30466.00v	107588.33v	31492999.20v	1002781.80v	100286.00v	896.00	82372.00v
9	50792.00v	358192.60v	52426890.73v	1643847.00v	146164.00v	930.00v	125560.00v
10	179957.00v	1978561.20v	153209099.87v	4624326.53v	210485.00v	1379.00v	285883.00v

Tabla 4.6: Perform test utilizando Comparison field: Serialized_Model_Size.

Data set	Wins	Losses
6	57	0
5	56	1
10	40	18
8	38	19
3	33	25
9	31	27
4	18	41
1	14	44
7	2	57
2	1	58

Tabla 4.7: Ranking test for data sets.

Capítulo 5

Análisis de los resultados

A continuación, se analizan los principales resultados obtenidos en la investigación realizada en esta Tesis Doctoral. Para ello, en primer lugar, se detallarán los resultados de cada uno de los experimentos realizados descritos en el capítulo anterior. En segundo lugar, se describe la interpretación semántica de las clasificaciones obtenidas. Finalmente, se evalúa la metodología obtenida como sistema de recuperación de información.

5.1. Resultados experimentales

De la clasificación realizada en el Experimento 1, como puede verse en la Tabla 4.1, mediante el uso de los algoritmos SGDtext y NaiveBayesMultinomial para datos textuales, se obtuvo que el porcentaje de aciertos para el algoritmo SGText es de 98.36 mientras que utilizando el algoritmo NaiveBayesMultinomial es de 98.46. Ambos resultados son muy similares sin que exista diferencia estadística entre ellos.

Por su parte, de los resultados obtenidos en el Experimento 2, como se refleja en la Tabla 4.2, se desprende que haciendo SToWVector junto con feature selection mediante el wrapper MOES (estrategia de búsqueda) y NaiveBayesMultinomial (evaluador) con ACC (métrica), se obtienen mejores resultados con el clasificador NaiveBayesMultinomial que con:

1. otros métodos de feature selection (BestFirst + Correlation Feature Selection y Ranker + Information Gain), evaluados con los clasificadores J48, Randon Forest, Logistic, LibSVM, MLPClassifier, NaiveBayesMultinomial y ZeroR

2. los clasificadores de texto SGDText y NaiveBayesMultinomialText.

Por último, en cuanto al Experimento 3, tal como queda patente en las Tablas 4.6, 4.5, y 4.4. Para el conjunto de datos de Euronews, el modelo con los mejores resultados fue (euronews_ preprocessed−StringToWordVec−tor−Kmeans2−MOES−NBM−ACC) con 2 clústeres y 5 atributos, de manera que cuando se utiliza stringToWordVector y

se realiza clustering con el algoritmo K-means y Feature selection MultiObjectiveEvolutionarySearch (tamaño de la población 100, generaciones 10000, reportfrecuency 10000, algoritmo ENORA) y WrapperSubsetEval (NaiveBayesMultinomial con los parámetros por defecto y accuracy como métrica) se obtiene el mejor modelo con un 100 % de probabilidad de acierto.

5.2. Interpretación semántica

Luego de haber evaluado e identificado el mejor data set (euronews_preprocessed−StringToWordVector−Kmeans2−MOES−NBM−ACC) se realiza la interpretación semática siguiendo los siguientes pasos:

1. Validación del clustering extrayendo una muestra aleatoria de 100 instancias.

```
http <= 0: cluster1 (8584.0)
http > 0: cluster2 (1604.0/1.0)

Number of Leaves :   2

Size of the tree :   3

Correctly Classified Instances     10187          99.9902 %
Incorrectly Classified Instances       1           0.0098 %
```

Figura 5.1: Resultados con el data set euronews.

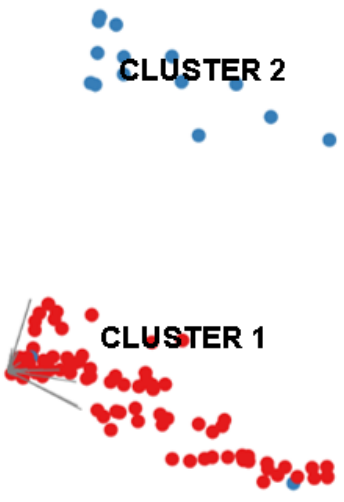


Figura 5.2: Resultado gráfico de clústeres con el data set Euronews.

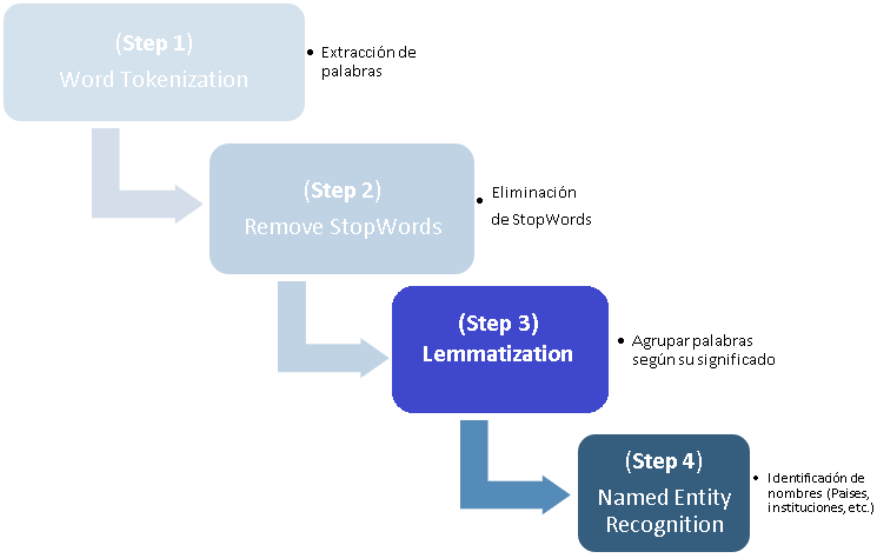


Figura 5.3: Pasos del análisis PNL.

2. Se aplican técnicas de procesamiento del lenguaje natural para realizar la interpretación semántica del conjunto de datos, ver figura 5.3.

Drinking, very, beverages, , probably, , causes, cancer, , scientists, have, claimed, ., , ?, world, news, newsamericabrazil, Brazil, Australia, s, Tyler, won, the, women, s, season, -, long, this, week, following, a, second, placed, finish, at, Roxy, Pro, France, ., newsasiasyria, Syria, insiders, world, news, UN, Security, Council, members, discuss, Iran, Behind, closed, doors, , Iran, s, nuclear, programme, is, being, discussed, today, by, the, five, permanent, of, Council, ., They, aim, to, work, out, , ?, World, newseuroperussia, Russia, Mexico, s, Caravan, For, has, reached, the, country, s, most, dangerous, city, , Ciudad, Juarez, ., World, newsamericamexico, Mexico, newsasiaphilippines, A, German, court, has, rejected, an, attempt, by, to, stop, a, controversial, biography, of, the, former, Chancellor, from, being, published, ., Germany, newsasiasyria, Syria, insiders, The, party, the, PVV, (, The, People, s, Freedom, Party,), , led, by, Geert, Wilders, , has, polledM, -, BM-, 17 %, coming, second, European, , ?, ?, Europe, brussels, bureau, Jerusalem, :, a, major, obstacle, to, peace, , During, its, long, and, often, brutal, history, , control, of, the, city, Jerusalem, has, changed, many, times, ., Its, religious, significance, remains, key, , ?, World, newsamericaguatemala, Guatemala, David, Cameron, the, Conservatives, Blair, He, may, be, just, 39, years, old, , but, his, supporters, say, he, is, man, to, take, driving, seat, of, a, Conservative, party, looking, for, new, way, forward, , ?, ?, World, newseuroperussia, The, end, of, the, summer, tourist, season, saw, number, people, registering, as, unemployed, shoot, up, by, a

Tabla 5.1: Paso 1 Tokenización.

En la Tabla 5.1 se presenta una muestra del proceso de tokenización realizado, donde se separan 2197 palabras.

Drinking, beverages, probably, causes, cancer, scientists, claimed, world, news, newsamericabrazil, Brazil, Australia, s, Tyler, won, women, s, season, long, week, following, second, placed, finish, Roxy, Pro, France, ., newsasiasyria, Syria, insiders, world, news, UN, Security, Council, members, discuss, Iran, closed, doors, Iran, s, nuclear, programme, discussed, today, permanent, Council, ., aim, work, World, newseuroperussia, Russia, Mexico, s, Caravan, reached, country, s, dangerous, city, Ciudad, Juarez, World, newsamericamexico, Mexico, newsasiaphilippines, German, court, rejected, attempt, stop, controversial, biography, Chancellor, published, ., Germany, newsasiasyria, Syria, insiders, party, PVV, (People, s, Freedom, Party,),led, Geert, Wilders, polledM, BM, 17 %, coming, second, European, Europe, brussels, bureau, Jerusalem, major, obstacle, peace, long, brutal, history, control, city, Jerusalem, changed, times, religious, significance, remains, key, , World, newsamericaguatemala, Guatemala, David, Cameron, Conservatives, Blair, 39, years, old, , supporters, man, driving, seat, Conservative, party, looking, new, way, forward, World, newseuroperussia, end, summer, tourist, season, saw, number, people, registering, unemployed, shoot, expected, 96, Business, economy, newseuropespain, bomb, blast, Madrid, Saturday, claimed, casualties, dealt, bitter, blow, city, s, hopes, hosting, 2012, Olympics, World, newseuroperussia, Russia, Niemeyer, icon, architecture, dies, legendary, Oscar, Niemeyer, died, , shy, 105th, birthday, days, life, Brazilian, architect, designed, committee, drafting, Iraq, s, new, decided, time, ruled, delay, afer, President, Jalal, Talabani, World, newseuroperussia, Russia, newsasiasyria, France, s, proved, strong, opponents, Wujiang, Athlete, secure

Tabla 5.2: Paso 2 Remoción de stop words.

Con la aplicación del paso 2, ver Tabla 5.2 han sido removidas un 33 % de las palabras obtenidas del paso anterior, por no aportar significado al conjunto de datos.

(Drinking, 'drink', 'VERB'), (beverages, 'beverage', 'NOUN'), (, ' ', 'SPACE'), (probably, 'probably', 'ADV'), (, ' ', 'SPACE'), (causes, 'cause', 'NOUN'), (cancer, 'cancer', 'NOUN'), (scientists, 'scientist', 'NOUN'), (claimed, 'claim', 'VERB'), (., '.', 'PUNCT'), (world, 'world', 'NOUN'), (news, 'news', 'NOUN'), (newsamericabrazil, 'newsamericabrazil', 'PROPN'), (Brazil, 'Brazil', 'PROPN'), (Australia, 'Australia', 'PROPN'), (s, 's', 'PART'), (Tyler, 'Tyler', 'PROPN'), (won, 'win', 'VERB'), (women, 'woman', 'NOUN'), (s, 's', 'PART'), (season, 'season', 'NOUN'), (long, 'long', 'ADV'), (week, 'week', 'NOUN'), (following, 'follow', 'VERB'), (second, 'second', 'ADJ'), (placed, 'placed', 'ADJ'), (finish, 'finish', 'NOUN'), (Roxy, 'Roxy', 'PROPN'), (Pro, 'Pro', 'PROPN'), (France, 'France', 'PROPN'), (., '.', 'PUNCT'), (newsasiasyria, 'newsasiasyria', 'PROPN'), (Syria, 'Syria', 'PROPN'), (insiders, 'insider', 'VERB'), (world, 'world', 'NOUN'), (news, 'news', 'NOUN'), (UN, 'UN', 'PROPN'), (Security, 'Security', 'PROPN'), (Council, 'Council', 'PROPN'), (members, 'member', 'NOUN'), (discuss, 'discuss', 'VERB'), (Iran, 'Iran', 'PROPN'), (closed, 'closed', 'ADJ'), (doors, 'door', 'NOUN'), (Iran, 'Iran', 'PROPN'), (s, 's', 'PART'), (nuclear, 'nuclear', 'ADJ'), (programme, 'programme', 'NOUN'), (discussed, 'discuss', 'VERB'), (today, 'today', 'NOUN'), (permanent, 'permanent', 'ADJ'), (Council, 'Council', 'PROPN'), (., '.', 'PUNCT'), (aim, 'aim', 'VERB'), (work, 'work', 'VERB'), (, ' ', 'SPACE'), (World, 'World', 'PROPN'), (newseuroperussia, 'newseuroperussia', 'PROPN'), (Russia, 'Russia', 'PROPN'), (Mexico, 'Mexico', 'PROPN'), (s, 's', 'PART'), (Caravan, 'Caravan', 'PROPN'), (reached, 'reach', 'VERB'), (country, 'country', 'NOUN'), (dangerous, 'dangerous', 'ADJ'), (city, 'city', 'NOUN'), (Ciudad, 'Ciudad', 'PROPN'), (Juarez, 'Juarez', 'PROPN'),

Tabla 5.3: Paso 3 Lematización.

Lorrys ORG registration cancelled a week DATE before fatal bus crash in Russi GPE At least 18 CARDINAL people have been killed near Moscow GPE . European NORP finished up for the third ORDINAL straight session hitting a two-week DATE high.Big screen premieres: The Host and Trance In HostFinally they have something smile about: the children among 500 migrants QUANTITY who landed at Sicilian NORP port of Empedocle ORG last Sunday DATE . British NORP supermarket chain has posted quarterly DATE that was towards the bottom end of forecasts due to recent fall DATE in food price. German NORP photographer Helmut Newton PERSON was known for his risque and provocative work.A huge shook parts of on Saturday DATE May 24.At CebitIsabelle Huppert ORG is in Cannes to promote her new film In Country which was directed and written by Hong Sangsoo GPE . She plays a trio of.The secular party of President Moncef Marzouki PERSON has withdrawn its three CARDINAL ministers from the Islamist NORP led government.The President of Nicos Anastasiades ORG is today DATE seek an 11-th hour prevent the country s financial collapse. Gibraltar ORG has criticised Spanish NORP police who sent divers to inspect an artificial reef in waters claimed by the British NORP territory. Divers measured.Robots of all shapes and uses were on show at Japan s International Robot Exhibition FAC . The focus was industrial robots but electronic pets.A Ukrainian NORP checkpoint has been set up near in the Kherson GPE region south of country. Troops arrived after unconfirmed reports an.Industrial tension in against a background of chronically high unemployment came to head the Basque NORP country with forced eviction 70.With a on refugee crisis.

Figura 5.4: Paso 4 Reconocimiento de entidades.

A continuación se presenta el diccionario de conceptos generados a partir de la interpretación de las relaciones entre las instancias analizadas. Ver Tabla 5.4.

CONCEPTO	ATRIBUTOS	RANGO
GEOPOLITICAL CONGLOMERATE	GEOPOLITICAL AREA	{EU, G20, G77, MEC, REST OF THE WORLD}
UE-G20-G77-MEC	GEOPOLITICAL AREA	{EU, G20, G77, MEC}
REST OF THE WORLD	GEOPOLITICAL AREA	{REST OF THE WORLD}
NEWS	GEOPOLITICAL AREA	{EU, G20, G77, MEC, REST OF THE WORLD}
EUROPEAN UNION (EU)	GEOPOLITICAL AREA	{FRANCE, GERMANY, ITALY, SPAIN, UNITED KINGDOM, AUSTRIA, BELGIUM}
G20	GEOPOLITICAL AREA	{EU, NON-EU, G20 COUNTRIES}
NON-EU, G20 COUNTRIES	GEOPOLITICAL AREA	{AUSTRALIA, BRASIL, CANADA, CHINA, JAPAN, SOUTH KOREA, MEXICO, RUSIA, USA}
G77	GEOPOLITICAL AREA	{COLOMBIA, CUBA, HONDURAS, LIBYA, PERU, AFGANISTAN, NORTH KOREA, SINGAPURE, MALI, MYANMAR, THAILAND, BRAZIL, PALESTINE, SYRIA, YEMEN}
MIDDLE EAST COUNTRIES (MEC)	GEOPOLITICAL AREA	{PALESTINA, IRAN, IRAK, ISRAEL, SYRIA, YEMEN}
FRANCE	GEOPOLITICAL AREA	{FRANCE}

CONCEPTO	ATRIBUTOS	RANGO
GERMANY	GEOPOLITICAL AREA	{GERMANY}
ITALY	GEOPOLITICAL AREA	{ITALY}
SPAIN	GEOPOLITICAL AREA	{SPAIN}
UNITED KINGDOM	GEOPOLITICAL AREA	{UNITED KINGDOM}
AUSTRIA	GEOPOLITICAL AREA	{AUSTRIA}
BELGIUM	GEOPOLITICAL AREA	{BELGIUM}
AUSTRIA	GEOPOLITICAL AREA	{AUSTRIA}
BRAZIL	GEOPOLITICAL AREA	{BRAZIL}
CANADA	GEOPOLITICAL AREA	{CANADA}
CHINA	GEOPOLITICAL AREA	{CHINA}
JAPON	GEOPOLITICAL AREA	{JAPON}
SOUTH KOREA	GEOPOLITICAL AREA	{SOUTH KOREA}
RUSIA	GEOPOLITICAL AREA	{RUSIA}
MEXICO	GEOPOLITICAL AREA	{MEXICO}
USA	GEOPOLITICAL AREA	{USA}

CONCEPTO	ATRIBUTOS	RANGO
COLOMBIA	GEOPOLITICAL AREA	{COLOMBIA}
CUBA	GEOPOLITICAL AREA	{CUBA}
HONDURAS	GEOPOLITICAL AREA	{HONDURAS}
LIBYA	GEOPOLITICAL AREA	{LIBYA}
PERU	GEOPOLITICAL AREA	{PERU}
NORTH KOREA	GEOPOLITICAL AREA	{NORTH KOREA}
AFGANISTAN	GEOPOLITICAL AREA	{AFGANISTAN}
SINGAPURE	GEOPOLITICAL AREA	{SINGAPURE}
MALI	GEOPOLITICAL AREA	{MALI}
MYANMAR	GEOPOLITICAL AREA	{MYANMAR}
THAILAND	GEOPOLITICAL AREA	{THAILAND}
PALESTINA	GEOPOLITICAL AREA	{PALESTINA}
IRAN	GEOPOLITICAL AREA	{IRAN}
IRAK	GEOPOLITICAL AREA	{IRAK}
ISRAEL	GEOPOLITICAL AREA	{ISRAEL}
SYRIA	GEOPOLITICAL AREA	{SYRIA}
YEMEN	GEOPOLITICAL AREA	{YEMEN}

Tabla 5.4: Diccionario de conceptos.

5.3. Evaluación como sistema de recuperación

Para realizar la evaluación como sistema de recuperación de información, se utilizó una muestra aleatoria de 100 noticias clasificadas con el modelo propuesto. En este sentido, se trata de saber si los resultados obtenidos por el modelo propuesto en esta investigación se aproxima a lo que debería dar como resultado si la clasificación la realizara un humano. Para ello se conformó una matriz de confusión. Donde TP, TN, FP, y FN representan: true positive, true negative, false positive y false negative, respectivamente. Quedando esctruturada de la forma siguiente:

	UE-G20-G77-MEC	RESTO DEL MUNDO
UE-G20-G77-MEC	**TP**	**FP**
RESTO DEL MUNDO	**FN**	**TN**

Tabla 5.5: Matriz de confusión.

	UE-G20-G77-MEC	RESTO DEL MUNDO
UE-G20-G77-MEC	**TP = 79**	**FP= 2**
RESTO DEL MUNDO	**FN= 1**	**TN=18**

Tabla 5.6: Matriz de confusión resultado.

Mediante los resultados obtenidos en nuestra matriz de confusión, podemos evaluar el modelo propuesto a través de los parámetros estándar (precision, recall, accuracy y F-measure).

$$Precision = \frac{TP}{TP + FP} = \frac{79}{79 + 2} = \frac{79}{81} = 0,9753 \tag{5.1}$$

$$Recall = \frac{TP}{TP + FN} = \frac{79}{79 + 1} = \frac{79}{80} = 0,9875 \tag{5.2}$$

$$Fmeasure = 2x\frac{PR}{P + R} = 2x\frac{0,9631}{1,9628} = 2x0,4906 = 0,9812 \tag{5.3}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} = \frac{79 + 18}{79 + 2 + 18 + 1} = \frac{97}{100} = 0,97 \tag{5.4}$$

Al estar la muestra desbalanceada, la métrica que nos permite tener un valor más objetivo es F-measure. Teniendo en cuenta que un clasificador con un valor de F-measure = 1 es perfecto, nuestro modelo con un valor de F-measure = 0,9812 es de alta precisión destacando la poca ocurrencia tanto de FN como de FP (noticias incorrectamente clasificadas).

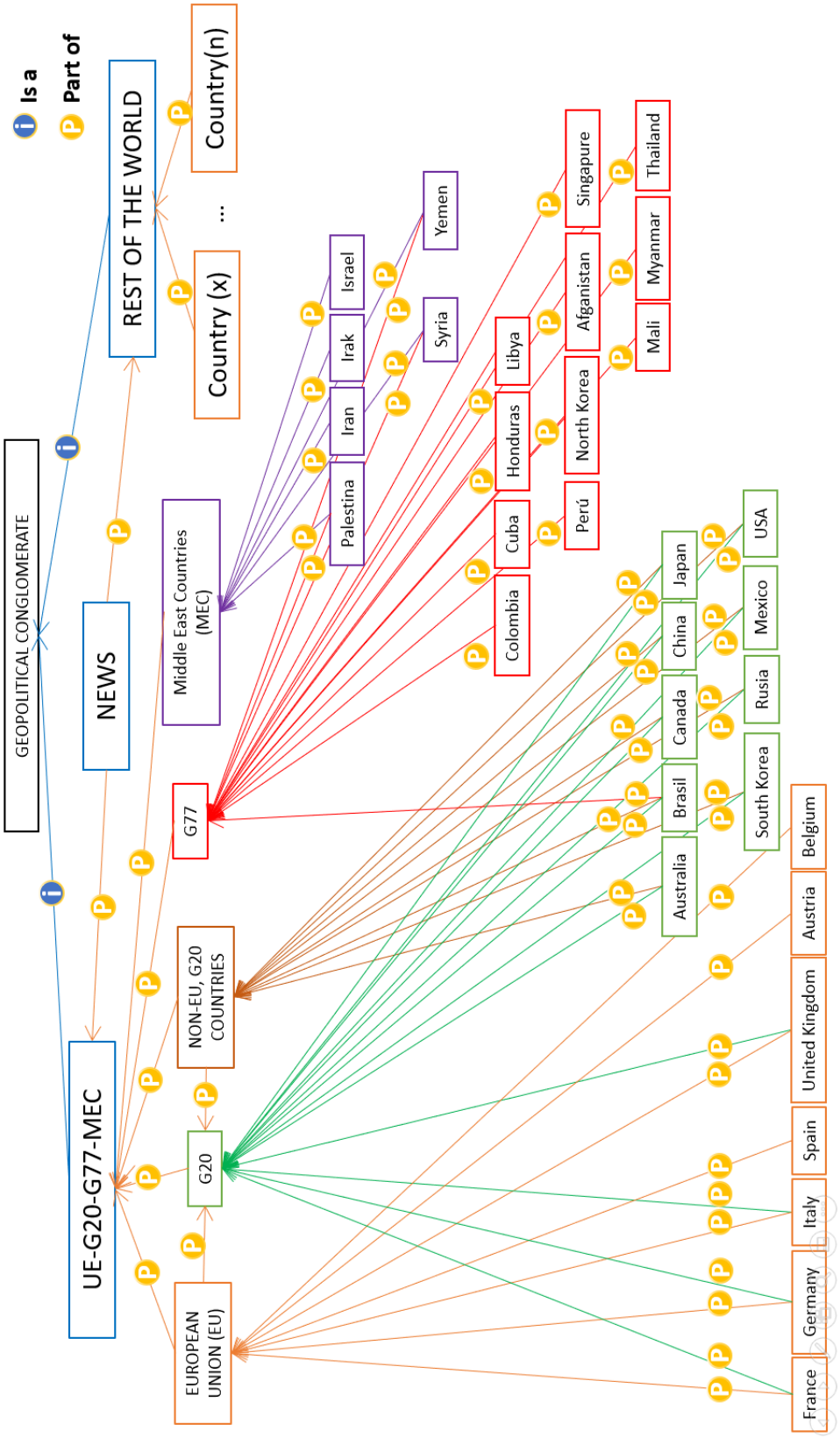


Figura 5.5: Representación de relaciones.

Capítulo 6

Conclusiones y trabajos futuros

El objetivo general de esta tesis fue combinar clustering, selección de atributos y métodos ontológicos para la clasificación semántica de texto. A continuación, se presentan las conclusiones obtenidas después de haber cumplido con cada uno de los objetivos específicos.

OE1. Redactar el estado del arte relacionado con la temática estudiada. La realización del estado del arte de esta tesis doctoral permitió establecer las bases teóricas de la investigación, así como también identificar las últimas técnicas utilizadas en el análisis de datos textuales al estudiar los trabajos relacionados.
OE2. Conformación de un conjunto de datos textuales lo suficientemente extenso para la aplicación de las diferentes técnicas de análisis de datos. La conformación del conjunto de datos propio permitió poner en práctica técnicas de extracción de información y de preprocesamiento de datos, siendo las primeras fases del proceso de minería de texto. Con el propósito de transformar la información no estructurada de los archivos de texto en una forma estructurada y ordenada, que luego pudo ser interpretada por los algoritmos de aprendizaje automático, permitiendo conformar una conjunto de datos textual con 104.434 instancias y 20 atributos.
OE3. Desarrollo de una metodología para la clasificación semántica de datos textuales.
El desarrollo de este objetivo específico permitió desarrollar la metodología propuesta, integrando técnicas de minería de texto e ingeniería ontológica para conformar una metodología a través de la cual se obtuvo un modelo con el cual se puede realizar clasificación semántica de texto de alta precisión.
OE4. Evaluar los resultados obtenidos.
La evaluación de los resultados nos ha permitido seleccionar el modelo más preciso tanto desde el punto de vista de Machine Learning como de Procesamiento del Lenguaje Natural. En ambos casos, se obtuvieron resultados de precisión superiores al 98 %.

En los últimos años, se ha vuelto difícil para los usuarios acceder a información precisa y confiable debido al aumento de la cantidad de información disponible en internet. En este estudio se ha propuesto un modelo para la clasificación automática de texto, utilizando técnicas de Machine Learning y de Web Semántica para dar significado a la clasificación obtenida. Se ha demostrado empíricamente que ha-

ciendo SToWVector junto con selección de atributos mediante el wrapper MOES (estrategia de búsqueda) y NaiveBayesMultinomial (evaluador) con ACC (métrica), se obtienen mejores resultados con el clasificador NaiveBayesMultinomial. Además, cuando se realiza clustering con el algoritmo K-means y la selección de atributos mediante MultiObjectiveEvolutio- narySearch, WrapperSubsetEval y NaiveBayes-Multinomial se obtiene un modelo altamente preciso 4.4. Combinando técnicas de machine learning y de ontologías se pudo dar significado a los dos clústeres obtenidos, encontrando un concepto para cada clúster. Clúster 1: UE-G20-G77-MEC y clúster 2: Resto del mundo. Lo que nos permitió establecer una relación entre los clústeres (el área con el área geopolítica). De manera que se pudo diseñar y evaluar una metodología efectiva para la clasificación e interpretación de datos textuales. En este trabajo, se han integrado dos enfoques paradigmáticos de la Inteligencia Artificial, como es la de aprendizaje de patrones, a través de la aplicación de diversas técnicas de Machine Learning, y la de procesamiento simbólico, mediante la aplicación de técnicas de ingeniería ontológica para representar el conocimiento. Así, se ha expresado, por un lado, conocimiento experto sobre el dominio a través de ontologías y también se ha dotado de significado semántico a los clústeres aprendidos (mediante técnicas de Machine Learning) haciendo uso de ingeniería ontológica.

Trabajos futuros

Como parte del trabajo futuro evidente de esta tesis, se planea implementar la metodología propuesta mediante una aplicación móvil multiplataforma completamente funcional, que permita clasificar si una noticia es de interés o no, clasificándola como relevante o irrelevante según el entrenamiento dado por el usuario, la cual se pondría a disposición de los usuarios a través de las plataformas de descarga comunmente utilizadas.

Bibliografía

[1] Antanas Verikas, Adas Gelzinis, and Marija Bacauskiene. Mining data with random forests: A survey and results of new tests. *Pattern recognition*, 44(2):330–349, 2011.

[2] S Thamarai Selvi, P Karthikeyan, A Vincent, V Abinaya, G Neeraja, and R Deepika. Text categorization using rocchio algorithm and random forest algorithm. In *2016 Eighth International Conference on Advanced Computing (ICoAC)*, pages 7–12. IEEE, 2017.

[3] Ned Horning. Introduction to decision trees and random forests. *Am. Mus. Nat. Hist*, 2:1–27, 2013.

[4] Moawia Elfaki Yahia and Badria Abaker Ibrahim. K-nearest neighbor and c4. 5 algorithms as data mining methods: advantages and difficulties. *Computer Systems and Applications*, page 103, 2003.

[5] Mohamed Osman Bashar et al. *Improving Students Academic Performance Using Hybrid Recommendation Techniques*. PhD thesis, Sudan University of Science & Technology, 2018.

[6] Xujuan Zhou, Raj Gururajan, Yuefeng Li, Revathi Venkataraman, Xiaohui Tao, Ghazal Bargshady, Prabal D Barua, and Srinivas Kondalsamy-Chennakesavan. A survey on text classification and its applications. In *Web Intelligence*, volume 18, pages 205–216. IOS Press, 2020.

[7] Shabnam Kumari, V Vani, Shaveta Malik, Amit Kumar Tyagi, and Sravanti Reddy. Analysis of text mining tools in disease prediction. In *International Conference on Hybrid Intelligent Systems*, pages 546–564. Springer, 2021.

[8] Wenqi Wang, Run Wang, Jianpeng Ke, and Lina Wang. Textfirewall: Omni-defending against adversarial texts in sentiment classification. *IEEE Access*, 9:27467–27475, 2021.

[9] Sunil Kumar, Arpan Kumar Kar, and P Vigneswara Ilavarasan. Applications of text mining in services management: A systematic literature review. *International Journal of Information Management Data Insights*, 1(1):100008, 2021.

[10] Amit Kumar Sharma, Sandeep Chaurasia, and Devesh Kumar Srivastava. Sentimental short sentences classification by using cnn deep learning model with fine tuned word2vec. *Procedia Computer Science*, 167:1139–1147, 2020.

[11] Dandan Tao, Pengkun Yang, and Hao Feng. Utilization of text mining as a big data analysis tool for food science and nutrition. *Comprehensive reviews in food science and food safety*, 19(2):875–894, 2020.

[12] Qin Li, Shaobo Li, Sen Zhang, Jie Hu, and Jianjun Hu. A review of text corpus-based tourism big data mining. *Applied Sciences*, 9(16):3300, 2019.

[13] Vallikannu Ramanathan and T Meyyappan. Twitter text mining for sentiment analysis on people's feedback about oman tourism. In *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)*, pages 1–5. IEEE, 2019.

[14] Yanshan Wang, Sunghwan Sohn, Sijia Liu, Feichen Shen, Liwei Wang, Elizabeth J Atkinson, Shreyasee Amin, and Hongfang Liu. A clinical text classification paradigm using weak supervision and deep representation. *BMC medical informatics and decision making*, 19(1):1–13, 2019.

[15] Bernhard Waltl, Georg Bonczek, Elena Scepankova, and Florian Matthes. Semantic types of legal norms in german laws: classification and analysis using local linear explanations. *Artificial Intelligence and Law*, 27(1):43–71, 2019.

[16] Hossein Hassani, Christina Beneki, Stephan Unger, Maedeh Taj Mazinani, and Mohammad Reza Yeganegi. Text mining in big data analytics. *Big Data and Cognitive Computing*, 4(1):1, 2020.

[17] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[18] Christos Boutsidis, Petros Drineas, and Michael W Mahoney. Unsupervised feature selection for the k-means clustering problem. In *Advances in Neural Information Processing Systems*, pages 153–161, 2009.

[19] Charu C Aggarwal and ChengXiang Zhai. A survey of text clustering algorithms. In *Mining text data*, pages 77–128. Springer, 2012.

[20] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering*, 17(4):491–502, 2005.

[21] Mohammed J Zaki, Wagner Meira Jr, and Wagner Meira. *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[22] Shyr-Shen Yu, Shao-Wei Chu, Chuin-Mu Wang, Yung-Kuan Chan, and Ting-Cheng Chang. Two improved k-means algorithms. *Applied Soft Computing*, 68:747–755, 2018.

[23] Adam Meyerson, Michael Shindler, and Alex Wong. Fast and accurate k-means for large datasets. *Submitted to NIPS*, 2011.

[24] Shiming Xiang, Feiping Nie, and Changshui Zhang. Learning a mahalanobis distance metric for data clustering and classification. *Pattern recognition*, 41(12):3600–3612, 2008.

[25] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[26] Pradeep Rai and Shubha Singh. A survey of clustering techniques. *International Journal of Computer Applications*, 7(12):1–5, 2010.

[27] George Nagy. State of the art in pattern recognition. *Proceedings of the IEEE*, 56(5):836–863, 1968.

[28] José M Pena, Jose Antonio Lozano, and Pedro Larranaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern recognition letters*, 20(10):1027–1040, 1999.

[29] Murat Erisoglu, Nazif Calis, and Sadullah Sakallioglu. A new algorithm for initial cluster centers in k-means algorithm. *Pattern Recognition Letters*, 32(14):1701–1705, 2011.

[30] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[31] Shi Yu, Leon Tranchevent, Xinhai Liu, Wolfgang Glanzel, Johan AK Suykens, Bart De Moor, and Yves Moreau. Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1031–1039, 2011.

[32] Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190, 2008.

[33] Dae-Won Kim, Ki Young Lee, Doheon Lee, and Kwang H Lee. Evaluation of the performance of clustering algorithms in kernel-induced feature space. *Pattern Recognition*, 38(4):607–611, 2005.

[34] Suman Tatiraju and Avi Mehta. Image segmentation using k-means clustering, em and normalized cuts. *Department of EECS*, 1:1–7, 2008.

[35] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[36] CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983.

[37] Jinwen Ma, Lei Xu, and Michael I Jordan. Asymptotic convergence rate of the em algorithm for gaussian mixtures. *Neural Computation*, 12(12):2881–2907, 2000.

[38] John Hartigan. Statistical clustering. *International Encyclopedia of the Social and Behavioral Sciences*, pages 15014–15019, 2001.

[39] Petr Skoda and Fathalrahman Adam. *Knowledge Discovery in Big Data from Astronomy and Earth Observation: Astrogeoinformatics*. Elsevier, 2020.

[40] Athman Bouguettaya, Qi Yu, Xumin Liu, Xiangmin Zhou, and Andy Song. Efficient agglomerative hierarchical clustering. *Expert Systems with Applications*, 42(5):2785–2797, 2015.

[41] Luiza Barbosa da Matta, Lívia Gracielle Oliveira Tomé, Caio Césio Salgado, Cosme Damião Cruz, and Letícia de Faria Silva. Hierarchical genetic clusters for phenotypic analysis. *Acta Scientiarum. Agronomy*, 37(4):447–456, 2015.

[42] John C Gower. A comparison of some methods of cluster analysis. *Biometrics*, pages 623–637, 1967.

[43] Ravinda Khatree and Dayanand N Naik. *Applied multivariate statistics with SAS software*. SAS Publishing, 1997.

[44] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[45] Weipeng Jing, Chuanyu Zhao, and Chao Jiang. An improvement method of dbscan algorithm on cloud computing. *Procedia computer science*, 147:596–604, 2019.

[46] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[47] Shaomin Wang, Shouxiang Wang, and Dan Wang. Combined probability density model for medium term load forecasting based on quantile regression and kernel density estimation. *Energy Procedia*, 158:6446–6451, 2019.

[48] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining: Concepts and techniques,‖ morgan kaufmann publishers inc. *San Francisco, CA, USA*, 2011.

[49] BG Obula Reddy and Dr Maligela Ussenaiah. Literature survey on clustering techniques. *IOSR Journal of Computer Engineering*, 3(1):1–50, 2012.

[50] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156, 1997.

[51] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–45, 2017.

[52] Hudhaifa Mohammed Abdulwahab, S Ajitha, and Mufeed Ahmed Naji Saif. Feature selection techniques in the context of big data: taxonomy and analysis. *Applied Intelligence*, 52(12):13568–13613, 2022.

[53] Urszula Stańczyk and Lakhmi C Jain. *Feature selection for data and pattern recognition*. Springer, 2015.

[54] R Kohavi and GH John. Wrappers for feature subset selection, artificial intelligence 97 (1-2)(1997), 273-324. *Google Scholar Google Scholar Digital Library Digital Library*.

[55] Aiguo Wang, Ning An, Guilin Chen, Lian Li, and Gil Alterovitz. Accelerating wrapper-based feature selection with k-nearest-neighbor. *Knowledge-Based Systems*, 83:81–91, 2015.

[56] Roberto Ruiz, Jose C Riquelme, and Jesus S Aguilar-Ruiz. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition*, 39(12):2383–2392, 2006.

[57] Pablo Bermejo, Luis de la Ossa, José A Gámez, and José M Puerta. Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. *Knowledge-Based Systems*, 25(1):35–44, 2012.

[58] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 856–863, 2003.

[59] J Weston, S Mukherjee, O Chapelle, M Pontil, T Poggio, and V Vapnik. Feature selection for svms: Advances in neural information processing systems. 2001.

[60] Wojciech Siedlecki and Jack Sklansky. A note on genetic algorithms for large-scale feature selection. In *Handbook of pattern recognition and computer vision*, pages 88–107. World Scientific, 1993.

[61] Hisao Ishibuchi and Tomoharu Nakashima. Multi-objective pattern and feature selection by a genetic algorithm. In *Proceedings of the 2nd annual conference on genetic and evolutionary computation*, pages 1069–1076, 2000.

[62] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.

[63] Fernando Jiménez, Gracia Sánchez, José M García, Guido Sciavicco, and Luis Miralles. Multi-objective evolutionary feature selection for online sales forecasting. *Neurocomputing*, 234:75–92, 2017.

[64] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[65] Wa'el Hadi, Qasem A Al-Radaideh, and Samer Alhawari. Integrating associative rule-based classification with naïve bayes for text classification. *Applied Soft Computing*, 69:344–356, 2018.

[66] Mingyang Jiang, Yanchun Liang, Xiaoyue Feng, Xiaojing Fan, Zhili Pei, Yu Xue, and Renchu Guan. Text classification based on deep belief network and softmax regression. *Neural Computing and Applications*, 29(1):61–70, 2018.

[67] Perumal Pitchandi and Mathivanan Balakrishnan. Document clustering analysis with aid of adaptive jaro winkler with jellyfish search clustering algorithm. *Advances in Engineering Software*, 175:103322, 2023.

[68] Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, pages 364–371. IEEE, 2017.

[69] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[70] Qasem A Al-Radaideh and Samya S Al-Khateeb. An associative rule-based classifier for arabic medical text. *International Journal of Knowledge Engineering and Data Mining*, 3(3-4):255–273, 2015.

[71] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

[72] Doris Hoogeveen, Li Wang, Timothy Baldwin, and Karin M Verspoor. Web forum retrieval and text analytics: A survey. *Foundations and Trends in Information Retrieval*, 12(1):1–163, 2018.

[73] Sanjay K Dwivedi and Chandrakala Arya. Automatic text classification in information retrieval: A survey. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, pages 1–6, 2016.

[74] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading, 2010.

[75] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis (foundations and trends (r) in information retrieval), 2008.

[76] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*, 2002.

[77] Colm O'Riordan and Humphrey Sorensen. Information filtering and retrieval: An overview. In *Proceedings of the 16th Annual International Conference of the IEEE, Atlanta, GA, USA*, pages 28–31. Citeseer, 1997.

[78] Chris Buckley. Implementation of the smart information retrieval system. Technical report, Cornell University, 1985.

[79] Charu C Aggarwal. Content-based recommender systems. In *Recommender systems*, pages 139–166. Springer, 2016.

[80] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

[81] Mani Maybury. *Advances in automatic text summarization*. MIT press, 1999.

[82] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Improving multi-document summarization via text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[83] Eitel JM Lauría and Alan D March. Combining bayesian text classification and shrinkage to automate healthcare coding: A data quality analysis. *Journal of Data and Information Quality (JDIQ)*, 2(3):1–22, 2011.

[84] Jinghe Zhang, Kamran Kowsari, James H Harrison, Jennifer M Lobo, and Laura E Barnes. Patient2vec: A personalized interpretable deep representation of the longitudinal electronic health record. *IEEE Access*, 6:65333–65346, 2018.

[85] Dolf Trieschnigg, Piotr Pezik, Vivian Lee, Franciska De Jong, Wessel Kraaij, and Dietrich Rebholz-Schuhmann. Mesh up: effective mesh text classification for improved document retrieval. *Bioinformatics*, 25(11):1412–1418, 2009.

[86] Bahadorreza Ofoghi and Karin Verspoor. Textual emotion classification: An interoperability study on cross-genre data sets. In *Australasian Joint Conference on Artificial Intelligence*, pages 262–273. Springer, 2017.

[87] Alicia L Nobles, Jeffrey J Glenn, Kamran Kowsari, Bethany A Teachman, and Laura E Barnes. Identification of imminent suicide risk among young adults using text messages. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2018.

[88] James W Pennebaker, Martha E Francis, and Roger J Booth. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001, 2001.

[89] Michael J Paul and Mark Dredze. Social monitoring for public health. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 9(5):1–183, 2017.

[90] B Yu and L Kwok. Classifying business marketing messages on facebook (2011).

[91] Mangi Kang, Jaelim Ahn, and Kichun Lee. Opinion mining using ensemble text hidden markov models for text classification. *Expert Systems with Applications*, 94:218–227, 2018.

[92] Howard Turtle. Text retrieval in the legal world. *Artificial Intelligence and Law*, 3(1):5–54, 1995.

[93] Alper Kursat Uysal and Serkan Gunal. The impact of preprocessing on text classification. *Information processing & management*, 50(1):104–112, 2014.

[94] Daniel Jurasky and James H Martin. Speech and language processing: An introduction to natural language processing. *Computational Linguistics and Speech Recognition. Prentice Hall, New Jersey*, 2000.

[95] Majid Hameed Ahmed, Sabrina Tiun, Nazlia Omar, and Nor Samsiah Sani. Short text clustering algorithms, application and challenges: A survey. *Applied Sciences*, 13(1):342, 2022.

[96] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive data sets*. Cambridge university press, 2020.

[97] Gaurav Gupta and Sumit Malhotra. Text document tokenization for word frequency count using rapid miner (taking resume as an example). *Int. J. Comput. Appl*, 975:8887, 2015.

[98] Tanu Verma, Renu Renu, and Deepti Gaur. Tokenization and filtering process in rapidminer. *International Journal of Applied Information Systems*, 7(2):16–18, 2014.

[99] Jonathan J Webster and Chunyu Kit. Tokenization as the initial phase in nlp. In *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*, 1992.

[100] G Grefenstette and P Tapanainen. "what is a word, what is a sentence? problems of tokenization", in proceedings of the 3rd conference on computational lexicography and text research (complex'94). 1994.

[101] Martin F Porter. An algorithm for suffix stripping. program: electronic library & information systems. 1980.

[102] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938, 2011.

[103] Jose Ramon Cano, Francisco Herrera, and Manuel Lozano. Strategies for scaling up evolutionary instance reduction algorithms for data mining. In *Evolutionary Computation in Data Mining*, pages 21–39. Springer, 2005.

[104] Jerome H Friedman. Data mining and statistics: What's the connection? *Computing science and statistics*, 29(1):3–9, 1998.

[105] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[106] Sascha Kaufmann. *CUBA: Artificial conviviality and user-behaviour analysis in web-feeds*. PhD thesis, University of Luxembourg, Luxembourg, Luxembourg, 2010.

[107] Egon S Pearson. Bayes' theorem, examined in the light of experimental sampling. *Biometrika*, 17(3/4):388–442, 1925.

[108] Bruce M Hill. Posterior distribution of percentiles: Bayes' theorem for sampling from a population. *Journal of the American Statistical Association*, 63(322):677–691, 1968.

[109] Liangxiao Jiang, Chaoqun Li, Shasha Wang, and Lungan Zhang. Deep feature weighting for naive bayes and its application to text classification. *Engineering Applications of Artificial Intelligence*, 52:26–39, 2016.

[110] Zhaowei Qu, Xiaomin Song, Shuqiang Zheng, Xiaoru Wang, Xiaohui Song, and Zuquan Li. Improved bayes method based on tf-idf feature and grade factor feature for chinese information classification. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 677–680. IEEE, 2018.

[111] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

[112] Jason D Rennie, Lawrence Shih, Jaime Teevan, and David R Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 616–623, 2003.

[113] Jeremy J Eberhardt. Bayesian spam detection. *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal*, 2(1):2, 2015.

[114] B Santhi and GR Brindha. Multinomial naive bayes using similarity based conditional probability. *Journal of Intelligent & Fuzzy Systems*, 36(2):1431–1441, 2019.

[115] Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. Technical report, Stanford InfoLab, 1997.

[116] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

[117] David D Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer, 1998.

[118] Shengyi Jiang, Guansong Pang, Meiling Wu, and Limin Kuang. An improved k-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, 39(1):1503–1509, 2012.

[119] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Ruili Wang. Efficient knn classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems*, 29(5):1774–1785, 2017.

[120] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[121] T Bailey, JAIN AK, et al. A note on distance-weighted k-nearest neighbor rules. 1978.

[122] K Gowda and G Krishna. The condensed nearest neighbor rule using the concept of mutual nearest neighborhood (corresp.). *IEEE Transactions on Information Theory*, 25(4):488–490, 1979.

[123] Geoffrey Gates. The reduced nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 18(3):431–433, 1972.

[124] Subhash C Bagui, Sikha Bagui, Kuhu Pal, and Nikhil R Pal. Breast cancer detection using rank nearest neighbor classification rules. *Pattern recognition*, 36(1):25–34, 2003.

[125] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *OTM Confederated International Conferences."n the Move to Meaningful Internet Systems*

, pages 986–996. Springer, 2003.

[126] Hamid Parvin, Hosein Alizadeh, and Behrouz Minaei-Bidgoli. Mknn: Modified k-nearest neighbor. In *Proceedings of the world congress on engineering and computer science*, volume 1. Citeseer, 2008.

[127] Yong Zeng, Yupu Yang, and Liang Zhao. Pseudo nearest neighbor rule for pattern classification. *Expert Systems with Applications*, 36(2):3587–3595, 2009.

[128] Zhou Yong, Li Youwen, and Xia Shixiong. An improved knn text classification algorithm based on clustering. *Journal of computers*, 4(3):230–237, 2009.

[129] Ting Liu, Andrew W Moore, Alexander Gray, and Claire Cardie. New algorithms for efficient high-dimensional nonparametric classification. *Journal of Machine Learning Research*, 7(6), 2006.

[130] Robert F Sproull. Refinements to nearest-neighbor searching in k-dimensional trees. *Algorithmica*, 6(1):579–589, 1991.

[131] Stan Z. Li, Kap Luk Chan, and Changliang Wang. Performance evaluation of the nearest feature line method in image classification and retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1335–1339, 2000.

[132] V Vapnik and A Ya Chervonenkis. A class of algorithms for pattern recognition learning. *Avtomat. i Telemekh*, 25(6):937–945, 1964.

[133] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[134] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020.

[135] Gou Bo and Huang Xianwu. Svm multi-class classification. *Journal of Data Acquisition & Processing*, 21(3):334–339, 2006.

[136] Lam Hong Lee, Chin Heng Wan, Rajprasad Rajkumar, and Dino Isa. An enhanced support vector machine classification framework by using euclidean distance function for text document categorization. *Applied Intelligence*, 37(1):80–99, 2012.

[137] Ji He, Ah-Hwee Tan, and Chew-Lim Tan. On machine learning methods for chinese document categorization. *Applied Intelligence*, 18(3):311–322, 2003.

[138] Edda Leopold and Jörg Kindermann. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46(1):423–444, 2002.

[139] Steven CH Hoi, Rong Jin, and Michael R Lyu. Batch mode active learning with applications to text categorization and image retrieval. *IEEE Transactions on knowledge and data engineering*, 21(9):1233–1248, 2009.

[140] Mani Arun Kumar and Madan Gopal. Text categorization using fuzzy proximal svm and distributional clustering of words. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 52–61. Springer, 2009.

[141] Mika Timonen et al. Term weighting in short documents for document categorization, keyword extraction and query expansion. 2013.

[142] Kunlun Li, Jing Xie, Xue Sun, Yinghui Ma, and Hui Bai. Multi-class text categorization based on lda and svm. *Procedia Engineering*, 15:1963–1967, 2011.

[143] Tao Peng, Wanli Zuo, and Fengling He. Svm based adaptive learning method for text classification from positive and unlabeled documents. *Knowledge and Information Systems*, 16(3):281–301, 2008.

[144] Roberto HW Pinheiro, George DC Cavalcanti, and Ren Tsang. Combining binary classifiers in different dichotomy spaces for text categorization. *Applied Soft Computing*, 76:564–574, 2019.

[145] Bassam Al-Salemi, Masri Ayob, Graham Kendall, and Shahrul Azman Mohd Noah. Multi-label arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms. *Information Processing & Management*, 56(1):212–227, 2019.

[146] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[147] Himani Bansal, Gulshan Shrivastava, Gia Nhu Nguyen, and Loredana-Mihaela Stanciu. *Social network analytics for contemporary business organizations*. IGI Global, 2018.

[148] Yanjun Qi. Random forest for bioinformatics. In *Ensemble machine learning*, pages 307–323. Springer, 2012.

[149] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[150] Ilya Oparin, Ondrej Glembek, Lukas Burget, and Jan Cernocky. Morphological random forests for language modeling of inflectional languages. In *2008 IEEE Spoken Language Technology Workshop*, pages 189–192. IEEE, 2008.

[151] Sameen Maruf, Kashif Javed, and Haroon A Babri. Improving text classification performance with random forests-based feature selection. *Arabian Journal for Science and Engineering*, 41(3):951–964, 2016.

[152] Thiago Salles, Marcos Gonçalves, Victor Rodrigues, and Leonardo Rocha. Improving random forests by neighborhood projection for effective text classification. *Information Systems*, 77:1–21, 2018.

[153] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[154] Han Liu and Alexander Gegov. Induction of modular classification rules by information entropy based rule generation. In *Innovative Issues in Intelligent Systems*, pages 217–230. Springer, 2016.

[155] Wei Dai and Wei Ji. A mapreduce implementation of c4. 5 decision tree algorithm. *International journal of database theory and application*, 7(1):49–60, 2014.

[156] Yashuang Mu, Xiaodong Liu, Zhihao Yang, and Xiaolin Liu. A parallel c4. 5 decision tree algorithm based on mapreduce. *Concurrency and Computation: Practice and Experience*, 29(8):e4015, 2017.

[157] Rutvija Pandya and Jayati Pandya. C5. 0 algorithm to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications*, 117(16):18–21, 2015.

[158] Zhenyu Lu, Xindong Wu, and Josh C Bongard. Active learning through adaptive heterogeneous ensembling. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):368–381, 2014.

[159] Pedro J García-Laencina, Pedro Henriques Abreu, Miguel Henriques Abreu, and Noémia Afonoso. Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete values. *Computers in biology and medicine*, 59:125–133, 2015.

[160] X Wu, V Kumar, J Ross Quinlan, J Ghosh, Q Yang, H Motoda, GJ McLachlan, A Ng, B Liu, PS Yu, et al. Top 10 algorithms in data mining knowledge and information systems, vol. 14, no. 1, 2007.

[161] Phu Vo Ngoc, Chau Vo Thi Ngoc, Tran Vo THi Ngoc, and Dat Nguyen Duy. A c4. 5 algorithm for english emotional classification. *Evolving Systems*, 10(3):425–451, 2019.

[162] Asmaa M Aubaid and Alok Mishra. A rule-based approach to embedding techniques for text document classification. *Applied Sciences*, 10(11):4009, 2020.

[163] Yanshu Sun. Deep data mining of student scores. 2022.

[164] Feyza Altunbey Ozbay and Bilal Alatas. Fake news detection within online social media using supervised artificial intelligence algorithms. *Physica A: Statistical Mechanics and its Applications*, 540:123174, 2020.

[165] Hossein Hematialam and Wlodek Zadrozny. Identifying condition-action statements in medical guidelines using domain-independent features. *arXiv preprint arXiv:1706.04206*, 2017.

[166] Sridevi UK. An ontology-based sentiment analysis model towards classification of drug reviews. 2021.

[167] M Gayathri and R Jagadeesh Kannan. Ontology based concept extraction and classification of ayurvedic documents. *Procedia Computer Science*, 172:511–516, 2020.

[168] Muhammad Ali Ibrahim, Muhammad Usman Ghani Khan, Faiza Mehmood, Muhammad Nabeel Asim, and Waqar Mahmood. Ghs-net a generic hybridized shallow neural network for multi-label biomedical text classification. *Journal of biomedical informatics*, 116:103699, 2021.

[169] Immanuel Kant. *Lectures on metaphysics*. Cambridge University Press, 2001.

[170] Usha Yadav, Gagandeep Singh Narula, Neelam Duhan, Vishal Jain, and BK Murthy. Development and visualization of domain specific ontology using protege. *Indian Journal of Science and Technology*, 9(16):1–7, 2016.

[171] MPS Bhatia, Akshi Kumar, and Rohit Beniwal. Ontologies for software engineering: Past, present and future. *Indian Journal of Science and Technology*, 9(9):1–16, 2016.

[172] Rafael Pedraza-Jimenez, Lluis Codina, and Cristofol Rovira. Semantic web and ontologies in document information processing. 2007.

[173] Oscar Corcho, Mariano Fernández-López, Asunción Gómez-Pérez, and Angel López-Cima. Construcción de ontologías legales con la metodología methontology y la herramienta webode. *Law and the Semantic Web. Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications*, pages 142–157, 2005.

[174] Yuefeng Liu, Minyong Shi, and Chunfang Li. Domain ontology concept extraction method based on text. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pages 1–5. IEEE, 2016.

[175] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[176] Gertjan Van Heijst, A Th Schreiber, and Bob J Wielinga. Using explicit ontologies in kbs development. *International journal of human-computer studies*, 46(2-3):183–292, 1997.

[177] Donald AB Lindberg, Betsy L Humphreys, and Alexa T McCray. The unified medical language system. *Yearbook of Medical Informatics*, 2(01):41–51, 1993.

[178] Ga Miller. Wordnet: A lexical database for english communications of the acm vol. 38. 1995.

[179] Christiane Fellbaum et al. Wordnet: An electronic lexical database mit press. *Cambridge, Massachusetts*, 1998.

[180] Riichiro Mizoguchi, Johan Vanwelkenhuysen, and Mitsuru Ikeda. Task ontology for reuse of problem solving knowledge. towards very large knowledge bases. *KnowledgeBuilding and Knowledge Sharing*, pages 46–59, 1995.

[181] Georg Klinker, Carlos Bhola, Geoffroy Dallemagne, David Marques, and John McDermott. Usable and reusable programming constructs. *Knowledge Acquisition*, 3(2):117–135, 1991.

[182] Nicola Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

[183] Thabet Slimani. Ontology development: A comparing study on tools, languages and formalisms. *Indian Journal of Science and Technology*, 8(24):1–12, 2015.

[184] Yingzhong Zhang, Xiaofang Luo, Jian Li, and Jennifer J Buis. A semantic representation model for design rationale of products. *Advanced Engineering Informatics*, 27(1):13–26, 2013.

[185] Brian McBride. The resource description framework (rdf) and its vocabulary description language rdfs. In *Handbook on ontologies*, pages 51–65. Springer, 2004.

[186] Benjamin N Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker. Description logic programs: Combining logic programs with description logic. In *Proceedings of the 12th international conference on World Wide Web*, pages 48–57, 2003.

[187] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[188] Kaarthik Sivashanmugam, John A Miller, Amit P Sheth, and Kunal Verma. Framework for semantic web process composition. *International Journal of Electronic Commerce*, 9(2):71–106, 2005.

[189] Jos de Bruijn, Holger Lausen, Axel Polleres, and Dieter Fensel. The web service modeling language WSML: an overview. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings*, volume 4011 of *Lecture Notes in Computer Science*, pages 590–604. Springer, 2006.

[190] Winston W Royce. Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering*, pages 328–338, 1987.

[191] Barry W. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, 1988.

[192] Jaime Alberto Guzmán Luna, Mauricio López Bonilla, and Ingrid Durley Torres. Metodologías y métodos para la construcción de ontologías. *Scientia et technica*, 2(50):133–140, 2012.

[193] Benjamin Peraketh, Christopher P Menzel, Richard J Mayer, Florence Fillion, and Michael T Futrell. Ontology capture method (idef5). Technical report, KNOWLEDGE BASED SYSTEMS INC COLLEGE STATION TX, 1994.

[194] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.

[195] York Sure, Steffen Staab, and Rudi Studer. Methodology for development and employment of ontology based knowledge management applications. *ACM Sigmod Record*, 31(4):18–23, 2002.

[196] Konstantinos Kotis and George A Vouros. Human-centered ontology engineering: The hcome methodology. *Knowledge and Information Systems*, 10(1):109–131, 2006.

[197] Werner Kunz and Horst WJ Rittel. *Issues as elements of information systems*, volume 131. Citeseer, 1970.

[198] Christoph Tempich. *Ontology engineering and routing in distributed knowledge management applications.* PhD thesis, Karlsruhe Institute of Technology, Germany, 2006.

[199] Katharina Siorpaes and Martin Hepp. Games with a purpose for the semantic web. *IEEE intelligent systems*, 23(3):50–60, 2008.

[200] Martin Hepp, Joerg Leukel, and Volker Schmitz. A quantitative analysis of product categorization standards: content, coverage, and maintenance of ecl@ ss, unspsc, eotd, and the rosettanet technical dictionary. *Knowledge and Information Systems*, 13(1):77–114, 2007.

[201] Simone Braun, Andreas P Schmidt, Andreas Walter, Gabor Nagypal, and Valentin Zacharias. Ontology maturing: a collaborative web 2.0 approach to ontology engineering. In *Ckc*. Citeseer, 2007.

[202] Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl_1):D258–D261, 2004.

[203] Rashmie Abeysinghe, Michael A Brooks, and Licong Cui. Leveraging non-lattice subgraphs to audit hierarchical relations in nci thesaurus. In *AMIA annual symposium proceedings*, volume 2019, page 982. American Medical Informatics Association, 2019.

[204] Cornelius Rosse and José LV Mejino Jr. A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of biomedical informatics*, 36(6):478–500, 2003.

[205] Pablo López-García, Martin Boeker, Arantza Illarramendi, and Stefan Schulz. Usability-driven pruning of large ontologies: the case of snomed ct. *Journal of the American Medical Informatics Association*, 19(e1):e102–e109, 2012.

[206] Luan Fonseca Garcia, Mara Abel, Michel Perrin, and Renata dos Santos Alvarenga. The geocore ontology: A core ontology for general use in geology. *Computers & Geosciences*, 135:104387, 2020.

[207] Simon Jupp, Tony Burdett, Danielle Welter, Sirarat Sarntivijai, Helen Parkinson, and James Malone. Webulous and the webulous google add-on-a web service and application for ontology building from templates. *Journal of Biomedical semantics*, 7(1):1–8, 2016.

[208] Andreas Hotho, Alexander Maedche, and Steffen Staab. Ontology-based text document clustering. *KI*, 16(4):48–54, 2002.

[209] Travis D Breaux and Joel W Reed. Using ontology in hierarchical information clustering. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 111b–111b. IEEE, 2005.

[210] Mor Peleg, Nuaman Asbeh, Tsvi Kuflik, and Mitchell Schertz. Onto-clust—a methodology for combining clustering analysis and ontological methods for identifying groups of comorbidities for developmental disorders. *Journal of biomedical informatics*, 42(1):165–175, 2009.

[211] Jingnian Chen, Houkuan Huang, Shengfeng Tian, and Youli Qu. Feature selection for text classification with naïve bayes. *Expert Systems with Applications*, 36(3):5432–5435, 2009.

[212] Fengxi Song, Shuhai Liu, and Jingyu Yang. A comparative study on text representation schemes in text categorization. *Pattern analysis and applications*, 8(1):199–209, 2005.

[213] Ya Xiong Li and Deng Pan. Text clustering based on domain ontology and latent semantic analysis. In *Applied Mechanics and Materials*, volume 556, pages 3536–3540. Trans Tech Publ, 2014.

[214] Qing Ju Guo, Wen Tian Ji, and Sheng Zhong. Ontology-based k-means clustering algorithm analysis. In *Applied Mechanics and Materials*, volume 380, pages 1290–1293. Trans Tech Publ, 2013.

[215] Alexandre Ribeiro Afonso and Cláudio Gottschalg Duque. Automated text clustering of newspaper and scientific texts in brazilian portuguese: analysis and comparison of methods. *JISTEM-Journal of Information Systems and Technology Management*, 11:415–436, 2014.

[216] Rupasingha AHM Rupasingha, Incheon Paik, and Banage TGS Kumara. Improving web service clustering through a novel ontology generation method by domain specificity. In *2017 IEEE international conference on web services (ICWS)*, pages 744–751. IEEE, 2017.

[217] Fuchao Liu and Guanyu Li. The extension of domain ontology based on text clustering. In *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 1, pages 301–304. IEEE, 2018.

[218] Mahdi Abdollahi, Xiaoying Gao, Yi Mei, Shameek Ghosh, and Jinyan Li. An ontology-based two-stage approach to medical text classification with feature selection by particle swarm optimisation. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 119–126. IEEE, 2019.

[219] B Selvalakshmi and M Subramaniam. Intelligent ontology based semantic information retrieval using feature selection and classification. *Cluster Computing*, 22(5):12871–12881, 2019.

[220] Giridhar Urkude and Manju Pandey. Design and development of density-based effective document clustering method using ontology. *Multimedia Tools and Applications*, 81(23):32995–33015, 2022.

[221] Michal Toman, Roman Tesar, and Karel Jezek. Influence of word normalization on text classification. *Proceedings of InSciT*, 4:354–358, 2006.

[222] José Ramon Méndez, Eva Lorenzo Iglesias, Florentino Fdez-Riverola, Fernando Díaz, and Juan M Corchado. Tokenising, stemming and stopword removal on anti-spam filtering domain. In *Conference of the Spanish Association for Artificial Intelligence*, pages 449–458. Springer, 2005.

[223] Abdullah Ayedh, Guanzheng Tan, Khaled Alwesabi, and Hamdi Rajeh. The effect of preprocessing on arabic document categorization. *Algorithms*, 9(2):27, 2016.

[224] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[225] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[226] Magnus Sahlgren. The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33–53, 2008.

[227] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[228] Anil Sharma and Suresh Kumar. Ontology-based semantic retrieval of documents using word2vec model. *Data & Knowledge Engineering*, 144:102110, 2023.

[229] Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. Randomized algorithms and nlp: Using locality sensitive hash functions for high speed noun clustering. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 622–629, 2005.

[230] Chenn-Jung Huang, Dian-Xiu Yang, and Yi-Ta Chuang. Application of wrapper approach and composite classifier to the stock trend prediction. *Expert Systems with Applications*, 34(4):2870–2878, 2008.

[231] Feng Tan, Xuezheng Fu, Yanqing Zhang, and Anu G Bourgeois. A genetic algorithm-based method for feature subset selection. *Soft Computing*, 12(2):111–120, 2008.

[232] César Guerra-Salcedo, Stephen Chen, Darrell Whitley, and Stephen Smith. Fast and accurate feature selection using hybrid genetic strategies. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 1, pages 177–184. IEEE, 1999.

[233] Daniel Peralta, Sara Del Río, Sergio Ramírez-Gallego, Isaac Triguero, Jose M Benitez, and Francisco Herrera. Evolutionary feature selection for big data classification: A mapreduce approach. *Mathematical Problems in Engineering*, 2015, 2015.

[234] Amira Sayed A Aziz, Ahmad Taher Azar, Mostafa A Salama, Aboul Ella Hassanien, and Sanaa El-Ola Hanafy. Genetic algorithm with different feature selection techniques for anomaly detectors generation. In *2013 Federated Conference on Computer Science and Information Systems*, pages 769–774. IEEE, 2013.

[235] Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pages 3–34. Springer, 2011.

[236] Nidamarthi Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[237] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, 18(4):577–601, 2013.

[238] Fernando Jimenez, Antonio F Gómez-Skarmeta, Gracia Sánchez, and Kalyanmoy Deb. An evolutionary algorithm for constrained multi-objective optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 2, pages 1133–1138. IEEE, 2002.

[239] Fernando Jiménez, Gracia Sánchez, and Pandian Vasant. A multi-objective evolutionary approach for fuzzy optimization in production planning. *Journal of Intelligent & Fuzzy Systems*, 25(2):441–455, 2013.

[240] Fernando Jiménez, Gracia Sánchez, and José M Juárez. Multi-objective evolutionary algorithms for fuzzy classification in survival prediction. *Artificial intelligence in medicine*, 60(3):197–219, 2014.

[241] Fernando Jiménez, Enrico Marzano, Gracia Sánchez, Guido Sciavicco, and Nicola Vitacolonna. Attribute selection via multi-objective evolutionary computation applied to multi-skill contact center data classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 488–495. IEEE, 2015.

[242] Fernando Jiménez, Horacio Pérez-Sánchez, José Palma, Gracia Sánchez, and Carlos Martínez. A methodology for evaluating multi-objective evolutionary feature selection for classification in the context of virtual screening. *Soft Computing*, 23(18):8775–8800, 2019.

[243] Fernando Jiménez, Carlos Martínez, Enrico Marzano, Jose Tomas Palma, Gracia Sánchez, and Guido Sciavicco. Multiobjective evolutionary feature selection for fuzzy classification. *IEEE Transactions on Fuzzy Systems*, 27(5):1085–1099, 2019.

[244] Fernando Jiménez, Carlos Martínez, Luis Miralles-Pechuán, Gracia Sánchez, and Guido Sciavicco. Multi-objective evolutionary rule-based classification with categorical data. *Entropy*, 20(9):684, 2018.

[245] Lipo Wang and Xiuju Fu. *Data mining with computational intelligence*. Springer Science & Business Media, 2006.

[246] UP Cambridge. Online edition (c) 2009 cambridge up an introduction to information retrieval christopher d, 2009.

[247] Steven L Salzberg. C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993, 1994.

[248] Nur Fadzilah Othman and WISW Din. Youtube spam detection framework using naïve bayes and logistic regression. *Indonesian Journal of Electrical Engineering and Computer Science*, 14(3):1508–1517, 2019.

[249] David Aha. Uci machine learning repository: Center for machinelearning intelligent systems. *ed*, 2017.

[250] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[251] Lei Xu and Michael I Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

[252] Vishal Gupta, Gurpreet S Lehal, et al. A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, 1(1):60–76, 2009.

[253] S Wold, K Esbensen, and P Geladi. Principal component analysis. chimometrics and intelligent laboratory systems. In *IEEE Conference on Emerging Technologies & Factory Automation Efta Volume*, pages 704–706, 1987.

[254] Niklas Kühl, Marius Mühlthaler, and Marc Goutier. Supporting customer-oriented marketing with artificial intelligence: automatically quantifying customer needs from social media. *Electronic Markets*, 30(2):351–367, 2020.

[255] Haneet Kour, Jatinder Manhas, and Vinod Sharma. Usage and implementation of neuro-fuzzy systems for classification and prediction in the diagnosis of different types of medical disorders: a decade review. *Artificial Intelligence Review*, 53(7):4651–4706, 2020.

[256] Roberta A Sinoara, Jose Camacho-Collados, Rafael G Rossi, Roberto Navigli, and Solange O Rezende. Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, 163:955–971, 2019.

[257] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: a comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40, 2021.

[258] Tianshi Wang, Li Liu, Naiwen Liu, Huaxiang Zhang, Long Zhang, and Shanshan Feng. A multi-label text classification method via dynamic semantic representation model and deep neural network. *Applied Intelligence*, 50(8):2339–2351, 2020.

[259] Artem Revenko, Victor Mireles, Anna Breit, Peter Bourgonje, Julian Moreno-Schneider, Maria Khvalchik, and Georg Rehm. Learning ontology classes from text by clustering lexical substitutes derived from language models 1. In *Towards a Knowledge-Aware AI*, pages 155–169. IOS Press, 2022.

[260] Mukesh Kumar, Bhisham Sharma, and Disha Handa. Building predictive model by using data mining and feature selection techniques on academic dataset. *IJMECS*, 14:16–29, 2022.