SEP **SES** **TNM**

INSTITUTO TECNOLÓGICO DE CHIHUAHUA II

“BÚSQUEDA DE INFORMACIÓN PÚBLICA UTILIZANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL”

TESIS
PARA OBTENER EL GRADO DE

MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:

JULIÁN GARCÍA ARZATE

DIRECTOR DE TESIS
M.I.S.C. ARTURO ALVARADO GRANADINO

CO-DIRECTOR DE TESIS
DR. HERNÁN DE LA GARZA GUTIÉRREZ

CHIHUAHUA, CHIH., MÉXICO; A JULIO DE 2019

Resumen

La presente tesis describe una metodología donde se utilizan herramientas de Inteligencia Artificial para realizar búsquedas y clasificación de información pública en idioma español, con el fin de obtener textos previamente almacenados en una Base de Datos y cuyo resultado contenga información similar a la petición inicialmente dada, todo esto a partir de métodos de Recuperación de Información (RI), Procesamiento de Lenguaje Natural (PLN) y algoritmos de Aprendizaje Automático.

En este contexto para alcanzar el fin que se persigue se consideran dos preceptos primordiales de la Inteligencia Artificial que son sus *Fundamentos* y sus *Ramas*; dentro de los fundamentos se encuentran la Lingüística Computacional y las bases Matemáticas; desde de la Lingüística Computacional se contempla al Procesamiento de Lenguaje Natural (PLN), este último contiene varios métodos de Recuperación y Extracción de Información en el que se destaca el TF-IDF.

Por otro lado en lo referente a sus *Ramas*, estas, contienen algoritmos de *Aprendizaje Automático* del tipo *supervisados* y *no supervisados*, dentro de los algoritmos no supervisados, están los Valores Singulares de Descomposición, mientras que del lado de los algoritmos supervisados se encuentra la Máquina de Soporte a Vectores, todo esto con el apoyo de bases matemáticas para llegar al objetivo que se persigue en este proyecto.

Abstract

This thesis describes a methodology where Artificial Intelligence tools are used to perform searches and classification of public information in Spanish language, in order to obtain texts previously stored in a Database and whose result contain information similar to the initially given request, all this based on methods of Information Recovery (IR), Natural Language Processing (NLP) and algorithms of Machine Learning.

In this context, two primordial precepts are considered to reach the end sought of the Artificial Intelligence that are it's *Foundations* and it's *Branches*; are found Computational Linguistics and Mathematical bases; from the Computational Linguistics is contemplated to Natural Language Processing (NLP), the latter contains several methods of Information Recovery and Extraction in which it stands out the TF-IDF.

On the other hand, in relation to their *Branches*, these contain *Machine Learning* algorithms of the *supervised* and *unsupervised* type, within the unsupervised algorithms, there are the Singular Value Decomposition, while on the side of the supervised algorithms is the Support Vector Machine, all this with the support of mathematical bases to reach the objective pursued in this project.

Dedicación

A mis padres y a mi familia.

Agradecimientos

A mi familia por su apoyo incondicional.

A todos los maestros que he conocido en mi vida, gracias por su aportación académica y personal, porque me ayudaron a percibir mis fortalezas, debilidades, aciertos y errores, eso me ha ayudado para seguir tratando de ser un mejor ser humano.

Contenido

Lista de Figuras

Lista de Tablas

Capítulo 1

Introducción

En el mundo actual, la información y el conocimiento es adquirido en mayor proporción debido a que contamos con mecanismos tecnológicos que brindan una manera más cómoda de almacenar, procesar y difundir grandes cantidades de datos. En el pasado no muy lejano, la información se registraba y se almacenaba ocupando gran volumen y además quien tenía acceso a la información, se enfrentaba con un reto muy grande, que era obtenerla de manera rápida y concisa. Hoy en día a pesar de que la humanidad almacena información de manera masiva, y de que tenemos herramientas avanzadas para su localización, no siempre obtenemos lo que realmente necesitamos, ya sea porque no sabemos plantear de manera adecuada la petición o por que nos proporcionan más información de la que verdaderamente necesitamos, y en lugar de satisfacer nuestras necesidades de conocimiento se amplían más las dudas y la incertidumbre.

Con base en lo anterior se plantea desarrollar un sistema computacional que sirva para agilizar las búsquedas basadas en una petición realizada en Español, y proporcionar textos almacenados en una Base de Datos, cuyo resultado contenga información que se aproxime a la petición dada, para esto, se utilizará Inteligencia Artificial con métodos de Procesamiento del Lenguaje Natural (PLN), básicamente un sistema de recuperación de información o también conocidos como "Búsqueda de Respuestas" (sistemas BR) o "Question Answering System" (QA system) por sus siglas en ingles.

Como intención primordial desde el sistema se podrá realizar una consulta expresada en Lenguaje Natural y devolver los registros con mayor similitud a la pregunta planteada, junto a su respuesta asociada, cuyo Corpus[1] ha sido almacenado en un sistema actualmente en operación

[1]Corpus Lingüístico o Corpus, es un conjunto de textos relativamente grande, creado independientemente de sus posibles formas o usos.

que funciona como un sistema FAQ (Frequently Asked Question), denominado Infomex, el cual tiene como función primordial el de registrar preguntas que realizan ciudadanos a dependencias gubernamentales y cuyas respuestas son devueltas al solicitante, quedando almacenadas en una Base de Datos resguardada por el Instituto de Transparencia del Estado de Chihuahua.

1.1 Planteamiento del problema

Existe una aplicación web llamada Plataforma Nacional de Transparencia *PNT* (a nivel nacional) e *Infomex* (a nivel estatal), la cual tiene como objetivo servir de instrumento de enlace y comunicación entre la ciudadanía y el aparato gubernamental. Con este mecanismo el ciudadano tiene la posibilidad de formular preguntas en español a cualquier dependencia gubernamental, las cuales deberán ser respondidas por estas, en tiempo y forma, en un periodo no mayor a diez días con la posibilidad de realizar una prorroga de tiempo hasta por quince días.

En la Base de Datos del sistema Infomex están almacenadas actualmente un aproximado de $68,000$ preguntas que los ciudadanos han realizado a las dependencias gubernamentales del Estado de Chihuahua a lo largo de diez años de entre 2007 a 2017. Este sistema es proveído por el INAI (Instituto Nacional de Acceso a la Información y Protección de Datos Personales) a todas las entidades federativas, y en cada estado del país existe un organismo similar que se encarga de garantizar que las dependencias gubernamentales denominados Sujetos Obligados[2] cumplan con la Ley de Transparencia y asegurar el acceso a la información de todas las personas en igualdad de condiciones, que en el caso del estado Chihuahua es denominado Instituto Chihuahuense de Transparencia y Acceso a la Información Pública (Ichitaip).

El Ichitaip es el responsable de resguardar la información de las solicitudes, las respuestas proporcionadas al solicitante, además de otros datos como son: el folio, el periodo de tiempo, el tipo y la clasificación de las solicitudes.

Por medio de una opción del sistema Infomex, cualquier persona puede realizar consultas a todas las solicitudes que se han hecho a las dependencias gubernamentales, con el fin de ver qué fue lo que se contestó, sin embargo, solo se despliegan registros de una dependencia gubernamental en particular, en un rango de fechas y de una sola clasificación de la pregunta. Los registros finales se muestran por folio y para consultar la respuesta hay que dar clic a

[2]Los entes públicos, los partidos políticos, las agrupaciones políticas, así como los entes privados que reciban recursos públicos y los demás que disponga la Ley.

cada elemento y leer el texto para identificar lo que se busca, lo que conlleva en adicionar un problema a la búsqueda en cuestión que es el de leer todos los registros para localizar algún tema que le interese al usuario, a pesar de que se obtiene un resultado final, el usuario prefiere realizar la pregunta directamente al sistema Infomex y esperar de diez a quince días para obtener la información.

En la figura 1.1 se muestra la secuencia para realizar consultas.

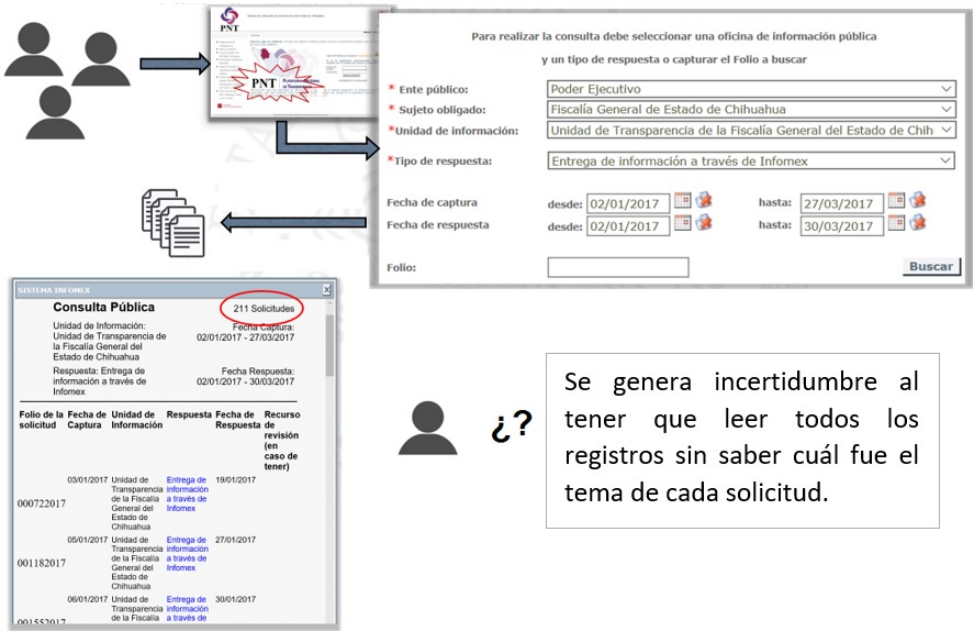


Figura 1.1. Consulta de solicitudes de cualquier dependencia gubernamental.

En algunos casos, al realizar una consulta se obtienen pocos registros y pueden ser consultados de manera rápida, pero, ¿qué sucedería si al realizar la búsqueda el sistema despliega cientos de registros?, aumentaría la incertidumbre de la necesidad inicial que es la de obtener información de forma rápida y satisfactoria. Tomando en cuenta que existen dependencias gubernamentales que reciben miles de solicitudes en un mes, como es el caso de los ayuntamientos más grandes como Chihuahua y Juárez, o como algunos de los Organismos Descentralizados Estatales que son las Universidades. En la figura 1.2 se puede mostrar la magnitud del problema, en cuyo caso se muestran dependencias gubernamentales del Poder Ejecutivo, propiamente el de la Fiscalía General del Estado.

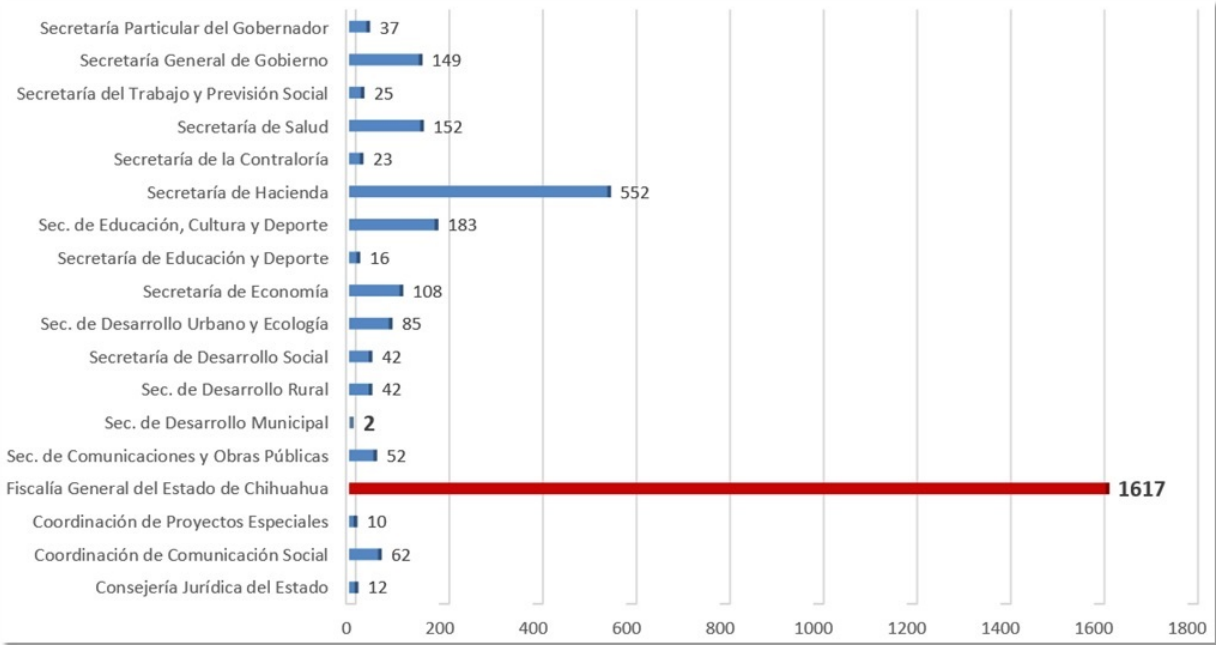


Figura 1.2. Solicitudes de información realizadas a las dependencias gubernamentales del Poder Ejecutivo en el 2016. (Énfasis: Fiscalía General del Estado de Chihuahua).

Como ejemplo se visualiza un tabulador de todas las solicitudes que se han realizado a las dependencias gubernamentales a lo largo de diez años. Ver Tabla 1.1.

Tabla 1.1. Total de solicitudes de información realizadas a lo largo de diez años.

Agrupador	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Ayuntamientos	965	1600	2701	1464	2099	1880	1571	2408	1683	2456	1930
Fideicomisos Mpales.	6	4	6	5	4	14	8	19	21	6	4
Fideicomisos Públicos	24	56	102	25	43	34	123	719	35	140	44
Org. Autónomos	305	213	552	467	374	210	315	428	213	633	175
Org. Desc. Estatales	733	884	1591	648	694	5122	921	2778	676	1151	494
Org. Desc. Municipales	82	103	162	37	175	267	155	316	131	97	68
Org. Desconcentrados	38	18	53	19	32	25	48	155	23	37	9
Org. no Gubernamentales	32	37	58	17	22	22	24	33	22	194	20
Partidos Políticos	159	83	149	151	60	109	49	43	81	154	49
Poder Ejecutivo	1106	1165	2389	1129	1264	1209	1518	2096	1904	3169	1430
Poder Judicial	120	174	376	120	179	383	220	426	265	325	242
Poder Legislativo	226	158	307	153	162	185	180	177	135	230	104
	3796	4495	8446	4235	5108	9460	5132	9598	5189	8592	4569

1.2 Alcances y limitaciones

La razón primordial del sistema informático a desarrollar es la de proporcionar de manera rápida y eficiente los resultados de la búsqueda de información solicitada y tratar de superar las expectativas deseadas, siempre y cuando se trate de aquella que sirva en tiempo y forma al solicitante.

a. Alcances

- Contar con una interfaz de usuario fácil de utilizar acorde a la tecnología actual para realizar las búsquedas necesarias.
- Solo se considerará la búsqueda de información de los años 2016 y 2017 de todas las dependencias gubernamentales, debido que se requiere de más tiempo para realizar el trabajo de normalizar el contenido de los textos de las solicitudes de información de los 9 años faltantes.
- La búsqueda se hará solo sobre el texto de la pregunta almacenada en la Base de Datos, si la pregunta hace referencia a un archivo adjunto,[3] no se considerará para el algoritmo de Valores Singulares de Descomposición.

b. Limitaciones

- La búsqueda de información se debe plantear en idioma español y sólo para cuestiones de índole gubernamental del estado de Chihuahua; además la intención inicial es aprovechar la información pública previamente procesada por la dependencia gubernamental y si tal información no satisface los requerimientos del usuario, tendría la posibilidad de realizar solicitudes de información por el mecanismo ya existente (sistema Infomex).
- Se deberá contar con el recurso humano que dará seguimiento a la tarea de preparación de nuevos documentos, con el objetivo de aplicar métodos de normalización de textos y preparación de estos para nuevas consultas.
- Se recomienda realizar la normalización de textos de las solicitudes de información de los años posteriores al 2017, ya que el interés primordial de los solicitantes es el de conocer sobre información más reciente.
- Muchas de las solicitudes de información están repetidas, ya que los solicitantes le dan un uso indebido al sistema Infomex realizando la misma petición varias veces al mismo Sujeto Obligado, en consecuencia se obtendría la misma respuesta para todos las solicitudes de información repetidas.

[3] El solicitante puede adjuntar un archivo con el fin de ampliar la pregunta solicitada, debido a que el espacio máximo para una pregunta es de 4000 carácteres.

1.3 Justificación

Actualmente por medio del sistema Infomex se pueden realizar consultas de las solicitudes de información que se han realizado a las distintas dependencias gubernamentales. Tales consultas, se hacen proporcionando un rango de fechas y seleccionando a un Sujeto Obligado en particular, y por medio de un listado de registros en pantalla, se muestran los folios con la pregunta y respuesta de dicha solicitud, sin embargo, el usuario tendría que leer el contenido de cada registro para saber si es lo que realmente necesita, ya que la consulta se realiza con parámetros estructurados y a pesar de que el sistema le ofrece un listado de registros, el usuario se tiene que dar a la tarea de clasificar la información, haciendo tediosa y tardada la consulta de la información buscada. A continuación se muestran las características que satisfacen la demanda requerida:

- Acceder a la información de forma ágil.
- Explotación de la información para el uso de la sociedad en general.
- Consultar información procesada para un fin específico.
- Obtener respuestas a nuestras preguntas con una fácil localización.
- Reducir tiempo de espera.

1.4 Objetivos

Desarrollar una aplicación adicional al sistema Infomex que permita optimizar las búsquedas de información pública ya existente en la Base de Datos del mismo sistema.

Esta aplicación informática deberá contribuir en el aprovechamiento y explotación más apropiada de la información ya existente, por medio de técnicas de Procesamiento de Lenguaje Natural y tratamiento de Minería de Textos. Se pretende acotar la distancia en el tiempo de espera al realizar la solicitud por medio del sistema Infomex con la ventaja que toda esa información es pública y se encuentra almacenada para su uso. A continuación se muestran los elementos que tendrá la aplicación informática antes mencionada:

a. Las búsqueda de información será con:

   - Datos estructurados.
   - Datos no estructurados (texto libre).

b. El tipo de colección:

- Documentos.
- Texto simple.

c. Tipo de dominio:

- Libre.

d. Modo de presentar la información:

- Tabuladores.
- Texto.

e. Tipo de usuario:

- Casuales.
- Expertos.

Capítulo 2

Estado del Arte

2.1 Derecho de acceso a la información pública

"La información pública, es un bien del dominio público en poder del Estado, cuya titularidad reside en la sociedad, misma que tendrá en todo momento la facultad de disponer de ella para los fines que considere" (Artículo 2 de la Ley de transparencia y acceso a la información pública del estado de Chihuahua). [1]

Se argumenta que las leyes de acceso a la información contribuyen al buen gobierno y al manejo de documentos administrativos. Lo anterior es porque las leyes fuerzan al gobierno a elaborar originalmente de mejor manera los documentos y archivos, así como a sistematizar su almacenamiento (Chapman y Hunt, 1987).

Bajo esta premisa, la sociedad tiene el derecho al acceso a la información pública y lo puede hacer de forma escrita, a través de un medio o sistema electrónico.

2.2 Ley de transparencia y acceso a la información pública del estado de Chihuahua

Es reglamentaria del artículo 4°, fracción II, de la Constitución Política del Estado de Chihuahua, y tiene por objeto garantizar el derecho de acceso a la información pública y establecer los principios, bases generales y procedimientos para ello.

2.3 Organismo que vigila y regula el acceso a la información pública

El Ichitaip, es un organismo público autónomo, creado por disposición expresa de la Constitución Política del Estado de Chihuahua [2], depositario de la autoridad en la materia, con personalidad jurídica, patrimonio y competencia propios (Art. 43 de la Ley de Transparencia).

a. Facultades:

- El ámbito de competencia del Instituto es en el estado de Chihuahua.
- Debe garantizar que los Sujetos Obligados cumplan la Ley de Transparencia.
- Debe garantizar el derecho de cualquier persona, no necesita ser Chihuahuense, ni mexicano(a), ni ser ciudadano(a), a solicitar y recibir información pública de Chihuahua, así como la protección de sus datos personales.
- Sancionará a los servidores públicos que incumplan la Ley (Art. 4° Constitucional y Art. 57 de la Ley de Transparencia).
- Solucionará los conflictos que se puedan dar entre las personas y los Sujetos Obligados en materia de información Pública y de protección de datos personales (Art. 50, fracción I, inciso f).
- Lograr la máxima apertura y transparencia en la administración pública y en el uso de recursos públicos en el Estado de Chihuahua, mediante una cultura de información clara, suficiente, oportuna, veraz, con perspectiva de género y que protege la información clasificada y los datos personales, con un sistema de información rápido, pertinente y gratuito, a fin de que la sociedad conozca y evalúe mejor el quehacer público, estimulando su participación democrática para mejorar su calidad de vida.

2.4 Solicitar información

Mediante el sistema web Infomex, cuyo fin es el de realizar la recepción, registro y seguimiento de las solicitudes de información, protección de datos personales, así como la recepción y registro de los recursos de revisión. Mediante este sistema, la sociedad en general realiza preguntas a los Sujetos Obligados concernientes a datos relevantes que deben quedar a disposición de quien los solicite; así como la posibilidad de preguntar en concreto lo que les interesaría saber respecto del quehacer de los Sujetos Obligados y de contar con la información de estos, de cómo gasta el recurso y qué tan efectivo fue el ejercicio de ese gasto, toda vez que los funcionarios públicos tienen la obligación de informar.

Una vez que las personas se registren con su usuario y contraseña al sistema Infomex, podrán ejercer el derecho a la información pública por este medio.

Al generar una solicitud de información se asignará un número de folio a cada solicitud que se presente. Este número de folio será único y con él el solicitante podrá dar seguimiento a sus solicitudes. En todos los casos, se hará entrega a la persona solicitante un acuse de recibo correspondiente en los términos que prevé la Ley y su Reglamento.

Realizada la solicitud el Servidor Público[1], que representa al Sujeto Obligado y está a cargo de la Unidad de Transparencia[2], reunirá la información solicitada y dará respuesta en tiempo y forma en un plazo no mayor a diez días hábiles contados a partir del día siguiente hábil de la creación de la solicitud, a su vez si el Servidor Público considera insuficiente el plazo para la entrega de información podrá ampliar el plazo de respuesta por cinco días hábiles más, extendiendo por única ocasión el plazo de entrega de la información hasta por quince días hábiles. La información entregada por titular de la Unidad de Transparencia deberá ser clara, veraz, oportuna, pertinente, verificable, completa, desagregada por género, en la forma y términos previstos en la Ley, lo que significa que las respuestas otorgadas en gran medida se realizan por especialistas en la materia. En el **Anexo A** se muestra el diagrama de flujo del proceso de una solicitud de información en el sistema Infomex.

[1]Toda persona física que desempeñe en un ente público, algún empleo, cargo o comisión de cualquier naturaleza, por elección, nombramiento o contrato.

[2]Órgano encargado de operar el sistema de información, cuyas funciones son las de registrar y procesar la información pública.

Capítulo 3

Marco Teórico

3.1 Inteligencia Artificial

Con los años, ha habido numerosos intentos de cómo definir con precisión el término Inteligencia Artificial. Sin embargo, nunca ha habido realmente una definición oficial aceptada, aun así, hay algunos conceptos más sofisticados que dan forma y definen su significado, las descripciones van desde el objetivo simple de que la IA pueda dotar de una verdadera inteligencia humana a las propias máquinas y computadoras. Desde esta óptica se puede decir que la IA es cualquier técnica que permita que las computadoras aporten significado a los datos de manera similar a un humano.

Mientras que la mayoría de las técnicas de la IA se centran en dominios de problemas específicos como en el caso del Procesamiento del Lenguaje Natural o la Visión por Computadora por citar unos ejemplos, la idea general de la IA es contar con una máquina que pueda realizar cualquier tarea tal como lo haríamos los humanos. [3]

a. Los objetivos principales de la IA incluyen:

- La deducción y el razonamiento.
- La representación del conocimiento.
- La planificación.
- El procesamiento del lenguaje natural.
- El aprendizaje.
- La percepción y la capacidad de manipular y mover objetos.

b. Los objetivos a largo plazo incluyen:

- El logro de la creatividad.
- La inteligencia social.
- La inteligencia general (a nivel humano).

3.2 Fundamentos de la Inteligencia Artificial

Existen varias disciplinas que han contribuido con ideas, puntos de vista y técnicas al desarrollo del campo de la Inteligencia Artificial, entre las que se destacan Lingüística, Matemáticas, Filosofía, Psicología, Neurociencia etc. [4]

3.2.1 Lingüística

La lingüística es la ciencia que estudia el lenguaje humano y estudio científico tanto de la estructura de las lenguas naturales y de aspectos relacionados con ellas, como de su evolución histórica, de su estructura interna y del conocimiento que los hablantes poseen de su propia lengua.

El conocimiento lingüístico es la base teórica para el desarrollo de una amplia gama de aplicaciones tecnológicas, por ejemplo, la búsqueda y el manejo de conocimiento, las interfaces en lenguaje natural entre el humano y las computadoras o los robots, y la traducción automática, entre otras.

La lingüística, como cualquier ciencia, construye los modelos y las descripciones de su objeto de estudio -el lenguaje natural-. La computación convierte a la lingüística en una ciencia exacta, además de presentarle nuevos retos y darle nueva motivación y nuevas direcciones de investigación. Esta transformación se puede comparar con la que en su época propiciaron las matemáticas en la física.

El amplio campo de intersección e interacción entre la lingüística y la computación se estructura a su vez en varias ciencias más específicas. Una de ellas es la lingüística computacional. Esta ciencia trata de la construcción de modelos de lenguaje –entendibles- para las computadoras, es decir, más formales que los modelos tradicionales orientados a los lectores humanos.

Procesamiento de Lenguaje Natural (PLN)

Pertenece al campo de las ciencias de la computación, IA y lingüística ya que se ocupa más de los aspectos técnicos, algorítmicos y matemáticos de la aplicación de dichos modelos a los grandes volúmenes de texto, con el fin de estructurarlos según la información contenida en ellos, de extraerles la información útil, de transformar esta información es decir, de traducirla a otro lenguaje.

El PLN es el campo mediante el cual se desarrollan sistemas informáticos que hacen posible la comprensión y el procesamiento de información expresado en el lenguaje humano, todo esto asistido por una computadora. No se trata de la comunicación por medio de lenguajes naturales de una forma abstracta, sino de diseñar mecanismos para comunicarse y que estos sean eficaces computacionalmente hablando, es decir que se puedan realizar por medio de programas que ejecuten o simulen la comunicación.

a. Conceptos

- **Recuperación de información (RI).** Es el proceso de encontrar en un repositorio grande de datos, material (generalmente documentos) de naturaleza no estructurada es decir texto o semiestructurada como páginas Web, que satisfaga una necesidad de información. [5]

  Las estrategias de recuperación de información involucran la transformación de texto en representaciones adecuadas de acuerdo a modelos específicos que cumplan con los propósitos de las búsquedas.

  Los modelos pueden ubicarse en categorías de acuerdo a dos posibles dimensiones como son con bases matemáticas y con sus propiedades.

  En la dimensión de bases matemáticas, el texto puede ser representado como: conjuntos de palabras o frases en donde las coincidencias se logran realizando operaciones de algebra booleana; modelos algebraicos que introducen parámetros e índices para recuperar información con metadatos, calificar y clasificar documentos en respuesta a una consulta, lo que lleva a modelos en espacios vectoriales, matriciales o agrupamientos irregulares; modelos probabilísticos que enfocan la solución de los problemas de búsqueda desde el punto de vista probabilístico, uno de los más utilizados es el teorema de Bayes.

- **Datos semiestructurados.** Están en documentos con marcas explícitas como el código HTML, XML o JSON. La información encontrada debe ser pertinente y relevante.[1]
- **Datos estructurados.** Son aquellos que se encuentran bien definidos como por ejemplo en una Base de Datos y pueden ser fechas, cantidades, catálogos etc.
- **Datos no estructurados.** No tienen un esquema claro, no están listos para procesar y son lo opuesto a los datos con un esquema estructurados, como el texto de un libro. [6]

b. Terminología

Las terminologías del PLN contienen elementos útiles que apoyan en el tratamiento inicial al que el texto se deberá someter; con esta transformación se hace más simple el traslado de los términos a un espacio vectorial, para así aplicar técnicas de Aprendizaje Automático propiamente a una Máquina de Soporte a Vectores. Ver figura 3.1.

Figura 3.1. Terminologías del Procesamiento del Lenguaje Natural.

[1] Pertinencia es la medida de cómo un documento se ajusta a una necesidad informativa. www.safaribooksonline.com

Descripción de terminologías del PLN:

- **Tokenización.** Es el proceso de extraer las palabras de un texto o frase y estas palabras se denominan tokens, es decir unidades indivisibles. Un tokenizador o segmentador es el primero de los componentes que se utiliza en el procesamiento de texto.
- **Stemming.** Es la raíz valida de una palabra. Utilizando algoritmos de radicación o stemming se puede apoyar en llevar a cabo el proceso de lematización cuyo objetivo es reducir una palabra a su raíz, de modo que las palabras clave de una consulta o documento se representen por sus raíces en lugar de por las palabras originales. El lema de una palabra comprende su forma básica, además de sus formas declinadas, esto permite representar de un mismo modo las distintas variantes de un término, a la vez que reducen el tamaño del vocabulario y mejoran, en consecuencia, la capacidad de almacenamiento de los sistemas y el tiempo de procesamiento de los documentos.
- **TF-IDF.** Técnicas de radicación (o stemming): la mayoría de las técnicas de recuperación de información utilizan el recuento de las frecuencias de los términos que aparecen en los documentos y las consultas. Esto implica la necesidad de normalizar dichos términos para que los recuentos puedan efectuarse de manera adecuada, tomando en consideración aquellos términos que derivan de un mismo lema o raíz.
- **Análisis semántico.** Compara palabras y frases de un conjunto de documentos para extraer su significado. Los primeros pasos para incorporar la semántica en el procesamiento del lenguaje natural son los que están muy relacionados al estudio del espacio vectorial y al álgebra lineal. Varios aspectos fundamentalmente semánticos desde el punto de vista léxico, están relacionados geométricamente por una noción de distancia.
- **Desambiguación.** Es el sentido de la palabra, es decir, consiste en descubrir con qué sentido se está usando una palabra en un contexto dado.[2] [7]
  La desambiguación es considerada como una "tarea intermedia" en algunas actividades de PLN, pero indispensable para lograr la mayoría de ellas.[3] La resolución de la ambigüedad de las palabras es una de las tareas más difíciles en el PLN. Esta resolución es necesaria en la medida en que una palabra pueda ser interpretada de diferentes formas, es decir, que posea más de un significado o sentido[4]. Lo que persigue la desambiguación del sentido de la palabra es la asignación automática de sentidos a las palabras de un texto.

[2]El contexto de la palabra es considerado como un conjunto de palabras que la acompañan, junto con las relaciones sintácticas y categorías semánticas (Vázquez et al., 2003).

[3](Wilks and Stevenson, 1996) Wilks, Y. and M. Stevenson. (1996). "The grammar of sense: Is word-sense tagging much more than part-of-speech tagging". CoRR, cmp-lg/9607028, 1996.

[4]Fenómeno lingüístico conocido como polisemia.

- **Modelos de tema.** Descubrir los temas que se encuentran en una colección de documentos.
- **Límites de la palabra.** Determina dónde empieza y termina una palabra.

c. Principales aplicaciones

- **Sistemas de búsqueda de respuesta.** Tienen como objetivo responder de manera automática a las preguntas de un usuario expresadas en lenguaje natural.
- **Corrección de ortografía.** Corregir ortografía de textos en cualquier idioma o en una disciplina en específico, ya sea área médica, ingeniería, química, humanidades etc.
- **Reconocimiento y síntesis de voz.** Consiste en que la computadora captura la señal de voz que emite una persona a través de un micrófono, convirtiéndola en información digital. El motor de voz debe ser capaz de reconocer las silabas de entre un conjunto de fonemas que ha recibido, y combinarlas para formar las palabras que se habían dicho anteriormente por el usuario.
- **Análisis de sentimientos.** Es la identificación y extracción de opiniones emitidas en textos con el objetivo de clasificarlos, mediante procesamiento computacional, según la polaridad de las emociones que expresan sobre determinados objetos, situaciones o personas.
- **Búsqueda de respuestas.** El proceso automático que realizan las computadoras para encontrar respuestas concretas a preguntas precisas formuladas por los usuarios. Los sistemas de búsqueda de respuestas no sólo localizan los documentos o pasajes relevantes, sino que también encuentran, extraen y muestran la respuesta al usuario final, evitándole la búsqueda o la lectura de la información relevante para encontrar de forma manual la respuesta final.
- **Generación de resúmenes automáticos.** Se define como el proceso de destilar la información más importante de una o varias fuentes para producir una versión abreviada destinada a un usuario determinado y para una o varias tareas.
- **Traducción automática.** Diseñados para traducir textos de una lengua a otra.
- **Recuperación de información.** Es el conjunto de tareas mediante las cuales el usuario localiza y recupera información que es pertinente para satisfacer su necesidad de información o la resolución de un problema; en este sentido, recuperar información significa obtener una información que alguna vez ha sido producida por alguien.
- **Sistemas de diálogos.** Son programas informáticos cuya finalidad es interactuar con los usuarios oralmente o de forma multimodal para proporcionarles determinados servicios, con el fin de aumentar la rapidez, efectividad y facilidad a la hora de realizar estas tareas de forma automática.

- **Clasificación automática de textos.** La clasificación de documentos de texto es una aplicación de la minería de textos que pretende extraer información de texto no estructurado. Por otro lado, la búsqueda semántica permite al usuario especificar en una consulta no solamente términos que deben aparecer en el documento, sino conceptos y relaciones, que pueden detectarse mediante el análisis de texto.

d. Componentes

El modelo del PLN comprende varios componentes que permiten analizar y extraer metadatos clave del texto, incluidas entidades, relaciones, conceptos, sentimientos y emociones. A continuación se describen cada uno de estos aspectos que se pueden extraer del Corpus el texto.

- **Entidades.** Es el caso de uso más común del PLN, se trata de personas, lugares, organizaciones y cosas en un texto.
- **Relaciones.** El PLN identifica si hay una relación entre múltiples entidades y determinar el tipo de relación que existe entre ellas.
- **Conceptos.** Es la extracción de conceptos generales del cuerpo del texto que puede no aparecer explícitamente en el Corpus.
- **Palabras clave.** Identifica las palabras clave importantes y relevantes en su contenido. Esto le permite crear una base de palabras del Corpus que son importantes para el valor comercial que intenta manejar.
- **Roles Semánticos.** Son los sujetos, las acciones y los objetos que actúan sobre en el texto. Ejemplo: “El equipo de futbol compró un estadio”. En esta sentencia el sujeto es “equipo”, la acción es “comprar” y el objeto es “estadio”, con esto el PLN puede analizar oraciones en estos roles semánticos para una variedad de usos comerciales.
- **Categorías.** Describe de qué trata una pieza de contenido en un alto nivel. El PLN puede analizar el texto y luego hacer la taxonomía[5], proporcionando categorías para usar en aplicaciones. Algunos ejemplos de categorías son deportes, finanzas, viajes, computación entre otros.
- **Emoción.** Detección de emociones en PLN ayuda a resolver problemas específicos, por ejemplo: el tratar de comprender la emoción transmitida por una publicación en las redes sociales y precisar si el contenido transmite ira, disgusto, miedo, alegría o tristeza. Identificar las emociones en un texto es extremadamente valioso en los negocios.
- **Sentimiento.** Proporcionar una puntuación en cuanto al nivel de positivo o sentimiento negativo del texto, muy útil en aplicaciones de atención a clientes ya que permite

[5]Ciencia que trata de los principios, métodos y fines de la clasificación

comprensión automática del sentimiento relacionado con su producto en una base continua.

3.2.2 Matemáticas

El término más utilizado define a las matemáticas como una ciencia formal que, partiendo de axiomas y siguiendo el razonamiento lógico, estudia las propiedades y relaciones entre entidades abstractas como números, figuras geométricas o símbolos.

Para entender y desarrollar los principales algoritmos que se utilizan en el campo de la Inteligencia Artificial, utilizando las matemáticas se deben tener nociones de:

- **Álgebra Lineal:** Estudia conceptos tales como vectores, matrices, tensores, sistemas de ecuaciones lineales y en su enfoque de manera más formal, espacios vectoriales y sus transformaciones lineales. Es esencial para entender y trabajar con muchos algoritmos de Machine Learning, y especialmente para los algoritmos de Deep Learning.
- **Cálculo:** Incluye el estudio de los límites, derivadas, integrales y series infinitas, y más concretamente se puede decir que es el estudio del cambio. Particularmente para el campo de la Inteligencia Artificial algunos conceptos que se deberían conocer incluyen: Cálculo Diferencial e Integral, Derivadas Parciales, Funciones de Valores Vectoriales, y Gradientes.
- **Optimización matemática:** Herramienta matemática que permite optimizar decisiones, es decir, seleccionar la mejor alternativa de un conjunto de criterios disponibles. Su comprensión es fundamental para poder entender la eficiencia computacional y la escalabilidad de los principales algoritmos de Machine Learning y Deep Learning, los cuales suelen trabajar con matrices dispersas de gran tamaño.
- **Probabilidad y estadística:** Rama de las matemáticas que trata con la incertidumbre, la aleatoriedad y la inferencia. Sus conceptos son fundamentales para cualquier algoritmo de Machine Learning o Deep Learning.

3.3 Ramas de la Inteligencia Artificial

Contiene varios elementos como son Deep Learning, Algoritmos Genéticos, Razonamiento Probabilístico, Machine Learning entre otros, este último, agrupa al Aprendizaje Supervisado y no Supervisado y al Aprendizaje por Refuerzo.

Debido a que el enfoque de esta tesis es trabajar con Aprendizaje Supervisado y no Supervisado, se mencionarán sus características muy particulares.

3.3.1 Machine Learning (ML)

ML por sus siglas en inglés o "Aprendizaje Automático". Busca usar datos (entradas) para descubrir relaciones y hacer predicciones (salidas). La computadora (máquina), aprende las relaciones en los datos, incluido el aprendizaje para hacer predicciones. Replicar la cognición humana no es un objetivo explícito de ML (como suele ser en la IA, al menos históricamente), y ML puede hacer predicciones que serían difíciles para cualquier persona. Como ejemplo, un algoritmo ML puede aprender a predecir salud o enfermedad mediante el análisis de todos los datos generados por especialistas médicos, cada uno de los cuales puede ver al paciente desde una sola perspectiva. La "máquina" puede ver a los pacientes desde una perspectiva mucho más completa mediante el análisis de todos los datos disponibles.
Los algoritmos de ML se dividen en tres categorías que se decriben a continuación:

a. Aprendizaje supervisado.
   También conocido como "clasificación", cuando se realiza el aprendizaje supervisado, tanto las entradas como las salidas son conocidas y etiquetadas, sin embargo, el aprendizaje supervisado puede ser costoso de realizar a gran escala debido a que etiquetar los datos a menudo es una tarea que se realiza de forma manual.

Algunos algoritmos de aprendizaje supervisado:

- **Máquinas de Soporte a Vectores (SVM)[6].** El primer algoritmo SVM es atribuido a Vladimir Vapnik en 1963, sin embargo las SVM son el resultado del trabajo de varias personas durante muchos años.
  SVM es un algoritmo de clasificación binario, donde: dado un conjunto de puntos de 2 clases en el lugar $\mathbb{R}^n$ dimensional, la SVM genera un hiperlano $\mathbb{R}^{n-1}$ dimensional para separar esos puntos en 2 grupos.
- **Árboles de Decisión.** Herramienta de apoyo a la decisión que utiliza un modelo similar a un árbol de decisiones cuyo objetivo es evaluar la probabilidad de tomar una decisión correcta en la mayoría del tiempo. Este método permite abordar el problema de una manera estructurada y sistemática para llegar a una conclusión lógica.

[6]Support Vector Machine, por sus siglas en inglés

- **Clasificación Naïve Bayes.** Son una familia de clasificadores probabilísticos simples, basados en la aplicación de Bayes "teorema con fuertes (Naïve) supuestos de independencia entre las características".
- **Regresión por Mínimos Cuadrados Ordinarios.** Es un método para realizar la regresión lineal cuyo propósito es ajustar una línea recta a través de un conjunto de puntos.
  Existen varias estrategias posibles para hacer esto, como la de "mínimos cuadrados ordinarios" donde se dibuja una línea y luego, para cada uno de los puntos de datos, mide la distancia vertical entre el punto y la línea para después sumarlos. La línea ajustada sería aquella en la que esta suma de distancias sea lo más pequeña posible.
- **Regresión Logística.** Poderosa manera estadística de modelar un resultado binomial con una o más variables explicativas. Mide la relación entre la variable dependiente categórica y una o más variables independientes estimando las probabilidades utilizando una función logística, que es la distribución logística acumulativa.

b. Aprendizaje no supervisado.
En esta metodología, los puntos de datos no tienen etiquetas asociadas a ellos, de tal suerte que el objetivo de estos algoritmos es el de organizar los datos de cierta manera o de describir su estructura. Esto puede significar agruparlos o buscar diferentes formas de examinar datos complejos para que parezcan más simples o más organizados. Una de sus cualidades más robustas es que reduce significativamente la necesidad de mano de obra humana.

Algunos algoritmos de aprendizaje no supervisado:

- **Valores Singulares de Descomposición (SVD)[7].** Los SVD proporcionan la base matemática de muchos algoritmos modernos en la ciencia de datos, incluidos Minería de Textos, Sistemas de Recomendación y Procesamiento de Lenguaje Natural. En el álgebra lineal, los SVD son una factorización de una matríz compleja real. Para una matríz $M \cdot n$ dada, existe una descomposición tal que $M = U \sum V$, donde U y V son matrices unitarias y $\sum$ es una matríz diagonal.
- **Algoritmos Clustering.** Es la tarea de agrupar un conjunto de objetos tales que los objetos en el mismo grupo (cluster) son más similares entre sí que los de otros grupos.

[7] Singular Value Decomposition, por sus siglas en inglés.

- **Análisis de Componentes Principales (PCA)[8].** Es un procedimiento estadístico que usa una transformación ortogonal para convertir un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables linealmente no correlacionadas llamadas componentes principales.
- **Análisis de Componentes Independientes (ICA)[9].** Es una técnica estadística para revelar los factores ocultos que subyacen a conjuntos de variables, mediciones o señales aleatorias. ICA define un modelo generativo para los datos multivariados observados, que se suele dar como una gran Base de Datos de muestras.

c. Aprendizaje por refuerzo.
   Es un algoritmo que dicta una acción basada en datos de entrada, y luego el sistema recibe una recompensa (o ausencia de recompensa). Con el tiempo, el algoritmo aprende de una manera basada en datos y crea una política tal que, con nuevos datos de entrada, el sistema seleccione la mejor acción siguiente.

[8]Principal Components Analysis, por sus siglas en inglés.
[9]Independents Components Analysis, pos sus siglas en inglés.

Capítulo 4

Metodología

4.1 Justificación de los métodos a utilizar

La idea principal del proyecto es localizar la similitud de un texto dado y compararlo contra textos almacenados en una Base de Datos, para realizar esta comparación se plantea el uso de un algoritmo de Machine Learning (Aprendizaje Automático) denominado Support Vector Machine (Máquina de Soporte a Vectores) la cual permitirá realizar la clasificación entre los textos involucrados tanto el de la búsqueda como los previamente almacenados. Esta idea se muestra de manera simple en la figura 4.1 donde se involucra el uso de la Inteligencia Artificial para un fin específico.

Figura 4.1. Idea simple para obtener similitud de textos usando Inteligencia Artificial.

Para llegar al objetivo que se persigue se consideran algunos métodos propios de Inteligencia Artificial mismos que se visualizan en el diagrama de la figura 4.2; dentro de los Fundamentos de la Inteligencia Artificial se encuentra la Lingüistica y las Matemáticas; en la Lingüistica se encuentra el Procesamiento de Lenguaje Natural y en cuyos conceptos está la recuperación y extracción de información, así mismo dentro de su Terminología se encuentran métodos como tokenización, stemming y TF-IDF (Frecuencia del Término y Frecuencia Inversa del Documento) con el fin de realizar el tratamiento y conversión del texto en una matríz de datos. Dentro de las Ramas de la Inteligencia Artificial se encuentran los algoritmos de Deep Learning, Algoritmos Genéticos y propiamente *Machine Learning* del cual se desprenden algoritmos de aprendizaje *supervisado* y *no supervisado*, dentro de los algoritmos no supervisados está el de Valores Singulares de Descomposición, mientras que del lado de algoritmos supervisados se encuentra la Máquina de Soporte a Vectores, todo esto con bases matemáticas y aplicadas de forma intensa para su operación.

En el diagrama se plantea de forma general la secuencia que se llevará para realizar la clasificación de los textos, sometiéndolos a tratamientos de PLN, cuyo objetivo es transformar los textos a una matríz de n componentes es decir $\mathbb{R}^n$ dimensiones y utilizando el algoritmo de SVD se realizará la reducción de dimensión de la matríz de $\mathbb{R}^n$ a $\mathbb{R}^2$ y al final, someter los vectores involucrados en la matríz bidimensional a una Máquina de Soporte a Vectores, donde se realizará un entrenamiento previo con las dos clases resultantes e iniciar la clasificación con los vectores que se vayan planteando, todo esto en tiempo real.

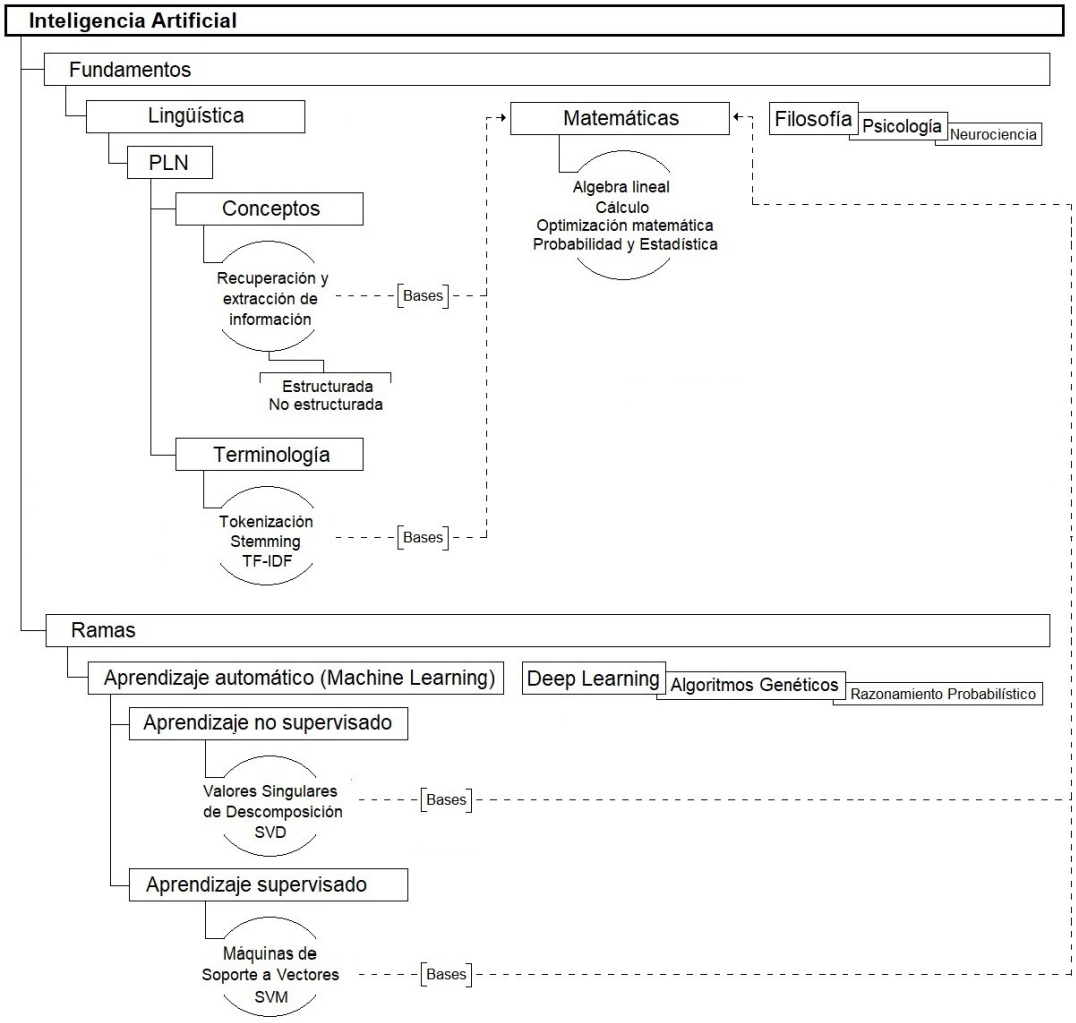


Figura 4.2. Diagrama general de los componentes principales de la Inteligencia Artificial.

4.2 Descripción de los métodos a utilizar

4.2.1 TF-IDF

Debido a que los textos son la representación más próxima a la expresión en lenguaje natural, es más complejo realizar búsqueda de oraciones con un significado específico, es decir dada una consulta, obtener todos aquellos documentos que traten de un tema en particular, ya que están clasificados como datos no estructurados y se caracterizan por tener una libre redacción

en su contenido, a diferencia de los datos estructurados, estos cuentan con un una palabra clave (atributos o características), asociados a un cierto elemento, el ejemplo más simple es el contenido de información en una Base de Datos, donde dado un nombre de atributo (campo o columna), se puede obtener una serie de registros que incluyen tal valor. Las palabras clave pueden combinarse en disyuntivas y conjunciones, proporcionando así una mayor expresividad de las consultas. Una consulta basada en palabras clave no puede identificar los documentos coincidentes de manera única y generalmente devuelve una gran cantidad de documentos, por lo tanto, en la Recuperación de Información es necesario clasificar los documentos por su relevancia[1] para la consulta.

Se pudiera decir que los textos en lenguaje natural están estructurados, lo cual es cierto, siempre y cuando se trate de la sintaxis del lenguaje, es decir su estructura gramatical, sin embargo, aun la transición al significado requiere de estructuración o comprensión semántica.

Modelo Espacio Vectorial

Define los documentos en vectores, así como transformar el texto en un modelo de espacio vectorial, donde los documentos son transformados en vectores y trasladados a un espacio multidimensional, donde las dimensiones son representadas por los términos que contiene cada documento. [8]

Nota: El modelo espacio vectorial no permite representar relaciones semánticas[2] entre las palabras.

Se emplean tres versiones básicas de representar el modelo de espacio vectorial y son: Booleano, TF e IDF, donde la idea principal es la de definir el valor de un término como una característica en una representación de documentos que puede ser el número de ocurrencias del término en el documento o en todo el Corpus.

Versiones básicas

a. Booleano
   Una de las ideas principales de la Recuperación de Información es el de recuperar documentos utilizando un criterio booleano simple y es cuando un término actúa como una

[1] La clasificación de relevancia es una diferencia importante con la consulta de datos estructurados donde el resultado de una consulta es un conjunto (colección no ordenada) de elementos de datos.

[2] Los espacios vectoriales semánticos se basan en la idea que el significado de una palabra puede ser aprendido de un entorno lingüístico y poseen dos enfoques, la semántica distribucional y la semántica composicional.

característica; en una representación del documento se debe comprobar si el término aparece o no en el documento, por lo tanto, el término se considera un atributo booleano.

Supongamos que existen n documentos $d_1, d_2, d_3 ..., d_n$ y m términos $t_1, t_2, t_3 ..., t_m$, donde n_{ij} denota el número de veces que el término t_i ocurre en el documento d_j. En la representación booleana, el documento d_j es representado como un vector con m componentes o características.

$$\vec{d}_j = d_j^1, d_j^2 \cdots d_j^m$$

Donde:

$$\vec{d}_j^i = \begin{cases} 0 & \text{si } n_{ij} = 0 \\ 1 & \text{si } n_{ij} > 0 \end{cases}$$

En el ejemplo de la Tabla 4.1 se muestran *siete* documentos que contienen *cinco* términos, considerando el documento *dos* se obtiene como resultado la siguiente representación: $\vec{d}_2 = (01100)$

Tabla 4.1. TF Booleano.

ID documento	t1	t2	t3	t4	t5
d1	0	1	0	1	1
d2	0	1	1	0	0
d3	0	0	0	0	0
d4	0	0	0	1	0
d5	0	1	1	0	0
d6	1	0	0	1	0
d7	0	0	0	0	1

b. TF (Term Frequency)
Frecuencia del término, es la cantidad de veces que una palabra se repite en un documento, parecido a la densidad de la palabra clave.
En la Tabla 4.2 se representa la forma más básica donde TF representa cuántas veces el término t_i aparece en el documento d_j.

Tabla 4.2. TF Frecuencia del Término.

ID documento	t1	t2	t3	t4	t5	Total de términos
d1	0	$\frac{2}{23}$	0	$\frac{3}{23}$	$\frac{1}{23}$	23
d2	0	$\frac{2}{18}$	$\frac{4}{18}$	0	0	18
d3	0	0	0	0	0	26
d4	0	0	0	$\frac{2}{31}$	0	31
d5	0	$\frac{2}{40}$	$\frac{5}{40}$	0	0	40
d6	$\frac{3}{16}$	0	0	$\frac{1}{16}$	0	16
d7	0	0	0	0	$\frac{2}{12}$	12

$$TF(t_i, d_j) = \begin{cases} 0 & \text{si } n_{ij} = 0 \\ \frac{n_{ij}}{\sum_{k=1}^{m} n_{kj}} & \text{si } n_{ij} > 0 \end{cases}$$

c. TF-IDF (Term Frecuency-Inverse Document Frecuency)
Frecuencia inversa del documento, se encarga de analizar la relevancia de una palabra clave con base en el número de veces que se repite en un documento y con base en otros documentos, pero con la variedad de que se compensa con la frecuencia del término en el Corpus, de tal suerte que ayuda en el control debido a que algunas palabras son por lo general más comunes que otras. [9]

Para este cálculo se utiliza la fórmula siguiente 4.1, donde el numerador es el número total de documentos en el Corpus, y el denominador es el número de documentos donde el término *t* aparece.

$$IDF(t, D) = log \frac{N}{|\{d \in D : t \in d\}|} \tag{4.1}$$

Si un término no aparece en el documento entonces se produciría un error por división entre cero. Se podría ajustar la fórmula como sigue:

$$IDF(t, D) = log \frac{N}{1 + |\{d \in D : t \in d\}|} \tag{4.2}$$

En la tabla 4.3 se representan los cálculos utilizados con el método *TF-IDF*.

Tabla 4.3. TF-IDF Frecuencia Inversa del Documento, cálculo 1.

ID documento	t1	t2	t3	t4	t5	Total de términos
d1	0	2	0	3	1	23
d2	0	2	4	0	0	18
d3	0	0	0	0	0	26
d4	0	0	0	2	0	31
d5	0	2	5	0	0	40
d6	3	0	0	1	0	16
d7	0	0	0	0	2	12
	0.8451	0.3680	0.5441	0.3680	0.5441	166

Como ejemplo se calcula el *IDF* con la fórmula anteriormente descrita y para el término $t2$, donde: $log(7/3) = 0.3680$, ya que se tienen 7 documentos y el término $t2$ aparece 3 veces y así subsecuentemente para todos los términos.

Posteriormente cada TF se multiplica por el *IDF* correspondiente a cada término, como ejemplo $d2$ y $t3$ se representaría como sigue:

$$TF\text{-}IDF(\text{“}t3\text{”}, d2)) = (4) \cdot (0.5441) = 2.1763$$

En la tabla 4.4 se muestran los resultados del total de las operaciones entre los documentos y los términos.

Tabla 4.4. TF-IDF Frecuencia Inversa del Documento, cálculo 2.

ID documento	t1	t2	t3	t4	t5	Total de términos
d1	0	0.7360	0	1.1039	0.5441	23
d2	0	0.7360	2.1763	0	0	18
d3	0	0	0	0	0	26
d4	0	0	0	0.7360	0	31
d5	0	0.7360	2.7203	0	0	40
d6	2.5353	0	0	0.3680	0	16
d7	0	0	0	0	1.0881	12
	0.8451	0.3680	0.5441	0.3680	0.5441	166

Utilidades:

- Ranking (ordenaciones) de enlaces en buscadores web.
- Generación de resúmenes de textos.
- Agrupación y clasificación de documentos textuales.
- Autentificación de autoría de un texto.

[10]

4.2.2 Reducción de Dimensionalidad

Es el proceso de reducir el número de variables aleatorias o atributos[3] bajo consideración. En la reducción de la dimensionalidad, los esquemas de codificación de datos se aplican para obtener una representación "comprimida" de los datos originales.

Este método proporcionan mejores resultados si los datos a analizar se han normalizado, es decir, se han escalado a un rango más pequeño, como $[0.0, 1.0]$.

Algunas técnicas típicas para el uso de la reducción de dimensionalidad son la *Transformada de Fourier Discreta* (DFT), *Transformadas de Wavelet Discretas* (DWT) y *Valores Singulares de Descomposición* (SVD) basada en el *Análisis de Componentes Principales* (PCA). [11]

4.2.3 Valores Singulares de Descomposición (SVD)

SVD es la técnica matemática subyacente a un tipo de recuperación de documentos y el método de similitud de palabras denominado *"análisis semántico latente"*. La idea que subyace al uso de SVD para estas tareas es que toma los datos originales, que generalmente consisten en alguna variante de un documento o matríz, y lo divide en componentes linealmente independientes. Estos componentes son en cierto sentido, una abstracción lejana de las correlaciones ruidosas encontradas en los datos originales a conjuntos de valores que mejor se aproximan a la estructura subyacente del conjunto de datos a lo largo de cada dimensión de forma independiente. [12]

Utilidades:

- La idea básica detrás de SVD es tomar un conjunto de datos de alta dimensión, y reducirlo a un espacio dimensional inferior que expone claramente la subestructura de los datos y ordena la mayoría de las variaciones al mínimo.

[3] Archivo de datos, que representa una característica o características de un objeto de datos. (*Término utilizado por profesionales de minería y bases de datos*).

- Práctico para las aplicaciones de PLN. Simplemente puede ignorar la variación debajo de un límite particular para reducir masivamente sus datos, con toda la certeza de que los datos originales se han preservado.

Para tales efectos se utilizará una matríz determinada en un espacio dimensional $\mathbb{R}^n$ con el objetivo de disminuir ese espacio a dos dimensiones $\mathbb{R}^2$ y cuya matríz resultante servirá para entrenar una Máquina de Soporte a Vectores.

En el álgebra lineal los SVD son una factorización de una matríz compleja real. Para una matríz $M \cdot n$ dada, existe una descomposición tal que $M = U \sum V$, donde U y V son matrices unitarias y $\sum$ es una matríz diagonal.

$$A = U \sum V^t \tag{4.3}$$

Donde:

U es la matríz ortogonal de eigenvectores de AA^t
V es la matríz ortogonal de eigenvectores de A^tA
Los componentes de $\sum$ son los eigenvalores de A^tA

Si se contara con 7 atributos y 9 documentos, se generaría una matríz de tamaño (7×9) como se muestra a continuación:

$$A= \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Para realizar la normalización de la matríz se toman en cuenta las siguientes consideraciones:

$$(n) = \begin{cases} Col_n : & \dfrac{valor}{\sqrt{\sum valor}} \\ ejemplo\ Col_4 = & \dfrac{1}{\sqrt{5}} = .45 \\ s.a.\ valor > 0 & \end{cases}$$

Donde se obtiene una matríz A normalizada:

$$A^{(n)}= \begin{bmatrix} 0 & .58 & 0 & .45 & .71 & 0 & .71 \\ 0 & .58 & .58 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .71 & .71 \\ 0 & 0 & 0 & .45 & 0 & 0 & 0 \\ 0 & .58 & .58 & 0 & 0 & 0 & 0 \\ .71 & 0 & 0 & .45 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .71 & .71 & 0 \\ 0 & 0 & .58 & .45 & 0 & 0 & 0 \\ .71 & 0 & 0 & .45 & 0 & 0 & 0 \end{bmatrix}$$

Al realizar las operaciones de las matrices de la ecuación 4.3, se obtiene la matríz ortogonal de eigenvectores U=AA^t:

$$U= \begin{bmatrix} -.698 & -.095 & 0.017 & \cdots & 0.144 \\ -.262 & 0.295 & 0.469 & \cdots & -.157 \\ -.352 & -.450 & -.103 & \cdots & -.049 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -.188 & 0.376 & -0.504 & \cdots & 0.034 \end{bmatrix}$$

La siguiente matríz es la ortogonal de eigenvectores representada por V=A^tA:

$$V= \begin{bmatrix} -.1687 & 0.4192 & -.5986 & \cdots & 0.2433 \\ -.4472 & 0.2255 & 0.4641 & \cdots & -.4987 \\ -.2692 & 0.4206 & 0.5024 & \cdots & 0.4451 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -.4702 & -.3037 & -.0507 & \cdots & 0.3407 \end{bmatrix}$$

Los componentes resultantes de $\sum$=A^tA (eigenvalores), se les genera una matríz diagonal S pero solo con los dos primeros elementos para obtener solo vectores en 2 dimensiones es decir de rango 2.

S rango 2:

$$S2= \begin{bmatrix} 1.585 & 0 & 0 & \cdots & 0 \\ 0 & 1.272 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Por último al realizar la multimplicación de las matrices resultantes de V y S se obtiene una matríz que determina los vectores de 2 dimensiones.

$$V \cdot S2= \begin{bmatrix} -0.2674 & 0.5333 & 0 & \cdots & 0 \\ -0.7088 & 0.2869 & 0 & \cdots & 0 \\ -0.4267 & 0.535 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -0.7452 & -0.3863 & 0 & 0 & 0 \end{bmatrix}$$

Cada par de vectores representa la ubicación de un documento en un espacio $\mathbb{R}^2$ dimensional.

4.2.4 Máquina de Soporte a Vectores (SVM)

La Máquina de Soporte a Vectores es un algoritmo de clasificación binario, donde dado un conjunto de datos de entrenamiento de 2 clases en el lugar $\mathbb{R}^n$ dimensional, la SVM genera un hiperlano $\mathbb{R}^{n-1}$ dimensional para separar esos puntos en 2 grupos. [13]

Las SVM obtienen un hiperplano óptimo utilizando vectores de soporte con la finalidad de encontrar el mayor margen que separe los datos de una clase con respecto a otra. El hiperplano óptimo se determina con un mapeo de valores iniciales, donde cada valor se representa como un vector en el espacio de características por ejemplo $\mathbb{R}^2$, por consiguiente, cada vector tiene una clasificación denominada “clase” y partiendo de estos datos originales se realiza el entrenamiento de la SVM, una vez finalizado el entrenamiento se puede asignar un nuevo vector en la ecuación del hiperplano óptimo, la clasificación de nuevo valor se obtiene del resultado final de la operación matemática, indicando a cuál clase pertenece dicho valor. [14]. En la figura 4.3, se muestra una máquina de Soporte a Vectores donde en (**a**) aparentemente se muestra un margen que separa ambas clases satisfactoriamente, mientras que en (**b**) se observa la separación de ambas clases con un hiperplano de márgen máximo.

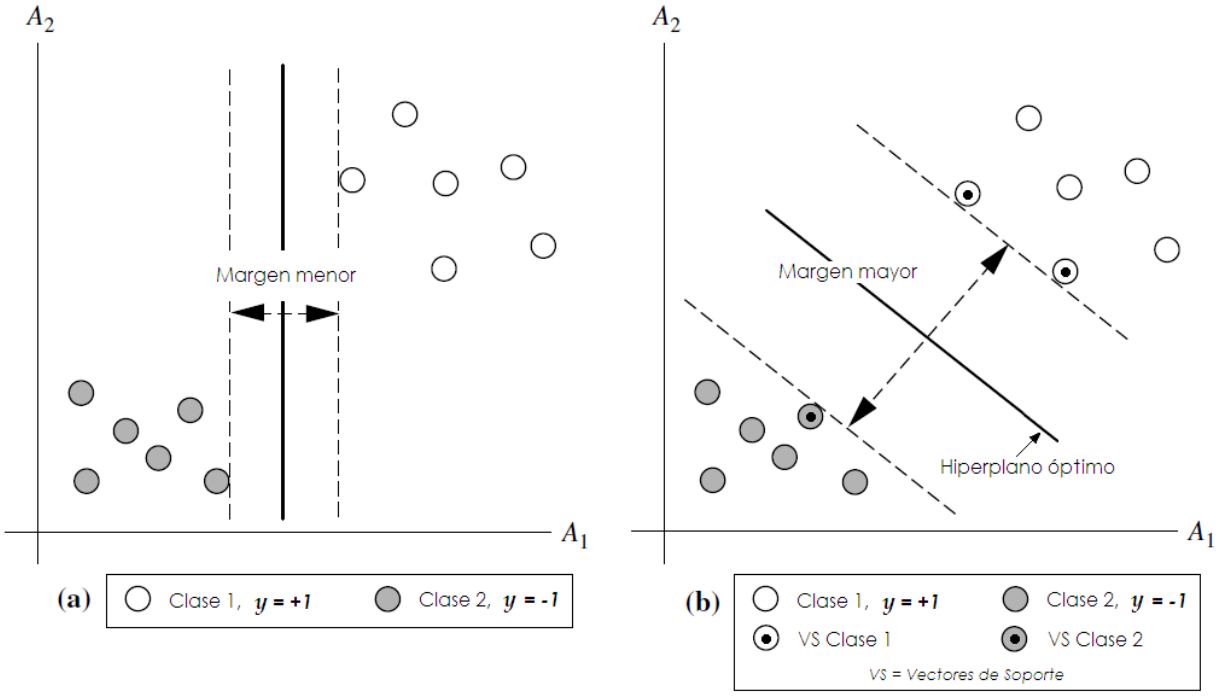


Figura 4.3. Comparación de márgenes de una SVM.

Donde se asocia cada vector $\mathbf{x}_i$ con la etiqueta y_i, que puede tener el valor $+1$ ó -1.

$$y_i \in \{1,-1\}_{i=1}^{m} \tag{4.4}$$

Dada la posición de $\mathbf{x}$ con respecto al hiperplano se puede predecir un valor para la etiqueta y ya que a cada punto de datos en un lado del hiperplano se le asignará la etiqueta -1, mientras que para los puntos de datos del lado contrario del hiperplano se les asignará la etiqueta $+1$.

Se define una función de hipótesis:

$$h(\mathbf{x}_i) = \begin{cases} +1 & \text{sí} \quad \mathbf{w}\cdot\mathbf{x}_i + b \geq 0 \\ -1 & \text{sí} \quad \mathbf{w}\cdot\mathbf{x}_i + b < 0 \end{cases}$$

Que es equivalente a:

$$h(\mathbf{x}_i) = \text{sgn}(\mathbf{w}\cdot\mathbf{x}_i + b) \tag{4.5}$$

Casos cuando los datos son linealmente separables:

El conjunto de datos $\mathscr{D}$ esta dada por $(\mathbf{x_1}, y_2)$, $(\mathbf{x_2}, y_2)$,...,$(\mathbf{x} \mid D \mid, y \mid D \mid)$, donde $\mathbf{x}_i$ es el vector de entrenamiento con una clase asociada y_i y cada y_i toma los valores correspondientes de $+1$, -1.

$$\mathscr{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, 1\}\}_{i=1}^{m} \tag{4.6}$$

Considerando dos atributos de entrada, A_1 y A_2 como se muestra en la figura 4.4, donde los datos están representados en un espacio de $\mathbb{R}^2$ y son separables linealmente ya que se puede dibujar una línea recta que separa los datos de la clase $+1$ de la clase -1.

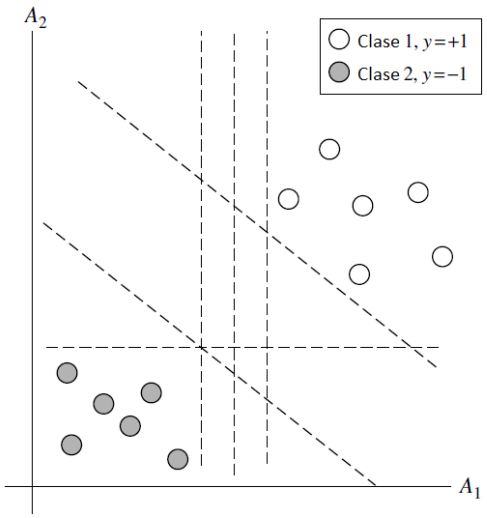


Figura 4.4. SVM, separable linealmente.

Hay un número infinito de líneas de separación que se pueden dibujar. La intención es encontrar la "mejor" línea, es decir, una que tenga el mínimo error de clasificación de los nuevos datos.

Encontrar el hiperplano óptimo

Considerando la figura 4.4 donde se representan varios hiperplanos de separación, se pudiera decir que todos cumplen con el objetivo de clasificar correctamente todos los datos dados. Sin embargo una SVM aborda este problema que es el de buscar el hiperplano óptimo en la fase de entrenamiento y se espera que el Hiperplano con el Mayor Margen (HMM) sea más preciso en la clasificación de futuras tuplas de datos y la mayor separación entre clases.

Una definición informal de margen, se puede decir que es la distancia más corta desde un hiperplano a un lado de su margen, a su vez, es la distancia más corta desde el hiperplano al otro lado de su margen, donde los "lados" del margen son paralelos al hiperplano.

Un hiperplano separador se puede representar con la siguiente ecuación:

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{4.7}$$

Donde **W** es un vector de peso, $\mathbf{W} = \{w_1, w_2, \cdots, w_n\}$; **n** es el número de atributos y **b** es el escalar o también conocido como *bias* (sesgo). Para ayudar en la visualización, consideremos dos atributos de entradas, A_1 y A_2, como se muestra en la figura 4.3(**b**).

Los vectores de entrenamiento están en un espacio de dimensiones $\mathbb{R}^2$ (e.g, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2)$), donde $\mathbf{x}_1$ y $\mathbf{x}_2$ son los valores de los atributos A_1 y A_2 respecto de **X**.

Si pensamos en **b** como un peso adicional, entonces se representaría como $\mathbf{w}_0$, y el resultado del producto punto de **W** con respecto a **X** donde $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2)$ y $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2)$ se reescribiría la ecuación 4.8 como sigue:

$$\mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 = 0 \tag{4.8}$$

Por lo tanto, cualquier punto que se encuentre sobre el hiperplano de separación satisface:

$$\mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 > 0 \tag{4.9}$$

De manera similar, cualquier punto que se encuentre debajo del hiperplano de separación satisface:

$$\mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 < 0 \tag{4.10}$$

Las ponderaciones se pueden ajustar de modo que los hiperplanos que definen los "lados" del margen. Se puede reescribir como:

$$\mathscr{H}_1 : \mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 \geq 1 \quad \text{para} \quad y_i = +1, \tag{4.11}$$

$$\mathscr{H}_2 : \mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 \leq -1 \quad \text{para} \quad y_i = -1 \tag{4.12}$$

Es decir, cualquier vector que cae en o por encima de $\mathscr{H}_1$ pertenece a la clase 1, y cualquier vector que cae en o por debajo de $\mathscr{H}_2$ pertenece a la clase -1. Combinando las dos desigualdades de las ecuaciones (4.11) y (4.12), obtenemos la ecuación 4.13

$$y_i(\mathbf{w}_0 + \mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2) \geq 1, \quad \forall_i \tag{4.13}$$

Cualquier vector de entrenamiento que caiga en los hiperplanos $\mathscr{H}_1$ o $\mathscr{H}_2$ satisfacen la ecuación 4.13 y se denominan vectores de soporte. Es decir, son igualmente cercanos a la separación (HMM). En la Figura 4.5, los vectores de soporte se muestran con un círculo remarcado en su centro. Esencialmente, los vectores de soporte son las tuplas más difíciles de clasificar, sin embargo son las que dan la mayor información con respecto a la clasificación.

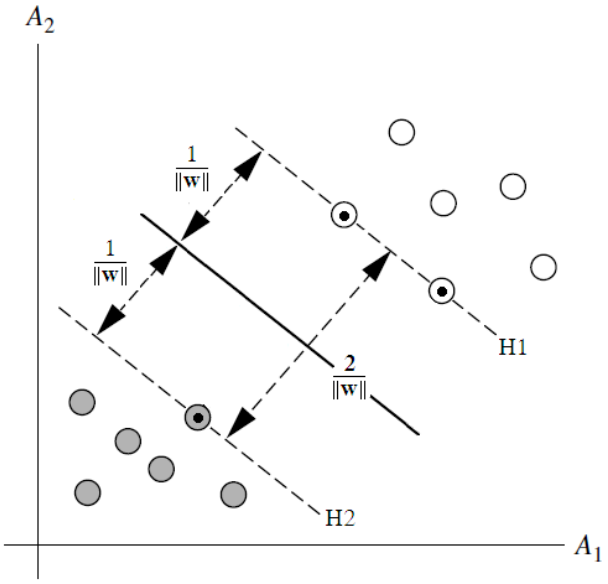


Figura 4.5. SVM, márgen máximo y vectores de soporte.

De esto, podemos obtener una fórmula para el tamaño del margen máximo. La distancia desde el hiperplano de separación a cualquier punto en $\mathscr{H}_1$ es $\frac{1}{||\mathbf{w}||}$, a su vez la distancia desde el hiperplano de separación para cualquier punto en $\mathscr{H}_2$ también se representa como $\frac{1}{||\mathbf{w}||}$, donde $||w||$ es la norma euclidiana de $\mathbf{W}$ que se calcula cómo $\sqrt{\mathbf{w} \cdot \mathbf{w}}$. Por lo tanto, el margen máximo es $\frac{2}{||\mathbf{w}||}$.

Vectores de Soporte

Usando una formulación Lagrangiana y luego resolviendo la solución usando condiciones de Karush-Kuhn-Tucker (KKT), se puede reescribir la ecuación (4.13) para que se convierta en un problema de optimización cuadrático. Para una lectura más avanzada sobre el tema consultar *R. Fletcher. Practical Methods of Optimization. JohnWiley & Sons, 1987. J. Nocedal and S. J. Wright. Numerical Optimization. Springer Verlag, 1999.*

Una vez entrenada la SVM, y basado en la formulación Lagrangiana mencionada anteriormente el HMM puede ser reescrito como el límite de decisión de la ecuación 4.14, donde los nuevos datos $\mathbf{X}^t$ se sustituyen en tal fórmula y se obtiene el resultado que indica la clase a la que pertenece el nuevo vector.

$$h(\mathbf{X}^t) = \text{sgn}\left(\sum_{i=1}^{S} \alpha_i y_i (\mathbf{X}_i \cdot \mathbf{X}^t) + b\right) \tag{4.14}$$

Donde y_i es la etiqueta de los *vectores de soporte* $\mathbf{X}_i$; mientras que $\mathbf{X}^t$ son los datos de entrenamiento; α_i y b son los parámetros numéricos que fueron determinados automáticamente por un algoritmo de optimización cuadrático señalado anteriormente; y S es el número de vectores de soporte. Los α_i son los multiplicadores *Lagrangianos*. Para datos linealmente separables, los vectores de soporte son un subconjunto de vectores de entrenamiento reales.

Dado un vector de prueba $\mathbf{X}^t$, incluido en la ecuación (4.14) y cuyo resultado de la operación aritmética nos indica en qué lado del hiperplano cae. Si el signo es positivo, entonces $\mathbf{X}^t$ cae por encima de la HMM, por lo que la SVM predice que $\mathbf{X}^t$ pertenece a la clase $+1$, si cae por debajo de la HMM la SVM predice que la $\mathbf{X}^t$ pertenece a la clase -1.

Si se observa la formulación *Lagrangiana* de la ecuación (4.14) contiene el producto punto entre el vector de soporte $\mathbf{X}_i$ y el vector de prueba $\mathbf{X}^t$, esto resultara muy útil para encontrar el HMM.

La complejidad del clasificador aprendido se caracteriza por la cantidad de los vectores de soporte en lugar de la dimensionalidad de los datos. Por lo tanto, los SVM tienden a ser menos propensos al exceso de ajustes que algunos otros métodos.

Si se eliminaran todos los demás datos de entrenamiento y se repitiera el entrenamiento, el mismo hiperplano de separación sería encontrado. Además, el número de vectores de soporte encontrados inicialmente se pueden utilizar para calcular un límite (superior) en la tasa de error esperada del clasificador SVM, que es independiente de la dimensionalidad de los datos. Una SVM con un pequeño número de vectores de soporte puede tener buena generalización, incluso cuando la dimensionalidad de los datos sea alta.

Casos cuando los datos no son linealmente separables:

¿Qué pasa cuándo los datos no son linealmente separables?, tal como se muestra en la figura 4.6.

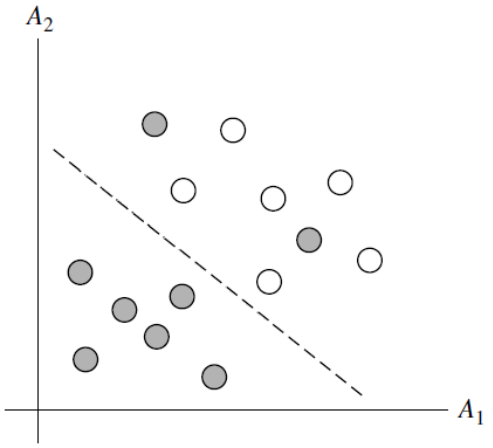


Figura 4.6. SVM, no linealmente separable.

El enfoque descrito para SVM lineales puede extenderse a crear SVM no lineales para la clasificación de datos linealmente inseparables.

La solución es transformar los datos de entrada originales en un espacio dimensional superior utilizando un mapeo no lineal. Una vez que los datos se han transformado en el nuevo espacio superior, se busca un hiperplano de separación lineal en el nuevo espacio. Nuevamente se termina con un problema de optimización cuadrático que se puede resolver usando formulación lineal.

El hiperplano de máximo márgen encontrado en el nuevo espacio corresponde a una hipersuperficie de separación no lineal en el espacio original, como se muestra en la figura 4.7.

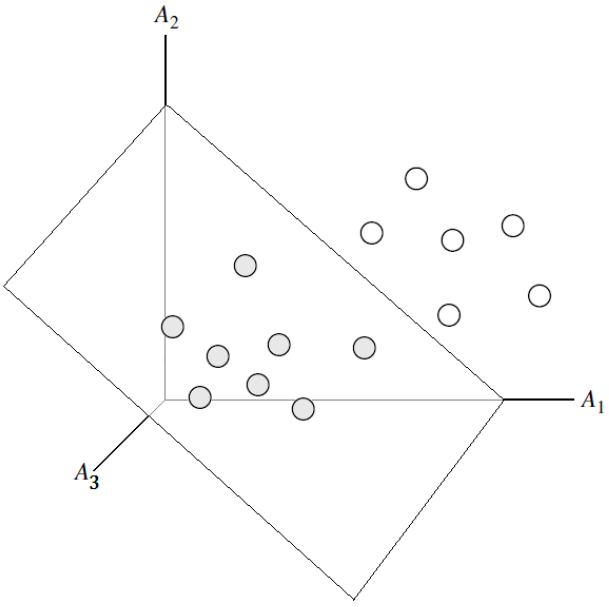


Figura 4.7. SVM, hiperplano de separación lineal en el nuevo espacio.

Transformación no lineal de datos de entrada originales en un espacio dimensional superior.

Considerando en vector de entrada en $\mathbb{R}^2$, $\mathbf{X} = (\mathbf{x_1}, \mathbf{x_2})$, se asigna en un espacio $\mathbb{R}^3$, denominado **Z**, representado como:

$$\phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \tag{4.15}$$

y usando los mapeos siguientes:

$$\phi_1(\mathbf{X}) = x_1^2, \quad \phi_2(\mathbf{X}) = \sqrt{2}x_1x_2, \quad \phi_3(\mathbf{X}) = x_2^2 \tag{4.16}$$

Un hiperplano de decisión en el nuevo espacio es $\mathbf{d}(\mathbf{Z}) = \mathbf{WZ} + b$, donde **W** y **Z** son vectores. Esto es lineal ya que se resuelve para **W** y b y luego se sustituyen los valores para que el hiperplano de decisión lineal sea el nuevo **Z**, el espacio corresponde a un polinomio de segundo orden no lineal en la entrada $\mathbb{R}^3$ del espacio original.

$$\mathbf{d}(\mathbf{Z}) = w_1x_1^2 + w_2\sqrt{2}x_1x_2 + w_3x_2^2 \tag{4.17}$$

$$= w_1z_1 + w_2z_2 + w_3z_3 \tag{4.18}$$

Sucede que al resolver el problema de optimización cuadrática de la SVM lineal (cuando se busca una SVM lineal en el nuevo espacio dimensional superior), los vectores de entrenamiento

aparecen solo en forma de producto punto $\phi(\mathbf{X}_i) \cdot \phi(\mathbf{X}^t)$ donde $\phi(\mathbf{X})$ es simplemente la función de mapeo no lineal aplicada para transformar los vectores de entrenamiento. En lugar de computar el producto punto en el vector transformado de datos, resulta que es matemáticamente equivalente a aplicar en su lugar una función de kernel $K(\mathbf{X}_i, \mathbf{X}^t)$ a los datos de entrada originales. Es decir,

$$\mathbf{K}(\mathbf{X}_i, \mathbf{X}^t) = \phi(\mathbf{X}_i) \cdot \phi(\mathbf{X}^t) \tag{4.19}$$

Por consiguiente la ecuación (4.14) se reescribe de la siguiente manera:

$$h(\mathbf{X}^t) = \text{sgn}\left(\sum_{i=1}^{S} \alpha_i y_i \mathbf{K}(\mathbf{X}_i \cdot \mathbf{X}^t) + b \right) \tag{4.20}$$

De esta manera, todos los cálculos se realizan en el espacio de entrada original, que es potencialmente de una dimensionalidad muy baja. Podemos evitarlos de forma segura, ya que ni siquiera tenemos que saber qué es el mapeo.

Después de aplicar el truco del Kernel, podemos proceder a encontrar una separación máxima.

Propiedades de los tipos de funciones kernel que podrían usarse para reemplazar el escenario del producto punto que se acaba de describir:

- Polinomial (grado h)

$$\mathbf{K}(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i \cdot \mathbf{X}_j + 1)^h \tag{4.21}$$

- (RBF) Función Gaussian Radial Basis

$$\mathbf{K}(\mathbf{X}_i, \mathbf{X}_j) = exp\left(- \frac{||\mathbf{X}_i - \mathbf{X}_j||^2}{2\sigma^2} \right) \tag{4.22}$$

- Sigmoide

$$\mathbf{K}(\mathbf{X}_i, \mathbf{X}_j) = tanh(k\mathbf{X}_i \cdot \mathbf{X}_j - \delta) \tag{4.23}$$

Cada uno de estos resulta ser un clasificador no lineal diferente en el espacio de entrada (original). No hay reglas de oro para determinar qué kernel admisible dará lugar a la SVM más precisa. En la práctica, el kernel elegido generalmente no hace una gran diferencia en la precisión resultante.

Un objetivo de investigación importante con respecto a los SVM es mejorar la velocidad en el entrenamiento y las pruebas para que los SVM se conviertan en una opción más viable

para conjuntos de datos muy grandes (por ejemplo, millones de vectores de soporte). Otras cuestiones incluyen determinar el mejor kernel para un conjunto de datos dado y encontrar métodos más eficientes para el caso multiclase.

Características de las SVM:

- Son altamente precisas debido a su capacidad para modelar límites complejos de decisión no lineales, a pesar de que pueden presentar lentitud en su entrenamiento, incluso de aquellas SVM más rápidas.
- Son mucho menos propensos a sobreajustes que otros métodos.
- Los vectores de soporte encontrados también proporcionan una descripción compacta de lo aprendido en el modelo.

Capítulo 5

Desarrollo

5.1 Modelo de negocio

El sistema de búsqueda de Información Pública con Técnicas de Inteligencia Artificial tiene la finalidad de utilizar métodos y algoritmos propios de la Inteligencia Artificial y resolver una problemática actual con herramientas diferentes a las empleadas tradicionalmente. Para iniciar con el desarrollo del proyecto se emplea una metodología única para la clasificación de textos donde se integran el uso de algunos métodos de Procesamiento de Lenguaje Natural como la Recuperación y Extracción de Información, a su vez, la utilización de un algoritmo de Aprendizaje Automático no supervisado denominado Valores Singulares de Descomposición para reducir la dimensión de una matríz determinada y finalmente para resolver la clasificación binaria de textos se considera un algoritmo de Aprendizaje Automático supervisado denominado Máquina de Soporte a Vectores.

El inicio del proyecto nace con el fin de contribuir en la resolución de la problemática mencionada en el apartado de "Planteamiento del Problema" de este documento y con la ventaja de que la información es pública y puede ser solicitada al Instituto Chihuahuense para la Transparencia y Acceso a la Información Pública, así mismo el proyecto es una propuesta del Instituto Tecnológico de Chihuahua II con el objetivo primordial de enriquecer el uso de técnicas de desarrollo de software concretamente en la creación de sistemas inteligentes que contribuyan en la resolución de problemas de la vida real.

Dada la naturaleza del proyecto se consideró en utilizar la metodología de desarrollo de software ágil, denominado, "Desarrollo dirigido por un plan" ya que se debe de considerar una serie

de pasos y métodos en cada etapa del desarrollo como son la Minería de Datos[1], antes de la codificación del software, así mismo incluir las "pruebas de desarrollo" que conllevan las "pruebas iniciales por unidad" y como etapa final las "pruebas de componentes".

5.2 Metodología de desarrollo dirigido por un plan

Para el desarrollo del sistema se consideró la metodología de desarrollo dirigido por un plan, ya que, para integrar la metodología planteada en el capítulo 4, es necesaria la preparación inicial de los datos que conllevan a una serie de pasos previos que permitirán estructurar los algoritmos, para con ello, obtener los resultados de cada operación, subsecuentemente, cada parte resultante se integrará en un componente final para la puesta en marcha. En cada etapa se harán las pruebas pertinentes ya que la metodología no se ha desarrollado en su conjunto en otros sistemas conocidos.

En la figura 5.1 se muestran los elementos de la metodología de desarrollo dirigido por un plan. [15]

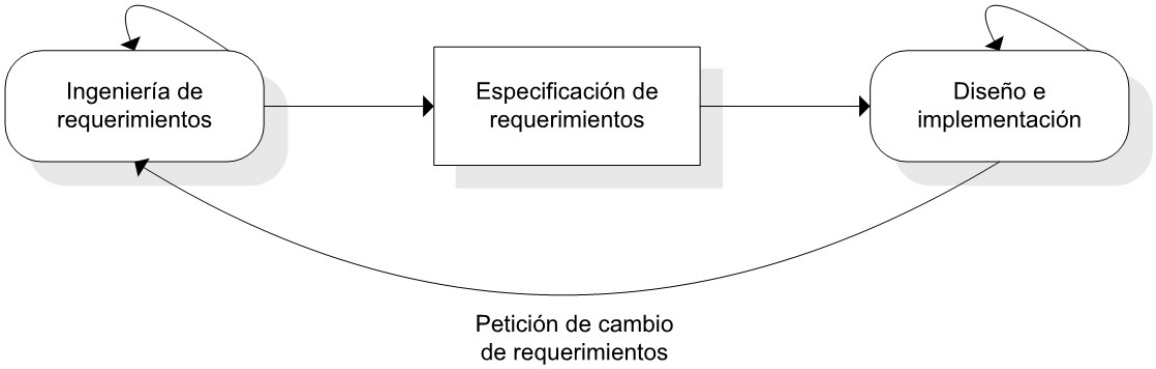


Figura 5.1. Metodología de desarrollo dirigido por un plan.

[1]Proceso de descubrir nuevas correlaciones significativas, patrones y tendencias por tamizado a través de grandes cantidades de datos almacenados en repositorios, utilizando tecnologías de reconocimiento de patrones, así como técnicas estadísticas y matemáticas. Gartner.

5.2.1 Estudio de factibilidad inicial

Con el objetivo de dar a conocer la factibilidad del sistema a desarrollar es necesario determinar las siguientes interrogantes:

a. ¿El sistema contribuye con los objetivos globales de la organización?
   Al integrar el sistema en la organización no solo contribuye en los objetivos globales de la organización, sino también en ofrecer una herramienta adicional para el acceso a la información pública, a su vez en tratar de brindar mayor certeza en los resultados de cada búsqueda.

b. ¿El sistema puede implementarse dentro de la fecha y con el presupuesto usando la tecnología actual?
   La fecha de implementación puede variar debido a que la información a buscar está acotada a ciertos años y puede provocar que se aplace el tiempo de entrega, un ejemplo de ello es la operación de normalización de los textos. Esta limitación se aborda en la sección de "Requerimientos funcionales" de este apartado.

   En lo que respecta al presupuesto, no hay inconveniente en este ámbito ya que solo se consideran costos mínimos operacionales como son el tiempo para realizar el proyecto y factores humanos para la preparación de los textos. A su vez, tanto el sistema actual como el sistema propuesto, funcionan bajo la plataforma Web y se ajustan a los requerimientos tecnológicos actuales.

c. ¿El sistema puede integrarse con otros sistemas que se utilicen?
   El sistema se integrará a un sistema que actualmente se encuentra en operación, es decir, funcionará a la par, simplemente será adicionado al sistema de operación actual. No existe ningún inconveniente que el sistema a implementarse se invoque desde el sistema actual, debido a que ambos sistemas comparten la misma plataforma es decir de tipo Web, pudiendo coexistir sin problema alguno.

5.2.2 Requerimientos funcionales

Definición de los requerimientos del usuario

1. Desde un navegador de internet, el usuario captura la URL [2] para ingresar al sistema.
2. El usuario seleccionará el año de la información a consultar.
3. el usuario selecciona el agrupador del Suejeto Obligado.

[2] Sigla del idioma inglés correspondiente a Uniform Resource Locator (Localizador Uniforme de Recursos).

4. El usuario selecciona al Sujeto Obligado que desea involucrar en la búsqueda.
5. Por medio de una entrada de texto (TextBox), el usuario captura una frase en lenguaje natural
6. Una vez que se muestran los resultados, el usuario seleccionará en la pantalla un hipervínculo con el folio resultante.
7. Por medio de un clic en el folio correspondiente se desplegara la información correspondiente.
8. Podrá realizar la cantidad de búsquedas que se desee.

La figura 5.2 muestra el diagrama de casos de uso de la definición de los requerimientos del usuario.

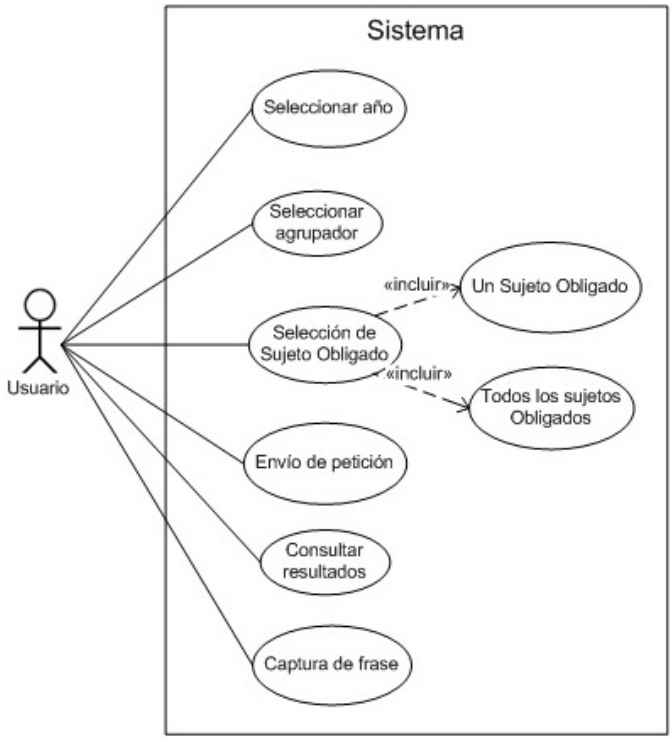


Figura 5.2. Diagrama de casos de uso de la definición de los requerimientos del usuario.

Especificación de los requerimientos del sistema

Una vez ingresada la URL por medio de un navegador de internet.

1. Permite la captura de una frase por medio de una entrada de texto (TextBox).
2. Permite incluir en la petición dada a un Sujeto Obligado en particular.
3. Permite incluir en la petición dada a todos los Sujetos Obligados.
4. Contiene un botón de envío para procesar la petición.
5. Sí el sistema encuentra similitud en las preguntas con la frase dada, desplegará un listado de folios.
6. Se crea un hipervínculo con cada folio resultante.
7. Al dar un clic en el hipervínculo del folio en cuestión se desplegará la información de las respuestas de una pregunta similar a la frase previamente capturada.
8. Permite restablecimiento de valores para una nueva consulta.

La figura 5.3 muestra el diagrama de casos de uso de la definición de los requerimientos del sistema.

5.2.3 Requerimientos no funcionales

Requerimientos del producto

- Se debe contar con conexión a internet para el ingreso al sistema.
- Versiones mínimas de los navegadores como Firefox 3.6.15, Internet Explorer 8.06, Opera 11.01, Chrome 10.0.
- Mínimo de memoria RAM: 200 MB.

Posibles fallas: El resultado de la búsqueda podría dar información diferente a la solicitada, ya que el algoritmo de Aprendizaje Automático Supervisado se entrena en tiempo real, es decir sin la asistencia humana.

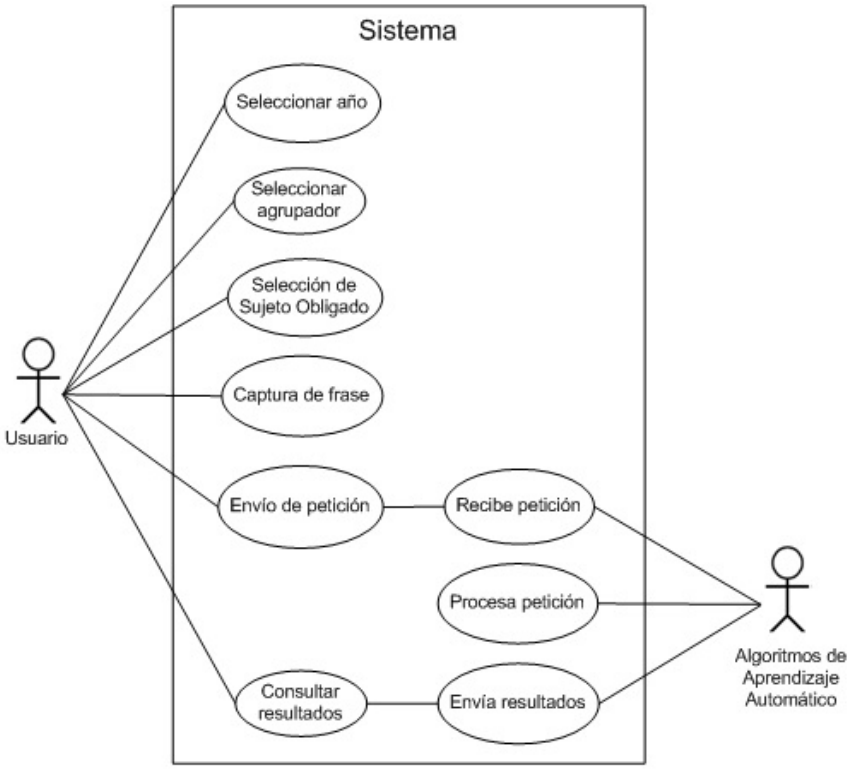


Figura 5.3. Diagrama de casos de uso de la definición de los requerimientos del sistema.

Requerimientos de la organización

- El sistema deberá estar disponible para cualquier usuario que tenga la inquietud de consultar las solicitudes de información previamente formuladas desde el sistema Infomex.

- El sistema de origen Infomex puede presentar cambios en la versión que modifiquen la operación inicialmente descrita y que indirectamente afecte el funcionamiento principal del nuevo sistema a implementar.

- Si existen cambios en el acceso a las solicitudes de información realizadas por medio del sistema Infomex, se deberá preveer que el sistema *Búsqueda de Información Pública con Técnicas de Inteligencia Artificial* esté disponible para realizar consultas de solicitudes de años anteriores.

Requerimientos externos

- La normalización de los textos en la Base de Datos solo se ha hecho con los folios de los años 2016 y 2017. Las pruebas se harán sobre los años en cuestión, para futuras búsquedas se debe de trabajar en la normalización de los años posteriores.

5.2.4 Modelo del sistema

A continuación se describen los estados generales y particulares del sistema. La descripción de los estados se refuerza con diagramas de actividades de cada proceso.

Proceso general del sistema

Tabla 5.1. Descripción de los estados del proceso general del sistema.

Estado	**Descripción**
Captura de texto.	Captura de texto a buscar desde la interfaz.
Normalizar texto.	Se le quitan todos aquellos caracteres no necesarios al texto, como símbolos, vocales acentuadas y todas aquellas palabras auxiliares que no cuenten con lema o raíz de la palabra, al final el texto es convertido a letras minúsculas.
Creación de matríz booleana.	Se crea una matríz con el mismo número de columnas como tokens tenga el texto capturado; se agregan filas por cada texto en el Corpus correpondiente a un año en particular y a cada celda se le agrega "1" si el término es igual al token correspondiente a la columna, de lo contrario se agrega "0".
Reducción de dimensionalidad con SVD. *(Algoritmo desarrollado por Sébastien Loisel)* [16]	La dimensión de la matríz es determinada por el número de tokens y número de filas. La matríz se reducira a 2 dimensiones para su tratamiento.
Cambiar vectores a positivos.	Una vez reducida la matríz a 2 dimensiones se cambiaran los valores de cada vector a números positivos.
Entrenamiento de SVM.	Se localizan los vectores de soporte y el máximo hiperplano separador entre las 2 clases resultantes. La clase -1 son los vectores que corresponden al texto capturado, mientras que la clase $+1$ son todos aquellos textos que no corresponden al grupo del texto capturado.
Clasificar textos.	Se toma cada vector de la matríz para realizar la clasificación por medio de la SVM.
Resultados.	Se despliegan los textos con mayor similitud al texto capturado.

En el **Anexo B** se muestra el diagrama de estados del proceso general del sistema.

Captura de texto

Tabla 5.2. Descripción de los estados del proceso de captura de texto.

Estado	Descripción
Captura texto.	El usuario captura una frase por medio de la interfaz.
Seleccionar año.	El usuario selecciona el año de la consulta.
Seleccionar sujetos obligados.	El usuario selecciona a todos los sujetos obligados o uno en particular para la búsqueda.
Enviar petición.	Se envía la petición dada para el proceso de búsqueda y despliegue de resultados.

En el **Anexo C** se muestra el diagrama de actividades la captura de texto.

Normalizar texto

Tabla 5.3. Descripción de los estados del proceso para normalizar el texto.

Estado	Descripción
Seleccionar texto.	Considerar todo el texto capturado.
Cambiar texto a minúsculas.	El texto se cambia a letras minúsculas.
Quitar stopwords del texto.	Al texto dado se le retiran los carácteres que no son necesarios como signos de interrogación, admiración y monetarios, así como vocales acentuadas entre otros símbolos inecesarios como para operaciones matemáticas, corchetes, paréntesis, etc.
Obtener tokens.	Se extraen los tokens del texto, después de eliminar los stopwords.
Sustitución de token por palabra raíz.	Se busca en la Base de Datos el token para obtener la palabra raíz.

En el **Anexo D** se muestra el diagrama de actividades del proceso para normalizar el texto.

Creación de matríz booleana

Tabla 5.4. Descripción de los estados del proceso de la creación de matríz booleana.

Estado	Descripción
Tokens.	Determinar número de tokens del texto dado.
Creación de matríz.	Se crea matríz donde el número de columnas será igual al número de tokens+1 con una sola fila.
Obtención de token y bigramas de texto dado.	Se hace un recorrido del texto para obtener los token y el token siguiente con el fin de obtener en el mismo recorrido el bigrama correspondiente.
Agregar el número de folios a la matríz booleana.	Por cada solicitud de información de un año en particular se incrementa una fila a la matríz y se guarda el número de folio en la fila *N*, columna 1.
Obtención de textos de las solicitudes de información de un año en particular de la Base de Datos.	De la Base de Datos se extraen los registros con el texto de cada solicitud de información de un año en particular.
Para cada registro de la Base de Datos.	Se toma el texto y se hace un recorrido de izquierda a derecha para extraer cada token; para obtener el bigrama se considera el token siguiente.
Buscar el término en la posición correcta en la matríz y asignar 1 ó 0.	Se toma el token y el bigrama creado y se busca la posición correspondiente en la columna de la matríz. Si existe correspondencia en el término o en el bigrama se asigna valor 1, de lo contrario 0.

En el **Anexo E** se muestra el diagrama de actividades de la creación de matríz booleana.

Reducción de dimensionalidad con SVD

Tabla 5.5. Descripción de los estados del proceso reducción de dimensionalidad con SVD.

Estado	Descripción
Normalizar matríz booleana.	Para cada valor de la referencia de la fila N, columna N se calcula tomando el valor de tal referencia y dividiendola por la raíz cuadrada de la sumatoria de la columna N.
Matríz ortogonal de eigenvectores de AA^t.	Por medio de un algoritmo externo se determina la matríz U.
Matríz ortogonal de eigenvectores de A^tA.	Por medio de un algoritmo externo se determina la matríz V.
Componentes de eigenvalores de A^tA.	Por medio de un algoritmo externo se determina la matríz S.
Matríz diagonal de S.	Por medio de un algoritmo se determina la diagonal de S.
Multiplicación de las matrices V por la diagonal de S.	Por medio de un algoritmo se realiza la operación de la multiplicación de V por diagonal de S de las 2 primeras filas. El resultado final es la creación de una matríz de 2 dimensiones.

En el **Anexo F** se muestra el diagrama de actividades del proceso de reducción de dimensionalidad con SVD.

Valor absoluto de los vectores

Tabla 5.6. Descripción de los estados del proceso para cambiar a valor absoluto los vectores.

Estado	Descripción
Realizar proceso a la matríz de 2 dimensiones.	Por medio de un ciclo se inicia el recorrido de la fila N por la columna N, si el valor es negativo se multiplica por -1. Así sucesivamente hasta que no existan valores negativos.

En el **Anexo G** se muestra el diagrama de actividades para cambiar a valor absoluto los vectores.

Entrenamiento de SVM

Tabla 5.7. Descripción de los estados del proceso de entrenamiento de SVM.

Estado	**Descripción**
Considerar matríz booleana (MB).	Se considera la matríz booleana para hacer recorrido de la misma. Necesaria para determinar los vectores de soporte.
Considerar matríz reducida a 2 dimensiones (SVD-2D).	Matríz obtenida del proceso de SVD reducida a 2 dimensiones.
Crear matríz de entrenamiento (ME).	Se crea matríz de entrenamiento para que exista y posteriormente ir agregando los datos que se vayan a considerar.
Recorrer matríz booleana (MB).	Recorrer la matríz booleana de inicio $(N+1)$ hasta el final para tomar los valores de las filas N, donde se determina el promedio de apariciones de valores en 1, si el valor es mayor o igual a 60%, se considera el folio de la matríz booleana (Columna 0), el folio se busca en la matríz (SVD-2D) para tomar los valores de los vectores de 2 dimensiones y agregarlos a la matríz de entrenamiento.
Matríz de entrenamiento (ME) con menos de 3 filas.	Si la matríz de entrenamiento cuenta con 1 fila de la clase -1 y más de la clase $+1$, se deberá agregar 4 vectores más para la clase -1 donde se suma .05 al vector de la clase -1 en diferentes magnitudes.

En el **Anexo H** se muestra el diagrama de actividades para el entrenamiento de SVM.

Determinar vectores de soporte

Tabla 5.8. Descripción de los estados del proceso determinar vectores de soporte.

Estado	**Descripción**
Considerar la matríz de entrenamiento (ME).	Se considera la matríz de entrenamiento del proceso anterior inmediato.
Crear matríz N filas por 6 columnas para vectores de soporte (VS).	Se crea una matríz de 6 columnas donde: Col_0=clase -1, Col_1=vector X_1 de la clase -1, Col_2=vector X_2 de la clase -1, Col_3=clase $+1$, Col_4=vector X_1 de la clase $+1$, Col_5=vector X_2 de la clase $+1$, Col_6=Distancia euclidiana de los vectores de la clase -1 y clase $+1$.
Ordenar matríz de entrenamiento (ME).	Ordenar la matríz de entrenamiento (ME) tomando por referencia la col_0, es decir por clases.
Recorrer matríz de entrenamiento (ME) "ciclo 1".	Se recorre la matríz de entrenamiento de inicio a fin para ubicar los vectores de la clase -1.
Mientras se recorre la matríz de entrenamiento (ME) "ciclo 1".	Si la Col_0 es clase -1 en matríz de entrenamiento (ME) se toman la columna 1 y 2.
Recorrer matríz de entrenamiento (ME) ciclo 2.	Se recorre la matríz de entrenamiento de inicio a fin para ubicar los vectores de la clase $+1$.
Mientras se recorre la matríz de entrenamiento (ME) "ciclo 2".	Si la Col_0 es de clase $+1$, se agrega una fila a la matríz de vectores de soporte (VS), almacenando los datos correspondientes provenientes de la matríz (ME) tanto del ciclo 1 como del ciclo 2, donde en la matríz (VS): Col_0=Col_0 de (ME) ciclo 1, Col_1=Col_1 de (ME) ciclo 1, Col_2=Col_2 de (ME) ciclo 1, Col_3=Col_0 de (ME) ciclo 2, Col_4=Col_1 de (ME) ciclo 2, Col_5=Col_2 de (ME) ciclo 2, Col_6=Distancia euclidiana de Col_1,Col_2 contra Col_3,Col_4.
Fin de recorrido de la matríz de entrenamiento (ME) ciclo 1.	Al finalizar el recorrido de la matríz de entrenamiento se determinan los vectores de soporte de la matríz de (VS).
Determinar vectores de soporte de la matríz (VS).	De la matríz (VS) resultante, se consideran 2 vectores de la clase -1 y 1 vector de la clase $+1$, para realizar las operaciones subsecuentes con los 3 vectores finales.

En el **Anexo I** se muestra el diagrama de actividades para determinar vectores de soporte.

Clasificar textos

Tabla 5.9. Descripción de los estados del proceso para clasificar textos.

Estado	**Descripción**
Tomar matríz (VS).	Se considera la matríz de vectores de soporte (VS).
Agregar bias.	Agregar bias, "un renglón adicional con el valor de 1 a cada vector de la matríz (VS)".
Matríz de Gramm	Se genera la matríz de Gramm con la matríz de (VS)
Obtener valores de α_1, α_2 y α_3.	Por medio del método de determinates se obtienen los valores todas las α_n.
Obtener los valores de w y b.	Se sustituyen los valores de alfas en la matríz para obtener los valores de w y b.
Iniciar recorrido de la matríz de 2 dimensiones.	Se recorre la matríz de 2 dimensiones para realizar la operación de la ecuación $(wx+b)$.
Clasifica el resultado de la ecuación.	Se obtiene el resultado de la ecuación para hacer la clasificación de la clase, donde se determina si pertenece a la clase -1 o $+1$.
Despliegue de resultados.	Despliegue de folios obtenidos clasificados.

En el **Anexo J** se muestra el diagrama de actividades para clasificar textos.

5.2.5 Arquitectura del sistema

En la figura 5.4 se muestra la Arquitectura del Sistema. Como primer paso, se inicia con el ingreso del texto desde la interfaz de usuario, enseguida se envía la petición vía Internet al servidor que recibirá la propuesta. Posteriormente se gestiona la petición del lado del servidor realizando la consulta a la Base de Datos con el objetivo de seleccionar los registros que cumplan con los "términos" contenidos en el texto ingresado, posteriormente se procesan los datos con el método TF del PLN, subsecuentemente por medio de Algoritmos de Aprendizaje Automático, se construye una matríz de 2 dimensiones creada por el algoritmo de SVD, esta matríz, será usada por el algoritmo de SVM para realizar la clasificación binaria de textos, los folios que cumplan con las características planteadas en el texto recibido, serán buscados en la Base de Datos con el fin de obtener el resto de la información como son, las preguntas y respuestas, así como los archivos adjuntos en caso de existir; al final el servidor envía los resultados al usuario.

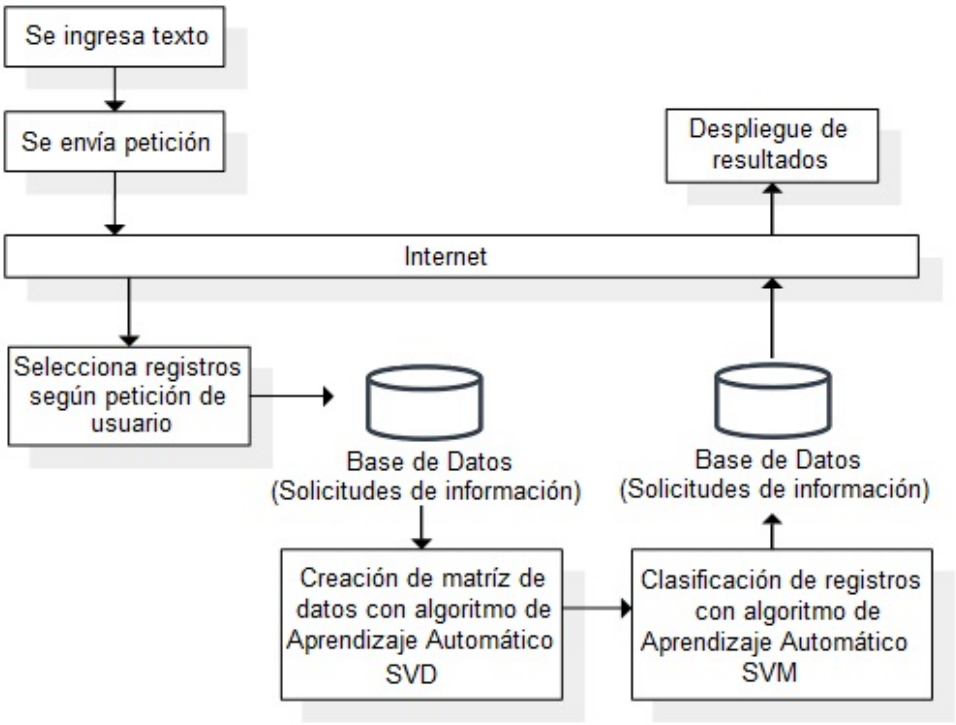


Figura 5.4. Arquitectura del sistema.

5.2.6 Apéndices

Para llevar a cabo el desarrollo del software a implementar, se consideran los siguientes elementos que a continuación se muestran:

Herramientas

Lenguajes de programación

- PHP (Hypertext Preprocessor). Lenguaje de programación de código abierto, adecuado para el desarrollo web y que puede ser incrustado en HTML.
  Version a utilizar: 5.6.30.
- Javascript. Lenguaje de programación que permite crear acciones en páginas Web, no requiere de compilación ya que el lenguaje funciona del lado del cliente.
- HTML (HyperText Markup Language). Lenguaje de Marcas de Hipertexto, define el contenido de las páginas Web.

Gestor de Base de Datos

- MySQL. Aplicación que permite gestionar archivos llamados desde Bases de Datos.
  Versión a utilizar: 5.0.11.

Requerimientos de la Base de Datos

En la figura 5.5 se muestra la Base de Datos a utilizar en el sofware a implementar.

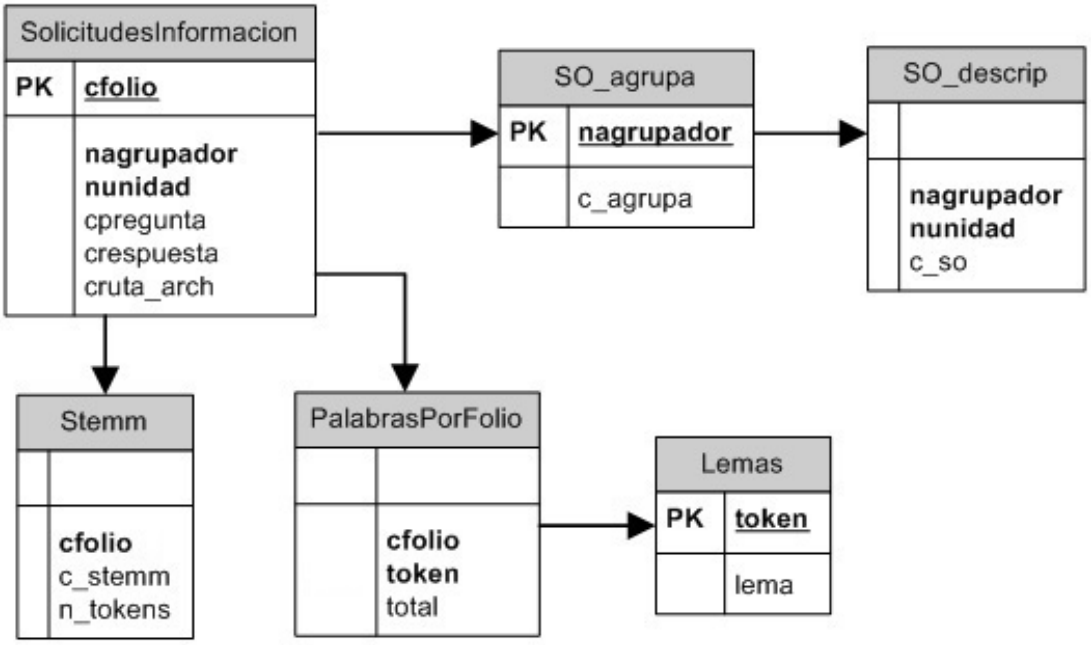


Figura 5.5. Base de Datos.

Estructura de datos y descripción de campos: A continuación se muestran la estructura de datos de las tablas de la Base de Datos, así como la descripción de los campos.

Archivo: SolicitudesInformación. Contiene los registros de las solicitudes de información realizadas en los años de 2016 y 2017 a los distintos sujetos obligados; en estos registros se encuentra la información siguiente: número del agrupador, número del sujeto obligado, pregunta, respuesta y ruta del archivo adjunto.

Tabla 5.10. Archivo SolicitudesInformacion.

Campo	Tipo	Longitud	Llave	Descripción
cfolio	char	9	primaria	Número de folio
nagrupador	decimal	2,0		Número de agrupador
nunidad	decimal	2,0		Número de sujeto obligado
cpregunta	mediumtext			Pregunta
crespuesta	longtext			Respuesta
cruta_arch	mediumtext			Ruta de archivo adjunto

Archivo: SO_agrupa. Catálogo que contiene los registros de los números y nombres de los agrupadores. Ver tabla 5.11.

Tabla 5.11. Archivo SO_agrupa.

Campo	Tipo	Longitud	Llave	Descripción
nagrupador	decimal	2,0	primaria	Número de agrupador
c_agrupa	char	80		Nombre del agrupador

Archivo: SO_descrip. Catálogo que contiene los registros del número y nombre de los sujetos obligados, así como el número del agrupador al que pertenece el sujeto obligado en cuestión. Ver tabla 5.12.

Tabla 5.12. Archivo SO_descrip.

Campo	Tipo	Longitud	Llave	Descripción
nagrupador	decimal	2,0		Número de agrupador
nunidad	decimal	2,0		Número de sujeto obligado
c_so	char	255		Nombre del sujeto obligado

Archivo: Stemm. Contiene los registros de las preguntas de los folios de las solicitudes de información de los años 2016 y 2017. El campo c_stemm contiene las preguntas que se han sometido al proceso de eliminar los elementos no necesarios en la oración, además cada token está reducido a la raíz de la palabra. Ver tabla 5.13.

Tabla 5.13. Archivo Stemm.

Campo	Tipo	Longitud	Llave	Descripción
cfolio	char	9		Número de folio
c_stemm	mediumtext	2,0		Texto en stemm
n_tokens	smallint	6		Número de tokens

Archivo: PalabrasPorFolio. Contiene los registros de las token de cada folio (proceso que se realizó con el apoyo de la tabla Stemm). Ver tabla 5.14.

Tabla 5.14. Archivo PalabrasPorFolio.

Campo	Tipo	Longitud	Llave	Descripción
cfolio	char	9		Número de folio
token	char	60		Palabra sola
total	int	6		Número de veces que el token se repite en la oración

Archivo: Lemas. Tabla de apoyo que sirvió para cambiar cada palabra de la oración a su raíz. Ver tabla 5.15.

Tabla 5.15. Archivo Lemas.

Campo	Tipo	Longitud	Llave	Descripción
token	char	30	primaria	Palabra sola
lema	char	20		Palabra lematizada

Interfaz de usuario

Por medio de esta interfaz, el usuario podrá seleccionar los elementos que mejor le complazcan con el fin de realizar la búsqueda de la frase previamente capturada. En la figura 5.6 se muestra la pantalla principal del sistema de *Búsqueda de Información Pública con Técnicas de Inteligencia Artificial.*

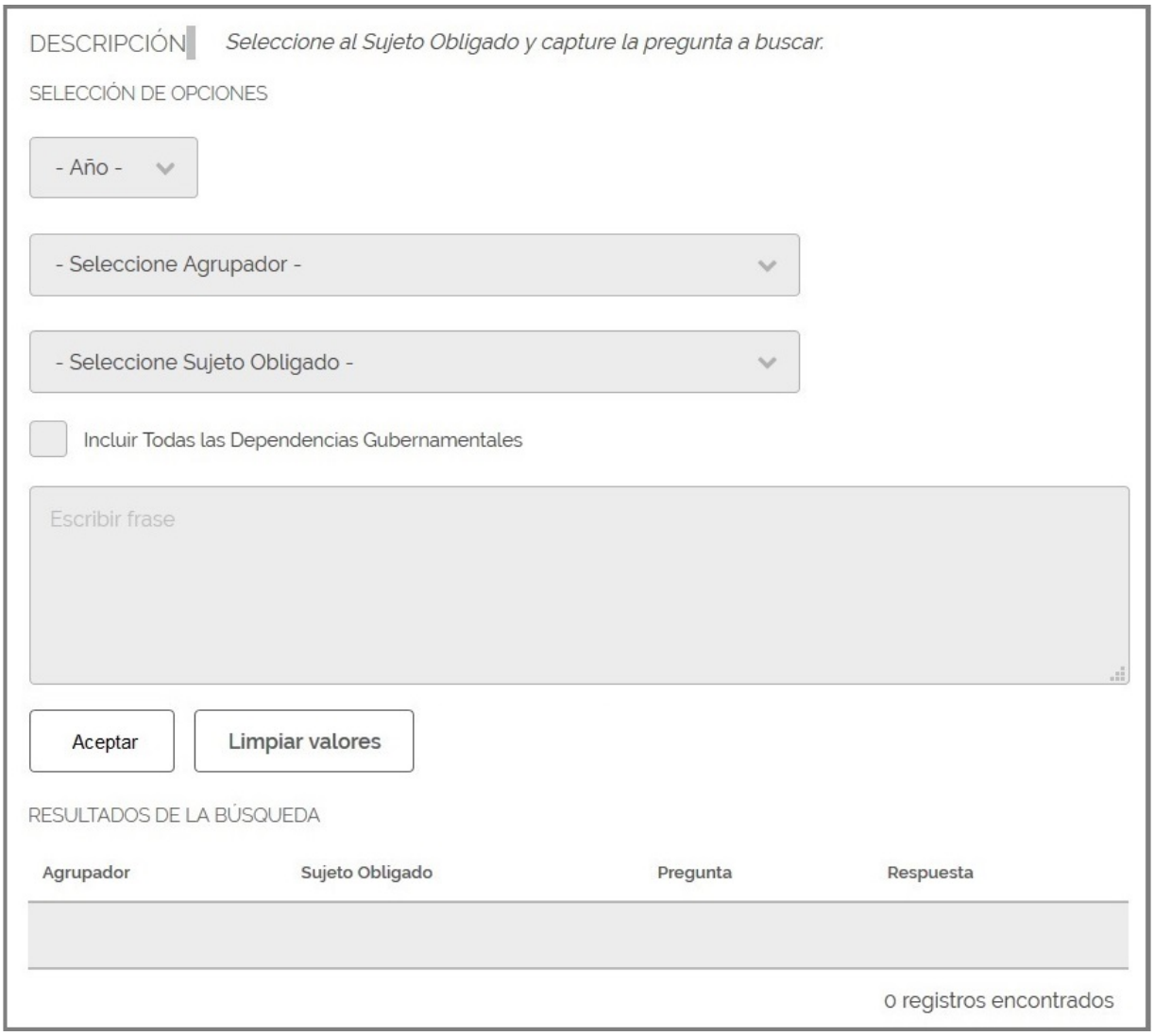


Figura 5.6. Pantalla principal del sistema de búsqueda de Información Pública con Técnicas de Inteligencia Artificial.

La interfaz de usuario se divide en 7 secciones y se describen a continuación:

a. Año. La figura 5.7 muestra que al dar clic sobre ese elemento permite seleccionar alguno de los años ya sea 2016 o 2017.

Figura 5.7. Selección de año.

b. Agrupador. La figura 5.8 muestra que al dar clic sobre ese elemento permite seleccionar algún agrupador.

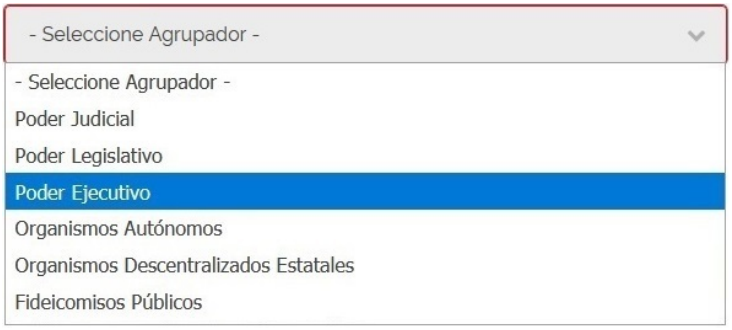


Figura 5.8. Selección de agrupador.

c. Sujeto obligado. La figura 5.9 muestra que al dar clic sobre ese elemento permite seleccionar algún sujeto obligado.

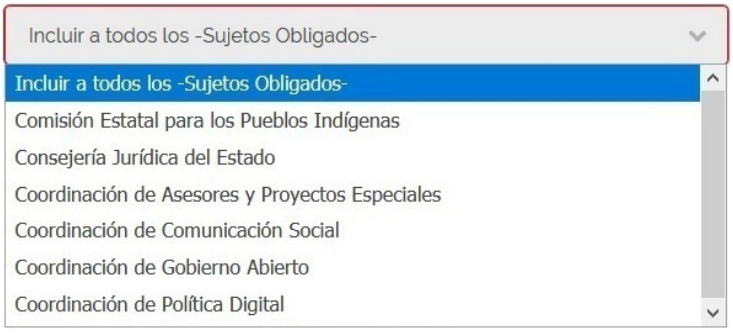


Figura 5.9. Selección de sujeto obligado.

d. Incluir todas las Dependencias Gubernamentales. Dar clic para activar el elemento de incluir a todas las Dependencias Gubernamentales, en caso de que se active esta opción las opciones de *Agrupador* y *Sujeto Obligado* se deshabilitarán. Ver figura 5.10.

Figura 5.10. Selecciona a todas las Dependencias Gubernamentales.

e. Frase. La figura 5.11 muestra el espacio designado para capturar la frase que se desea buscar.

Figura 5.11. Captura de frase.

f. Botones. La figura 5.12 muestra los botones para enviar la petición o limpiar los valores del formulario.

Figura 5.12. Enviar petición y limpiar valores.

g. Resultados. La figura 5.13 muestra el tabulador con los resultados de la petición solicitada. En caso de que la respuesta tenga archivo adjunto, este se verá como un enlace para ser consultado.

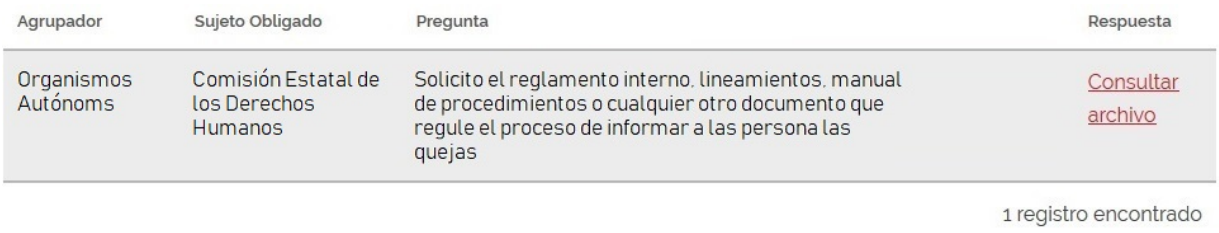


Figura 5.13. Resultados de la consulta.

5.3 Pruebas de desarrollo

A continuación se muestra una de las multiples pruebas que se realizaron antes de crear la interfaz de usuario.

La intención es mostrar las salidas de las pruebas y los resultados de las operaciones matemáticas de los componentes implicados en cada método. Los detalles de cada operación ya se describieron con detalle en el apartado *5.2.4 Modelo de sistema* de este capítulo, por lo que no es necesario ahondar en el tema.

5.3.1 Pruebas iniciales por unidad

A continuación se describen las pruebas de unidad que se realizaron por separado con la captura de una frase de prueba, considerando los registros de todos los sujetos obligados del año 2016.

Operaciones a modo de prueba para la captura de la frase

Se creó una interfaz temporal para realizar una de las pruebas con la siguiente frase escrita en lenguaje natural: *"Cuántas quejas de acoso sexual y hostigamiento hay en el estado de Chihuahua"*.

Por medio de la interfaz de usuario se envía la petición al servidor para que a su vez, éste, realice la gestión de los datos. Ver figura 5.14.

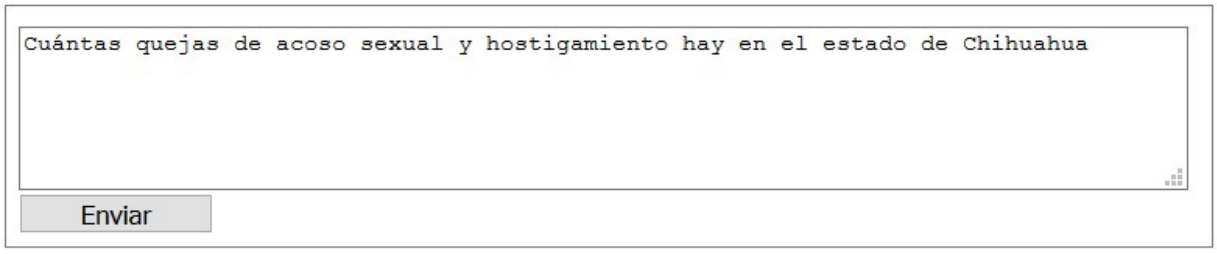


Figura 5.14. Interfaz de usuario temporal para la captura y envío de la frase al servidor.

Operaciones a modo de prueba para la creación de matríz TF

En la figura 5.15 se visualizan los resultados de la secuencia de operaciones que se hicieron sobre la frase inicial dada, tal frase, es sometida a una depuración quitando los *stopwords* y dejando

solo los *tokens lematizados* en minúsculas y posteriormente son adicionados los *bigramas*. El arreglo que se genera con los *tokens* y *bigramas*, sirve de referencia para colocarlo como encabezado de la matríz TF.

```
FRASE ORIGINAL:
cuantas quejas de acoso sexual y hostigamiento hay en el estado de chihuahua

FRASE SOLO TOKENS:
cuantas quejas acoso sexual hostigamiento estado chihuahua

SE CREA ARREGLO CON BIGRAMAS:
array ( 0 => 'cuanto-queja', 1 => 'queja-acosar', 2 => 'acosar-sexo', 3 => 'sexo-hostiga',
        4 => 'hostiga-estado', 5 => 'estado-chihuahua', )

SE AGREGA LA FRASE ORIGINAL CON BIGRAMAS:
array ( 0 => '', 1 => 'cuanto', 2 => 'queja', 3 => 'acosar', 4 => 'sexo', 5 => 'hostiga',
6 => 'estado', 7 => 'chihuahua', 8 =>'cuanto-queja', 9 => 'queja-acosar',
10 => 'acosar-sexo', 11 => 'sexo-hostiga', 12 => 'hostiga-estado', 13 => 'estado-chihuahua')
```

Figura 5.15. Creación de arreglo con tokens y bigramas.

La figura 5.16 está representada por una muestra de la matríz TF, así como la misma matríz normalizada.

```
SE CREA MATRÍZ TF: (Como encabezado el número de términos más bigramas)
===TF=========================
000000000 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
000012016 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
000052016 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
000062016 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
000072016 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 |
...

SE NORMALIZA LA MATRÍZ TF

===TF (Normalizada)=========================
000000000 | 0.031023602427885 | 0.11180339887499 | 0.35355339059327 | 0.039283710065919 | 0.074329414624717 |
0.028747978728803 | 0.028536507276767 | 0.27735009811261 | 1 | 0.44721359549996 | 0.70710678118655 | 1 |
0.04688072309385 |
000012016 | 0 | 0 | 0 | 0 | 0 | 0 | 0.028536507276767 | 0 | 0 | 0 | 0 | 0 | 0 |
000052016 | 0.031023602427885 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
000062016 | 0.031023602427885 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
000072016 | 0 | 0 | 0 | 0 | 0 | 0.028747978728803 | 0.028536507276767 | 0 | 0 | 0 | 0 | 0 | 0.04688072309385 |
...
```

Figura 5.16. Matríz TF y matríz TF normalizada.

Operaciones a modo de pruebas para la reducción de dimensionalidad con el algoritmo SVD

En la figura 5.17 se despliega una pequeña muestra del resultado de la matríz S rango 2 y la multiplicación de la matríz $U\times$ la diagonal de la matríz S rango 2. La explicación en detalle para obtener el resultado de las matrices S rango 2, U y V, se muestra en la sección 4.2.3 de este documento.

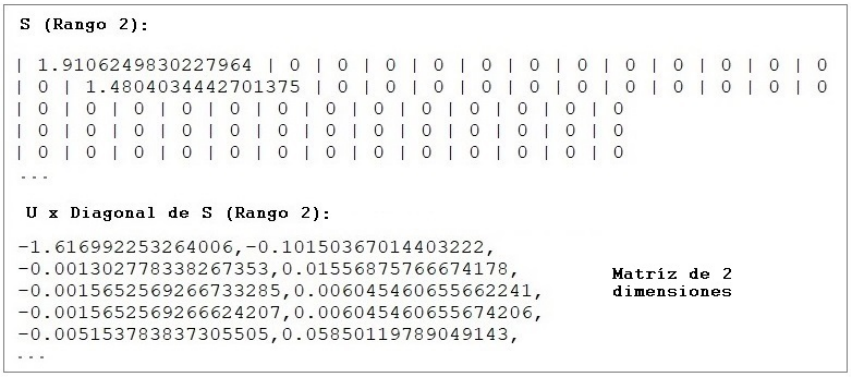


Figura 5.17. Resultados de la matrices S rango 2 y $U\times$ la diagonal de S rango 2.

Operaciones a modo de prueba para entrenar la SVM

En la figura 5.18 se muestra un arreglo con los vectores para hacer el entrenamiento de la MSV, donde se destaca de qué clase es cada vector, es decir *clase* -1 o *clase* $+1$.

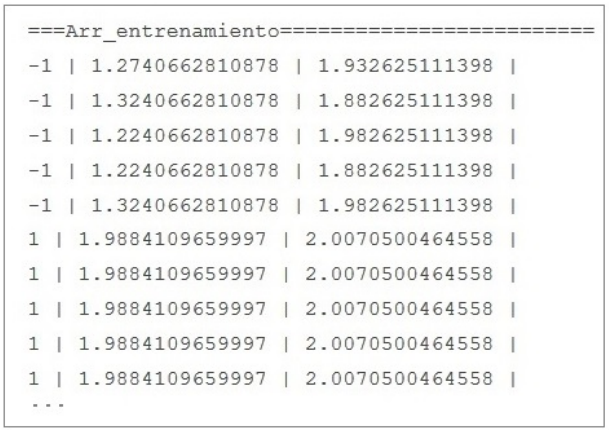


Figura 5.18. Matríz de entrenamiento.

Posteriormente en la figura 5.19 se muestran algunos de los registros de una matríz que sirve

para localizar los vectores de soporte.

```
===Arr_VS=========================
-1 | 1.3240662810878 | 1.982625111398 | 1 | 1.9884109659997 | 2.0070500464558 | 0.66479353022067 |
-1 | 1.3240662810878 | 1.982625111398 | 1 | 1.9884109659997 | 2.0070500464558 | 0.66479353022067 |
-1 | 1.3240662810878 | 1.982625111398 | 1 | 1.9884109659997 | 2.0070500464558 | 0.66479353022067 |
-1 | 1.3240662810878 | 1.982625111398 | 1 | 1.9884109659997 | 2.0070500464558 | 0.66479353022067 |
-1 | 1.3240662810878 | 1.982625111398 | 1 | 1.9884109659997 | 2.0070500464558 | 0.66479353022067 |
-1 | 1.3240662810878 | 1.982625111398 | 1 | 1.9884109659997 | 2.0070500464558 | 0.66479353022067 |
-1 | 1.3240662810878 | 1.982625111398 | 1 | 1.9884109659997 | 2.0070500464558 | 0.66479353022067 |
...
```

Figura 5.19. Matríz que sirve para localizar los vectores de soporte.

Una vez que se seleccionan los vectores de soporte, se crea una matríz que contiene 3 vectores. Un vector pertenece a la clase -1, mientras que los otros 2, pertenecen a la clase $+1$, al final se le agrega el *bias* (el número 1) a cada vector. Los vectores se visualizan de arriba hacia abajo y de izquierda a derecha. Ver figura 5.20.

```
===Arr_VSmatppal=========================
1.3240662810878 | 1.2740662810878 | 1.9971319647351 |
1.882625111398 | 1.932625111398 | 2.0216142183224 |
1 | 1 | 1 |
```

Figura 5.20. Matríz de vectores de soporte.

Para encontrar los valores de α_1, α_2, α_3, $\mathbf{w_1}$, $\mathbf{w_1}$ y b, se debe procesar la matríz de vectores de soporte con el método de *determinantes*. Ver figura 5.21.

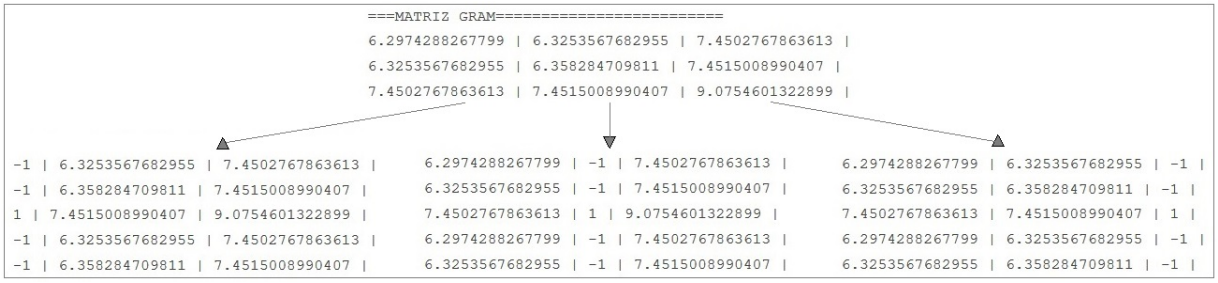


Figura 5.21. Cálculo por el método de determinantes.

El resultado de los valores de α_1, α_2, α_3, $\mathbf{w}$ y b se muestran en la figura 5.22.

```
alfa 1 = -319.84695108293
alfa 2 = 269.74754669842
alfa 3 = 41.201682631358
w1 = 2.4628880009193
w2 = 2.4628880009186
b = -8.8977217531535
```

Figura 5.22. Resultados de α_1, α_2, α_3, $\mathbf{w_1}$, $\mathbf{w_2}$ y b.

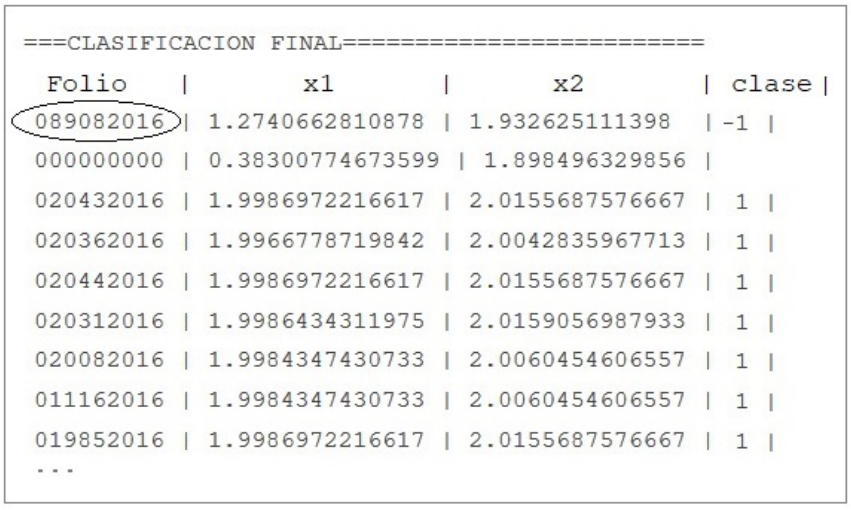
En la figura 5.23 se muestra una matríz con los vectores de entrenamiento y cuya clasificación ha sido determinada por la fórmula $\mathbf{w} \cdot \mathbf{x} + b$. Los nuevos vectores se someten a la operación de la fórmula anterior para determinar a qué clase pertenecen.

```
===ENTRENAMIENTO=========================
Clase N |     x1     |      x2       | Producto punto XW |      wx+b      |clasif|
-1 | 1.2740662810878 | 1.932625111398 | 7.897721753203 | -0.99999999995046 | NEG |
-1 | 1.3240662810878 | 1.882625111398 | 7.8977217532031 | -0.99999999995043 | NEG |
-1 | 1.2240662810878 | 1.982625111398 | 7.897721753203 | -0.9999999999505 | NEG |
-1 | 1.2240662810878 | 1.882625111398 | 7.6514329531111 | -1.2462888000424 | NEG |
-1 | 1.3240662810878 | 1.982625111398 | 8.1440105532949 | -0.75371119985857 | NEG |
1 | 1.9884109659997 | 2.0070500464558 | 9.8403729857159 | 0.94265123256247 | POS |
1 | 1.9884109659997 | 2.0070500464558 | 9.8403729857159 | 0.94265123256247 | POS |
1 | 1.9884109659997 | 2.0070500464558 | 9.8403729857159 | 0.94265123256247 | POS |
1 | 1.9884109659997 | 2.0070500464558 | 9.8403729857159 | 0.94265123256247 | POS |
1 | 1.9884109659997 | 2.0070500464558 | 9.8403729857159 | 0.94265123256247 | POS |
1 | 1.9884109659997 | 2.0070500464558 | 9.8403729857159 | 0.94265123256247 | POS |
1 | 1.9971319647351 | 2.0216142183224 | 9.897721752991 | 0.99999999983754 | POS |
1 | 1.9884109659997 | 2.0070500464558 | 9.8403729857159 | 0.94265123256247 | POS |
1 | 1.9884109659997 | 2.0070500464558 | 9.8403729857159 | 0.94265123256247 | POS |
1 | 1.9884109659997 | 2.0070500464558 | 9.8403729857159 | 0.94265123256247 | POS |
```

Figura 5.23. Clasificación de los vectores de entrenamiento con la fórmula: $\mathbf{w} \cdot \mathbf{x} + b$.

Operaciones a modo de prueba para realizar la clasificación de textos

En la figura 5.24 se despliegan los resultados de la clasificación que tuvieron los folios asociados a su vector correspondiente. La pregunta del folio 089082016, se clasificó como clase -1 en la SVM, debido a que el contexto de la pregunta es el más similar a la frase inicialmente dada.



```
===CLASIFICACION FINAL=========================
  Folio    |        x1        |        x2        | clase|
 089082016 | 1.2740662810878 | 1.932625111398  |-1 |
 000000000 | 0.38300774673599 | 1.898496329856 |
 020432016 | 1.9986972216617 | 2.0155687576667 | 1 |
 020362016 | 1.9966778719842 | 2.0042835967713 | 1 |
 020442016 | 1.9986972216617 | 2.0155687576667 | 1 |
 020312016 | 1.9986434311975 | 2.0159056987933 | 1 |
 020082016 | 1.9984347430733 | 2.0060454606557 | 1 |
 011162016 | 1.9984347430733 | 2.0060454606557 | 1 |
 019852016 | 1.9986972216617 | 2.0155687576667 | 1 |
 ...
```

Figura 5.24. Clasificación de folios en la SVM.

Se hace un recorrido de la matríz que contiene los resultados de la clasificación de clases y se van seleccionando los folios que pertenezcan a la *clase* -1, en el ejemplo de prueba, se encontró el folio "089082016", el cual, es considerado para ser integrado en una sentencia SQL[3] con el objetivo de obtener la pregunta y la respuesta asociada a dicho folio.

Sentencia SQL:

```
select S.cfolio, A.c_agrupa, U.c_so, S.cpregunta,
       S.crespuesta, S.cruta_arch
 from solicitudesinformacion as S
  left join so_agrupa A on S.nagrupador=A.nagrupador
  left join so_descrip as U on U.nagrupador=S.nagrupador and U.nunidad=S.nunidad
 where right(S.cfolio,4)="2016"
       and S.cfolio in ("089082016")
 order by A.c_agrupa, U.c_so, S.cfolio
```

[3]SQL (Structured Query Language), es un lenguaje de programación estandar e interactivo para la obtención y actualización de información desde una Base de Datos.

El resultado de la sentencia SQL se muestra en la tabla 5.16.

Tabla 5.16. Resultado de la sentencia SQL.

Campo	Dato
cfolio	089082016
c_agrupa	Organismos Autónomos
c_so	Comisión Estatal de los Derechos Humanos
c_pregunta	cuantas quejas se han presentado ante ese organismo en relacion al acoso sexual u hostigamiento sexual en nuestra entidad federativa en los ultimos cinco años y lo que va a la fecha.
c_respuesta	Respuesta anexa como documento adjunto.
cruta_arch	$\cdots$\74d06f02\3fd62ecd\e4b79726\c7c5f98a\respuesta.pdf

5.3.2 Pruebas de componentes

Una vez que se realizan las pruebas por unidad es necesario unir todos los componentes para integrarlos en un solo elemento, así como realizar las siguientes configuraciones de la conexión a la Base de Datos, llamadas a funciones y programas externos.

Integración de las unidades en el programa principal del sistema:

```
<html>
 ...
 <select name="nm_year">...</select>      // Unidad 1 (Año)
 <select name="nm_agrupa">...</select>    // Unidad 2 (Agrupador)
 <select name="nm_so">...</select>        // Unidad 3 (SO)
 <input type="checkbox" name="checkbox">  // Unidad 4 (Todos los SO)
 <textarea name="nm_frase">...</textarea> // Unidad 5 (Frase)
 <input type="button"/> ...               // Unidad 6 (Botones)
 <div><? require("proceso1.php");?></div> // Unidad 7 (TF)
 <div><? require("proceso2.php");?></div> // Unidad 8 (SVD)
 <div><? require("proceso3.php");?></div> // Unidad 9 (SVM)
 <div id="id_resultados">...</div>        // Unidad 10 (Resultados)
 ...
</html>
```

Posteriormente se llevaron a cabo las pruebas de componentes, arrojando el mismo resultado que en las pruebas por unidad, con la diferencia de que los resultados de las operaciones se desplegaron en una sola pantalla. No es necesario mostrar un ejemplo por obvias razones.

5.3.3 Pruebas del sistema

Ya integradas las unidades en un solo componente, se realizan las adecuaciones finales para que el sistema permita aceptar y enviar los datos desde la interfaz de usuario. La figura 5.25 muestra como se realizó la prueba directamente en la interfaz de usuario; dando el mismo resultado final tal cómo se realizó en las pruebas de unidad.

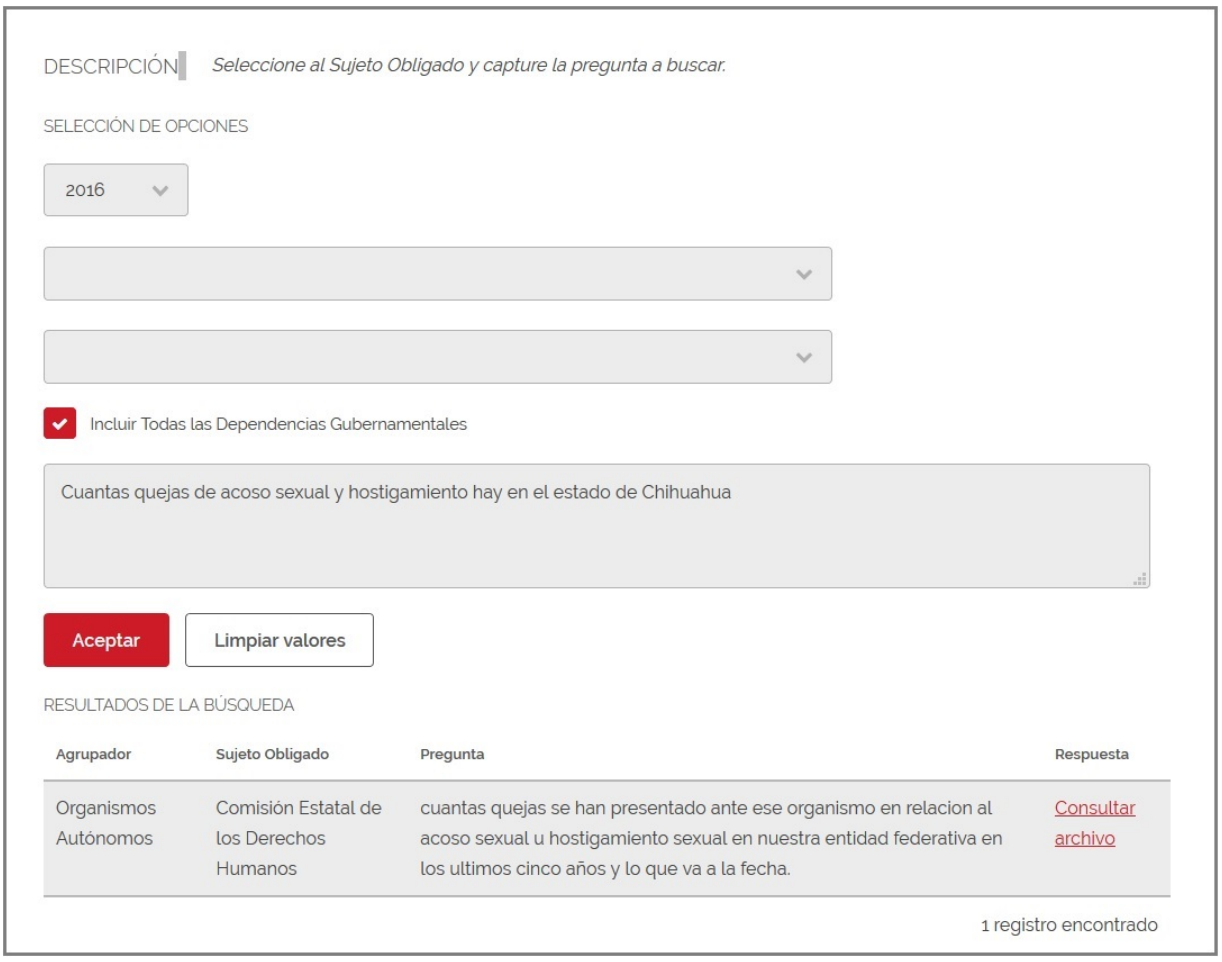


Figura 5.25. Pruebas del sistema en la interfaz de usuario.

Capítulo 6

Pruebas y resultados

6.1 Pruebas con el sistema a implementar

Para realizar las pruebas pertinentes se consideró el Corpus del año 2016 de las Solicitudes de Información del sistema Infomex, que suman un total de $8,592$ preguntas (textos) y $10,540$ *términos*, dentro de los cuales se determinaron $1,904$ *lemas* donde se seleccionaron 15 preguntas al azar. Por cada pregunta seleccionada, se realizó el entrenamiento de una Máquina de Soporte a Vectores en tiempo real (sin la supervisión humana), y en cuyo resultado arrojó un promedio de 88% de efectividad en el entrenamiento y clasificación; considerando un mínimo de 4 vectores para la clase -1 y un máximo de 20 vectores para la clase $+1$.

En la tabla 6.1 se observan los resultados de un muestreo no probabilístico por juicio donde, se incluye el tiempo de entrenamiento de la SVM, los textos clasificados y los textos que tienen similitud con el planteado inicialmente, así como el porcentaje de efectividad en la clasificación.

Tabla 6.1. Muestreo de entrenamiento y clasificación de textos.

No. Pregunta (Texto)	Tiempo de entrenamiento y clsificación (seg.)	Textos clasificados	Textos similares	% Efectividad
1	2.51	1	1	100
2	11.43	1	1	100
3	2.07	3	2	66.67
4	2.11	9	6	66.67
5	5.09	1	1	100
6	3.14	7	6	85.71
7	1.63	2	2	100
8	12.47	3	3	100
9	1.24	5	5	100
10	8.49	4	3	75
11	5.84	20	18	90
12	2.01	2	2	100
13	4.36	3	2	66.67
14	2.06	1	1	100
15	10.06	12	9	75

En la figura 6.1 se visualiza la comparación entre los textos clasificados contra los textos encontrados con mayor similitud, mientras que la figura 6.2 muestra el tiempo de entrenamiento y clasificación contra el porcentaje de efectividad.

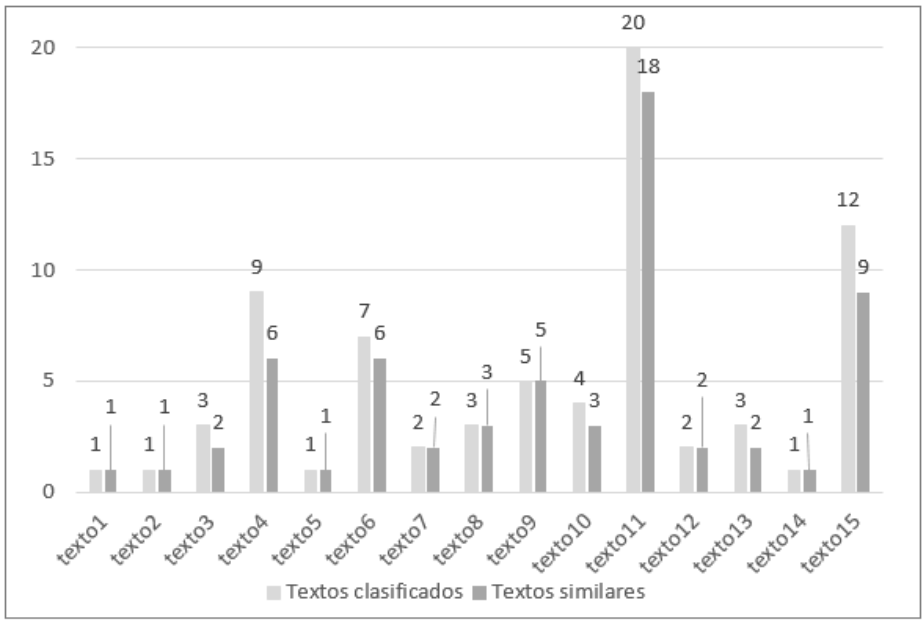


Figura 6.1. Textos clasificados contra textos similares.

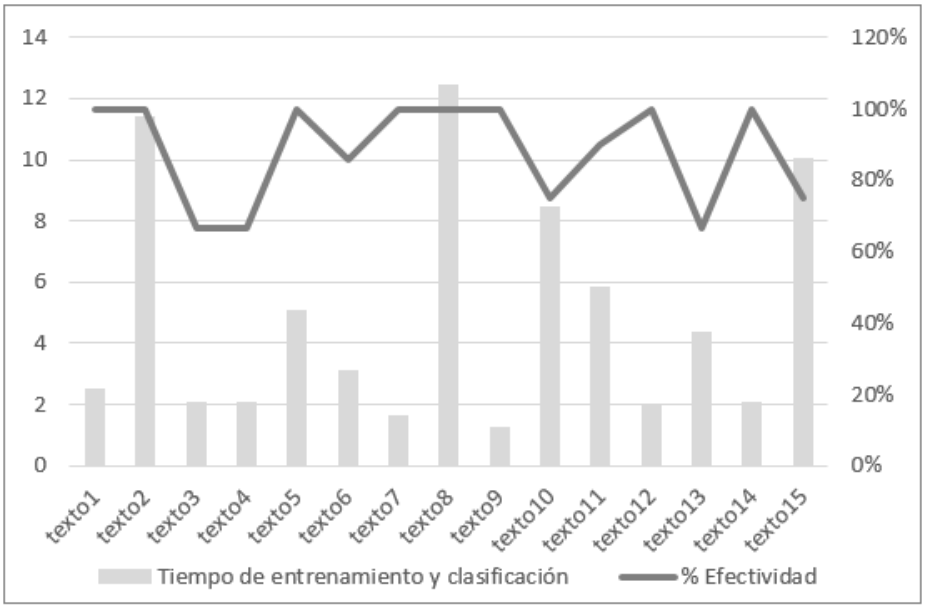


Figura 6.2. Tiempo de entrenamiento y clasificación contra porcentaje de efectividad.

6.2 Realización de pruebas con el sistema a implementar vs. el sistema Infomex

A continuación se muestran varias pruebas que se realizaron sobre ambos sistemas.

Se hace el supuesto de que se quiere saber si existe algún folio que tenga cierta similitud con alguna pregunta escrita en lenguaje natural a alguna Dependencia Gubernalemental. A continuación se muestran las frases que se desean buscar en ambos sistemas:

1. “Número de homicidios en Ciudad Juárez”.
2. “Convenios que ha celebrado Pensiones Civiles”.
3. “Número de becas otorgadas por la UACH”.

En la tabla 6.2 se muestran los resultados para localizar la frase planteada en el sistema de búsqueda de Información Pública con Técnicas de Inteligencia Artificial.

Tabla 6.2. Resultados de la búsqueda del año 2016, con el sistema a implementar.

Frase	**No. registros encontrados**	**No. registros con similitud**	**No. registros sin similitud**
“Número de homicidios en Ciudad Juárez”	11	11	0
“Convenios que ha celebrado Pensiones Civiles del Estado”	34	4	30
“Número de becas otorgadas por la UACH”	2	1	1

En la tabla 6.3 se muestran los resultados para localizar la frase planteada.

Tabla 6.3. Resultados de la búsqueda del año 2016, con el sistema Infomex.

Frase	**No. registros encontrados**	**No. registros con similitud**	**No. registros sin similitud**
“Número de homicidios en Ciudad Juárez”	538	indefinido	indefinido
“Convenios que ha celebrado Pensiones Civiles del Estado”	69	indefinido	indefinido
“Número de becas otorgadas por la UACH”	144	indefinido	indefinido

Como se puede observar en el resultado de ambas pruebas, son menos los resultados que arroja el sistema *búsqueda de Información Pública con técnicas de Inteligencia Artificial*, que si bien es cierto, no todos los registros tienen alguna similitud con la frase que se busca, pero es más fácil localizar la más parecida en el listado que emite el sistema.

Por otro lado, los resultados otorgados por el sistema *Infomex* no tienen certeza, ya que se tiene que ingresar a cada folio para saber que fué lo que se preguntó, además el usuario debe especificar por medio de un catálogo el tipo de respuesta antes enviar la petición para conocer los resultados.

En la figura 6.3 se muestra la pantalla donde se realizan las búsquedas de folios de alguna Dependencia Gubernamental en particular.

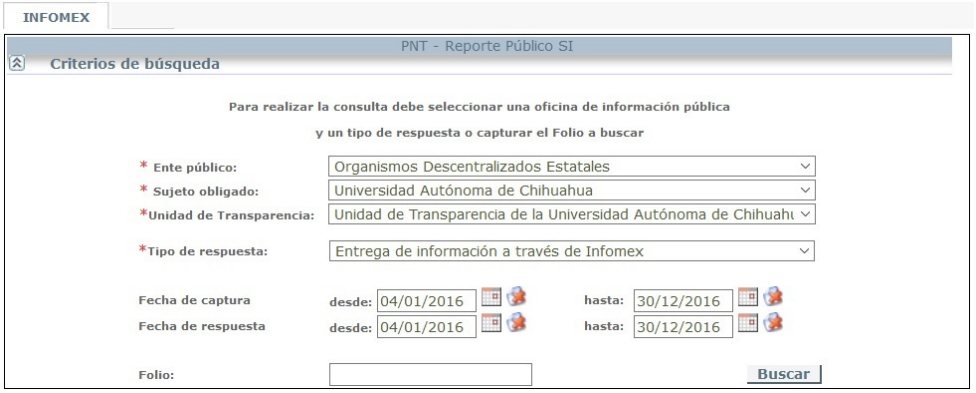


Figura 6.3. Pantalla para consulta de folios desde el sistema Infomex.

Una vez enviada la petición se despliega una ventana con los registros encontrados. Ver figura 6.4.

Folio de la solicitud	Fecha de Captura	Unidad de Información	Respuesta	Fecha de Respuesta	Recurso de revisión (en caso de tener)
000842016	11/01/2016	Unidad de Transparencia de la Universidad Autónoma de Chihuahua	Entrega de información a través de Infomex	25/01/2016	
001892016	20/01/2016	Unidad de Transparencia de la Universidad Autónoma de Chihuahua	Entrega de información a través de Infomex	11/02/2016	
003052016	26/01/2016	Unidad de Transparencia de la Universidad Autónoma de Chihuahua	Entrega de información a través de Infomex	10/02/2016	
003062016	26/01/2016	Unidad de Transparencia de la Universidad Autónoma de Chihuahua	Entrega de información a través de Infomex	10/02/2016	
003072016	26/01/2016	Unidad de Transparencia de la Universidad Autónoma de Chihuahua	Entrega de información a través de Infomex	10/02/2016	
003082016	26/01/2016	Unidad de Transparencia de la Universidad Autónoma de Chihuahua	Entrega de información a través de Infomex	10/02/2016	
003102016	26/01/2016	Unidad de Transparencia de la Universidad Autónoma de Chihuahua	Entrega de información a través de Infomex	11/02/2016	
003432016	28/01/2016	Unidad de Transparencia de la Universidad Autónoma de Chihuahua	Entrega de información a través de Infomex	19/02/2016	

Figura 6.4. Pantalla de despliegue de resultados del sistema Infomex.

Capítulo 7

Conclusiones

En el caso de los textos de Información Pública, específicamente las Solicitudes de Información que se almacenan desde el sistema Infomex, los registros se van incrementando cada año, y por lo tanto, muy interesante pensar en la manera de integrar la clasificación de los textos desde cualquier tipo y no solo por una pregunta en particular, es decir, realizar clasificaciones para cualquier contexto o mejor aún, realizar clasificaciones multiclase involucrando todos los algoritmos de Procesamiento de Lenguaje Natural y Aprendizaje Automático descritos en este documento.

Un objetivo de investigación importante con respecto a las Máquinas de Soporte a Vectores es mejorar la velocidad en el entrenamiento y las pruebas para que éstas se conviertan en una opción más viable para conjuntos de datos más grandes.

No hay reglas de oro para determinar qué dimensión admisible se deba utilizar en una SVM para dar lugar a un resultado más preciso. En la práctica, la dimensión elegida generalmente no hace una gran diferencia en la precisión resultante.

En comparación con otros clasificadores como las Redes Neuronales, se puede observar que los hiperplanos de decisión resultantes encontrados para las SVM no lineales son del mismo tipo.

Por ejemplo, una SVM con una Función de Base Radial Gaussiana (RBF) proporciona el mismo plano de decisión que un tipo de Red Neuronal conocida como Red de Funciones de Base Radial. Una SVM con un Kernel Sigmoide es equivalente a una Red Neuronal Simple de dos capas conocida como Perceptrón Multicapa (sin capas ocultas).

La formación de la SVM siempre encuentra una solución global, a diferencia de las Redes Neuronales con Retropropagación donde a medida que crece la cantidad de ejemplos aumenta la complejidad de la topología y el espacio de búsqueda se vuelve cada vez más complejo provocando que la función de error contenga cada vez más mínimos locales.

Anexo A

Diagrama de flujo para realizar una solicitudes de información

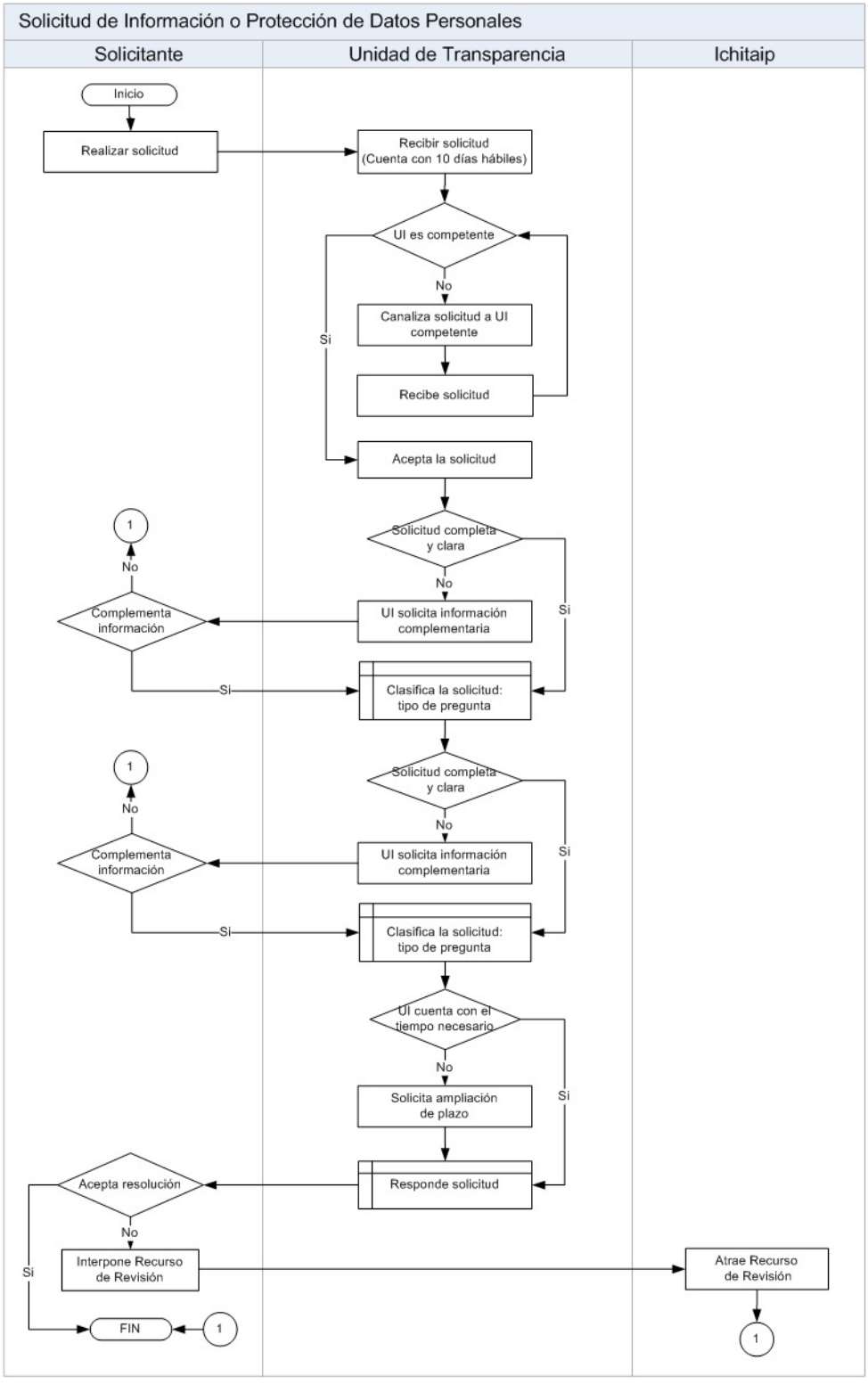


Figura A.1. Diagrama de flujo para realizar solicitudes de información en el sistema Infomex.

Anexo B

Diagrama de estados: Proceso general del sistema

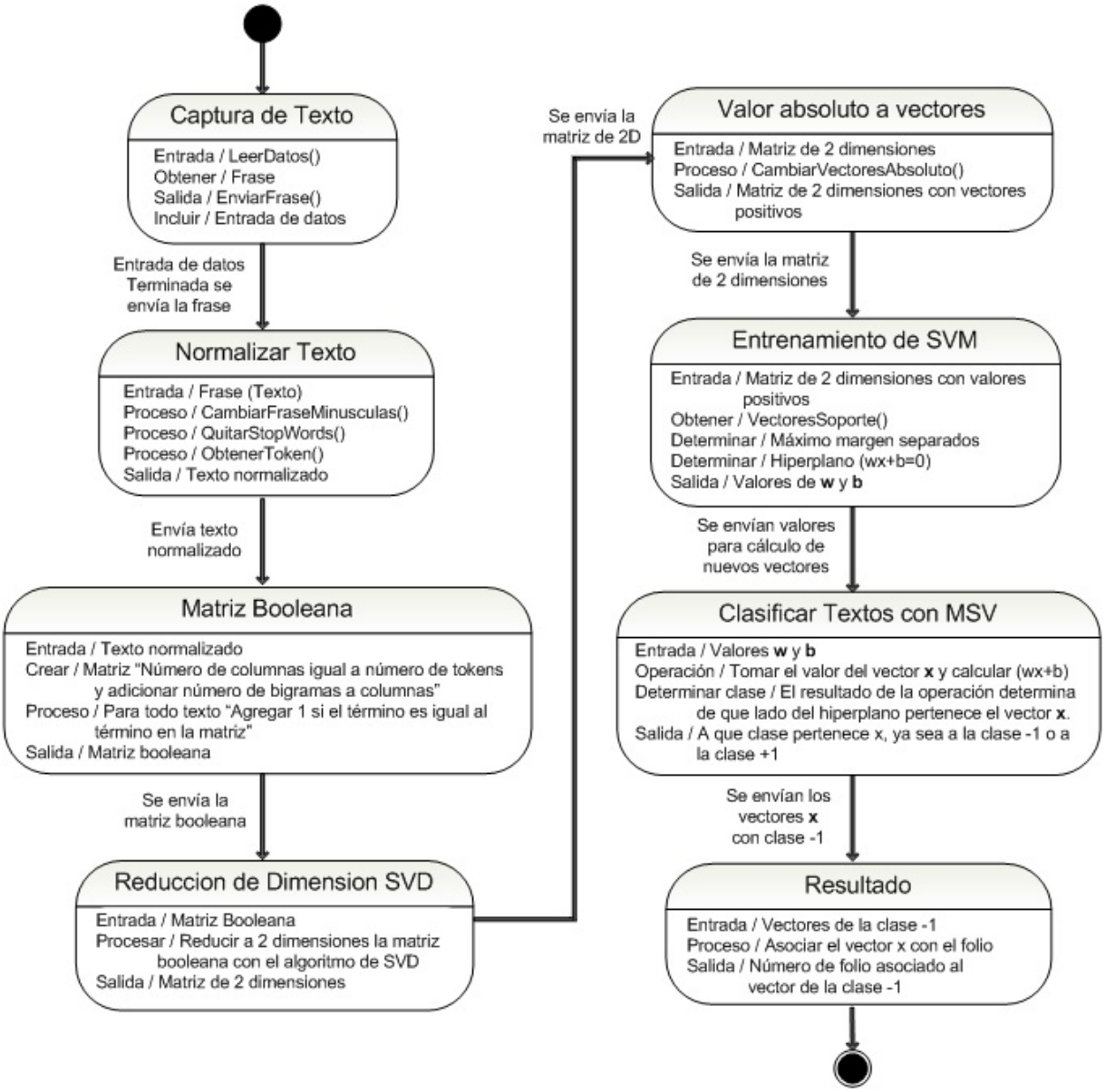


Figura B.1. Diagrama de estados: Proceso general del sistema.

Anexo C

Diagrama de actividades: Captura de texto

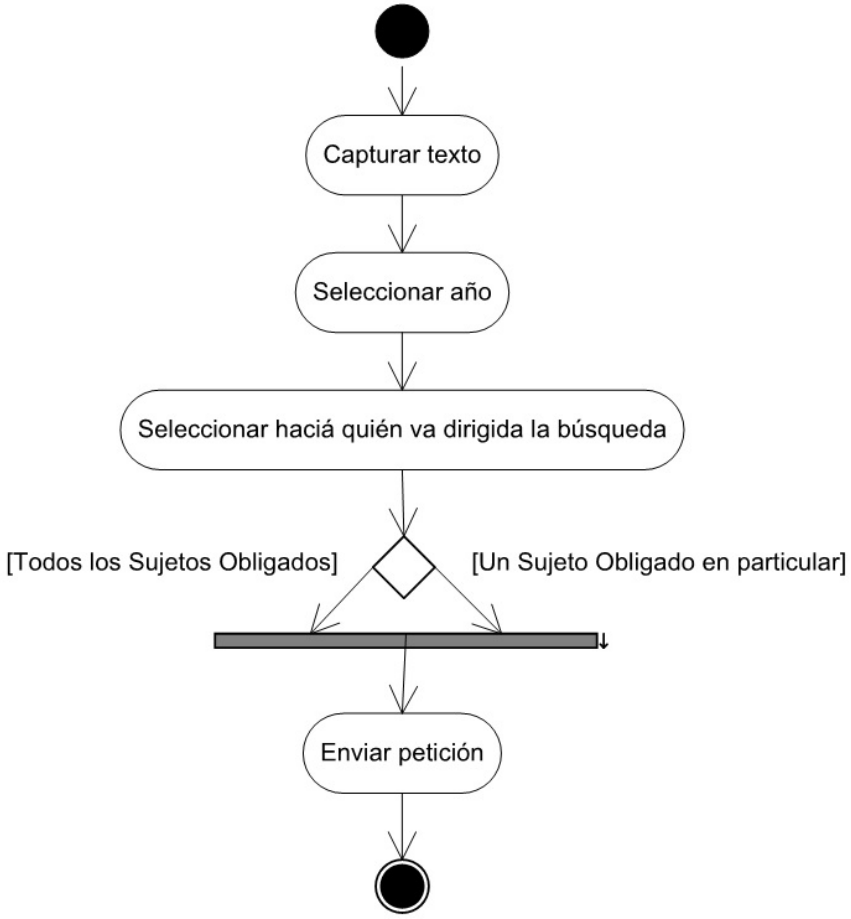


Figura C.1. Diagrama de actividades: Captura de texto.

Anexo D

Diagrama de actividades: Normalizar texto

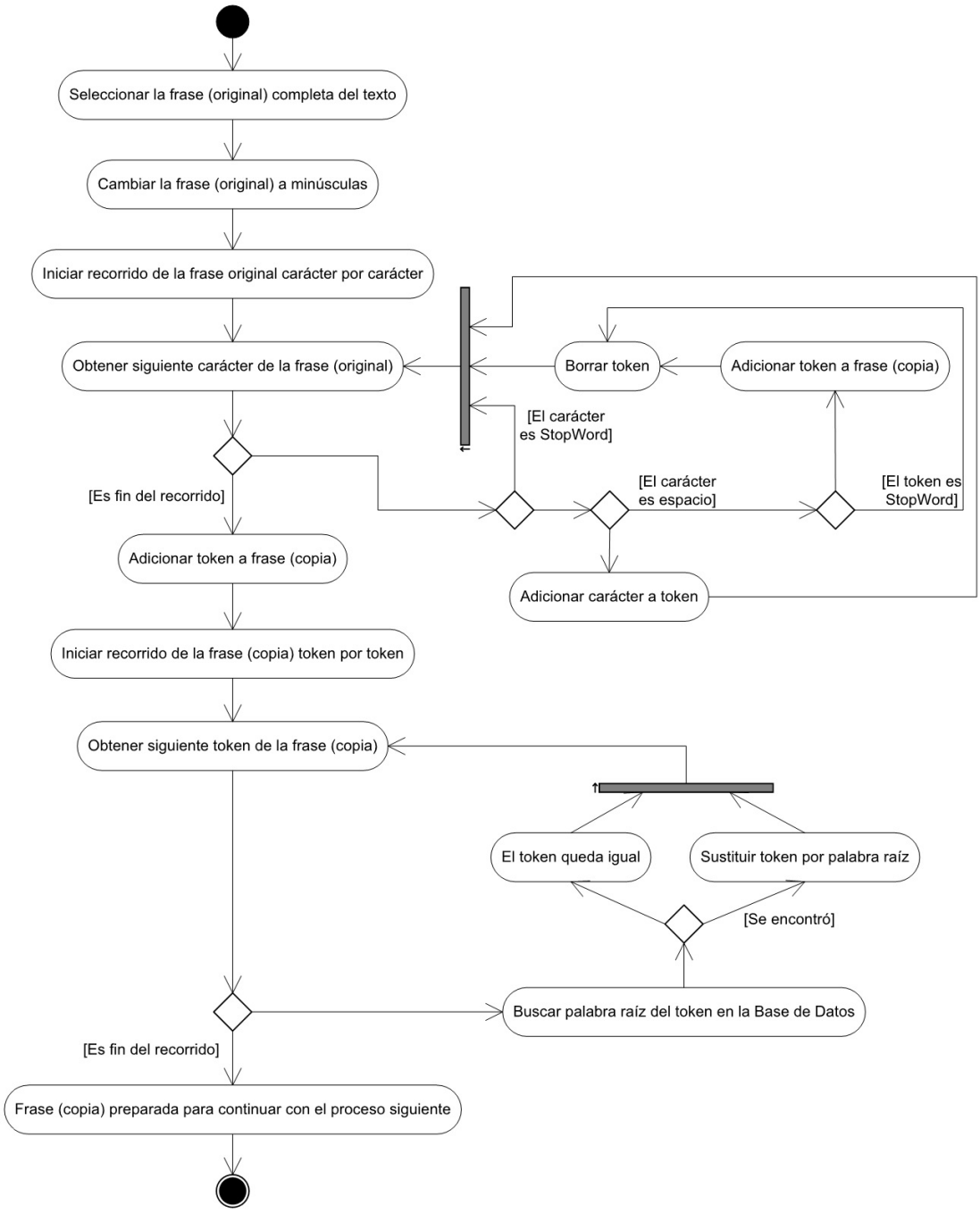


Figura D.1. Diagrama de actividades: Normalizar texto.

Anexo E

Diagrama de actividades: Creación de matríz booleana

Figura E.1. Diagrama de actividades: Creación de matríz booleana.

Anexo F

Diagrama de actividades: Reducción de dimensionalidad con SVD

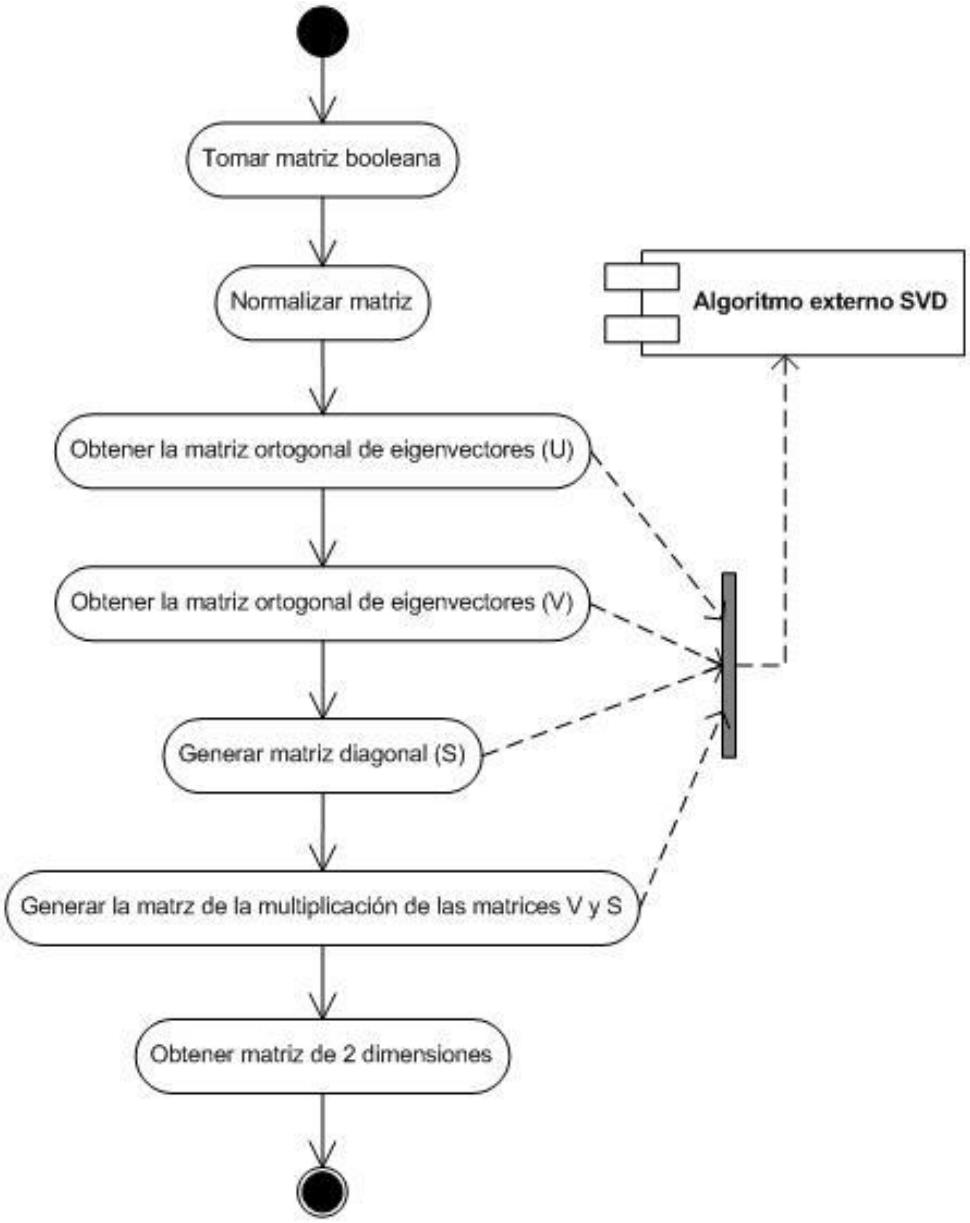


Figura F.1. Diagrama de actividades: Reducción de dimensionalidad con SVD.

Anexo G

Diagrama de actividades: Cambiar a valor absoluto los vectores

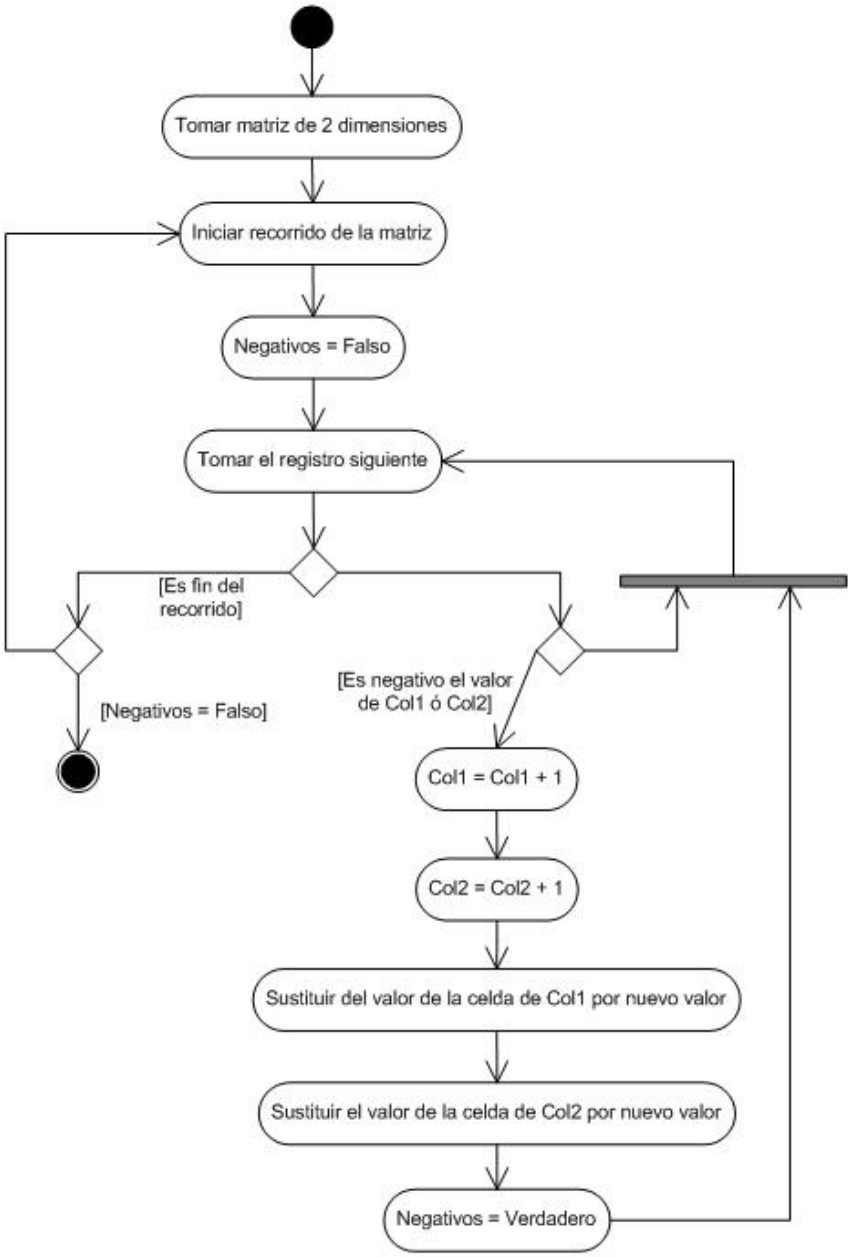


Figura G.1. Diagrama de actividades: Cambiar a valor absoluto los vectores.

Anexo H

Diagrama de actividades: Entrenamiento de SVM

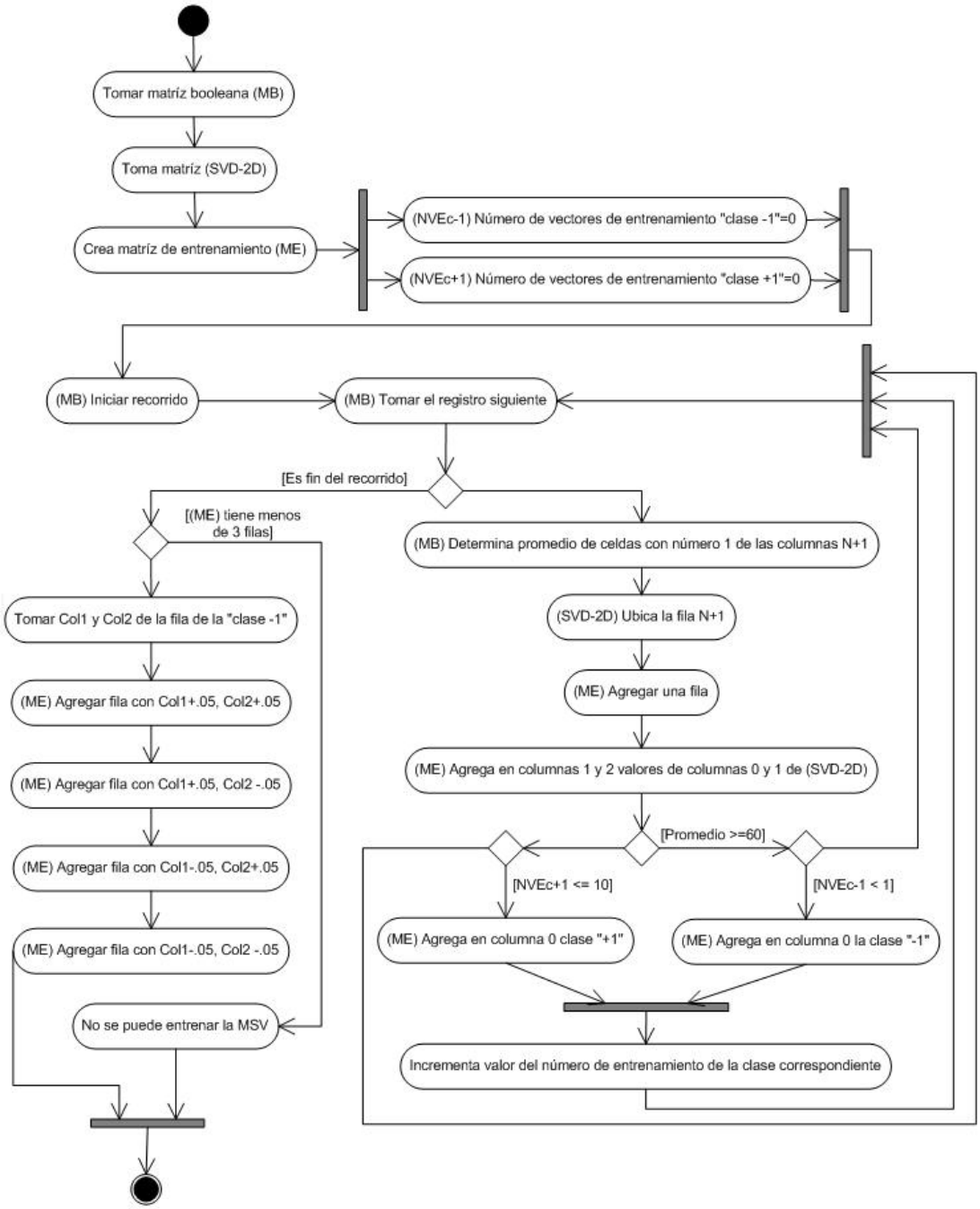


Figura H.1. Diagrama de actividades: Entrenamiento de SVM.

Anexo I

Diagrama de actividades: Determinar vectores de soporte

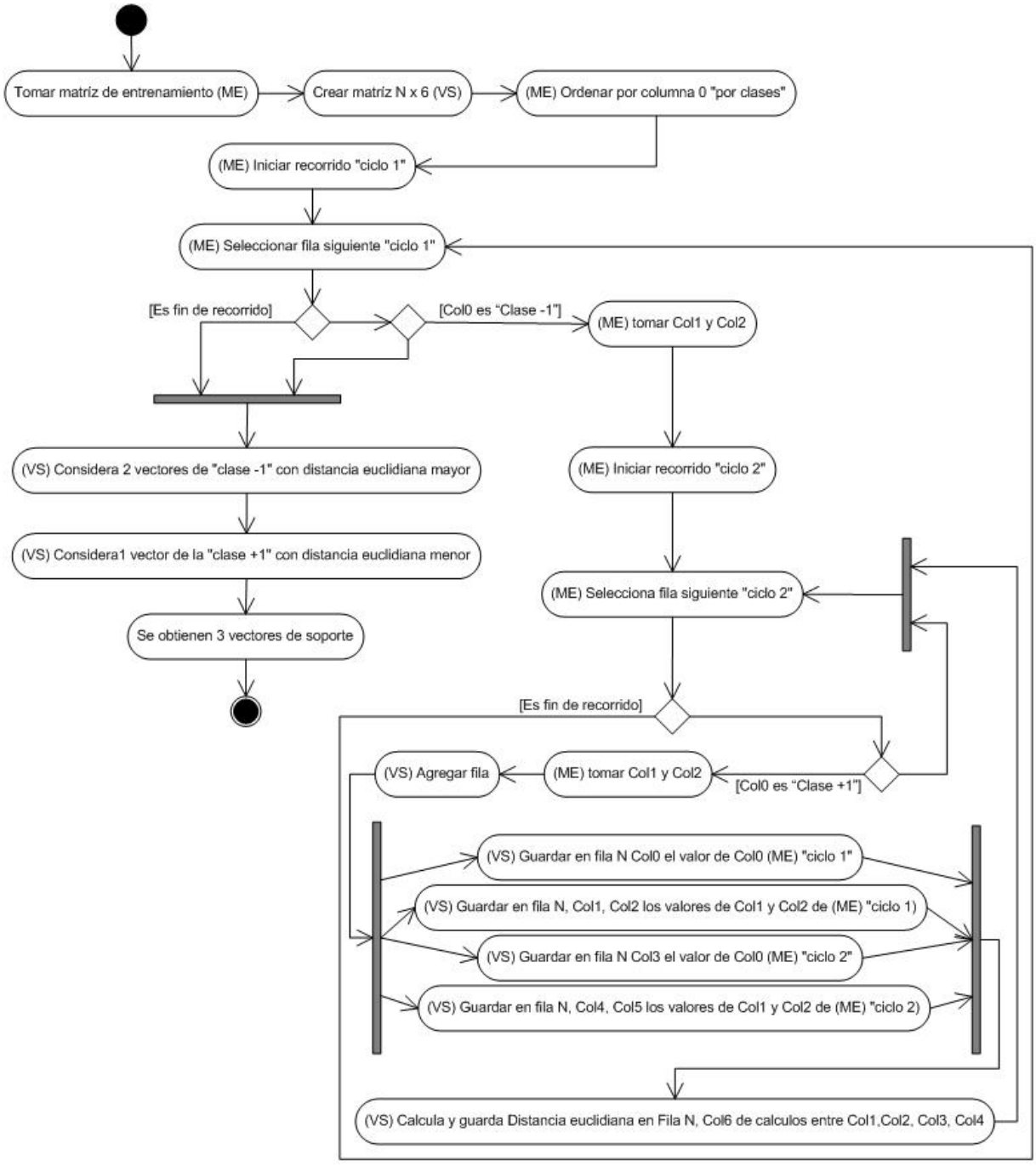


Figura I.1. Diagrama de actividades: Determinar vectores de soporte.

Anexo J

Diagrama de actividades: Clasificar textos

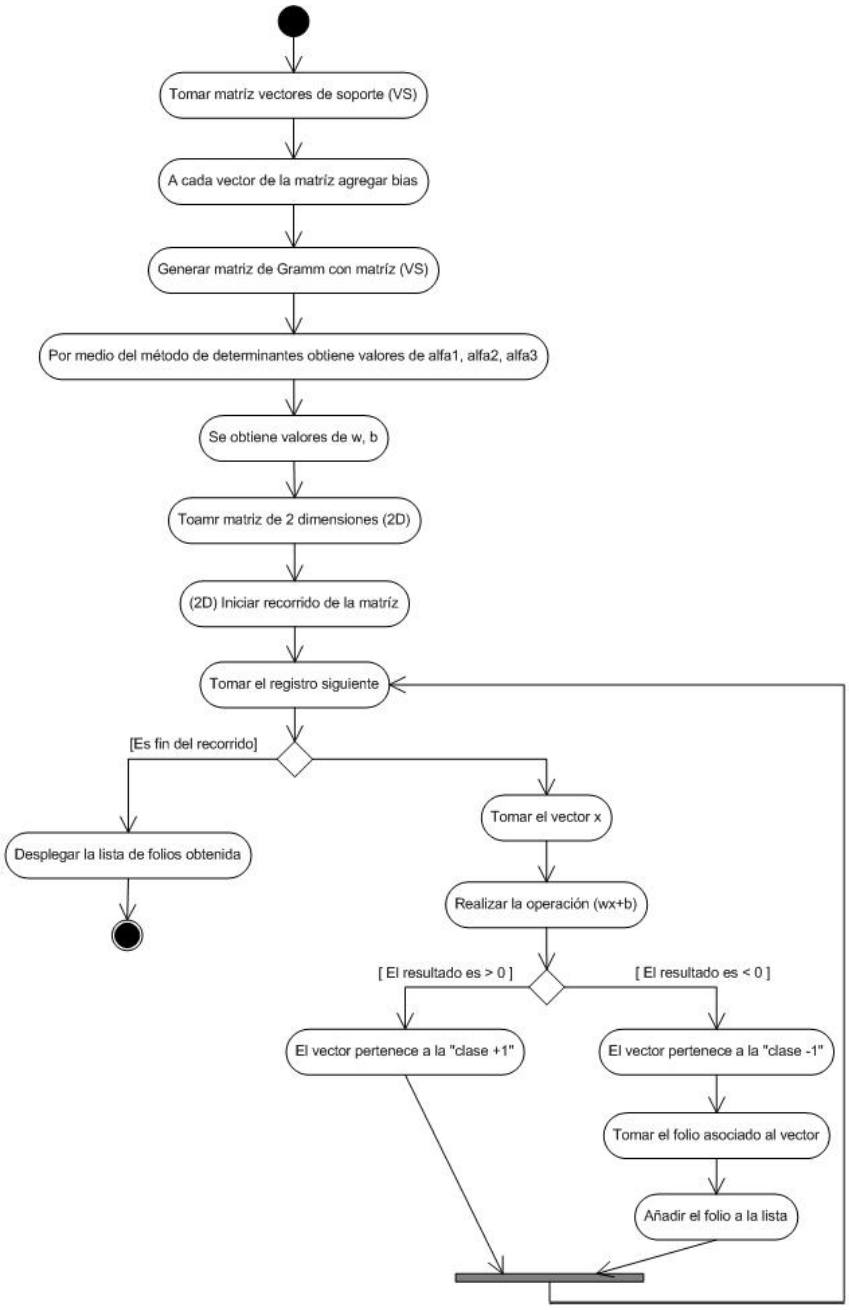


Figura J.1. Diagrama de actividades: Clasificar textos.

Referencias

[1] H. C. del Estado de Chihuahua, "Ley de transparencia del estado de chihuahua." http://www.congresochihuahua2.gob.mx/biblioteca/leyes/archivosLeyes/1175.pdf, Enero 2018. Acceso en 2018-01-02.

[2] H. C. del Estado de Chihuahua, "Constitución política del estado de chihuahua." http://www.congresochihuahua2.gob.mx/biblioteca/constitucion/archivosConstitucion/actual.pdf, Junio 1950. Acceso en 2018-01-02.

[3] T. Markiewicz and J. Zheng, *Getting Started with Artificial Intelligence, a Practical Guide to Building Enterprise Applications*. O'Reilly Media, Inc., 2018.

[4] R. E. L. Briega, "Iaar comunidad argentina de inteligencia artificial." https://iaarbook.github.io/, 2018. Acceso en 2018-01-02.

[5] P. R. C. D. Manning and H. Schütze, "Introduction to information retrieval." www.safaribooksonline.com, Julio 2008. Acceso en 2018-01-02.

[6] R. Korfhage, *Information storage and retrieval*. John Wiley, 2007.

[7] A. M. Vázquez, S. and G. Rigau., "Método de desambiguación léxica basada en el recurso léxico: dominios relevantes.." http://www.sepln.org/revistaSEPLN/revista/31/31-Pag141.pdf, 2003. Acceso en 2018-01-02.

[8] J. H. M. Daniel Jurafsky, *Speech and Language Processing*. Prentice Hall, 2008.

[9] Wikipedia, "Tf-idf.." https://es.wikipedia.org/wiki/tf-idf, 2018. Acceso en 2018-01-02.

[10] D. T. L. Zdravko Markove, *Data Mining the web, uncovering patterns in web content, structure, and usage*. Wiley-Interscience, 2007.

[11] J. P. Jiawei Han, Micheline Kamber, *Data Mining Concepts and Techniques*. Morgan Kaufmann Publisher, 2012.

[12] E. Ientilucci, "Using the singular value decomposition." http://www.cis.rit.edu/~ejipci/Reports/svd.pdf, 2003. Acceso en 2018-01-02.

[13] D. J. Alexandre Kowalczyk, *Support Vector Machines Succinctly*. Syncfusion, Inc., 2017.

[14] M. A. H. Ian H. Witten, Eibe Frank, *Data Mining Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[15] I. Sommerville, *Ingeniería de Software*. Addison Wesley, (Pearson), 2011.

[16] S. Loisel, "Departament of mathematics, heriot-watt university, uk.." http://www.numericjs.com/index.php/, 2018. Acceso en 2018-01-02.