PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

ANÁLISIS DE SENTIDO EN TEXTOS

JORGE ANDRÉS GÁLVEZ GAJARDO

Profesor Guía: **Rodrigo Alfaro Arancibia**

INFORME FINAL DEL PROYECTO

PARA OPTAR AL TÍTULO PROFESIONAL DE

INGENIERO DE EJECUCIÓN EN INFORMÁTICA

NOVIEMBRE, 2010

Agradezco a mis padres Jorge e Isabel,
mis hermanos Felipe y Francisco
y todos los que de alguna u otra forma
me acompañaron en este largo camino.

Índice

Lista de Abreviaturas

GNU GPL	: Licencia Pública General
HTML	: Lenguaje de Marcado de Hipertexto
PDF	: Formato de documento portátil
PHP	: Php Hypertext Pre-processor
RI	: Recuperación de información
SQL	: Lenguaje de Consulta Estructurado
SVM	: Máquina de Soporte Vectorial
UML	: Lenguaje Unificado de Modelado
URL	: Localizador uniforme de recursos
WML	: Wireless Markup Language
WWW	: World Wide Web

Lista de Símbolos

$\prod$: Productoria

$\sum$: Sumatoria

$<$: Menor que

$>$: Mayor que

$\leq$: Menor o igual que

$\geq$: Mayor o igual que

ϵ : Pertenencia

Lista de Figuras

Lista de Tablas

Resumen

Los procesos de toma de decisiones requieren de la utilización de información oportuna y veraz para su implementación exitosa en los negocios. El análisis de información generada por internautas en la web permite apoyar los procesos de toma de decisiones en empresas de diversos tamaños en distintas áreas de negocio, tales como seguimiento de estrategias de marketing o análisis del grado de aceptación por parte del usuario de algún nuevo producto o servicio. Asimismo, el auge de la web 2.0, en la cual la información es entregada por los propios usuarios, nos da la posibilidad de utilizar técnicas automatizadas para poder procesar y extraer conocimiento de esta gran cantidad de textos.

El objetivo de esta investigación es utilizar técnicas de clasificación automática, las cuales se han comportado de mejor forma en distintos estudios utilizando un set de datos conocido, sobre comentarios de películas clasificadas en positivo o negativo. Para su pre procesamiento y posterior clasificación se desarrolló un software, el cual entrega resultados de la aplicación de las técnicas.

A partir de la obtención de los resultados, se puede plantear que Bayes Ingenuo logra una menor precisión (72.2%) que las Máquinas de Soporte Vectorial (85%), lo que es corroborado por otros resultados publicados en la literatura

Palabras Claves: Análisis de Sentido, Máquinas de Aprendizaje, Maquinas de Soporte Vectorial, Bayes Ingenuo.

Abstract

The process of decision making requires the use of opportune and truthful information for a successful implementation in business. The analysis of information generated by users on the web can support the processes of decision making in companies of different sizes in different business areas, such as monitoring of marketing strategies or analysis of the degree of acceptance by the user of a new product or service. Also, the boom of Web 2.0, in which information is given by users, gives us the possibility to use automated techniques to be able to process and extract knowledge of this large amount of information data.

The objective of this research is to use techniques of automatic classification, which have behaved in the best shape in different studies using a known set of data, about comments of movies classified as positive or negative. For pre-processing and subsequent classification software has been developed, which delivers results of the application of techniques.

From the obtain results, it can be argued that Naïve Bayes achieved less accuracy (72.2%) that Support Vector Machine (85%), which is corroborated by other published results in literature.

Keywords: Sentiment Analysis, Machine Learning, Support Vector Machine, Naïve Bayes.

1 DESCRIPCIÓN DEL PROYECTO

1.1 Introducción

El Análisis de Sentido ha logrado un gran interés, ayudado por la gran cantidad de información que se dispone gracias a la Internet, la que es una excelente fuente de información. El análisis automático de sentido en opiniones generadas en Internet es muy útil para compañías o instituciones para medir la percepción de los usuarios frente a determinado bien o servicio. Uno de los recursos son blogs donde se comenta sobre ciertos objetos al que hace referencia determinado post en el cual todos los usuarios pueden comentar, otro recurso son reviews donde sitios especializados en ciertos temas dan a conocer su apreciación sobre determinado producto.

Generalmente la persona que comenta en la web ha tenido previo conocimiento del producto o servicio, lo cual le da facultad para emitir una opinión positiva o negativa, el cual es una influencia para las personas que leen estos comentarios, ya sea empresas de marketing o consumidores. Por esto, extraer esta información es muy valioso.

Estas opiniones son extraídas en tiempo real, por ejemplo, al momento de lanzar un producto tecnológico inmediatamente se da a conocer en blogs y páginas de especialidad tecnológica donde se entregan las características de este producto y gente va emitiendo sus opiniones pudiendo cambiar la(s) estrategia(s) de marketing sobre determinado producto.

Se trabajará con Maquinas de Aprendizaje para poder diferenciar opiniones enfocada en comentarios de películas. Es posible diferencia estas opiniones ya que en éstas se encuentran palabras comunes al hacer una crítica, por ejemplo “malo”, “pésimo”, “bueno”, “bonito” y todo esto enmarcado en un lenguaje acotado, el que hace referencia al cine.

Las técnicas elegidas en Máquinas de Aprendizaje han sido Máquinas de Soporte Vectorial y Bayes Ingenuo, al comportarse de mejor manera en la clasificación de texto [1]. Para el pre procesamiento de datos se utilizará Stop Words, el cual elimina las conexiones entre oraciones y Stemming, el cual trabaja las palabras en su forma de raíz para no repetir palabras que tienen igual significado y están escritas en distintos tiempos verbales.

El software desarrollado tendrá la opción de utilizar las técnicas de Máquinas de Aprendizaje elegidas, pudiendo optar entre sus distintos parámetros y si los textos son pre procesados utilizando Stop Words y/o Stemming para su posterior análisis de resultados.

1.2 Definición de Objetivos

1.2.1 Objetivo General

- Utilizar Maquinas de Aprendizaje para categorizar textos en positivo o negativo, según sea su sentido.

1.2.2 Objetivos Específicos

- Utilizar Máquinas de Aprendizaje, eligiendo Bayes Ingenuo y Maquinas de Soporte Vectorial como técnicas para clasificar en positivo o negativo, según sea su sentido, distintos textos extraídos de comentarios sobre películas.
- Desplegar resultados obtenidos y comparar técnicas utilizadas con el fin de analizar cual forma es la más eficiente para determinado tipo de texto.
- Habilitar una interfaz web en la cual se pueda almacenar textos para poder ser utilizados en un entrenamiento o prueba de categorización de sentido.

1.3 Plan de Trabajo

Según el plan de trabajo diseñado, se comienza con una investigación general de *Machine Learning* y todo lo que conlleva, tales como extracción de la información que se ocupará (en este caso textos cortos), el pre procesado de esta información para poder ser procesada por los algoritmos elegidos para este fin, la elección de la técnica que se usará para ver cómo se comporta la clasificación de estos textos y el posterior análisis de los resultados. Luego viene una etapa de investigación más específica, donde ya se eligió los dos algoritmos de Machine Learning que se utilizaran, siendo estos Naïve Bayes y Máquinas de Soporte Vectorial, utilizando software libres para poder hacer pruebas con set de datos ya ocupados en otros estudios y disponibles para su libre uso, una vez terminado comparar resultados obtenidos con los ya publicados por los autores, viendo cómo se comportan con distintas técnicas, por ejemplo, kernels en máquinas de soporte vectorial y pre procesado de los textos.

Durante la segunda parte del proyecto de tesis, se lleva a cabo un Blog en donde se va plasmando todo lo aprendido durante este tiempo referente al tema, el que va siendo actualizado cada dos semanas, con temas elegidos del propio informe o investigado con el fin de ser aplicado más adelante. Este desarrollo tiene como fin el poder preparar un post e investigar muy bien sobre lo que se pondrá.

En paralelo a las reuniones semanales con profesor guía se irá desarrollando lo que será el software de muestra para el proyecto 2. Este se irá desarrollando con el fin de poder aplicar los conocimientos adquiridos en la investigación, como un software científico y no de uso masivo.

	Nombre de tarea	Duración	Comienzo	Fin	Pre
1	**Inicio**	**86 días?**	**mié 05-08-09**	**vie 20-11-09**	
2	**Investigacion General**	**30 días**	**mié 05-08-09**	**mar 15-09-09**	
3	Machine Learning	15 días	mié 05-08-09	mar 25-08-09	
4	Representación Lenguaje Natural	15 días	mié 26-08-09	mar 15-09-09	
5	**Informe de Avance**	**8 días?**	**mié 16-09-09**	**vie 25-09-09**	
6	Comenzar Redacción	5 días?	mié 16-09-09	mar 22-09-09	
7	Reajustar Redacción Informe de Avance	2 días?	mié 23-09-09	jue 24-09-09	6
8	Entrega Informe de Avance	1 día?	vie 25-09-09	vie 25-09-09	7
9	**Investigación Especifica**	**18 días?**	**mié 07-10-09**	**mar 27-10-09**	
10	Representación Lneguaje Natural	3 días?	mié 07-10-09	vie 09-10-09	
11	Reducción Dimensionalidad de Vectores	2 días?	dom 11-10-09	lun 12-10-09	10
12	Similitud entre Vectores	3 días?	mié 14-10-09	vie 16-10-09	11
13	Maquinas de Soporte Vectorial	3 días?	dom 18-10-09	mar 20-10-09	12
14	Naïve Bayes	3 días?	mié 21-10-09	vie 23-10-09	13
15	Software Usados	3 días?	dom 25-10-09	mar 27-10-09	14
16	Adaptación de Algoritmos a usar	4 días?	mié 28-10-09	sáb 31-10-09	15
17	Busqueda de material de test	3 días?	lun 02-11-09	mié 04-11-09	16
18	Pruebas con Maquina de Aprendizaje	3 días?	jue 05-11-09	sáb 07-11-09	17
19	Analizar Resultados	2 días?	lun 09-11-09	mar 10-11-09	18
20	Diseñar entrega de Resultados	3 días?	mié 11-11-09	vie 13-11-09	19
21	**Informe Final**	**19 días?**	**lun 02-11-09**	**vie 20-11-09**	
22	Comenzar Redaccion	14 días?	lun 02-11-09	dom 15-11-09	
23	Reajustar Redacción Informe de Avance	4 días?	lun 16-11-09	jue 19-11-09	22
24	Entrega Informe de Avance	1 día?	vie 20-11-09	vie 20-11-09	23

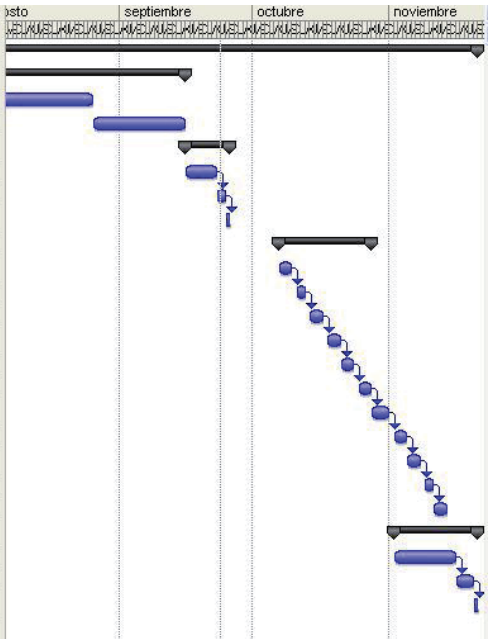

Figura 1.1. Plan de Trabajo Proyecto 1

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	**Inicio**	**72 días?**	**dom 07-03-10**	**lun 14-06-10**	
2	**Reuniones**	**31 días?**	**jue 11-03-10**	**jue 22-04-10**	
3	Reunion N°1	1 día	jue 11-03-10	jue 11-03-10	
4	Reunion N°2	1 día?	jue 18-03-10	jue 18-03-10	3
5	Reunion N°3	1 día?	jue 25-03-10	jue 25-03-10	4
6	Reunion N°4	1 día?	jue 01-04-10	jue 01-04-10	5
7	Reunion N°5	1 día?	jue 08-04-10	jue 08-04-10	6
8	Reunion N°6	1 día?	jue 15-04-10	jue 15-04-10	7
9	Reunion N°7	1 día	jue 22-04-10	jue 22-04-10	8
10	**Blog**	**72 días?**	**dom 07-03-10**	**lun 14-06-10**	
11	Actualización N°1	5 días?	dom 07-03-10	jue 11-03-10	
12	Actualización N°2	1 día?	lun 22-03-10	lun 22-03-10	11
13	Actualización N°3	1 día?	lun 05-04-10	lun 05-04-10	12
14	Actualización N°4	1 día?	lun 19-04-10	lun 19-04-10	13
15	Actualización N°5	1 día?	lun 03-05-10	lun 03-05-10	14
16	Actualización N°6	1 día?	lun 17-05-10	lun 17-05-10	15
17	Actualización N°7	2 días?	lun 31-05-10	mar 01-06-10	16
18	Actualización N°8	1 día?	lun 14-06-10	lun 14-06-10	17
19	**Informe**	**11 días?**	**vie 09-04-10**	**vie 23-04-10**	
20	Informe de Avance	9 días?	vie 09-04-10	mié 21-04-10	
21	Resumen Ejecutivo	2 días?	jue 22-04-10	vie 23-04-10	20
22	Desarrollo	66 días?	jue 11-03-10	jue 10-06-10	

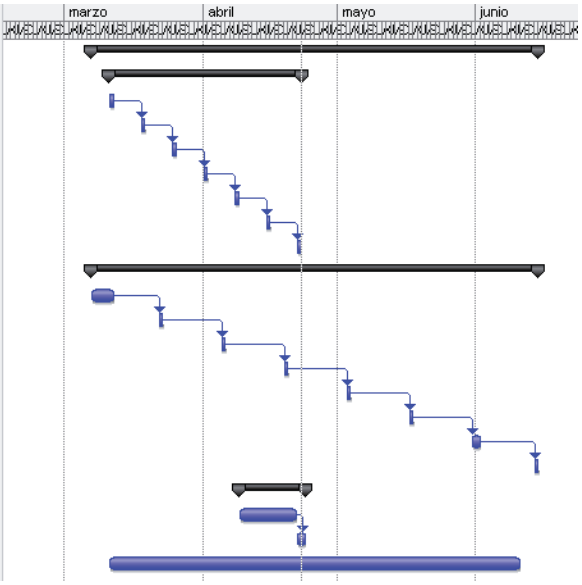

Figura 1.2. Plan de Trabajo Proyecto 2

1.4 Estado del Arte

La Web es una fuente de gran información con textos que siguen distintas formas de estructura. Analizar esos textos es de gran importancia y mucho mejor si se puede extraer información que sea útil al momento de analizar lo que se escribe sobre cierta marca, producto, persona, etc. La clave de esta información esta expresada a través de opiniones que usuarios desde cualquier punto con conexión a Internet puede escribir. Muchas empresas

buscan público o consumidores para saber que opinan sobre sus productos y servicios. En la web 2.0, donde la web se basa en comunidades de usuarios y servicios como redes sociales, blogs o wikis donde se fomenta la colaboración y el intercambio de información entre usuarios nos da una fuente inagotable de información que puede ser usada para el estudio de estas opiniones y posterior toma de decisión por parte de ejecutivos sobre estrategias de marketing y como mejorar la llegada de su producto o servicio a los consumidores. Este contenido es una gran influencia para los lectores que al encontrar muchas opiniones positivas sobre determinado producto lleva a influir en la decisión final de compra, al igual que las opiniones negativas pueden llevar a no adquirir ese producto.

Uno de los mayores investigadores para determinar la orientación de las opiniones es Fabrizio Sebastiani, el cual ha usado distintos métodos para lograrlos, determinando términos subjetivos y términos de orientación en las opiniones [1] donde tiene un set de términos positivos como por ejemplo “good”, “nice” y términos negativos como por ejemplo “bad”, “nasty”. Ayudado de WordNet el cual es una enorme base de datos léxica del idioma Inglés el cual agrupa las palabras en conjuntos de sinónimos llamados ”Synsets”, proporcionando definiciones cortas y generales, y almacenando las relaciones semánticas entre estos conjuntos de sinónimos. El propone la construcción de un SentiWordNet junto con Andrea Esuli [2], la que es una adaptación de los llamados “Synsets” para clasificar la polaridad. Otra técnica que usa Sebastiani junto con Shlomo Argamon, Kenneth Bloom y Andrea Esuli [3] es el uso de “Appraisal Theory” que se ocupa de los recursos lingüísticos por medio de los cuales los textos llegan a expresar, negociar y naturalizar determinadas posiciones intersubjetivas y en última instancia, ideológicas. [4]

2 REPRESENTACION DE LENGUAJE NATURAL

2.1 Presentación

Con el incremento del número de documentos en formato electrónico, se hace necesario contar con herramientas informáticas adecuadas para la recuperación de documentos. Las técnicas manuales han demostrado ser ineficaces, básicamente consisten en la elaboración manual de una descripción del contenido temático de cada uno de los documentos, siendo éste un trabajo costoso en tiempo, que además adolece de abundantes inconsistencias. Por otra parte, las búsquedas conocidas como en texto libre (búsqueda de subcadenas a fin de cuentas) se han mostrado incapaces de resolver dos problemas básicos: la sinonimia (relación semántica de identidad o semejanza de significados entre determinadas palabras u oraciones) y la polisemia (cuando una misma palabra o signo lingüístico tiene varias acepciones). Por ello, la investigación en recuperación de información (RI) busca diseñar sistemas que acepten consultas en lenguaje natural y proporcionen documentos adecuados a tales consultas, ordenados según algún criterio del sistema, de acuerdo a las características de los documentos y a las necesidades informativas expresadas por el usuario en su consulta. Uno de los modelos más conocido y difundido para el sistema de recuperación es el llamado modelo vectorial.

2.2 Conceptos Básicos de la recuperación de información

La Recuperación de información (Information Retrieval) es el estudio que ayuda a los usuarios a encontrar la información que se ajuste a sus necesidades. Técnicamente, la recuperación de la información sigue los estudios de adquisición, organización, almacenamiento, recuperación y distribución de la información. Históricamente la recuperación de información de documentos, hace hincapié en el documento como la unidad básica. La figura muestra una arquitectura general de un sistema de recuperación de información.

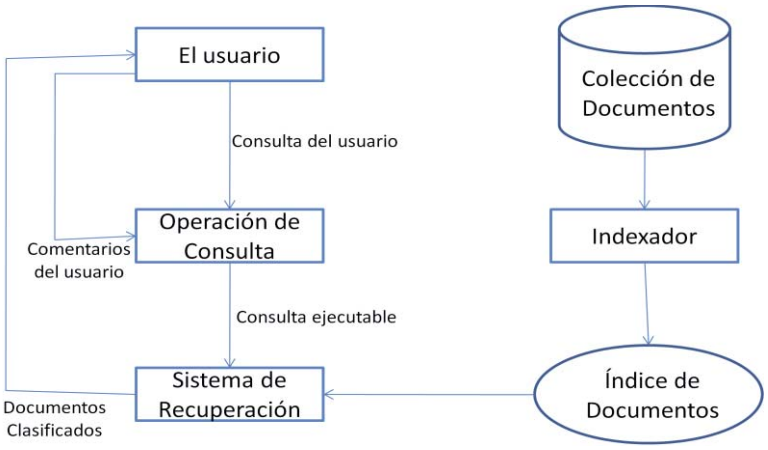

Figura 2.1. Arquitectura general de un sistema de Recuperación de Información

En la figura, el usuario necesita información de las consultas (*consulta de usuario*) para el *sistema de recuperación* a través del modulo *operación de consulta.* En el *modulo de*

recuperación se utiliza el *índice de documentos* para recuperar los documentos que contengan algunos de los términos de la consulta (tales documentos pueden ser relevantes para la consulta), calcula los resultados de relevancia para ellos y entrega los documentos recuperados de acuerdo a los resultados. Los *Documentos Clasificados* son los que se presentan al usuario. La *Colección de Documentos* se llama también Base de Datos de texto, que esta indexado por el indexador para una recuperación más eficiente.

Una consulta de usuario representa la información que el usuario necesita, que puede ser de las siguientes formas:

- **Consulta por palabra clave:** el usuario expresa sus necesidades de información con una lista de al menos una palabra clave (o término) con el objetivo de encontrar si el documento contiene alguno de ellos (al menos uno) o todos los términos de la consulta. Los términos de la lista se supone que están relacionados con una "suave" versión de la lógica AND. Por ejemplo, si uno está interesado en encontrar información acerca de "web mining", se puede ejecutar la consulta "web mining" en un sistema de recuperación de información o motor de búsqueda. "Web Mining" es buscado como "Web AND Mining". El sistema de recuperación luego de encontrar los documentos relevantes y clasificarlos adecuadamente los presenta al usuario. Hay que tener en cuenta que los documentos recuperados no tienen por qué contener todos los términos de la consulta. En algunos sistemas de recuperación de información, el orden de las palabras si es importante y afecta en los resultados de la recuperación.
- **Consultas Booleanas:** el usuario puede utilizar operadores booleanos como AND, OR y NOT para construir consultas más complejas. Por lo tanto, dichas consultas consisten en términos y operadores booleanos. Por ejemplo, "data OR web" es una consulta booleana, que solicita documentos que contengan la palabra "data" o "web". La consulta es devuelta si la consulta booleana es verdadera, por ejemplo si hay una concordancia exacta. Uno puede escribir consultas booleanas complejas utilizando los tres operadores, pero los usuarios rara vez escriben este tipo de consultas.
- **Consultas de frases:** tales consultas constan de una secuencia de palabras que componen una frase. Cada documento devuelto debe contener al menos una instancia de esa frase. En los motores de búsqueda una consulta de frase es normalmente cerrada con comillas dobles. Por ejemplo se puedo ejecutar la siguiente frase (incluyendo las comillas dobles) "Las técnicas de minería y aplicaciones web" para encontrar documentos que contengan la frase exacta.
- **Consultas de Proximidad:** la proximidad de consultas es una versión relajada de las consultas y la frase puede ser una combinación de términos y frases. Consultas de proximidad busca los términos de consulta dentro de la cercanía de uno u otro. La cercanía de las frases es utilizada como un factor de ranking de los documentos devueltos. Por ejemplo, un documento que contenga todos los términos de la consulta juntas se considera más relevante que un documento que contenga los términos de la consulta mas separados. Algunos sistemas permiten que el usuario especifique la máxima distancia permitida entre los términos de la consulta. La mayoría de los motores de búsqueda tienen en cuenta la proximidad y el orden de los términos en la recuperación de información.
- **Consulta a Documentos Completos:** cuando la consulta es un documento completo, el usuario desea encontrar otros documentos que son similares al documento de la

consulta. Algunos motores de búsqueda (como Google) permiten al usuario proporcionar la URL de la página a consultar. Además en los resultados devueltos por el motor de búsqueda, cada fragmento puede tener un enlace "mas de esto" o "páginas similares". Cuando se devuelve el usuario hace clic en el enlace y un conjunto de páginas similares a esa página son presentadas.

- **Consultas en Lenguaje Natural:** este es el caso más complejo y también el caso ideal. El usuario expresa su necesidad de información como una pregunta en lenguaje natural. Sin embargo, dichas consultas siguen siendo difíciles de entender debido a la dificultad de comprensión del lenguaje natural. Sin embargo, esta es un área activa de investigación, llamado "responder a la pregunta". Algunos sistemas de búsqueda están empezando a ofrecer servicios de respuestas a la pregunta sobre algunos tipos específicos de preguntas, las preguntas "definición" por ejemplo, que piden las definiciones de términos técnicos. Las definiciones de las preguntas suelen ser más fácil de contestar por que existen fuertes patrones lingüísticos que indican frases definición, por ejemplo "se define como", "se refiere a", etc.

El modulo de *Operación de Consultas* puede ser desde muy simples hasta muy complejas. En el caso más sencillo, no hace nada, pero solo tiene que pasar la consulta al motor de la recuperación después de algún simple pre procesamiento, por ejemplo stop-words (palabras que aparecen con mucha frecuencia en el texto y que tienen poco significado, por ejemplo "el", "a", "en", etc.). En casos más complejos, es necesario transformar las consultas de lenguaje natural en consultas ejecutables. Esto generalmente se denomina retroalimentación pertinente (relevance feedback).

El *Indexador* es el modulo que indexa los índices de los documentos originales para permitir una recuperación más eficiente. El resultado es el *índice de documentos*.

El *Sistema de Recuperación* calcula el nivel de importancia de cada documento indexado a la consulta. De acuerdo con sus puntuaciones pertinentes, los documentos son clasificados y presentados al usuario. Por lo general no se puede comparar la consulta del usuario con todos los documentos de la colección, ya que es muy ineficiente. En cambio, solo un pequeño subconjunto de los documentos que contiene al menos un término de la búsqueda y se encuentra en el índice de la consulta del usuario se toma en cuenta.

2.3 Modelos de Recuperación de información

Un modelo de recuperación de información rige de la forma como un documento y una consulta son representadas y como se comporta ese documento a la consulta definida por el usuario. Hay cuatro modelos de recuperación de información principal: modelo booleano, modelo de espacio vectorial, modelo de lenguaje y modelo probabilístico. Los modelos más comunes utilizados en los sistemas de recuperación de información son los tres primeros modelos.

Aunque estos tres modelos representan los documentos y consultas de manera distinta, se utilizan en el mismo marco. Todos ellos tratan cada documento o consulta como un "saco" de palabras o términos (bag of words). La secuencia y la posición de una frase o un documento

son ignorados. Es decir, un documento es descrito por un conjunto de términos distintos. Un término es simplemente una palabra, cuya semántica ayuda a recordar los temas principales del documento. Hay que señalar que aquí el término no puede ser una palabra de la lengua natural. Cada término está asociado con un peso. Dada una colección de documentos D, sea V = $\{t_1, t_2, \dots, t_{|V|}\}$ el conjunto de términos distintos de una colección, donde t es un término. V se suele llamar el **vocabulario** de la colección y |V| es su tamaño, es decir, el numero de términos en V. un peso $w_{ij} > 0$ está asociado a cada termino t_i de un documento d_j Є D. Por un término que no aparece en el documento d_j, $w_{ij} = 0$. Cada documento d_j está representado con un vector de expresión,

$$d_j = (w_{1j}, w_{2j}, \dots, w_{|V|j})$$

Donde cada peso w_{ij} corresponde a el termino t_i Є V y cuantifica el nivel de importancia de t_i en el documento d_j. La secuencia de los componentes (o términos) en el vector no es significativa. Con esta representación vectorial, una colección de documentos es simplemente representada como una tabla relacional (o una matriz). Cada término es un atributo. En diferentes modelos de recuperación, w_{ij} se calcula de una manera distinta.

2.3.1 Modelo Booleano

El modelo booleano es uno de los primeros modelos y más simple en recuperación de información. Utiliza el concepto de concordancia exacta para que coincida con los documentos a la consulta del usuario. Tanto la consulta como la recuperación se basan en algebra de booleanos.

- **Documento de representación:** en el modelo booleano, documentos y consultas se representan como conjuntos de términos. Es decir, cada término solo se considera presente o ausente en el documento. Usando la representación anteriormente vista, el peso w_{ij} (Є {0, 1}) del termino t_i en el documento d_j es 1 si t_i está presente en el documento y 0 en caso contrario, es decir,

$$w_{ij} = \begin{cases} 1 & si\ t_i\ esta\ presente\ en\ el\ documento\ d_j \\ 0 & en\ otro\ caso \end{cases}$$

- **Consultas Booleanas:** como se ha mencionado con anterioridad, los términos en las consultas se combinan lógicamente utilizando los operadores booleanos AND, OR y NOT, que tienen su semántica usual en la lógica. Por lo tanto, una consulta booleana tiene semántica precisa. Por ejemplo, la consulta (X AND Y) y (NOT Z) dice que un documento recuperados debe contener tanto los términos x e y, pero no z. Como ejemplo, la consulta (X OR Y) significa que al menos uno de los dos debe estar en cada documento recuperado. En este caso, suponemos que X, Y y Z son los términos.
- **Recuperación de Documento:** dada una consulta booleana, el sistema recupera todos los documentos que hace la consulta. Así la recuperación se basa en el criterio de decisión binaria, es decir, un documento puede ser relevante o

irrelevante. Esto se llama coincidencia exacta. No hay ninguna noción de coincidencia parcial o rankeada en los documentos recuperados. Esto es uno de las principales desventajas del modelo booleano, que a menudo conduce a resultados de recuperación pobres. Es evidente que la frecuencia de los términos y su proximidad contribuyen significativamente a la relevancia de un documento. Debido a este problema, el modelo booleano es rara vez utilizado, solo en la práctica. La mayoría de los motores de búsqueda utiliza formas limitadas de recuperación mediante inclusión explicita de operadores booleanos por ejemplo, la siguiente consulta puede ser hecha en Google, 'mining – data + "equipment Price"', donde + es inclusión y – es exclusión, operadores similares a AND y NOT, respectivamente.

2.3.2 Modelo de espacio Vectorial

Este modelo es tal vez el más conocido y más ampliamente utilizado para el modelo de recuperación de información.

Representación de Documentos: un documento en el modelo de espacio vectorial de representa como un vector de peso, en la que el peso de cada componente se calcula sobre la base de algunas variaciones de TF (frecuencia de termino) o TF-IDF (frecuencia de termino – frecuencia de documento inverso). El peso w_{ij} de el termino t_i en el documento d_j ya no es $\{0, 1\}$ como en el modelo booleano, sino que puede ser cualquier numero.

- **Frecuencia de Termino (TF):** en este método, el peso de un término t_i en un documento d_j se identifica como f_{ij}.Es posible aplicar normalización. Las deficiencias que tiene este sistema es que no tiene en cuenta la situación en la que un término aparece en muchos documentos de la colección. Donde este término puede ser discriminatorio.
- **TF-IDF:** este es el esquema de ponderación más conocido, donde TF se entiende como la frecuencia del término e IDF como la frecuencia inversa del documento. Existen varias variaciones a este sistema. Sea N el número total de documentos en el sistema o la colección y df_i el número de documentos en que el termino t_i aparece al menos una vez. Sea f_{ij} el recuento de las frecuencias del término t_i en el documento d_j. Entonces, la frecuencia normalizada (denotada por tf_{ij}) de t_i en d_j esta dada por

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|V|j}\}}$$

Donde el máximo se calcula sobre todos los términos que aparecen en el documento d_j. Si el termino t_i no aparece en d_j entonces $tf_{ij} = 0$. |V| es el tamaño del vocabulario de la colección. La frecuencia inversa de documento (denotada por idf_i) del termino t_i esta dada por

$$idf_i = \log \frac{N}{df_i}$$

Si un término aparece en un gran número de documentos en la colección, probablemente no es importante o no discriminatorio. El TF-IDF final está dado por:

$$w_{ij} = tf_{ij} \text{ x } idf_i$$

Consultas (Queries): Una consulta Q se representa de la misma forma como un documento en la colección de documentos. El peso del término w_{iq} de cada término t_i en Q se puede calcular de la misma manera como en un documento normal, o ligeramente diferente. Por ejemplo, Salton y Buckley [5] sugiere lo siguiente:

$$w_{iq} = \left(0.5 + \frac{0.5\, f_{iq}}{\max\{ f_{1q}, f_{2q}, \ldots, f_{|V|q}\}}\right) x\, log \frac{N}{df_i}$$

Recuperación de documentos y ranking por relevancia: A menudo es difícil tomar la decisión binaria sobre si un documento es relevante para una consulta determinada. A diferencia del modelo booleano, el modelo de espacio vectorial no toma una decisión. En cambio, los documentos se ordenan de acuerdo a sus grados de relevancia a la consulta. Una forma de calcular ese grado de relevancia es la de calcular la similitud de la consulta Q en cada documento d_j en la colección de documentos D. Hay muchas medidas de similitud. El más conocido es la similitud del coseno, que es el ángulo entre el vector de la consulta q y el vector del documento d_j,

$$\text{Coseno } (d_j, q) = \frac{\langle d_j \cdot q \rangle}{\left|\left|d_j\right|\right| \times ||q||} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}}$$

El producto escalar de dos vectores es otra medida de similitud,

$$\text{sim}(d_j, q) = \langle d_j \cdot q \rangle$$

La clasificación de los documentos de realiza utilizando los valores de similitud. Los documentos destacados se consideran más relevantes para la consulta.

Otra forma de evaluar el grado de importancia es calcular una puntuación para cada documento de la consulta. El método **Okapi** y sus variantes son técnicas populares en este entorno. La formular de recuperación de Okapi que se basa en [6, 7]. Se ha demostrado que las variaciones de Okapi son más eficaces que el coseno de consulta para la recuperación de consultas cortas.

Dado que es más fácil presentar la formula directamente a través de la bolsa de palabras (bag of words) de la notación de los documentos de vectores, el documento d_j se denota por d_j y la consulta q se denota por q. Anotaciones adicionales son las siguientes:

t_i Es un término.

f_{ij} Es el número de la frecuencia del término t_i en el documento d_j.

f_{iq} Es el número de la frecuencia del término t_i en la consulta q.

N Es el número total de documentos en la colección.

df_i Es el número de documentos que contienen el término t_i.

dl_j Es la longitud del documento en bytes de d_j.

$avdl$ Es la longitud promedio de los documentos de la colección.

El nivel de relevancia de Okapi de un documento d_j para una consulta q es:

$$\text{Okapi } (d_j, \text{q}) = \sum_{t_i \in q, d_j} \ln \frac{N - df_i + 0.5}{df_i + 0.5} \times \frac{(k_1 + 1) f_{ij}}{k_1 \left(1 - b + b \frac{dl_j}{avdl}\right) + f_{ij}} \times \frac{(k_2 + 1) f_{iq}}{k_2 + f_{iq}}$$

Donde k_1 (entre 1.0 – 2.0), b (usualmente 0.75) y k_2 (entre 1 – 1000) son parámetros. Sin embargo otra función es el "pivoted nornalization weighting" denotado por pnw [7]

$$\text{Pnw } (d_j, \text{q}) = \sum_{t_i \in q, d_j} \frac{1 + \ln (1 + \ln(f_{ij}))}{(1 - s) + s \frac{dl_j}{avdl}} \times f_{iq} \times ln \frac{N+1}{df_i}$$

Donde s es un parámetro (usualmente 0.2). Hay que tener en cuenta que estas funciones son empíricas basadas en intuiciones y evaluaciones experimentales. Hay muchas variaciones de estas funciones en la práctica.

2.3.3 Modelado de Lenguaje Estadístico

Modelado de lenguaje estadístico (o simplemente modelados de lenguaje) se basan en la probabilidad y tienen bases en la teoría estática. La idea básica de este enfoque de la recuperación es simple. En primer lugar, se estima un modelo de lenguaje para cada documento y clasifica el documento por la posibilidad de la consulta dado el modelo de lenguaje. Ideas similares han sido previamente utilizadas en el procesamiento del lenguaje natural y el reconocimiento de voz. La recuperación de información usando modelos de lenguaje fue propuesto por Ponte y Croft [8].

La consulta q es una secuencia de términos, q = $q_1, q_2, \ldots, q_m$ y la colección de documentos D un conjunto de documentos, D = $\{d_1, d_2, \ldots, d_N\}$. En el enfoque de modelado de lenguaje, consideramos que la probabilidad de que una consulta q como "generada" por un modelo probabilístico basado en un documento d_j, es decir, Pr(q|d_j). Para clasificar los documentos en la recuperación, estamos interesados en estimar la posterior probabilidad Pr(d_j|q). Usando las reglas de Bayes, tenemos

$$\text{Pr } (d_j|\text{q}) = \frac{\text{Pr}(q|d_j)\text{Pr } (d_j)}{\text{Pr } (q)}$$

Para la clasificación, $\Pr(q)$ no es necesario ya que es el mismo para cada documento. $\Pr(d_j)$ Generalmente se considera uniforme y con ello no afecta a la clasificación. Solo tenemos que calcular $\Pr(q|d_j)$.

El modelo de lenguaje utilizado en la mayor parte de los trabajos se basa en unigramas, es decir, solo en términos individuales (palabras) son considerados. Es decir, el modelo asume que cada termino (palabra) se genera de forma independiente, que es esencialmente una distribución multinomial sobre las palabras. El caso genera es el modelo N-Gramas, donde el enésimo termino está condicionado al termino anterior (n – 1).

Basados en la distribución multinomial y el modelo unigrama, tenemos

$$\Pr(\mathrm{q} = q_1, q_2, \ldots, q_m \mid d_j) = \prod_{i=1}^{m} \Pr(q_i \mid d_j) = \prod_{i=1}^{|V|} \Pr(t_i \mid d_j)^{f_{iq}}$$

Donde f_{iq} es el número de veces que el termino t_i ocurre en q y $\sum_{i=1}^{|V|} \Pr(t_i \mid d_j) = 1$. El problema de recuperación se reduce a la estimación de $\Pr(t_i \mid d_j)$, que puede ser la frecuencia relativa

$$\Pr(t_i \mid d_j) = \frac{f_{ij}}{|d_j|}$$

Recordemos que f_{ij} es el número de veces que el término t_i aparece en el documento d_j. $|d_j|$ Representa el número total de palabras en d_j.

Sin embargo, un problema con esta estimación es que un término que no aparece en d_j tiene la probabilidad de cero, lo que subestima la probabilidad de que el termino este oculto en el documento. Esta situación es similar en la clasificación de texto utilizando el modelo Naive Bayes. Un non-zero de probabilidad es típicamente asignado a cada término invisible en el documento, que se llama "smoothing". Smoothing ajusta la estimación de las probabilidades de producir las probabilidades más precisas. El nombre "smoothing" proviene del hecho de que estas técnicas tienden a hacer la distribución más uniforme, mediante el ajuste de bajas probabilidades como cero las probabilidades hacia arriba y las probabilidades de alta a la baja. No solo los métodos de smoothing previenen el cero probabilidad, sino que también trata de mejorar la exactitud del modelo en su conjunto. Tradicionalmente el aditivo smoothing es

$$Pr_{add}(t_i \mid d_j) = \frac{\lambda + f_{ij}}{\lambda |V| + |d_j|}$$

Donde $\boldsymbol{\lambda} = 1$, es el "Laplace Smoothing" and donde $0 < \boldsymbol{\lambda} < 1$, es el "Lidstone smoothing".

2.4 Relevancia de Retroalimentación

Para mejorar la eficacia en la recuperación, han propuesto muchas técnicas. Relevancia de Retroalimentación es uno bien efectivo. Es el proceso donde el usuario identifica algunos documentos relevantes e irrelevantes en la lista inicial de los documentos recuperados y el

sistema se crea una consulta ampliada mediante la extracción de algunos de los términos tradicionales de la muestra y los documentos pertinentes irrelevantes para una segunda ronda de recuperación. El sistema también puede producir otro modelo donde el usuario identifica los documentos relevantes e irrelevantes para clasificar documentos en la colección de documentos en los documentos relevantes e irrelevantes. El proceso de retroalimentación pertinente puede repetirse hasta que el usuario esté satisfecho con los resultados recuperados.

2.4.1 El método de Rocchio

Es uno de los algoritmos de relevancia de retroalimentación más efectiva. Se basa en el primer acercamiento anterior. Es decir, que utiliza al usuario para identificar documentos relevantes e irrelevantes para ampliar la consulta anterior. La nueva (o ampliada) consulta se utiliza para llevar a cabo la recuperación nuevamente. El vector original de la consulta es q, el conjunto de documentos relevantes seleccionados por el usuario es D_r y el conjunto de documentos irrelevantes es D_{ir}. La consulta expandida q_e se calcula de la siguiente forma,

$$q_e = \alpha\, q + \frac{\beta}{|D_r|} \sum_{d_r \in D_r} d_r - \frac{\gamma}{|D_{ir}|} \sum_{d_{ir} \in D_{ir}} d_{ir}$$

Donde α, β y γ son parámetros. La ecuación simplemente aumenta a la ecuación original de la consulta q los términos adicionales de los documentos pertinentes. Originalmente la consulta q sigue siendo necesaria, ya que refleja directamente la necesidad de información del usuario. Los documentos relevantes se consideran más importantes que los documentos irrelevantes. La resta se utiliza para reducir la influencia de esos términos que no sean discriminatorios (es decir que aparecen en los documentos relevantes e irrelevantes) y los que términos que aparecen en los documentos irrelevantes solamente. Los tres parámetros se establecen de manera empírica. Hay que tener en cuenta que una ligera variación del algoritmo es sin la normalización de $|D_r|$ y $|D_{ir}|$. Estos dos métodos son simples y eficientes y por lo general dan buenos resultados.

2.4.2 Métodos de Maquinas de Aprendizaje

Puesto que tenemos un conjunto de documentos relevantes e irrelevantes, se puede construir un modelo de clasificación de los mismos. Entonces, el problema de retroalimentación pertinente se convierte en un problema de aprendizaje. Cualquier método de aprendizaje supervisado puede ser utilizado, por ejemplo Naïve Bayes y SVM. Una variación del Método Rocchio anterior, llamado Método de Clasificación de Rocchio se puede utilizar para este fin. La construcción de un Clasificar de Rocchio se realiza mediante la construcción de un vector prototipo c_i para cada case i , que puede ser relevante o irrelevante en este caso (elementos negativos o componentes del vector c_i se suelen fijar en 0):

$$c_i = \frac{\propto}{|D_i|} \sum_{d \in D_i} \frac{d}{||d||} - \frac{\beta}{|D - D_i|} \sum_{d \in D - D_i} \frac{d}{||d||}$$

Donde D_i es la colección de documentos de la clase i y α y β son parámetros. Usando la TF-IDF la ponderaciones de los términos α = 16 y β = 4, por lo general con estos valores

trabaja muy bien. En la clasificación se aplica la similitud del coseno. Es decir, cada documento de prueba d_t, se compara con cada prototipo $c_i$basado en la similitud del coseno. d_t Se asigna a la clase con el que el valor más alto de similitud. Algoritmo:

For cada clase i **do**

Construir un vector prototipo c_i ocupando la ecuación anteriormente vista

End for

For cada test de documento d_i **do**

La clase de d_t es $argmax_i$ cosine (d_t, c_i)

End for

2.5 Medidas de Evaluación

En la Recuperación de información, generalmente no se toma la decisión si el documento es relevante o irrelevante para una consulta. En su lugar, una clasificación de documentos se produce para el usuario.

La colección de documentos en la Base de datos es D y el número de total de documentos en D es N. dada una consulta de un usuario, el algoritmo de recuperación calcula primero resultados importantes para todos los documentos en D y luego elabora un ranking R_q de los documentos basados en la puntuación de relevancia, es decir

$$R_q : < d_1^q, d_2^q, \dots , d_N^q >$$

Donde d_1^q ∈ D es el documento más relevante de la consulta q y d_N^q ∈ D es el documento mas irrelevante a la consulta q.

Sea D_q (⊆ D) el conjunto de los documentos más relevantes a la consulta q en D. se puede calcular la precisión y los valores a recordar de cada uno de d_i^q en el ranking.

Recall: el ranking de la posición i o el documento d_i^q (denotado por $r(i)$) es la fracción de los documentos relevantes desde d_1^q a d_i^q en R_q si $s_i \leq |D_q|$ ($|D_q|$ es el tamaño de D_q). Entonces

$$r(i) = \frac{s_i}{|D_q|}$$

La **precisión** en la posición i del ranking o el documento d_i^q (denotado por $p(i)$) es la fracción de los documentos desde d_1^q hasta d_i^q en R_q que son relevantes:

$$p(i) = \frac{s_i}{i}$$

Ejemplo: Tenemos una colección de documento D con 20 documentos. Dada la consulta q, sabemos que ocho documentos son relevantes a q (con signo positivo). Un algoritmo de recuperación produce el ranking (de todos los documentos en D), como se muestra la tabla

Tabla 2.1. Precisión y Recall de los valores de cada posición en el ranking

Rank *i*	+/–	*p*(*i*)	*r*(*i*)
1	+	1/1 = 100%	1/8 = 13%
2	+	2/2 = 100%	2/8 = 25%
3	+	3/3 = 100%	3/8 = 38%
4	–	3/4 = 75%	3/8 = 38%
5	+	4/5 = 80%	4/8 = 50%
6	–	4/6 = 67%	4/8 = 50%
7	+	5/7 = 71%	5/8 = 63%
8	–	5/8 = 63%	5/8 = 63%
9	+	6/9 = 67%	6/8 = 75%
10	+	7/10 = 70%	7/8 = 88%
11	–	7/11 = 63%	7/8 = 88%
12	–	7/12 = 58%	7/8 = 88%
13	+	8/13 = 62%	8/8 = 100%
14	–	8/14 = 57%	8/8 = 100%
15	–	8/15 = 53%	8/8 = 100%
16	–	8/16 = 50%	8/8 = 100%
17	–	8/17 = 53%	8/8 = 100%
18	–	8/18 = 44%	8/8 = 100%
19	–	8/19 = 42%	8/8 = 100%
20	–	8/20 = 40%	8/8 = 100%

En la columna 1, 1 representa el más alto ranking y 20 representa el ranking más bajo. "+" y "-" en la columna 2 indican que un documento es relevante o irrelevante, respectivamente. La precisión p(i) y el recuerdo r(i) son los valores en cada posición i que se muestra en las columnas 3 y 4.

Precisión media: a veces queremos una precisión única para comparar diferentes algoritmos de recuperación en una consulta q. una precisión promedio (p_{avg}) puede calcularse sobre la base de la precisión en cada documento relevante en el ranking.

$$P_{avg} = \frac{\sum_{d_i^q \in D_q} p(i)}{|D_q|}$$

Para el ranking de la figura 6.3, del ejemplo, la precisión promedio es 81%:

$$P_{avg} = \frac{100\%+100\%+100\%+80\%+71\%+67\%+70\%+62\%}{8} = 81\%$$

Curva de Precision-Recall: Basándose en la precisión y recall de cada posición en el ranking, podemos dibujar una curva de precisión-recall con recall en el eje x y precisión en el eje y en lugar de utilizar la precisión y recall en cada posición del ranking, la curva es frecuentemente representada utilizando 11 niveles, 0%, 10%, 20%, … 100%. Ya que no se puede obtener

estos niveles exactamente desde el ranking, es necesaria la interpolación para obtener los niveles necesarios tanto de precisión y recall

Ejemplo, podemos obtener la interpolación de los 11 niveles que se muestra en la tabla 2.1

i	$p(r_i)$	r_i
0	100%	0%
1	100%	10%
2	100%	20%
3	100%	30%
4	80%	40%
5	80%	50%
6	71%	60%
7	70%	70%
8	70%	80%
9	62%	90%
10	62%	100%

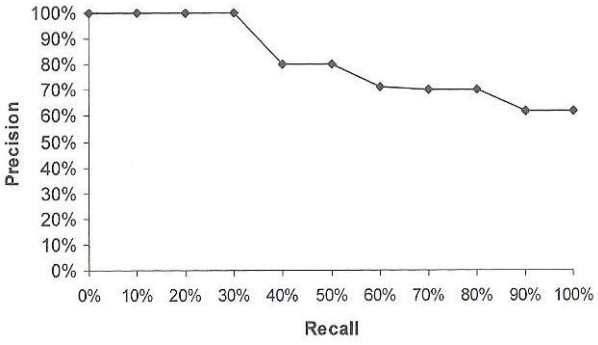

Figura 2.2. Curva Precision-Recall

2.6 Pre Procesado de Texto

Antes de que una colección de documentos sea usada, se deben realizar algunas tareas de pre procesamiento. Las tareas son remover Stop Words, Stemming y manipular los dígitos, guiones y puntuaciones.

2.6.1 Remover Stop Words

Son palabras insignificantes que ayudan a construir oraciones, pero no representan ningún contenido en los documentos. Artículos, preposiciones, conjunciones y algunos pronombres son algunos candidatos. Un Stop Words en ingles incluye:

A, able, about, above, abst, accordance, according, accordingly, across, act, actually, added, adj, adopted, affected, affecting, affects, after, afterwards, again, against, ah, all, almost, alone, along, already, also, although, always, am, among, amongst, an, and, announce, another, any, anybody, anyhow, anymore, anyone, anything, anyway, anyways, anywhere, apparently, approximately, are, aren, aren't, arise, around, as, aside, ask, asking, at, auth,, available, away, awfully, b, back, be, became, because, become, becomes, becoming, been, before, beforehand, begin, beginning, beginnings, begins, behind, being, believe, below, beside, besides, between, beyond, boil, both, brief, briefly, but, by, c, ca, came, can, cannot, cant, cause, causes, certain, certainly, co, com, come, comes, contain, containing, contains, could, couldnt, d, date, did, didn't, different, do, does, doesn't, doing, done, don't, down, downwards, due, during, e, each, ed, edu, effect, eg, eight, eighty, either, else, elsewhere, end, ending, enough, especially, et, et-al, etc, even, ever, every, everybody, everyone, everything, everywhere, ex, except, f, far, few, ff, fifth, first, five, fix, followed, following, follows, for, former, formerly, forth, found, four,

from, further, furthermore, g, gave, get, gets, getting, give, given, gives, giving, go, goes, gone, got, gotten, h, had, happens, hardly, has, hasn’t, have, haven’t, having, he, hed, hence, her, here, hereafter, hereby, herein, heres, hereupon, hers,, herself, hes, hi, hid, him, himself, his, hither, home, how, howbeit, however, hundred, I, id, ie, if, i'll, im, immediate, immediately, importance, important, in, inc, indeed, index, information, instead, into, invention, inward, is, isn’t, it, itd, it'll, its, itself, i've, j, just, k, keep, keeps, kept, keys, kg, km, know, known, knows, l, largely, last, lately, later, latter, latterly, least, less, lest, let, lets, like, liked, likely, line, little, 'll, look, looking, looks, ltd, m, made, mainly, make, makes, many, may, maybe, me, mean, means, meantime, meanwhile, merely, mg, might, million, miss, ml, more, moreover, most, mostly, mr, mrs, much, mug, must, my, myself, n, na, name, namely, nay, nd, near, nearly, necessarily, necessary, need, needs, neither, never, nevertheless, new, next, nine, ninety, no, nobody, non, none, nonetheless, no one, nor, normally, nos, not, noted, nothing, now, nowhere, o, obtain, obtained, obviously, of, off, often, oh, ok, okay, old, omitted, on, once, one, ones, only, onto, or, ord, other, others, otherwise, ought, our, ours, ourselves, out, outside, over, overall, owing, own, p, page, pages, part, particular, particularly, past, per, perhaps, placed, please, plus, poorly, possible, possibly, potentially, pp, predominantly, present, previously, primarily, probably, promptly, proud, provides, put, q, que, quickly, quite, qv, r, ran, rather, rd, re, readily, really, recent, recently, ref, refs, regarding, regardless, regards, related, relatively, research, respectively, resulted, resulting, results, right, run, s, said, same, saw, say, saying, says, sec, section, see, seeing, seem, seemed, seeming, seems, seen, self, selves, sent, seven, several, shall, she, shed, she'll, shes, should, shouldn’t, show, showed, shown, showns, shows, significant, significantly, similar, similarly, since, six, slightly, so, some, somebody, somehow, someone, somethan, something, sometime, sometimes, somewhat, somewhere, soon, sorry, specifically, specified, specify, specifying, state, states, still, stop, strongly, sub, substantially, successfully, such, sufficiently, suggest, sup, sure, t, take, taken, taking, tell, tends, th, than, thank, thanks, thanx, that, that'll, that’s, that've, the, their, theirs, them, themselves, then, thence, there, thereafter, thereby, thered, therefore, therein, there'll, thereof, there’re, theres, thereto, thereupon, there've, these, they, theyd, they'll, theyre, they've, think, this, those, thou, though, though, thousand, through, through, throughout, thru, thus, til, tip, to, together, too, took, toward, towards, tried, tries, truly, try, trying, ts, twice, two, u, un, under, unfortunately, unless, unlike, unlikely, until, unto, up, upon, ups, us, use, used, useful, usefully, usefulness, uses, using, usually, v, value, various, 've, very, via, viz, vol, vols, vs, w, want, wants, was, wasn’t, way, we, wed, welcome, we'll, went, were, weren’t, we've, what, whatever, what'll, whats, when, whence, whenever, where, whereafter, whereas, whereby, wherein, wheres, whereupon, wherever, whether, which, while, whim, whither, who, whod, whoever, whole, who'll, whom, whomever, whos, whose, why, widely, willing, wish, with, within, without, wont, words, world, would, wouldn’t, www, x, y, yes, yet, you, youd, you'll, your, youre, yours, yourself, yourselves, you've, z, zero

Y uno en español incluye:

Un , una, unas, unos, uno, sobre, todo, también, tras, otro, algún, alguno, alguna, algunos, algunas, ser, es, soy, eres, somos, sois, estoy, esta, estamos, estáis, están, como, en, para, atrás, porque, por qué, estado, estaba, ante, antes, siendo, ambos, pero, por, poder, puede, puedo,

podemos, podéis, pueden, fui, fue, fuimos, fueron, hacer, hago, hace, hacemos, hacéis, hacen, cada, fin, incluso, primero desde, conseguir, consigo, consigue, consigues, conseguimos, consiguen, ir, voy, va, vamos, vais, van, vaya, ha, tener, tengo, tiene, tenemos, tenéis, tienen, el, la, lo, las, los, su, aquí, mío, tuyo, ellos, ellas, nos, nosotros, vosotros, vosotras, si, dentro, solo, solamente, saber, sabes, sabe, sabemos, sabéis, saben, ultimo, largo, bastante, haces, muchos, aquellos, aquellas, sus, entonces, tiempo, verdad, verdadero, verdadera, cierto, ciertos, cierta, ciertas, intentar, intento, intenta, intentas, intentamos, intentáis, intentan, dos, bajo, arriba, encima, usar, uso, usas, usa, usamos, usáis, usan, emplear, empleo, empleas, emplean, empleáis, valor, muy, era, eras, éramos, eran, modo, bien, cual, cuando, donde, mientras, quien, con, entre, sin, trabajo, trabajar, trabajas, trabaja, trabajamos, trabajáis, trabajan, podría, podrías, podríamos, podrían, podríais, yo, aquel.

Estas palabras deben ser removidas antes de indexar y almacenar los documentos.

2.6.2 Stemming

En muchos lenguajes, una palabra tiene varias formas sintácticas en función del contexto en que es utilizada. Por ejemplo, en Ingles, los sustantivos tienen formas plurales, los verbos tienen formas gerundio si se le agrega "ing" y utilizando verbos en tiempo pasado son diferentes al de tiempo presente. Esto es considerado como variaciones sintácticas en la forma de la raíz de la palabra. Estas variaciones causan un bajo recall de un sistema debido a que en un documento de referencia puede contener una variación de una palabra en la consulta y no la palabra exacta. Este problema puede ser parcialmente tratado por steamming.

Stemming es el proceso de reducción de las palabras a su stems (tallo) o root (raíz). Un stem es la parte de una palabra que queda después de la eliminación de los prefijos y sufijos. En ingles, la mayoría de las variantes de una palabra se generan por la introducción de sufijos (en lugar de prefijos). Por lo tanto en Ingles, significa generalmente remover sufijos. Por ejemplo, "computer", "computing" y "compute" son reducidos a "comput". "Walks", "walking" y "Walker" son reducidos a "walk". Stemming permite distintas variaciones de la palabra para ser considerados en la recuperación, lo que mejora el "recall". Existen varios algoritmos derivados, también conocidos como stemmers. En ingles el stemmers mas popular es "Martin Porter´s stemming algorithm".

Durante muchos años, investigadores han evaluado las ventajas y desventajas del uso de stemming. Claramente, stemming incrementa el "recall" y reduce el tamaño de la estructura indexada. Sin embargo, puede hacerle un daño a la precisión, porque muchos documentos irrelevantes, pueden considerarse relevantes. Por ejemplo, "cop" y "cope" son reducidos al stem "cop". Sin embargo si uno sabe que el documento es sobre policías, si contiene "cope" probable que sea relevante. Muchos experimentos han sido realizados por investigadores, pero aun no hay pruebas concluyentes que sea mejor de una manera u otra.

2.6.3 Otras tareas de Pre Procesamiento de Texto

Dígitos: Números y términos que contengan números son removidos de los documentos, excepto en algunos casos específicos, por ejemplo fechas, horas y otros tipos predefinidos

expresados con expresiones regulares. Sin embargo, en motores de búsquedas, estos son usualmente indexados.

Guiones: Suelen ser objeto de incoherencia su uso. Por ejemplo algunas personas usan "estado-del-arte", pero otros usan "estado del arte". Si los guiones en el primer caso son removidos, eliminamos el problema de inconsistencia. Sin embargo algunas palabras que tienen guiones son parte integral de una palabra, por ejemplo "Y-21". Así, en general, el sistema puede seguir una regla general, por ejemplo remover todos los guiones y otra regla sería tener algunas excepciones. También tenemos dos tipos para remover los guiones, reemplazando el guion por un espacio "estado-del-arte" quedaría como "estado del arte" y remover el guion sin ser reemplazado, donde "estado-del-arte" quedaría como "estadodelarte". En algunos sistemas es muy difícil de determinar cuál será la técnica correcta, por ejemplo si "pre-procesado" esa convertido a "pre procesado", puede ser relevante en un sentido distinto a si la palabra quedara como "preprocesado".

2.7 Indexación semántica Latente

Los modelos de recuperación discutidos hasta ahora se basan en palabras claves o coincidencias, es decir, igualando términos en la consulta del usuario con la de los documentos. Sin embargo, muchos conceptos u objetos pueden ser descritos de muchas maneras (con otras palabras), debido al contexto y los hábitos del idioma de la persona. Si una consulta de un usuario usa palabras distintas a las palabras utilizadas en un documento, el documento no se recuperará, aunque puede ser relevante porque en el documento se utilizan algunos sinónimos de las palabras consultadas por el usuario. Esto causa un bajo "recall". Por ejemplo, "picture", "image" y "photo" son sinónimos en el contexto de cámaras digitales. Si la consulta del usuario solo tiene la palabra "picture", los documentos relevantes que contengan "image" o"photo" no se recuperaran.

La indexación semántica latente, propuesta por Deerwester [9], tiene como objetivo hacer frente a este problema mediante la identificación de las asociaciones estáticas de los términos. Se utiliza una técnica llamada "descomposición de valor singular (SVD, por sus siglas en ingles) [10], para estimar la estructura latente y para eliminar el ruido. Los resultados de esta descomposición son descripciones de términos y documentos basados en la estructura semántica latente derivada de SVD. Esta estructura también se llama el oculto "concepto" del espacio, que asocia sintácticamente diferentes, pero semánticamente similares términos y documentos. Estos términos se transforman y los documentos con el "concepto" son usados para la recuperación, no así los términos o el documento original. Por otra parte, la consulta también se transforma en el "concepto" espacio antes de la recuperación.

Sea D la colección d textos, el numero de palabras distintivo en D por m y el numero de documentos en D es n. LSI se inicia con una matriz-documento A de m x n. Cada fila de A representa un término y cada columna representa el documento. La matriz puede ser calculada de diferentes maneras, por ejemplo, usando la frecuencia de términos o los valores de TF-IDF. Así cada entrada o celda de la matriz A, denotada por A_{ij} es el número de veces que el termino se encuentra en el documento j.

2.7.1 Descomposición de valor Singular

Que provoca que la matriz A (de MxN) sea un producto de tres matrices, es decir

$$A = U\Sigma V^T$$

Donde

- U es una matriz $m\,x\,r$ y sus columnas, denominadas vectores de derecho singular, son vectores propios asociados con r no cero vealores propios de AA^T. Por otra parte, las columnas de U son los vectores unitarios ortogonales, es decir $U^TU = I$ (matriz identidad).
- V es una matriz $n\,x\,r$ y sus columnas, denominadas vectores de derecho singular, son vectores propios asociados con r no cero, valores propios de AA^T. Las columnas de V son también vectores unitarios ortogonales, es decir $V^TV = I$
- Σ es una matriz diagonal $r\,x\,r$, Σ = diag (σ_1, σ_2, …, σ_r), σ_1> = 0, σ_1, σ_2, …, y σ_r son raíces cuadradas no negativas de los r (no negativos) , valores propios de AA^T. Están en orden decreciente, es decir $\sigma_1 \geq \sigma_2 \geq$, … , $\geq \sigma_r \geq 0$. Tomamos nota de que en un principio U es en realidad una matriz $m\,x\,n$ y V una matriz de $n\,x\,n$ y Σ una matriz diagonal $m\,x\,n$. Σ es diagonal y consta de valores propios no negativos de AA^T o A^TA. Sin embargo, debido a los valores propios ceros, Σ tiene ceros los valores de filas y columnas. La multiplicación de matrices nos dice que aquellos con valores ceros en filas y columnas en Σ se pueden quitar. Entonces, las columnas pasadas $m - r$ en U y las columnas pasadas $n - r$ en V se pueden eliminar.

m es el número de filas (términos) en A, reprentando el número de términos.

N es el número de columnas en A, representando el número de documentos.

r es el ranking de A, $r \leq \min(m, n)$.

La descomposición de valor singular de A siempre existe y es único a

1. Las permutaciones admisibles de las columnas de U y V y elementos de Σ dejados en diagonal, es decir, las columnas i y j de Σ pueden intercambiarse si y solo si la fila i y j de Σ son intercambiadas y las columnas i y j de U y V se intercambian
2. Se cambian los signos (+/-)U y V

Una característica importante en SVD es que podemos eliminar algunas dimensiones insignificantes en el concepto de transformación del espacio en forme optima (en el sentido de los mínimos cuadrados) aproximados a la matriz A. La importancia de las dimensiones indicadas por las magnitudes de los valores singulares en Σ, que ya están ordenados. En el contexto de recuperación de información, las dimensiones insignificantes pueden significar "ruido" en los datos y debe ser eliminado. Vamos a usar solo los valores de k mayor singular en E y establecer los restantes a cero. La matriz de aproximación de A se denota por A_k.

También se puede reducir el tamaño de las matrices Σ, U y Veliminando las ultimas $r-k$ filas y columnas en Σ, las ultimas $r-k$ columnas en U y las ultimas $r-k$ columnas en V. Podemos obtener

$$A_k = U_k\ \Sigma_k\ V_k^T$$

Lo que significa que usamos el k más grande a la aproximación original a los términos de la matriz A. El nuevo espacio se llama k-concepto. La figura muestra las matrices originales y las matrices reducidas de forma esquemática.

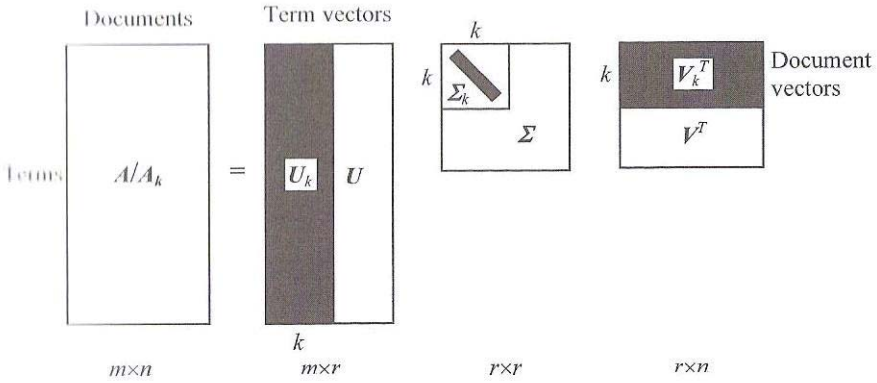

Figura 2.3. Representación semántica de A y A_k

2.7.2 Consulta y Recuperación

Dado una consulta q del usuario (representada por un vector de columnas como las de A), se convierte primero en el documento de el espacio k-concepto, denotado por q_k. Esta transformación es necesaria porque SVD se ha transformado de los documentos originales en el espacio k-concepto y almacenados en V_k. La idea es que q es tratado como un nuevo documento en el espacio original representado como una columna en A y se le asigna a q_k (un vector de filas) como un documento adicional (o columna) en V. Es fácil ver que

$$\text{q} = U_k\ \Sigma_k q_k^T$$

Donde las columnas de U son vectores unitarios ortogonales, $U_k^T\ U_k$ = I. Así

$$U_q^T\ q = \Sigma_k\ q_k^T$$

Como la inversa de una matriz diagonal es todavía una matriz diagonal y cada entrada en la diagonal es $1/\sigma_i$ (1≤ i ≤ k), si se multiplica en ambos lados de la ecuación anterior obtenemos

$$\Sigma_k^{-1}\ U_k^T q\ \ = q_k^T$$

Por último, tenemos lo siguiente

$$q_k = q^T U_k \Sigma_k^{-1}$$

Para la recuperación, simplemente q_k se compara con cada documento (fila) en V usando similitud, por ejemplo la ecuación de similitud del coseno. Este método ha sido utilizado tradicionalmente.

2.7.3 Ejemplo

Se usara el ejemplo ubicado en [9] para ilustrar el proceso. La colección de documentos incluye los siguientes nueve documentos. Las primeras cinco están relacionadas con la interacción persona computador y los últimos cuatro documentos están relacionados con los gráficos. Para reducir el tamaño del problema, solo los términos subrayados son utilizados en nuestros cálculos.

c_1 : Human machine interface for Lab ABC computer applicarions

c_2: A survey of user opinión of computer system response time

c_3: The EPS user interface management system

c_4: System and human system engineering testing of EPS

c_5: Relation of user-perceived response time to error measurement

m_1: The generation of random, binary, unordered tres

m_2: The intersection graph of paths in tres

m_3: Graph minors IV: Widths of tres and well-quasi-ordering

m_4: Graph minors: A survey

Los términos de los documentos dentro de la matriz es la siguiente, donde la matriz es de 9x12

$$A = \begin{array}{c} \begin{matrix} c_1 & c_2 & c_3 & c_4 & c_5 & m_1 & m_2 & m_3 & m_4 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array} \begin{array}{l} \\ \textit{human} \\ \textit{interface} \\ \textit{computer} \\ \textit{user} \\ \textit{system} \\ \textit{response} \\ \textit{time} \\ \textit{EPS} \\ \textit{survey} \\ \textit{trees} \\ \textit{graph} \\ \textit{minors} \end{array}$$

Después de realizar SVD, se obtienen tres matrices U, Σ y V^T, que se muestran a continuación. Los valores singulares de la diagonal de Σ son en orden decreciente.

$$U = \begin{pmatrix} 0.22 & -0.11 & 0.29 & -0.41 & -0.11 & -0.34 & 0.52 & -0.06 & -0.41 \\ 0.20 & -0.07 & 0.14 & -0.55 & 0.28 & 0.50 & -0.07 & -0.01 & -0.11 \\ 0.24 & 0.04 & -0.16 & -0.59 & -0.11 & -0.25 & -0.30 & 0.06 & 0.49 \\ 0.40 & 0.06 & -0.34 & 0.10 & 0.33 & 0.38 & 0.00 & 0.00 & 0.01 \\ 0.64 & -0.17 & 0.36 & 0.33 & -0.16 & -0.21 & -0.17 & 0.03 & 0.27 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.27 & 0.11 & -0.43 & 0.07 & 0.08 & -0.17 & 0.28 & -0.02 & -0.05 \\ 0.30 & -0.14 & 0.33 & 0.19 & 0.11 & 0.27 & 0.03 & -0.02 & -0.17 \\ 0.21 & 0.27 & -0.18 & -0.03 & -0.54 & 0.08 & -0.47 & -0.04 & -0.58 \\ 0.01 & 0.49 & 0.23 & 0.03 & 0.59 & -0.39 & -0.29 & 0.25 & -0.23 \\ 0.04 & 0.62 & 0.22 & 0.00 & -0.07 & 0.11 & 0.16 & -0.68 & 0.23 \\ 0.03 & 0.45 & 0.14 & -0.01 & -0.30 & 0.28 & 0.34 & 0.68 & 0.18 \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} 3.34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.54 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.35 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.64 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.56 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.36 \end{pmatrix}$$

$$V^T = \begin{pmatrix} 0.20 & -0.06 & 0.11 & -0.95 & 0.05 & -0.08 & 0.18 & -0.01 & -0.06 \\ 0.61 & 0.17 & -0.50 & -0.03 & -0.21 & -0.26 & -0.43 & 0.05 & 0.24 \\ 0.46 & -0.13 & 0.21 & 0.04 & 0.38 & 0.72 & -0.24 & 0.01 & 0.02 \\ 0.54 & -0.23 & 0.57 & 0.27 & -0.21 & -0.37 & 0.26 & -0.02 & -0.08 \\ 0.28 & 0.11 & -0.51 & 0.15 & 0.33 & 0.03 & 0.67 & -0.06 & -0.26 \\ 0.00 & 0.19 & 0.10 & 0.02 & 0.39 & -0.30 & -0.34 & 0.45 & -0.62 \\ 0.01 & 0.44 & 0.19 & 0.02 & 0.35 & -0.21 & -0.15 & -0.76 & 0.02 \\ 0.02 & 0.62 & 0.25 & 0.01 & 0.15 & 0.00 & 0.25 & 0.45 & 0.52 \\ 0.08 & 0.53 & 0.08 & -0.03 & -0.60 & 0.36 & 0.04 & -0.07 & -0.45 \end{pmatrix}$$

Ahora vamos a elegir solo los valores más singulares de Σ, es decir k = 2. Así, el concepto de espacio solo tiene dos dimensiones. Las otras dos matrices también están truncadas.

$$A_k = \begin{matrix} U_k \\ \begin{pmatrix} 0.22 & -0.11 \\ 0.20 & -0.07 \\ 0.24 & 0.04 \\ 0.40 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.30 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{pmatrix} \end{matrix} \begin{matrix} \Sigma_k \\ \begin{pmatrix} 3.34 & 0 \\ 0 & 2.54 \end{pmatrix} \end{matrix} \begin{matrix} V_k^T \\ \begin{pmatrix} 0.20 & 0.61 & 0.46 & 0.54 & 0.28 & 0.00 & 0.02 & 0.02 & 0.08 \\ -0.06 & 0.17 & -0.13 & -0.23 & 0.11 & 0.19 & 0.44 & 0.62 & 0.53 \end{pmatrix} \end{matrix}$$

Ahora que la debemos trabajar la consulta q de búsqueda, para obtener los documentos pertinentes. El documento de consulta transformada q_k de la consulta q en el k-concepto se calcula de la siguiente manera.

$$q_k = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}^T \begin{pmatrix} 0.22 & -0.11 \\ 0.20 & -0.07 \\ 0.24 & 0.04 \\ 0.40 & 0.06 \\ 0.64 & -0.17 \\ 0.27 & 0.11 \\ 0.27 & 0.11 \\ 0.30 & -0.14 \\ 0.21 & 0.27 \\ 0.01 & 0.49 \\ 0.04 & 0.62 \\ 0.03 & 0.45 \end{pmatrix} \begin{bmatrix} 3.34 & 0 \\ 0 & 2.54 \end{bmatrix}^{-1} = (0.179 \quad -0.004)$$

q_k se compara con cada vector - documento V_k utilizando la similitud del coseno. Los valores de similitud son los siguientes:

c_1: 0.964

c_2: 0.957

c_3: 0.968

c_4: 0.928

c_5: 0.922

m_1: - 0.022

m_2: 0.023

m_3: 0.010

m_4: 0.127

Donde obtenemos el ranking final como (c_3, c_1 , c_2 , c_4, c_5, m_4, m_2, m_3, m_1)

3 MAQUINAS DE APRENDIZAJE

3.1 Presentación

Con el avance de las tecnologías en computación, tenemos la posibilidad de procesar una gran cantidad de datos. Por ejemplo una tienda comercial almacena todas las compras y datos de los consumidores en cada uno de sus sucursales. Ellos saben exactamente qué persona compro cada uno de los productos y puede ver los gustos de cada una de las personas filtrando por edad, sexo, etc. Con toda esta información podríamos construir por ejemplo comportamientos de un consumidor y así no ofrecerle cosas completamente al azar. No podemos identificar completamente los procesos pero si podríamos construir una muy buena aproximación. Identificar los procesos completamente no es posible, pero podríamos detectar una cierta regularidad en nuestros modelos. Este es el nicho de Machine Learning, entender procesos o poder crear predicciones a partir de los patrones encontrados, asumiendo que en el futuro no variara mucho de los ejemplos obtenidos en los datos del pasado. Aplicaciones de Machine Learning a gran cantidad de información es llamado Data Mining. Estas aplicaciones son abundantes, por ejemplo en finanzas los bancos analizan datos del pasado para construir modelos a usar en aplicaciones de créditos, detectar fraudes. En medicina programas de aprendizaje son usados para diagnósticos médicos. Machine Learning es una parte de la Inteligencia Artificial, ya que los sistemas deben tener la habilidad de aprender y adaptarse a los distintos cambios [11].

Ejemplos de Aplicaciones en Machine Learning:

En el caso de retail, por ejemplo, para una cadena de supermercado, una aplicación de machine learning sería analizar la canasta. Buscando asociaciones entre productos comprado por consumidores. Si una persona compra un producto X, posiblemente pueda comprar un producto Y entonces elegimos puestos objetivos para suplir esas necesidades.

Para encontrar estas reglas de asociación, debemos aprender una probabilidad condicional de la forma P(Y|X) donde Y es el producto que nos gustaría sobre la condición X, que es el producto o el set de productos que sabemos el consumidor ya ha comprado. Nosotros sobre los datos podemos calcular P(papas fritas | cerveza) = 0.7 con esto podemos definir la siguiente regla: El 70 por ciento de los consumidores que compran cerveza también compran papas fritas. Podemos hacer distinción entre los consumidores, P(Y | X, D) donde D es el set de atributos del consumidor, por ejemplo, genero, edad, estado civil, etc. asumiendo que uno tiene acceso a toda esta información. En el caso de un portal web que ofrezca estos servicios podríamos ofrecer solo los productos que le interesan a estimado tipo de usuario y tener un mejor acceso a sus compras.

Un crédito es una cantidad de dinero prestado por una institución financiera, por ejemplo un banco presta dinero con intereses, generalmente en cuotas. Es de vital importancia para el banco poder predecir el riesgo asociado con el préstamo, donde el riesgo es la probabilidad de que la persona no devuelva el préstamo. El banco calcula el riesgo dado los datos del crédito y la información sobre el consumidor. La información sobre el consumidor incluye datos a los cuales tenemos acceso y que será relevante para poder calcular el riesgo o su capacidad financiera (ingresos, ahorros, profesión, edad, pasado financiero y así sucesivamente. Desde

esta información se infiere en una regla general que asocia entre los atributos del consumidor y los riesgos. Es un sistema de machine learning el que busca un modelo con los datos del pasado capaz de calcular el riesgo y decidir si acepta o no el crédito. Este es un ejemplo de clasificación de problema donde se tienen dos clases, bajo riesgo y alto riesgo. La información del cliente son los datos de entrada para poder clasificar y asignar en una de estas dos clases. Antes del entrenamiento con los datos del pasado, una regla de regla de aprendizaje puede ser de la forma:

IF sueldo > Θ1 Y ahorro > Θ2 ENTONCES bajo riego ELSE alto riesgo

Este es un ejemplo de discriminante, donde separa los ejemplos de diferentes clases.

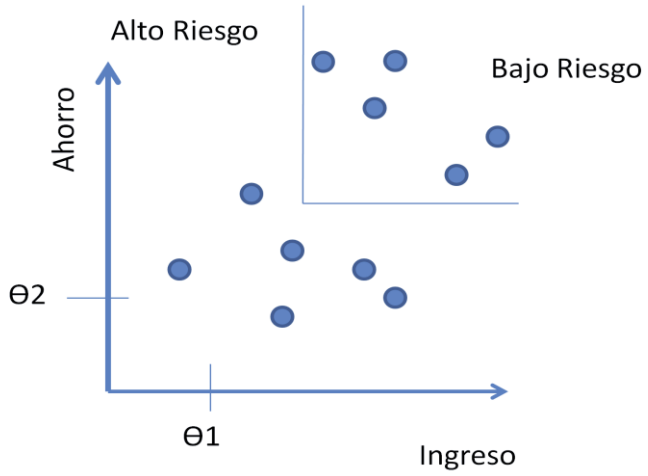

Figura 3.1. División en dos clases de set de datos

Teniendo estas reglas la aplicación es predictivo, nosotros tenemos estas reglas según los datos del pasado, si el futuro es similar al pasado, entonces podríamos hacer correctas predicciones para nuevas instancias.

Las reglas de conocimiento desde los datos también permite la extracción de conocimiento. La regla es un simple modelo que explica los datos y mirando ese modelo nosotros tenemos una explicación más simple que los datos, requiriendo almacenar menor memoria y menores procesos computacionales. Entonces tenemos las reglas de adición y no necesitamos recordar la totalidad de todos los posibles pares de números.

Otro uso de machine learning es detectar la parte aislada, donde buscamos la instancia que no cumple las reglas y es excepción. Donde puede implicar una anomalía y puede requerir atención, por ejemplo un fraude.

Si queremos tener un sistema que haga una predicción del precio de un auto usado, las entradas son los atributos del auto, marca, año, capacidad del motor, kilometraje y otra información que nosotros creamos que puede afectar en el precio del auto, la salida será el precio del auto. Este problema donde la salida es número es un problema de regresión. X denota los atributos del auto e Y el precio del auto. La función puede ser de la forma:

$$y = wx + w0$$

para valores adecuados de w y w0. Regresión y clasificación son problemas de aprendizaje supervisado, donde X es una entrada e Y es una salida y la tarea es aprender la asignación desde la entrada a la salida. El enfoque en machine learning es que nosotros asumimos un modelo definido sobre un set de parámetros

$$y = g(x \mid \Theta)$$

Donde g(.) es el modelo y Θ son sus parámetros. Y es un numero en regresión y es una clase de código, por ejemplo 0/1 en el caso de clasificación. G(.) es la función de regresión o clasificación, es la función discriminante deparando las distintas instancias de clases. Machine learning optimiza los parámetros Θ, de tal manera que el error sea mínimo.

3.2 Máquinas de Soporte Vectorial

3.2.1 Presentación

La teoría de las Maquinas de Soporte Vectorial (SVM) es una técnica de clasificación y está basada en la idea de minimización de riesgo estructural. En muchas aplicaciones, las SVM han mostrado tener un gran desempeño, más que las maquinas de aprendizaje tradicional como las redes neuronales y han sido introducidas como herramientas poderosas para resolver problemas de clasificación. Una SVM primero mapea los puntos de entrada a un espacio de características de una dimensión mayor (por ejemplo si los puntos de entrada están en R^2 entonces son mapeados por la SVM a R^3) y encuentra un hyperplano que los separe y maximice el margen *m* entre las clases en este espacio.

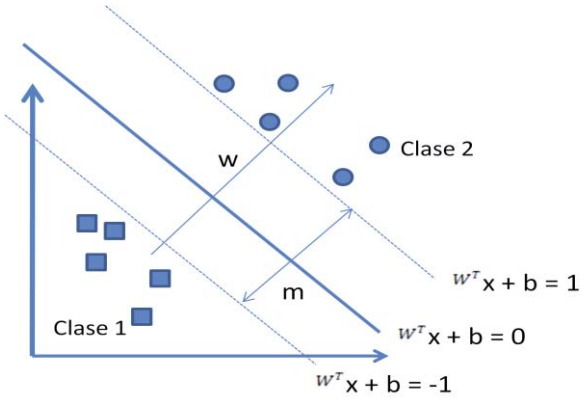

Figura 3.2. Separaciones del hyperplano y márgenes de SVM

Maximizar el margen *m* es un problema de programación cuadrática (QP) y puede ser resuelto por su problema dual introduciendo multiplicadores de Lagrange. Sin ningún conocimiento del mapeo, la SVM encuentra el hyperplano óptimo utilizando el producto punto con funciones en el espacio de características que son llamadas kernels. La solución del

hyperplano óptimo puede ser escrita como la combinación de unos pocos puntos de entrada que son llamados vectores de soporte.

3.2.2 Caso Linealmente Separable

Supongamos que nos dan un conjunto S de puntos etiquetados para entrenamiento como muestra la figura

$$(y_1 , x_1) , \dots , (y_i , x_i) \qquad (1)$$

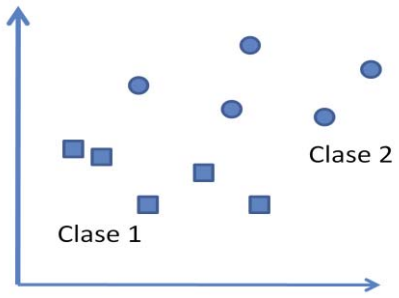

Figura 3.3. Caso de dos clases linealmente separables

Cada punto de entrenamiento $x_1 \in R^n$ pertenece a alguna de las dos clases y se le ha dado una etiqueta $y_1 \in \{-1, 1\}$ para i = 1,.....,l. En la mayoría de los casos, la búsqueda de un hyperplano adecuado en un espacio de entrada es demasiado restrictiva para ser de uso práctico. Una solución a esta situación es mapear el espacio de entrada en un espacio de características de una dimensión mayor y buscar el hyperplano óptimo allí. Sea z = Φ(x) la notación del correspondiente vector en el espacio de características Z. Deseamos encontrar el hyperplano

$$w \cdot z + b = 0 \qquad (2)$$

Definido por el par (w , b) tal que podamos separar el punto x_1 de acuerdo a la función

$$f(x_i) = \text{sign}(w \cdot z_i + b) = \begin{cases} 1 & y_i = 1 \\ -1 & y_i = -1 \end{cases} \qquad (3)$$

Donde $w \in Z$ y $b \in \Re$. Mas precisamente, el conjunto S se dice que es linealmente separable si existe (w , b) tal que las inecuaciones

$$\begin{cases} (w \cdot z_i + b) \geq 1 \ , \ y_i = 1 \\ (w \cdot z_i + b) \leq -1 \ , \ y_i = -1 \end{cases} \quad i = 1,\dots,l \qquad (4)$$

Sean válidas para todos los elementos del conjunto S. para el caso linealmente separable de S, podemos encontrar un único hyperplano optimo, para el cual, el margen entre las proyecciones de los puntos de entrenamiento de dos diferentes clases es maximizado.

3.2.3 Caso linealmente no separable

Si el conjunto S no es linealmente separable, violaciones a la clasificación deben ser permitidas en la formulación de la SVM

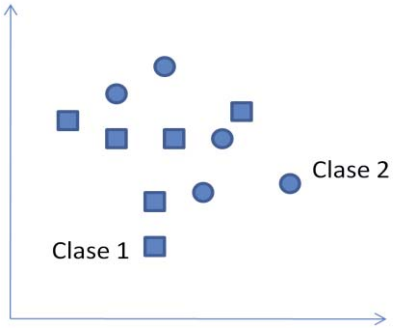

Figura 3.4. Caso de dos clases linealmente no separables

Para tratar con datos que no son linealmente separables, el análisis previo puede ser generalizado introduciendo algunas variables no-negativas $\xi_i \geq 0$ de tal modo que la ecuación (4) es modificada a

$$y_i \, (w \cdot z_i + b) \geq 1 - \xi_i \, , \quad i = 1,\ldots,l \qquad (5)$$

Los $\xi_i \neq 0$ en (5) son aquellos para los cuales el punto x_i no satisface (4). Entonces el término $\sum_{i=1}^{l} \xi_i$ puede ser tomado como algún tipo de medida del error en la clasificación.

El problema del hyperplano óptimo es entonces redefinido como la solución al problema

$$\min \left\{ \frac{1}{2} \, w \cdot w + C \, \sum_{i=1}^{l} \xi_i \right\}$$

$$\text{s.a} \quad y_i \, (w \cdot z_i + b) \geq 1 - \xi_i \, , \quad i = 1,\ldots,l \qquad (6)$$

$$\xi_i \geq 0 \, , \quad i = 1,\ldots,l$$

Donde C es una constante. El parámetro C puede ser definido como un parámetro de regularización. Este es el único parámetro libre de ser ajustado en la formulación de SVM. El ajuste de este parámetro puede hacer un balance entre la maximización del margen y la violación a la clasificación.

Buscando el hyperplano óptimo en (6) es un problema QP, que puede ser resuelto construyendo un Lagrangiano y transformándolo en el dual

$$\text{Max } W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \; \alpha_j \; y_i \; y_j \; z_i \cdot z_j \qquad (7)$$

$$\text{s.a} \sum_{i=1}^{l} \alpha_i = 0 \, , \; 0 \leq \alpha_i \leq C \, , \quad i = 1,\ldots,l$$

Donde $\alpha = (\alpha_1,\ldots, \alpha_l)$ es un vector de multiplicadores de Lagrange positivos asociados con las constantes en (5).

El teorema de Khun-Tucker juega un papel importante en la teoría de las SVM. De acuerdo a este teorema, la solución $\overline{\alpha_i}$ del problema (7) satisface:

$$\overline{\alpha_i} \, (\, y_i \, (\, \overline{w} \cdot z_i + \overline{b} \,) - 1 + \overline{\xi_i} \,) = 0 \, , \quad \text{í} = 1,\ldots,l \qquad (8)$$

$$(C - \overline{\alpha_i} \,) \, \xi_i = 0 \, , \quad \text{í} = 1,\ldots,l \qquad (9)$$

De esta igualdad de deduce que los únicos valores $\overline{\alpha_i} \neq 0$ (9) son aquellos que para las constantes en (5) son satisfechas con el signo de igualdad. El punto x_i correspondiente con $\overline{\alpha_i} > 0$ es llamado *vector de soporte*. Pero hay dos tipos de vectores de soporte en un caso no separable. En el caso $0 < \overline{\alpha_i} < C$, el correspondiente vector de soporte x_i satisface las igualdades $y_i \, (\, \overline{w} \cdot z_i + \overline{b} \,) = 1$ y $\xi_i = 0$. En el caso $\overline{\alpha_i} = C$, el correspondiente ξ_i es diferente de cero y el correspondiente vector de soporte x_i no satisface (4). Nos referimos a estos vectores de soporte como errores. El punto x_i correspondiente con $\overline{\alpha_i} = 0$ es clasificado correctamente y está claramente alejado del margen de decisión.

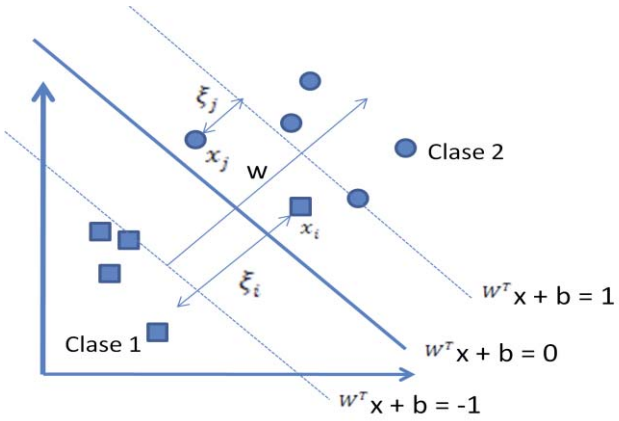

Figura 3.5. Caso de dos clases linealmente no separables con puntos de error

Para construir el hyperplano optimo $\overline{w} \cdot z + \overline{b}$, se utiliza

$$\overline{w} = \sum_{i=1}^{l} \overline{\alpha_i} \, y_i \, z_i \qquad (10)$$

Y el escalar *b* puede ser determinado de las condiciones de Kuhn-Tucker (9). La función de designación generalizada de (3) y (10) es tal que

$$f(x) = \text{sign} \, (\, w \cdot z + b \,) = \text{sign} \left(\sum_{i=1}^{l} \alpha_i \, y_i \, z_i \cdot z + b \right) \qquad (11)$$

3.2.4 Truco del Kernel para el caso linealmente no separable

Las formulaciones de SVM discutidas hasta ahora requieren que los ejemplos positivos y negativos puedan ser separados linealmente, es decir, el límite de decisión debe ser un hiperplano. Sin embargo, para muchos set de datos de la vida real, los límites de decisiones no son lineales. Para hacer frente a los datos linealmente no separables, la misma fórmula y técnica de solución que para el caso lineal se siguen utilizando. Solo transformar los datos de

entrada en su espacio original a otro espacio (generalmente de un espacio tridimensional mucho más alto) para que un límite de decisión lineal pueda separar ejemplos positivos y negativos en el espacio transformado, el que se llama "espacio de características". El espacio de datos original se llama "espacio de entrada". Así, la idea básica es que el mapa de datos en el espacio X de entrada a un espacio de características F, a través de un mapeo no lineal φ,

$$\Phi : X \rightarrow F$$

$$X \rightarrow \varphi(x)$$

Después del mapeo, los datos de entrenamiento originales $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ se convierte en:

$$\{(\phi(x_1), y_1), (\phi(x_2), y_2), \ldots, (\varphi(x_n), y_n)\}$$

El mismo método de SVM de solución lineal se aplica a F. la Figura ilustra el proceso. En el espacio de entrada (figura del lado izquierdo) los ejemplos de formación no pueden ser linealmente separados. En el espacio que aparece transformado (figura de la derecha) pueden ser separados linealmente.

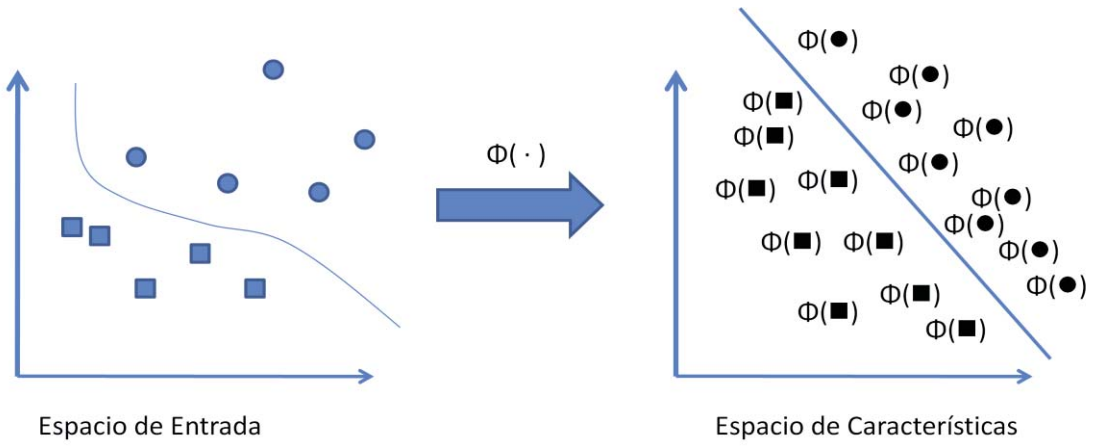

Figura 3.6. Transformación sobre el espacio de entrada al espacio de características

Con la transformación, el problema de optimización se convierte en

$$\text{Mínimo: } \frac{\langle w \cdot w \rangle}{2} + C \sum_{i=1}^{n} \xi_i$$

$$\text{Sujeto a: } y_i(\langle w \cdot \phi(x_i) \rangle + b) \geq 1 - \xi_i \,, i = 1, 2, \ldots, n$$

$$\xi_i \geq 0 \,, i = 1, 2, \ldots, n$$

Y su correspondiente doble es

$$\text{Máximo: } L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i \, y_j \, \alpha_i \, \alpha_j \langle \phi(x_i) \cdot \phi(x_j) \rangle$$

$$\text{Sujeto a: } \sum_{i=1}^{n} y_i \, \alpha_i = 0$$

$$0 \leq \alpha_i \leq C \,, i = 1, 2, \ldots, n$$

La regla de decisión final para la clasificación (prueba) es

$$\sum_{i=1}^{n} y_i \, \alpha_i \, \langle \phi \, (x_i) \cdot \, \phi \, (x) \rangle + b$$

El problema con este enfoque potencia de transformar los datos de entrada de forma explícita a un espacio de características. El número de dimensiones en el espacio de características puede ser enorme, con algunas transformaciones útiles, incluso con un número razonable de los atributos en el espacio de entrada. Esto hace que sea computacionalmente imposible de manejar.

Afortunadamente, las transformaciones explicitas se pueden evitar si nos damos cuenta de que en la doble representación, tanto la construcción del hiperplano optimo en F y la evaluación de la correspondiente función decisión/clasificación solo requiere la evaluación del producto escalar $\langle \phi(x) \cdot \phi(z) \rangle$ y nunca el vector $\phi(x)$ se asigna en su forma explícita. Este es un punto crucial. Así, tenemos una forma de calcular el producto escalar $\langle \phi(x) \cdot \phi(z) \rangle$ en el espacio de características F utilizando los vectores de entrada x y z, entonces no sería necesario conocer el vector de características $\phi(x)$ o incluso la función de mapeo ϕ. En SVM, esto de hace mediante el uso de "la función Kernel", que se denomina como K,

$$K(x, z) = \langle \phi(x) \cdot \phi(z) \rangle$$

Que son exactamente las funciones para el cálculo de productos punto en el espacio de características transformado mediante vectores de entrada x y z. Un ejemplo de una función Kernel es "polynomial kernel",

$$K(x, z) = \langle x \cdot z \rangle^d$$

Se puede sustituir todos los productos punto $\langle \phi(x) \cdot \phi(z) \rangle$ con la función del núcleo K(x, z). Esta estrategia de utilizar directamente una función de núcleo para sustituir productos punto en el espacio de función se llama "kernel trick". Donde nunca tendría que saber explícitamente lo que es ϕ.

Sin embargo, la pregunta es saber como sabemos si una funciones un kernel sin realizar una derivación. Es decir, saber como sabemos que la función del kernel es de hecho un producto punto de un espacio de características. Esta pregunta es contestada por "Mercer´s theorem" [12].

Es claro que la idea del kernel generaliza el producto punto en el espacio de entrada. El producto punto también es un kernel con la función de mapa que está en la identidad.

$$K(x , z) = \langle x \cdot z \rangle$$

Donde comúnmente los más utilizados son:

Polynomial: $K(x, z) = (\langle x \cdot z \rangle + \theta)^d$

Gaussian RBF: $K(x, z) = e^{-||x-z||^2/2\sigma}$

Donde $\theta \in \Re$, $d \in N$ y $\sigma > 0$

3.2.5 Resumen

Las maquinas de soporte vectorial es un sistema de aprendizaje lineal que encuentra el máximo margen de decisión para separar ejemplos positivos y negativos. El aprendizaje se formula como un problema de optimización cuadrática. Los límites de decisión no lineales se encuentran a través de una transformación de los datos originales a un espacio de mucha mayor característica dimensional. Sin embargo, esta transformación no está explícitamente hecha. En cambio, las funciones del kernel se utilizan para calcular productos punto que requiere el aprendizaje sin necesidad de siquiera saber la función de transformación. Debido a la separación del algoritmo de aprendizaje y las funciones del kernel, los kernels se pueden estudiar de forma independiente del algoritmo de aprendizaje. Uno puede diseñar y experimentar con diferentes funciones del kernel, sin tocar el algoritmo de aprendizaje subyacente.

SVM también tiene lagunas limitaciones:

1. Solo funciona en el espacio de valores reales. Para un atributo categórico, tenemos que convertir sus valores categóricos a valores numéricos. Una forma de hacerlo es crear atributos binarios adicionales para cada valor categórico y establecer el valor del atributo a 1 si es valor categórico y 0 en otros casos.
2. Solo permite dos clases, es decir, clasificación binaria. Para problemas de clasificación de varias clases la estrategia puede se aplicar, por ejemplo uno contra el resto y error corrección de la codificación de salida. [13]
3. El hyperplano producido por SVM es difícil de entender por los usuarios. Es difícil de imaginar que el hyperplano es en un espacio de grandes dimensiones. El asunto se agrava por los kernels. Por lo tanto, SVM se utiliza comúnmente en aplicaciones que no requiere comprensión humana.

3.3 Naïve Bayes

3.3.1 Presentación

Es un método importante, no solo porque ofrece un análisis cualitativo de los atributos y valores que pueden intervenir en el problema, sino porque da cuenta también de la importancia cuantitativa de esos atributos. En el aspecto cualitativo podemos representar como se relacionan esos atributos ya sea de una forma causal, o señalando simplemente de la correlación que existe entre esas variables (o atributos). Cuantitativamente (y este es el gran

aporte de los métodos bayesianos), da una medida probabilística de la importancia de esas variables en el problema (y por lo tanto una probabilidad explicita de las hipótesis que se formulan). Esta es quizás una de las diferencias fundamentales que ofrecen las redes bayesianas con respecto a otros métodos, como pueden ser los arboles de decisión y las redes neuronales, que no dan una medida cuantitativa de esa clasificación. Además de estas consideraciones, el aprendizaje basado en redes bayesianas es especialmente adecuado en ciertas tareas como puede ser la clasificación de textos, siendo incluso más eficiente que los otros métodos ya señalados, y ofrece una medida para el estudio y comprensión de estos métodos.

Entre las características que poseen los métodos bayesianos en tareas de aprendizaje se pueden resaltar las siguientes:

- Cada ejemplo observado va a modificar la probabilidad de que la hipótesis formulada sea correcta (aumentándola o disminuyéndola). Es decir, una hipótesis que no concuerda con un conjunto de ejemplos más o menos grande no es desechada por completo sino que lo que harán será disminuir esa probabilidad estimada para la hipótesis.
- Estos métodos son robustos al posible ruido presente en los ejemplos de entrenamiento y a la posibilidad de tener entre esos ejemplos de entrenamiento datos incompletos o posiblemente erróneos.
- Los métodos bayesianos permiten tener en cuenta en la predicción de la hipótesis el conocimiento a priori o conocimiento del dominio en forma de probabilidades. El problema puede surgir al tener que estimar ese conocimiento estadístico sin disponer de datos suficientes. Esta dificultad ha sido estudiada por Kanehmann y Tversky, que analizaron los sesgos que se producen en los sujetos en la estimación subjetiva de las probabilidades de un suceso.

3.3.2 Clasificador de Patrones

Cualquier sistema de clasificación de patrones se basa en lo siguiente: dado un conjunto de datos (que dividiremos en dos conjuntos de entrenamiento y de test) representados por pares < atributo, valor >, el problema consiste en encontrar una función $f(x)$ (llamada hipótesis) que clasifique dichos ejemplos.

La idea es usar el teorema de Bayes en cualquier problema de aprendizaje automático (en especial los de clasificación) es que podemos estimar las probabilidades a posteriori de cualquier hipótesis consistente con el conjunto de datos de entrenamiento para así escoger la hipótesis más probable. Para estimar estas probabilidades se han propuesto numerosos algoritmos, entre los que cabe destacar el algoritmo Naïve Bayes.

3.3.3 Clasificación basada en el algoritmo Naïve Bayes

Dado un ejemplo x representado por k valores el clasificador Naïve Bayes se basa en encontrar la hipótesis más probable que describa a ese ejemplo. Si la descripción de ese ejemplo viene dada por los valores $< \alpha_1 , \alpha_2 , \dots , \alpha_n >$, la hipótesis más probable será aquella que cumpla:

$$\text{Vmap} = argmax_{vj} \in V \ P(v_j \mid \alpha_1 , \dots , \alpha_n)$$

Es decir, la probabilidad de que conocidos los valores que describen a ese ejemplo, este pertenezcan a la clase v_j (donde v_j es el valor de la función de clasificación f(x) en el conjunto finito V). Por el Teorema de Bayes:

$$\text{Vmap} = argmax_{vj} \in V \ \frac{P(\alpha_1 ,\dots,\alpha_n \mid v_j) \, p(v_j)}{P(\alpha_1 ,\dots,\alpha_n)} = argmax_{vj} \in V \ P(\alpha_1 , \dots , \alpha_n \mid v_j) \, p(v_j)$$

Podemos estimar P(v_j) contando las veces que aparece el ejemplo v_j en el conjunto de entrenamiento y dividiéndolo por el número total de ejemplos que forman este conjunto. Para estimar el termino $P(\alpha_1 , \dots , \alpha_n \mid v_j)$, es decir, las veces que para cada categoría aparecen los valores del ejemplo x, debo recorrer todo el conjunto de entrenamiento. Este cálculo resulta impracticable para un número suficientemente grande de ejemplos por lo que se hace necesario simplificar la expresión. Para ello se recurre a la hipótesis de independencia condicional con el objeto de poder factorizar la probabilidad. Esta hipótesis dice lo siguiente:

Los valores a_j que describen un atributo de un ejemplo cualquiera x son independientes entre sí conocido el valor de la categoría a la que pertenecen. Así la probabilidad de observar la conjunción de atributos a_j dada una categoría a la que pertenece justamente el producto de las probabilidades de cada valor por separado [14]:

$$\text{P}(a_1, \dots , a_n \mid v_j) = \prod_i \ P(a_i \mid v_j)$$

3.3.4 Clasificación de Texto con Naïve Bayes

La clasificación o categorización de texto es problema de los modelos de aprendizaje de clasificación de documentos marcado con clases pre definidas. Aprendiendo que los modelos se utilizan para clasificar los documentos futuros. Por ejemplo, tenemos un conjunto de artículos de noticias de las categorías Deporte, Política y Ciencia. Queremos enseñar a una clasificar que sea capaz de clasificar artículos en una de las clases ya predefinidas.

Debido al rápido crecimiento de documentos en línea, la clasificación automatizada de documentos es un problema importante. Esta sección se basa principalmente en [15].

Marco Probabilístico

El método de aprendizaje Naïve Bayes para la clasificación de texto se basa en el modelo generativo de probabilidad. Se supone que cada documento se ha generado por una distribución paramétrica regida por un conjunto de parámetros ocultos. Los datos de entrenamiento se utilizan para estimar los parámetros. Los parámetros se aplican para clasificar cada documento de prueba utilizando la regla de Bayes, mediante el cálculo de la probabilidad posterior de la distribución asociada a una clase que generó el documento dado. La clasificación se convierte en una simple cuestión de selección de la clase más probable.

El modelo generativo se basa en dos hipótesis

1. Los datos son generados por el modelo de mezcla
2. Existe una correspondencia uno a uno entre los componentes de la mezcla de clases del documento.

Un modelo de la mezcla de modelos de datos con un número de distribuciones estadísticas. Intuitivamente, cada una de las distribuciones corresponde a un conjunto de datos y los parámetros de la distribución proporcionan una descripción de la agrupación correspondiente. Cada distribución en un modelo de mezcla también se le llama "componente de la mezcla". En la figura se pueden ver dos funciones de densidad de probabilidad de dos distribuciones gaussianas que generan datos unidimensionales de dos clases, una distribución por clase, cuyo parámetros (denotados por θ_i) son la media (μ_i) y la desviación estándar (σ_i), es decir $\theta_i = (\mu_i, \sigma_i)$.

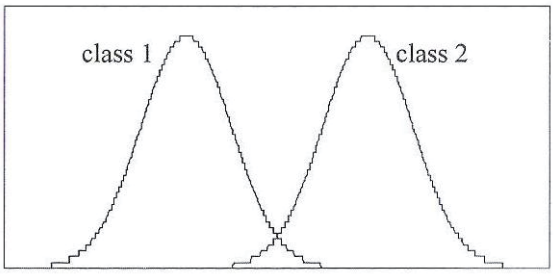

Figura 3.7. Funciones de densidad de probabilidad de dos distribuciones en el modelo de mezcla

Si el número de componentes de la mezcla de componentes en un modelo de mezcla es K y la distribución de la j-esimo distribución tiene los parámetros θ_j. Sea Θ el conjunto de parámetros de todos los componentes, $\Theta = \{\varphi_1, \varphi_2, \dots, \varphi_k, \theta_1, \theta_2, \dots, \theta_k\}$, donde φ_j es el peso de la mezcla (o probabilidad de la mezcla) de la componente de mezcla j y θ_j es el set de parámetros de componente j. El peso de la mezcla esta sujeto a la restricción $\sum_{j=1}^{k} \varphi_j = 1$.

Veremos como el modelo de mezcla genera una colección de documentos. Recordemos la clase C en nuestro problema de clasificación son $c_1, c_2, \dots, c_{|c|}$. Puesto que suponemos que existe la una correspondencia uno a uno entre los componentes de la mezcla y las clases, cada clase corresponde a un componente de la mezcla. Así |C| = K y el componente de la mezcla j-esimo puede ser representado por su clase c_j y parametrizada por θ_j. La mezcla de los pesos

son probabilidades de clases a priori es decir, φ_j = Pr(c_j|Θ). El modelo de mezcla genera cada documento d_j por:

1. Primero la selección de una componente se mezcla de acuerdo a la clase de la probabilidad a priori, es decir, el peso de la mezcla φ_j = Pr(c_j|Θ);
2. Luego de haber seleccionado este componente de mezcla c_j genera un documento d_i de acuerdo a sus parámetros, con una distribución Pr($d_i|c_j; \Theta$) o mas precisamente Pr($d_i|c_j; \Theta_j$)

La probabilidad de que un documento d_i sea generado por el modelo de mezcla puede ser escrita como la suma de probabilidades total sobre todos los componentes de la mezcla. Hay que tener en cuenta que para simplificar la notación, usamos c_j en vez de C = c_j

$$\Pr(d_i|\Theta) = \sum_{j=1}^{|C|} \Pr(c_j|\Theta) \Pr(d_i|c_j; \Theta)$$

Ya que cada uno de los documento se adjunta con su etiqueta de clase, podemos obtener el modelo Naïve Bayes para clasificación de texto. Hay que tener en cuenta que en las expresiones de probabilidad anteriores se incluye Θ para representar su dependencia de Θ que contamos con un modelo generativo. En una aplicación real, no tenemos que estar preocupados con Θ, es decir puede ser ignorado.

Modelo Naïve Bayesian

Un documento de texto consiste en una secuencia de oraciones y cada frase se compone de una secuencia y sus relaciones, varios supuestos se realizan en la derivación de clasificador bayesiano. Por esto también se le llama el modelo de clasificación final, clasificación bayesiano "ingenuo".

En concreto, la clasificación bayesiana trata cada documento como una bolsa de palabras (bag of words). El modelo generativo hace los siguientes supuestos:

1. Las palabras de un documento se generan de forma independiente del contexto, es decir, independientemente de las otras palabras del mismo documento, dada la etiqueta de clase.
2. La probabilidad de una palabra es independiente de su posición en el documento. Por ejemplo, la probabilidad de ver la palabra "estudiante" en la primera posición del documento es el mismo de verlo en cualquier otra posición. La longitud del documento es elegido independiente de su clase.

Con estos supuestos, cada documento puede considerarse como generado por una distribución multinomial. En otras palabras, cada documento se extrae de una distribución multinomial de palabras con tantas pruebas independientes como la longitud del documento.

Un ensayo multinomial es un proceso que se puede producir en cualquiera de las resultados de k, donde k $\geq$ 2. Cada uno de los resultados de un ensayo multinomial tiene la probabilidad de ocurrencia. Las probabilidades de los resultados de k se notan por p_1, p_2, …, p_k. Por ejemplo el balanceo de morir es una prueba multinomial, con seis resultados posibles 1, 2, 3, 4, 5 y 6. Para un dado $p_1 = p_2 = \ldots = p_6$ = 1/6.

Ahora supongamos que n pruebas independientes se llevan a cabo, cada uno con k posibles resultados y k probabilidades para $p_1, p_2, \ldots, p_k$. Teniendo los resultados 1, 2, 3, …, k. para cada uno de los resultados, X_t denota el numero de ensayos que dan lugar a los resultados. Entonces $X_1, X_2, \ldots, X_k$ son variables discretas aleatorias. La colección de $X_1, X_2, \ldots, X_k$ se dice que tiene una distribución multinomial con parámetros n, $p_1, p_2, \ldots, p_k$.

En nuestro contexto, n corresponde a la longitud de un documento y los resultados corresponden a todas las palabras en el vocabulario V (k = |V|). $p_1, p_2, \ldots, p_k$corresponden a las probabilidades de ocurrencia de las palabras en el documento V, que son $\Pr(w_t|c_j; \Theta)$. X_t es una variable aleatoria que representa el número de veces que la palabra w_t aparece en un documento. De este modo podemos aplicar directamente la función de probabilidad de la distribución multinomial para encontrar la probabilidad de un documento dado su clase (incluyendo la probabilidad del largo del documento, $\Pr(|d_i|)$, suponiendo que sea independiente de la clase):

$$\Pr(d_i|c_j; \Theta) = \Pr(|d_i|)\ |d_i|! \prod_{t=1}^{|V|} \frac{\Pr(w_t|c_j; \Theta)^{N_{ti}}}{N_{ti}!}$$

Donde N_{ti} es el número de veces que la palabra w_t se produce en el documento d_i y

$$\sum_{t=1}^{|V|} N_{ti} = |d_i| \ \text{ y } \ \sum_{t=1}^{|V|} \Pr(w_t|c_j; \Theta) = 1$$

El parámetro θ_j del componente generativo para cada clase c_j son las probabilidades de todas las palabras w_t en V, escrito como $\Pr(w_t|c_j; \Theta)$ y las probabilidades de longitud del documento, que son los mismos para todas las clases debido a nuestra hipótesis.

Estimación de parámetros

Los parámetros se pueden estimar a partir de la formación de datos D = $\{D_1, D_2, \ldots, D_{|C|}\}$, donde D_j es el subconjunto de datos para la clase c_j (|C| es el numero de clases). El vocabulario V es el conjunto de todas las palabras distintas en D. Hay que tener en cuenta que no necesitamos estimar la probabilidad de cada longitud del documento, ya que no se utiliza en nuestro clasificador. La estimación de Θ es escribe como $\hat{\Theta}$. Los parámetros se estiman con basado en el recuento empírico. La probabilidad estimada de la palabra w_t dada la clase c_j es simplemente el número de veces que w_t se produce en los datos de entrenamiento D_j (de la clase c_j) dividido por el número total de ocurrencias de palabras en los datos de formación de esa clase:

$$\Pr(w_t \mid c_j\ ; \hat{\Theta}\ ) = \frac{\sum_{i=1}^{|D|} N_{ti} \Pr(c_j|d_i\ )}{\sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{si} \Pr(c_j|d_i)}$$

En la ecuación anterior no se usa D_j de forma explícita. En cambio, se incluye $\Pr(c_j \mid d_i\ )$ para conseguir el mismo efecto por que $\Pr(c_j \mid d_i) = 1$ para cada documento en D_j y $\Pr(c_j \mid d_i) = 0$ para los documentos de otras clases. Una vez más, N_{ti} es el número de veces que la palabra w_t se produce en el documento d_j. Para poder manejas palabras con poca

frecuencia o que no aparecen en el conjunto de entrenamiento y que pueden aparecer en el set de pruebas tenemos que facilitar la probabilidad de evitar la probabilidad de 0 o 1. Esto es para aumentar el recuento de cada palabra distintivo con una pequeña cantidad λ ($0 \leq \lambda \leq 1$) o una fracción de una palabra tanto en el numerador como en el denominador. Por lo tanto, cualquier palabra tendrá al menos una probabilidad muy pequeña de ocurrencia.

$$\Pr(w_t \mid c_j ; \hat{\theta}) = \frac{\lambda + \sum_{i=1}^{|D|} N_{ti} \Pr(c_j \mid d_i)}{\lambda |V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N_{si} \Pr(c_j \mid d_i)}$$

Esto se llama **Lidstone Smoothing.** Cuando λ = 1, el smoothing es conocido como Laplace smoothing. Muchos experimentos han demostrado que λ < 1 funciona mejor para la clasificación de texto [16]. El mejor valor de λ para un conjunto de datos puede consultarse a través de experimentos utilizando un conjunto de validación o a través de validaciones cruzadas. Por último, las probabilidades de clases antes, que son mezclas de pesos φ_j, puede ser fácilmente calculado con los datos de entrenamiento.

$$\Pr(c_j \mid \hat{\theta}) = \frac{\sum_{i=1}^{|D|} \Pr(c_j \mid d_i)}{|D|}$$

Clasificación: Teniendo en cuenta los parámetros estimados, en el momento de la clasificación, tenemos que calcular la probabilidad de cada clase c_j para el documento de prueba d_i. Es decir, calcular la probabilidad de que en un determinado componente de la mezcla c_j generado dado el documento d_i. Tenemos:

$$\Pr(c_j \mid d_i ; \hat{\theta}) = \frac{\Pr(c_j \mid \hat{\theta}) \Pr(d_i \mid c_j ; \hat{\theta})}{\Pr(d_i \mid \hat{\theta})} = \frac{\Pr(c_j \mid \hat{\theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j ; \hat{\theta})}{\sum_{r=1}^{|C|} \Pr(c_j \mid \hat{\theta}) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j ; \hat{\theta})}$$

Donde $w_{d_i,k}$ es la palabra en la posición k del documento d_i (que es el mismo usando w_t y N_{ti}). Si el clasificador final es clasificar cada documento es una sola clase, de selecciona la clase con la probabilidad más alta posteriormente:

$$argmax_{c_j \in C} \Pr(c_j \mid d_i ; \hat{\theta})$$

Discusión

La mayoría de las suposiciones hechas por el aprendizaje naïve bayesian son violadas en la práctica. Por ejemplo, las palabras en un documento claramente no son independientes unas de otras. El supuesto modelo de mezcla uno-a-uno entre clases y componente de la mezcla no podría ser verdad, ya sea porque una clase puede contener documentos de varios temas. A pesar de tales violaciones, los investigadores han demostrado que el aprendizaje naïve bayesian produce modelos muy precisos.

El aprendizaje Naïve Bayes es también muy eficiente. Analiza los datos de entrenamiento solo una vez para calcular todas las probabilidades requeridas para la

clasificación. Puede ser utilizado como un algoritmo incremental. El modelo se puede actualizar fácilmente con los nuevos datos porque viene en las probabilidades que pueden ser convenientemente ser revisadas. Es ampliamente utilizado para la clasificación de texto. La formulación naïve bayes que se ha presentado se basa en una mezcla de distribución multinomial. Hay también una formulación basada en las distribuciones de Bernoulli que dice que cada palabra en el vocabulario es un binario, es decir, o aparece o no aparece en el documento. Por lo tanto no se considera el número de veces que aparece una palabra en el documento. Comparaciones experimentales demuestran que la formulación multinomial produce constantemente clasificadores más precisos [15]

4 SOFTWARE

4.1 Herramientas Básicas

En un proyecto de desarrollo de software es muy importante elegir las herramientas y tecnologías más adecuadas para trabajar en él, ya que estas son de gran apoyo en el desarrollo de las distintas actividades que el proyecto posee.

A continuación se detallan algunas herramientas utilizadas para el desarrollo de esta tesis:

Microsoft Word

Utilizado para la generación de documentos e informes, debido a que es un procesador de texto conocido y fácil de utilizar. Una de las ventajas que posee este procesador de texto es:

Revisión mejorada y funcionalidad de comentarios, permitiéndole comunicarse y compartir la información con más eficacia

Microsoft PowerPoint

Utilizado para la presentación del proyecto, mejora la manera de crear, presentar y colaborar en las presentaciones. Utiliza capacidades multimedia mejoradas para proporcionar presentaciones con mejor impacto.

Microsoft Visio

Ayuda en la creación de diagramas técnicos y de negocio, de modo que se visualiza de un modo sencillo sistemas complejos o de comunicación pudiendo además modificar sus procesos de negocio. Se utiliza menos tiempo diseñando, documentando y manteniendo procesos y sistemas.

Microsoft Project

Utilizado para generar la carta Gantt, utilizada en la planificación de las actividades involucradas en el proyecto. Permite programar y organizar recursos y tareas, con el fin de generar proyectos a tiempo.

Acrobat Reader Professional

Utilizado para el envió de documentación. El software Adobe Acrobat Professional es la forma avanzada para crear, controlar y enviar documentos PDF de Adobe de gran calidad de manera segura. Combina archivos electrónicos o en papel (incluso sitios Web, diseños de ingeniería y correo electrónico) en documentos PDF fiables que se pueden compartir fácilmente con otros mediante el software gratuito Adobe Reader.

4.2 Herramientas de Modelado

Lenguaje Unificado de Modelado (UML)

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema, incluyendo aspectos conceptuales tales como: procesos de negocios, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

El punto importante para notar aquí es que UML es un "lenguaje" para especificar y no un método o un proceso. UML se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir, es el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

4.3 Tecnologías

4.3.1 Motor de Base de Datos

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

4.3.2 Lenguaje de Programación PHP

PHP (Pre-Procesador de Hipertexto) es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos Web y el cual puede ser embebido en páginas HTML. La meta de este lenguaje es permitir escribir a los creadores de páginas Web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP.

PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

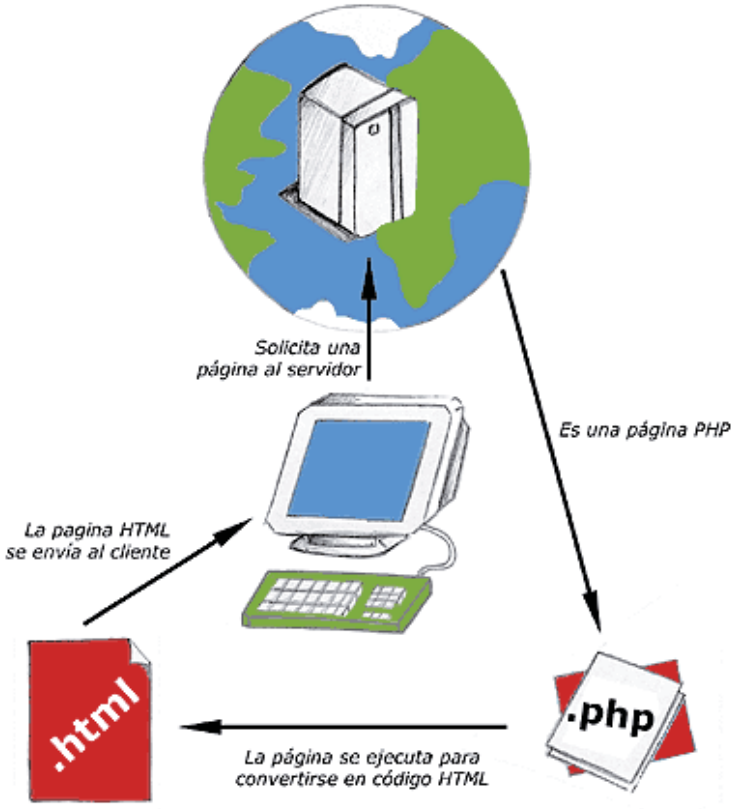

Figura 4.1. Funcionamiento de PHP

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.

Usos de PHP:

Programación de páginas dinámicas en servidores

Puede ser utilizado como lenguaje de scripting en consola.

Permite escribir aplicaciones de interfaz gráfica.

Ventajas de PHP:

Multiplataforma
Fácil de aprender
Seguridad
Funciones nativas para la mayoría de bases de datos.
Es veloz.
Es de código abierto.

Desventajas de PHP:

No existe Debugger
Legibilidad del código
Ineficiencia cuando aumentan las solicitudes.
PHP es interpretado.

JavaScript

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, orientado a las páginas web, con una sintaxis semejante a la del Lenguaje Java y el Lenguaje C.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de Herencia, es más bien dirigido por eventos. Estará listo para actuar en cuanto un evento (un clic en un botón, por ejemplo) sea ejecutado. Aún así Javascript implementa una sencilla interfaz de objetos/propiedades/métodos.

El lenguaje Javascript se integra dentro del código HTML de las páginas Web. Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el navegador al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Javascript es muy fácil de aprender para quien ya conoce lenguajes similares como el C++ o Java, pero, dada su simplicidad sintáctica y su manejabilidad, no es tampoco difícil para quien se acerca por primera vez a este lenguaje. Sin embargo, esto puede ser un arma de doble filo porque la simplicidad se basa en una disponibilidad de objetos limitada, por lo que algunos procedimientos, aparentemente muy sencillos, requieren script bastante complejos.

La característica principal de Javascript, de hecho, es la de ser un lenguaje de scripting, pero, sobre todo, la de ser el lenguaje de scripting por excelencia y, sin lugar a dudas, el más usado. Esta particularidad conlleva una notable serie de ventajas y desventajas según el uso que se le deba dar.

- Ventajas y Desventajas:

El lenguaje de scripting es seguro y fiable porque es claro y hay que interpretarlo, por lo que puede ser filtrado; para el mismo Javascript, la seguridad es casi total y sólo en su primera versión el CIAC (Computer Incident Advisory Committee) señaló problemas de leve entidad, entre ellos la lectura de la caché y de los sitios visitados, de la dirección e-mail y de los file presentes en el disco. Sin embargo, estos fallos se corrigieron ya en las versiones de Netscape sucesivas a la 2.0.

Los script tienen capacidades limitadas, por razones de seguridad, por lo cual no es posible hacer todo con Javascript, sino que es necesario usarlo conjuntamente con otros lenguajes evolucionados, posiblemente más seguros, como Java y PHP. Dicha limitación es aún más evidente si queremos operar en el hardware del ordenador, como, por ejemplo, la fijación en automático de la resolución vídeo o la impresión de un documento.

Un problema importante es que el código es visible y puede ser leído por cualquiera, incluso si está protegido con las leyes del copyright. Esto, que en mi opinión es una ventaja, representa el precio que tiene que pagar quien quiere utilizar el web: la cuestión de los derechos de autor ha asistido a una verdadera revolución con la llegada de Internet (citamos, como ejemplo más representativo, el MP3). La tutela que proporcionan las leyes actuales resulta débil e inadecuada, por lo que la única solución es tomarse las cosas con filosofía.

El código Javascript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido; un script ejecutado en el servidor, sin embargo, sometería a éste a dura prueba y los servidores de capacidades más limitadas podrían resentir de una continua solicitud por un mayor número de usuarios.

El código del script debe descargarse completamente antes de poderse ejecutar y ésta es la otra cara de la moneda de lo que se ha dicho anteriormente: si los datos que un script utiliza son muchos (por ejemplo, una recopilación de citas que se mostrará de manera casual), el tiempo que tardará en descargarse será muy largo, mientras que la interrogación de la misma base de datos en el servidor sería más rápida.

4.4 Decisión de Herramientas a Utilizar

Las decisiones de las herramientas a utilizar en la construcción del sistema, son el lenguaje PHP y base de datos MySQL.

Los argumentos para basar la decisión en PHP y MySQL, es porque son conocidas tecnologías de código abierto que resultan muy útiles para diseñar de forma rápida y eficaz aplicaciones Web dirigidas a bases de datos. PHP es un potente lenguaje de secuencia de comandos diseñado específicamente para permitir a los programadores crear aplicaciones Web con distintas prestaciones de forma rápida. MySQL es una base de datos rápida y fiable que se integra a la perfección con PHP y que resulta muy adecuada para aplicaciones dinámicas basadas en Internet.

Javascript, es un lenguaje interpretado orientado a páginas Web, dirigido por eventos, con una sintaxis similar a la de java y C. Tradicionalmente, se utiliza en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. Por lo tanto constituye una manera muy conveniente de trasladar ciertas labores de, validación de la información por ejemplo, al ordenador del cliente descargando así al servidor de esta tarea.

A continuación se lista algunas de las ventajas que también fueron consideradas en la elección de las herramientas.

Rendimiento: El rendimiento de PHP es muy bueno y verdaderamente eficiente, utilizando un servidor modesto puedes atender millones de peticiones al día. Además de ello si necesitas mejorar este rendimiento Zend Technologies ha desarrollado versiones especiales para incrementar este rendimiento.

Bajo Costo: El precio para utilizar PHP es cero, PHP es gratuito y lo puedes descargar desde www.php.net. Incluso si contratas un hosting verás que sale más barato uno con soporte PHP comparado con el que tiene soporte ASP o ASP.NET.

Es Open Source, lo puedes modificar: PHP es Open Source es decir que se tiene acceso al código fuente. Si deseas agregar o modificar algo para obtener un funcionamiento de acuerdo a tus necesidades puede hacerlo con total libertad. Esto a diferencia de las aplicaciones comerciales en las cuales solo queda esperar versiones mejoradas de la empresa desarrolladora. Este punto es importante también pues teniendo acceso al código miles de desarrolladores detectan bugs y van corrigiendo y mejorando PHP, logrando tener una aplicación muy segura y constantemente mejorada.

Librerías Incluidas: PHP fue diseñada para trabajar sobre la web por ello trae un conjunto muy amplio de funciones para ser utilizadas en diferentes tares relacionadas con la web. Se puede conectar con bases de datos, conectar a web services, parsear XML, enviar email, generar PDFs, generar imágenes, etc. Basadas en estas librerías existen clases implementadas para facilitar el trabajo de los desarrolladores. Otro punto es que hay desarrolladores que agregan librerías especializadas para extender las funcionalidades de PHP.

Portabilidad: PHP está disponible para la mayoría de sistemas operativos existentes. Desde Unix, Linux, Microsoft Windows, MAC, entre otros. Una vez desarrollado tu aplicación PHP esta puede funcionar cualquiera de estos sistemas operativos sin necesidad de modificar el código.

Soporte para OOP: La versión 5 de PHP está diseñada para soporte de características de programación orientada a objetos. Características como herencia, métodos y atributos públicos o privados, clases y métodos abstractos, constructores, interfaces y destructores. Si tienes conocimientos de C++ o Java estas características te serán muy familiares con una sintaxis muy similar.

Soporte para gran variedad de Bases de Datos: PHP tiene soporte para conectarse a una gran variedad de base de datos como: MySQL, PostgreSQL, mSQL, Oracle, dbm, FilePro, HyperWave, Informix, InterBase, Sybase entre otras. Las base de datos hacen que una aplicación sea más robusta y con este soporte tu aplicación puede conectarse con facilidad a tu base de datos existente.

4.5 Metodología, paradigma y herramienta de modelado

Para abordar de manera óptima la solución que se ha propuesto, primero se debe considerar el cómo abordar el problema y definir cuáles son los pasos a seguir que éste implicará. Además de seleccionar las herramientas que facilitarán de cierta manera el trabajo para generar la solución óptima al problema. Para esto se hará una pequeña descripción de determinadas metodologías, paradigmas y herramientas, que ayudarán a tomar la decisión sobre cuál de estas se ajustan a las necesidades.

4.6 Metodología

Para abordar la problemática que se ha presentado se ha seleccionado la metodología de Orientación a Objetos, considerando que esta es una de las más conocidas y de mayor eficiencia a la hora de enfrentar los problemas de un proyecto.

4.6.1 Orientación a Objetos

Dentro del amplio mundo de la orientación a objetos se pueden destacar características importantes tales como que es un conjunto de objetos (instancias de una clase), que corresponden a una abstracción del mundo real (problema). Un objeto se comunica con otro objeto enviando un mensaje a algún método de este último, el cual le puede retornar otro objeto como resultado.

El procesamiento del mensaje puede alterar el estado del objeto. Un Objeto tiene estado, comportamiento e identidad, significa que un objeto puede tener datos internos (los cuales le otorgan este estado), métodos (para producir comportamiento), y cada objeto puede ser únicamente distinguido de otro objeto (cada objeto tiene una única dirección en memoria).

Principales Características:

La orientación a objetos posee algunas características que hacen de esta metodología algo gratificante. Algunas de estas son:

- **Polimorfismo**, que quiere decir que la misma operación es resuelta de diferente forma según el objeto que recibe el mensaje.
- **Herencia**, es la relación que hay entre las clases de objetos que permiten incluir (rehusar) los atributos y operaciones definidas en otra clase más general de la cual se hereda o deriva.
- **Encapsulamiento,** que posibilita el ocultamiento de información y reduce el impacto asociado a cambios.

A demás posee una serie de ventajas de las que se pueden sacar provecho, algunas de estas son:

- Fomenta la reutilización y extensión del código.
- Permite crear sistemas más complejos.
- Relacionar el sistema al mundo real.
- Facilita la creación de programas visuales.
- Construcción de prototipos.
- Agiliza el desarrollo de software.
- Facilita el trabajo en equipo.
- Facilita el mantenimiento del software.

4.7 Paradigma

Para abordar el problema se ha seleccionado UP. La cual se detallará a continuación.

4.7.1 Proceso Unificado

La elección de un paradigma o modelo de proceso es tal vez una de las decisiones más importantes en un proyecto de software, y no es para menos ya que, éstos establecen las actividades necesarias para transformar los requisitos de un usuario en software. Es un paradigma, por lo tanto, una representación abstracta de un proceso de software y la utilización de un proceso inadecuado, probablemente reducirá la calidad o la utilidad del producto de software que se va a desarrollar.

Al no existir un paradigma "ideal", hay que buscar el más conveniente y que ayude a cumplir con las necesidades del proyecto.

Para este proyecto se ha escogido el proceso unificado el cual tiene como principales características:

- Es dirigido por casos de uso: son una herramienta para especificar los requerimientos del sistema y además sirven como base para guiar el resto del proceso;
- Es centrado en la arquitectura: desarrollo de la arquitectura constantemente,
- Es iterativo e incremental: divide el proyecto en mini-proyectos que mediante iteraciones van mejorando.

En base a lo expuesto anteriormente se pueden identificar algunas ventajas que fueron decisivas a la hora de la elección:

- Identificación temprana de atrasos.
- Resolución en las primeras etapas de los riesgos.
- Obtención de resultados a corto plazo
- Retroalimentación de lo hecho en iteración anterior
- Adaptación al cambio de requerimientos

Además de estas ventajas, la elección de proceso unificado se debe a la experiencia previa, a la mayor cantidad y calidad de documentación existente y a que el proceso unificado hace uso de UML.

Las fases que posee el proceso unificado son las siguientes.

- Inicio
- Elaboración.
- Construcción.
- Transición.

INICIO

Su propósito general es establecer los objetivos para el ciclo de vida del producto. Durante esta fase se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y casos de uso. Se desarrolla un plan de negocio para determinar qué recursos deben ser asignados al proyecto.

Los objetivos específicos de esta fase son:

- Establecer el ámbito del proyecto y sus límites.
- Encontrar los casos de uso críticos del sistema, los escenarios básicos que definen la funcionalidad.
- Mostrar al menos una arquitectura candidata para los escenarios principales.
- Estimar el costo en recursos y tiempo de todo el proyecto.
- Estimar los riesgos, las fuentes de incertidumbre.

El hito en esta fase finaliza con el establecimiento del ámbito del producto, e identificación de los principales riesgos y la viabilidad del proyecto.

ELABORACIÓN

Su objetivo general es plantear la arquitectura para el ciclo de vida del producto. Se construye un modelo de la arquitectura, que se desarrolla en iteraciones sucesivas hasta obtener el producto final, el sistema debe contener los casos de uso críticos que fueron identificados en la fase de inicio. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

Los objetivos específicos de esta fase son:

- Definir, validar y establecer la arquitectura.
- Completar la visión.
- Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones. Debe incluir los costos si procede.
- Demostrar que la arquitectura propuesta soportará la visión con un costo razonable y en un tiempo razonable.

El hito en la fase de elaboración finaliza con la obtención de una línea base de la arquitectura del sistema, la captura de la mayoría de los requerimientos y la reducción de los riesgos importantes así como permitir la escalabilidad del equipo del proyecto durante la fase de construcción.

CONSTRUCCIÓN

El objetivo general de esta fase es alcanzar la capacidad operacional del producto de forma incremental, a través de las sucesivas iteraciones. En esta fase todas las características, componentes, y requerimientos deben ser integrados, implementados, y probados en su totalidad, obteniendo una versión aceptable del producto comúnmente llamada versión beta.

Se hace énfasis en controlar las operaciones realizadas, administrando los recursos eficientemente, de tal forma que se optimicen los costos, los calendarios y la calidad.

Los objetivos específicos de esta fase son:

- Minimizar los costos de desarrollo mediante la optimización de recursos y evitando el tener que rehacer un trabajo o incluso desecharlo.
- Conseguir una calidad adecuada tan rápido como sea práctico.
- Conseguir versiones funcionales (alfa, beta, y otras versiones de prueba) tan rápido como sea práctico.

El hito en esta fase culmina con el desarrollo del sistema con calidad de producción y la preparación para la entrega al equipo de transición. Toda la funcionalidad debe haber sido implementada y las pruebas para el estado beta de la aplicación completadas. Si el proyecto no cumple con estos criterios de cierre, entonces la transición deberá posponerse una iteración.

TRANSICIÓN

Tiene como objetivo general entregar el producto funcional en manos de los usuarios finales una vez realizadas las pruebas de aceptación por un grupo especial de usuarios, para lo que se requerirá desarrollar nuevas versiones actualizadas del producto, entrenar a los usuarios en el manejo del sistema, completar la documentación, y en general tareas relacionadas con la configuración, instalación y usabilidad del producto.

Los objetivos específicos de esta fase son:

- Garantizar que el usuario aprenda a operar y mantener el sistema.
- Conseguir un producto final que cumpla los requerimientos esperados.

El hito en la fase de transición corresponde a haber decidido si los objetivos se cumplieron y el comienzo de otro ciclo de desarrollo. El cliente debe haber revisado y aceptado los artefactos que le han sido entregado.

4.8 Herramienta de Modelado

Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un “plano” del sistema, incluyendo aspectos conceptuales tales como: procesos de negocios, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

El punto importante para notar aquí es que UML es un "lenguaje" para especificar y no un método o un proceso. UML se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir -es el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

4.9 Desarrollo de software

Para aplicar parte del conocimiento que se ha logrado en la etapa de investigación, tanto de representación de lenguaje natural como de máquinas de aprendizaje, es indispensable construir un software el cual pueda servir tanto para la etapa de entrenamiento como la etapa de pruebas. Se manejarán dos niveles de usuarios, un administrador y un usuario normal el cual podrá hacer pruebas con los textos que ha ingresado. El fin de tener este usuario que solo pueda hacer pruebas, es que cualquier usuario con conexión a internet pueda ver si el comentario ingresado lo clasifica o no bien en positivo o negativo y a la vez alimentar la base de datos con estos textos ingresados para futuros entrenamientos por parte del administrador. Así el software puede ir escalando a ir recolectando comentarios en distintos idiomas y categorías las cuales estarán bien etiquetadas para un futuro entrenamiento de la máquina.

4.10 Casos de Uso

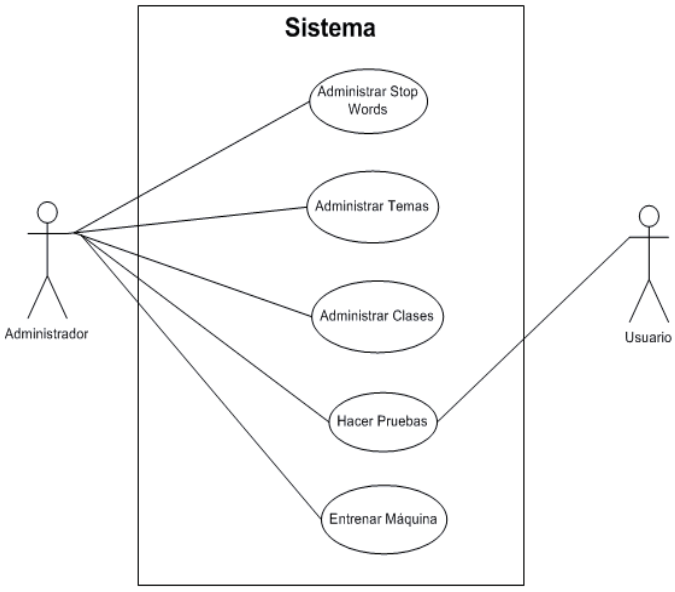

Figura 4.2. Caso de Uso de Alto Nivel

Aquí podemos ver dos tipos de usuarios que tendrán acceso al sistema, Administrador, el cual será restringido por un login y podrá administrar los Stop Words (agregar, modificar y eliminar), los temas en los cuales hay comentarios en el sistema, como películas, deporte, etc, las clases (en un primer caso, positivo y negativo), entrenar la máquina con los distintos textos almacenados en el sistema y aprobar o rechazar textos agregados por usuarios. Mientras tanto el usuario normal podrá subir textos y hacer pruebas con esos textos, con la máquina ya entrenada.

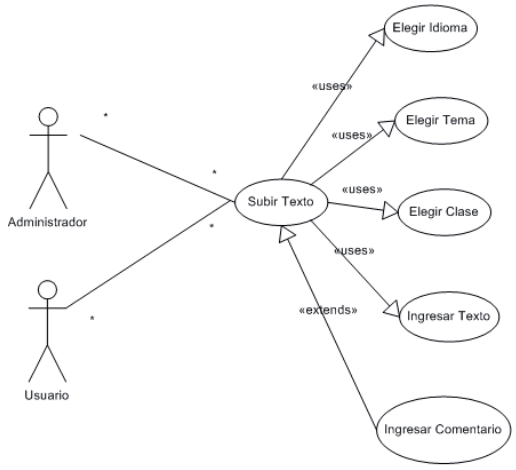

Figura 4.3. Caso de Uso subir texto

Tanto el usuario normal, como el administrador podrán subir textos, el cual lo deben etiquetar con el idioma, tema del que habla y su clase (positivo o negativo), además podrá ingresar optativamente un comentario para ese texto subido. De esta manera se irá consiguiendo una base de datos cada vez mayor con textos ya etiquetados que servirán para futuros estudios de clasificación.

Figura 4.4. Caso de Uso entrenar máquina

Para realizar un entrenamiento con textos almacenados en la base de datos el usuario debe estar registrado como administrador y podrá seleccionar entre los algoritmos SVM o Naïve Bayes, además deberá seleccionar el número de textos, el tema y el idioma que se usará. Al elegir como algoritmo SVM, se tendrán las opciones de ingresar distintos tipos de kernels y a la vez ingresar los parámetros asociados.

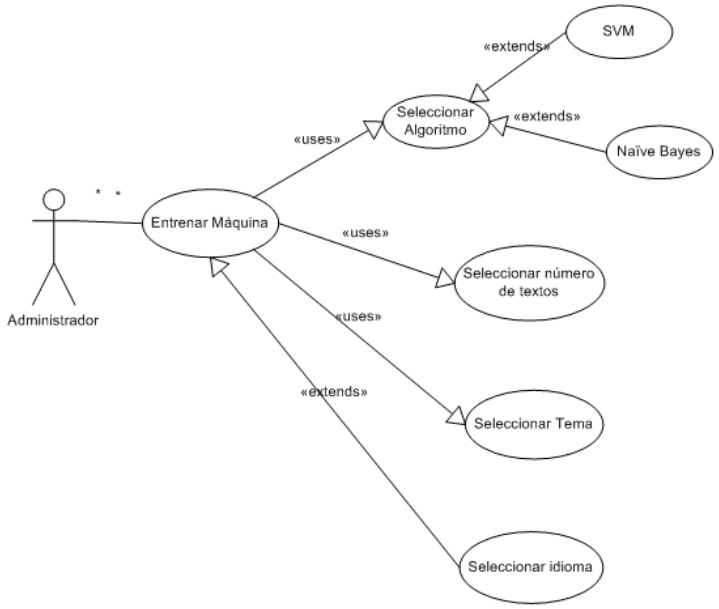

Figura 4.5. Caso de Uso Generar Modelo Automático

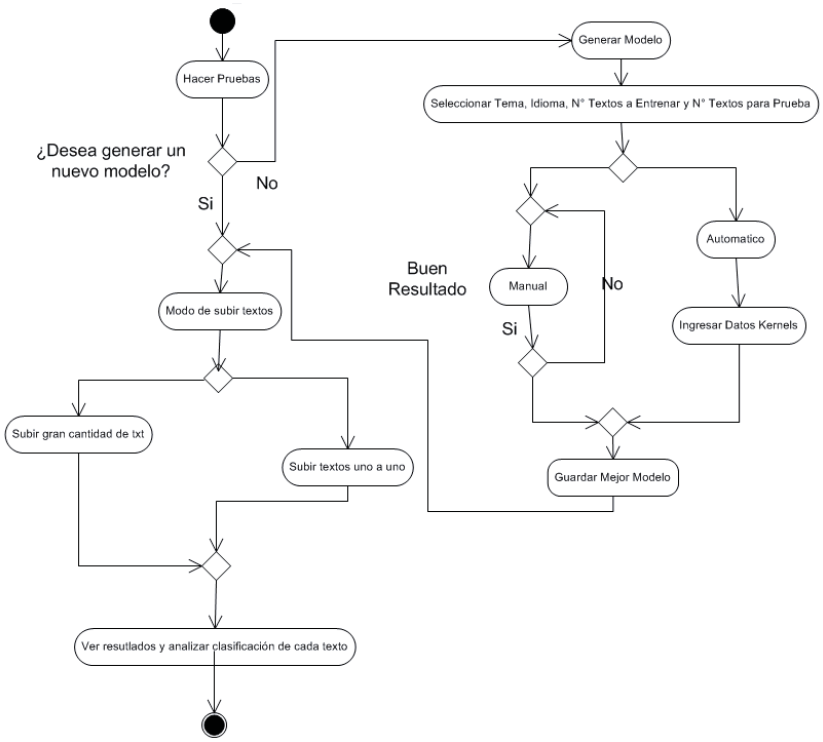

Figura 4.6. Diagrama de Actividad hacer Pruebas

4.11 Base de Datos

Una de las mayores preocupaciones a la hora de hacer un sistema web es que maneje una base de datos relacional, simple y que no se tengan que hacer consultas muy pesadas a la hora de ser ocupado el sistema.

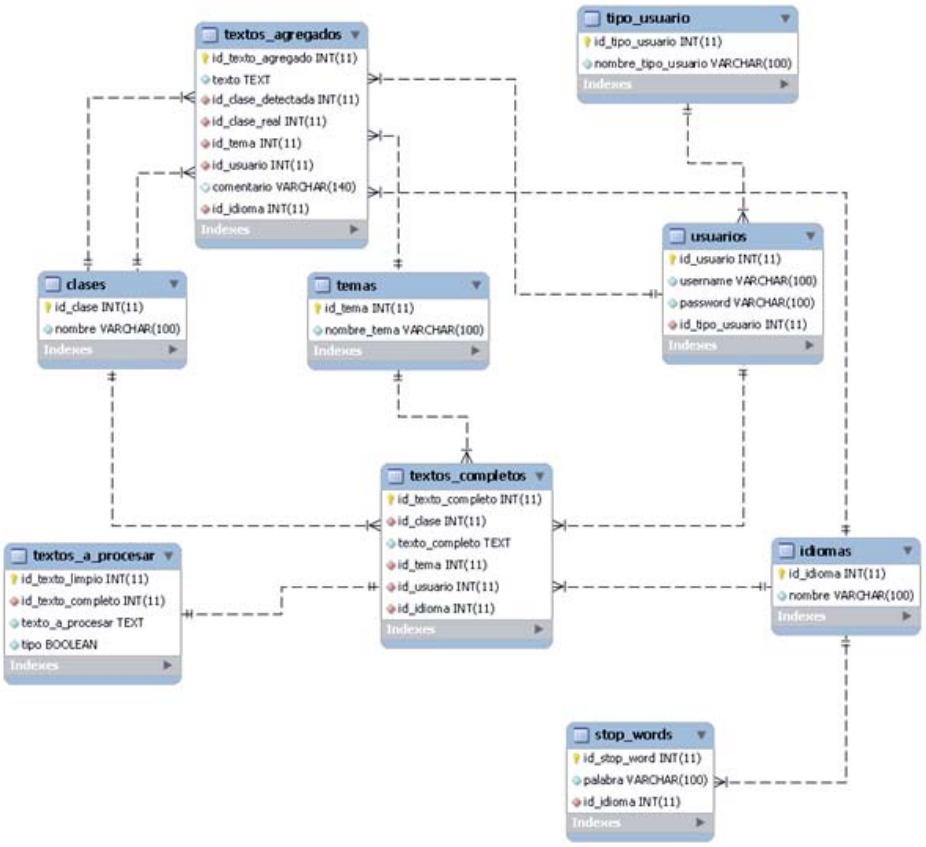

Figura 4.7. Modelo Base de Datos

4.12 Diagrama de Actividad Entrenar Máquina

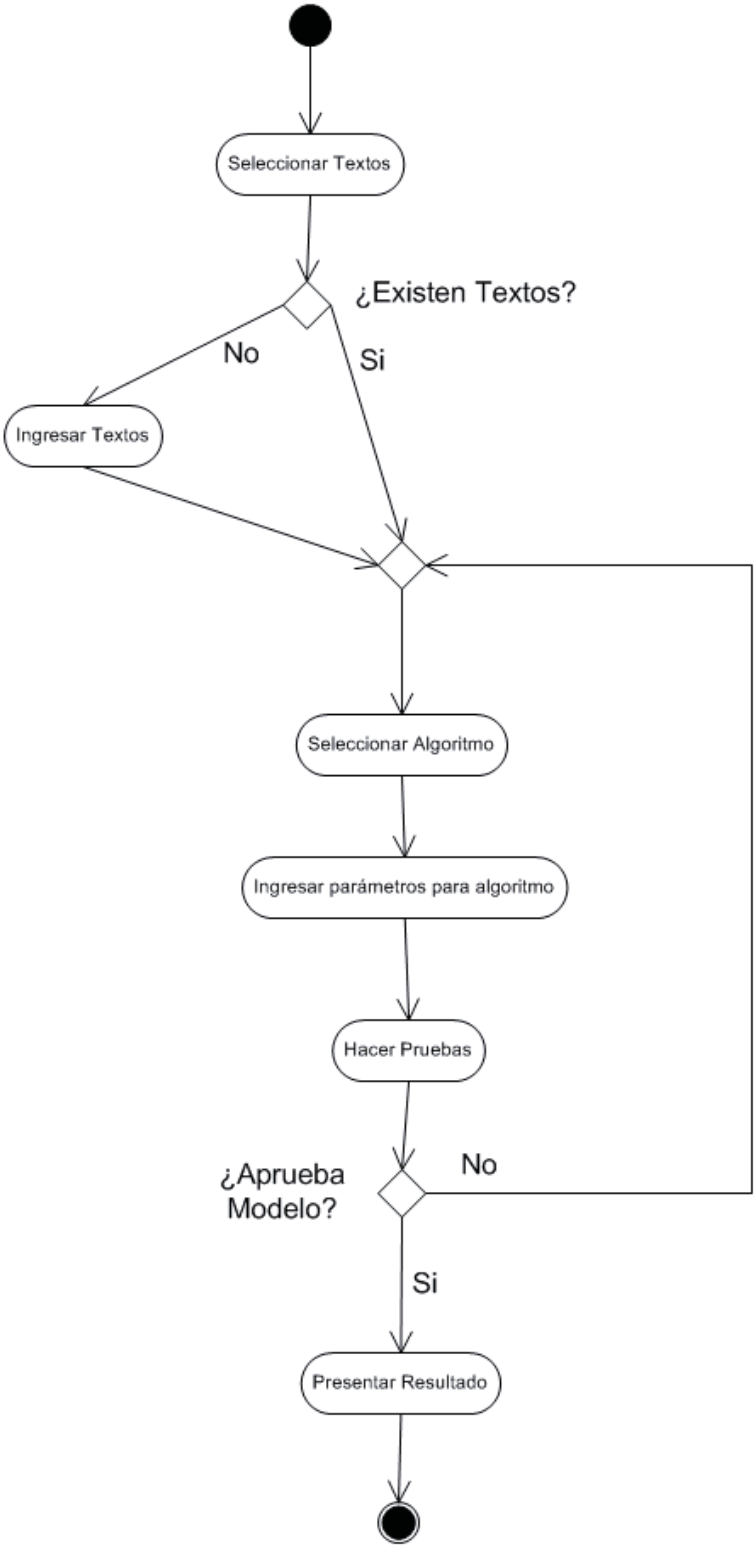

Figura 4.8. Diagrama de actividad entrenar y hacer pruebas a máquina

5 CASO DE ESTUDIO

5.1 Introducción

Hoy en día, grandes cantidades de información están disponibles en documentos en línea. Como parte del esfuerzo para organizar mejor esta información para los usuarios, los investigadores han estado trabajando activamente el problema de la categorización automática de textos. La mayor parte de esa labor se ha centrado en la categorización de actualidad, tratando de ordenar los documentos según su objeto (por ejemplo, los deportes frente a la política). Sin embargo, los últimos años han crecido rápidamente los grupos de discusión en línea y sitios de reviews, cuando una característica crucial de los artículos publicados es su sentimiento u opinión global sobre el tema, por ejemplo si una revisión del producto habla positiva o negativamente. El etiquetado de estos artículos con su sentimiento proporcionaría un valor agregado a los lectores, de hecho, estas etiquetas son parte del atractivo y el valor agregado de sitios como www.rottentomatoes.com, que etiquetan la película que no contienen indicadores explícitos de calificación y normaliza la distintos sistemas de calificación que los encuestados orienten su sentido. También sería útil en aplicaciones de inteligencia empresarial y los sistemas de recomendación, donde la entrada del usuario y la retroalimentación puede ser rápidamente resumida. Por otra parte, también hay aplicaciones potenciales para el filtrado de mensajes, por ejemplo, uno podría ser capaz de utilizar la información para reconocer el sentido y desechar comentarios que no le interesa leer. En este trabajo se examina la eficacia de la aplicación con técnicas de aprendizaje automático para el problema de clasificación de sentido. Un aspecto difícil de este problema que parece distinguirla de la clasificación tradicional basada en temas es que si bien los tópicos son a menudo identificables por palabras clave, el sentido puede ser expresado de una manera más sutil.

5.2 El dominio de comentarios sobre películas

Para la investigación, se optó por trabajar con comentarios de películas. Este dominio es experimentalmente conveniente porque hay grandes colecciones en línea y porque los usuarios suelen resumir su sentimiento general con un indicador de número extraíble por una maquina, como el número de estrellas, por lo que no se tiene necesidad de etiquetar a mano los datos para el aprendizaje supervisado o fines de evaluación. También se debe considerar que Turney (2002) encontró las críticas de películas el más difícil de varios dominios para la clasificación de sentido, la presentación de informes con una precisión de 65,83% sobre un documento de 120 términos. Pero insistimos en que la máquina de aprendizaje y las características que se utiliza no son específico de críticas de películas y debe ser fácilmente aplicable a otros ámbitos, siempre y cuando los datos de entrenamiento sean suficientes para demostrarlo. La fuente de datos fue la Internet Movie Database (IMDb.com), el cual es un archivo de la rec.arts.movies.reviews, el que es un grupo de noticias. Se seleccionaron sólo revisiones cuando la valorización del autor se expresó con estrellas o un valor numérico. Los comentarios fueron extraídos de forma automática y se convierten en una de tres categorías: positiva, negativa o neutra. Para el problema descrito en este trabajo, se concentra solamente en distinguir entre las opiniones positivas y negativas.

5.3 Una Mirada al problema

Un experto al usar máquinas de aprendizaje para la categorización de textos tiene un rendimiento relativamente bajo previsto de los métodos automáticos. Por otra parte, parece que la distinción positiva por parte de las críticas negativas es relativamente fácil para los seres humanos, especialmente en comparación con el problema de categorización de texto estándar, donde los temas pueden estar estrechamente relacionados. Y se puede sospechar que hay personas que usan ciertas palabras para expresar sentimientos fuertes, de modo que puede ser suficiente para producir una lista de palabras por la introspección y dependen de ellos para clasificar los textos. Muchos estudios indican que vale la pena explorar técnicas basadas en corpus, en lugar de confiar en las intuiciones previas, para seleccionar las características buenas de sentimiento y para realizar la clasificación en general.

5.4 Manejo del set de datos

Debido al carácter más informal de los comentarios de películas con los cuales se trabajará, es más crítico hacer una limpieza de los textos ya que se encuentra una gran cantidad de palabras que están escrita de forma distinta o con la misma raíz, pero terminada en otro tiempo.

El primer acercamiento de limpieza en los textos escogidos es aplicar Stop Words, el cual consiste en eliminar todas las palabras que son "conexiones" entre palabras (artículos, conjunciones y algunos pronombres). Para ello se utiliza la siguiente función:

```
function hacerStopWord($id_idioma) {
    $SQL = "SELECT * FROM stop_words WHERE id_idioma = ".$id_idioma."";
    $this->consulta = $this->db->consulta($SQL);
    $this->numeroFilas = $this->db->num( $this->consulta );

    for ( $x = 0 ; $x < $this->numeroFilas ; $x++ ) {
        $datos = $this->db->fetch( $this->consulta );
        $palabra = addslashes($datos['palabra']);
        $SQL3[$x] = "DELETE FROM temporal WHERE palabra = '".$palabra."'";
    }

    for ( $i = 0 ; $i < $x ; $i++ ) {
        $this->consulta = $this->db->consulta($SQL3[$i]);
    }
}
```

El cual recibe el id del idioma en el cual se está trabajando (en este caso son comentarios de películas escritas en ingles) y hace la consulta de todos los Stop Words almacenados en Base de Datos de este idioma. Si una de esas palabras es encontrada en el texto, almacenado en la tabla temporal, es borrada y no tomada en cuenta para su futuro análisis.

Luego se realiza un stemming, el cual es un proceso de reducción de las palabras a su stem (tallo) o root (raíz). El stem es la parte que queda después de la eliminación de los prefijos y sufijos. Para eso se ocupa la siguiente función:

```
function hacerStemming() {

    $SQL = "SELECT * FROM temporal";
    $this->consulta = $this->db->consulta($SQL);
    $this->numeroFilas = $this->db->num( $this->consulta );
    $i = 0;
    for ( $x = 0 ; $x < $this->numeroFilas ; $x++ ) {
        $datos = $this->db->fetch( $this->consulta );
        $sinStemming = addslashes($datos['palabra']);
        $id_temporal = addslashes($datos['id_temporal']);
        $conStemming = PorterStemmer::Stem(trim($sinStemming)); //veo si se puede hacer stemming a esa palabra
        if($conStemming!=$nose){
            //se le hizo stemming
            $SQL3[$i] = "UPDATE temporal SET palabra = '".$conStemming."' WHERE id_temporal = ".$id_temporal."";
            $i++;
        }
    }
    for ( $j = 0 ; $j < $i ; $j++ ) {
        $this->consulta = $this->db->consulta($SQL3[$j]);
    }
}
```

La cual selecciona todas las palabras del texto almacenada en la tabla temporal (habiendo o no aplicado Stop Words anteriormente) donde se analizan cada una de estas para ver si se le puede o no hacer Stemming, mediante los cuatro pasos siguientes:

```
private static function steplab($word)
{
    // Parte a
    if (substr($word, -1) == 's') {

        self::replace($word, 'sses', 'ss')
     OR self::replace($word, 'ies', 'i')
     OR self::replace($word, 'ss', 'ss')
     OR self::replace($word, 's', '');
    }

    // Parte b
    if (substr($word, -2, 1) != 'e' OR !self::replace($word, 'eed', 'ee', 0)) {
        $v = self::$regex_vowel;

        // ing and ed
        if (   preg_match("#$v+#", substr($word, 0, -3)) && self::replace($word, 'ing', '')
            OR preg_match("#$v+#", substr($word, 0, -2)) && self::replace($word, 'ed', '')) {


            if (    !self::replace($word, 'at', 'ate')
                AND !self::replace($word, 'bl', 'ble')
                AND !self::replace($word, 'iz', 'ize')) {


                if (    self::doubleConsonant($word)
                    AND substr($word, -2) != 'll'
                    AND substr($word, -2) != 'ss'
                    AND substr($word, -2) != 'zz') {

                    $word = substr($word, 0, -1);

                } else if (self::m($word) == 1 AND self::cvc($word)) {
                    $word .= 'e';
                }
            }
        }
    }

    return $word;
}
```

```
private static function step2($word)
{
    switch (substr($word, -2, 1)) {
        case 'a':
                self::replace($word, 'ational', 'ate', 0)
            OR self::replace($word, 'tional', 'tion', 0);
            break;

        case 'c':
                self::replace($word, 'enci', 'ence', 0)
            OR self::replace($word, 'anci', 'ance', 0);
            break;

        case 'e':
            self::replace($word, 'izer', 'ize', 0);
            break;

        case 'g':
            self::replace($word, 'logi', 'log', 0);
            break;

        case 'l':
                self::replace($word, 'entli', 'ent', 0)
            OR self::replace($word, 'ousli', 'ous', 0)
            OR self::replace($word, 'alli', 'al', 0)
            OR self::replace($word, 'bli', 'ble', 0)
            OR self::replace($word, 'eli', 'e', 0);
            break;

        case 'o':
                self::replace($word, 'ization', 'ize', 0)
            OR self::replace($word, 'ation', 'ate', 0)
            OR self::replace($word, 'ator', 'ate', 0);
            break;

        case 's':
                self::replace($word, 'iveness', 'ive', 0)
            OR self::replace($word, 'fulness', 'ful', 0)
            OR self::replace($word, 'ousness', 'ous', 0)
            OR self::replace($word, 'alism', 'al', 0);
            break;

        case 't':
                self::replace($word, 'biliti', 'ble', 0)
            OR self::replace($word, 'aliti', 'al', 0)
            OR self::replace($word, 'iviti', 'ive', 0);
            break;
    }

    return $word;
}
```

```php
private static function step3($word)
{
    switch (substr($word, -2, 1)) {
        case 'a':
            self::replace($word, 'ical', 'ic', 0);
            break;

        case 's':
            self::replace($word, 'ness', '', 0);
            break;

        case 't':
              self::replace($word, 'icate', 'ic', 0)
            OR self::replace($word, 'iciti', 'ic', 0);
            break;

        case 'u':
            self::replace($word, 'ful', '', 0);
            break;

        case 'v':
            self::replace($word, 'ative', '', 0);
            break;

        case 'z':
            self::replace($word, 'alize', 'al', 0);
            break;
    }

    return $word;
}
```

```
private static function step4($word)
{
    switch (substr($word, -2, 1)) {
        case 'a':
            self::replace($word, 'al', '', 1);
            break;

        case 'c':
              self::replace($word, 'ance', '', 1)
            OR self::replace($word, 'ence', '', 1);
            break;

        case 'e':
            self::replace($word, 'er', '', 1);
            break;

        case 'i':
            self::replace($word, 'ic', '', 1);
            break;

        case 'l':
              self::replace($word, 'able', '', 1)
            OR self::replace($word, 'ible', '', 1);
            break;

        case 'n':
              self::replace($word, 'ant', '', 1)
            OR self::replace($word, 'ement', '', 1)
            OR self::replace($word, 'ment', '', 1)
            OR self::replace($word, 'ent', '', 1);
            break;

        case 'o':
            if (substr($word, -4) == 'tion' OR substr($word, -4) == 'sion') {
              self::replace($word, 'ion', '', 1);
            } else {
               self::replace($word, 'ou', '', 1);
            }
            break;

        case 's':
            self::replace($word, 'ism', '', 1);
            break;

        case 't':
              self::replace($word, 'ate', '', 1)
            OR self::replace($word, 'iti', '', 1);
            break;

        case 'u':
            self::replace($word, 'ous', '', 1);
            break;

        case 'v':
            self::replace($word, 'ive', '', 1);
            break;

        case 'z':
            self::replace($word, 'ize', '', 1);
            break;
    }

    return $word;
}
```

En donde se eliminan terminaciones de sufijos y prefijos las cuales no nos interesaran para un posterior análisis. Finalmente estas palabras en su raíz son reemplazadas por la que había originalmente en el texto.

Un ejemplo de cuánto podría disminuir las palabras es el siguiente cuadro en el cual se han elegido 25 textos al azar, donde en la primera fila se aplica Stop Words mas Stemming llegando a un vector de palabras de 4.878 y en la siguiente fila se aplica solamente Stop Words donde llegamos a tener 6.050 palabras en nuestro vector. Lo cual refleja de gran manera el beneficio que entrega al asociar palaras que significan lo mismo (19.371% menos de palabras), pero están escritas en su forma prefija o sufija.

Tabla 5.1. Comparación de limpieza de textos con Stemming y Stop Words

Técnica	**Total de Palabras**
Con Stemming + Stop Words	4878
Sin Stemming + Stop Words	6050

Si uno va pre procesando más y más textos deberían llegar a un tope en que se estabiliza la cantidad de palabras recolectadas en el total de textos. Se hicieron pruebas con set de 100, 200 y 400 textos en los cuales se hizo limpieza de su corpus con Stop Words y Stemming y luego se procede a un BoW.

Tabla 5.2. Tiempos de pre-procesado y cantidad de palabras obtenidas

Cantidad de Textos	**Tiempo de Stop Words + Stemming**	**Tiempo de BoW**	**Total de Palabras**
50 pos + 50 neg	193,9 s	3.259,1 s	7.609
100 pos + 100 neg	456,0 s	11.270,9 s	11.265
200 pos + 200 neg	758,5 s	27.426,7 s	15.998

Cabe destacar que todas estas pruebas fueron hechas con un Notebook DELL Inspiron 600m

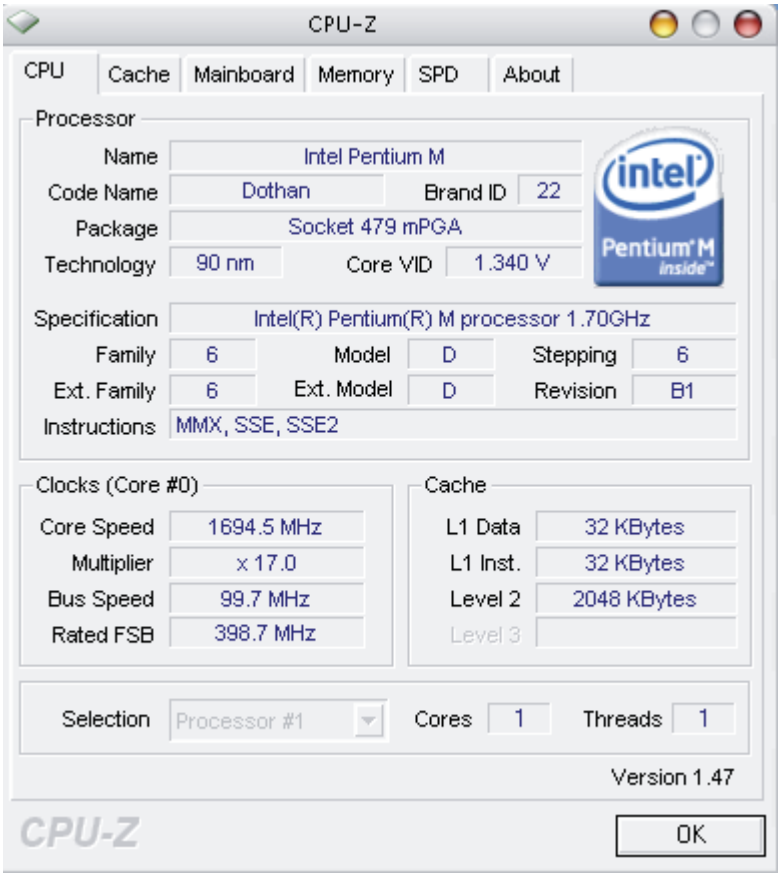

Figura 5.1. Características Notebook DELL Inspiron 600m

5.5 Vector de Palabras

Para la construcción del vector de palabras se van agregando todas las palabras encontradas en los textos ya limpios (Stop Words, Stemming) y no repitiéndose ninguna de estas. Estas palabras son almacenadas en la etapa de entrenamiento.

La función que se ocupa al crear este vector de palabras es el siguiente:

```
function llenarPalabras($palabra) {
    $SQL = "SELECT * FROM palabras WHERE palabra = '".$palabra."'";
    $this->consultal = $this->db->consulta($SQL);

    if(($this->db->num($this->consultal))==0){
        $SQL = "INSERT INTO palabras (palabra) VALUES ('".$palabra."')";
        $this->consulta3 = $this->db->consulta($SQL);
    }
}
```

En donde se recibe como parámetro la palabra que se está analizando y revisa en la tabla palabras si ya esta agregada o no.

5.6 Matriz de Frecuencia

Para construir esta matriz se toma en cuenta la frecuencia de cada una de las palabras dentro del texto, habiendo ya aplicado limpieza (Stop Words, Stemming) por lo tanto no habrá frecuencias de palabras demasiado altas y que no nos interesa en los análisis posteriores. Cabe destacar que este formato es el usado ya que se complementa correctamente con LibSVM, la cual es una herramienta gratuita para aplicar técnicas de Maquinas de Soporte Vectorial a cualquier set de datos.

La función para crear esta matriz de frecuencia es la siguiente:

```
function hacerBoW($texto,$sentido){

    if($sentido==1)
        $linea="1 ";
    else
        $linea="-1 ";

    $SQL = "SELECT * FROM palabras";
    $this->consulta = $this->db->consulta($SQL);
    $this->numeroFilas = $this->db->num( $this->consulta );

    for ( $j = 0 ; $j < $this->numeroFilas ; $j++ ) { // llega a la cantidad de filas
        $datos = $this->db->fetch( $this->consulta );
        $palabras[$j] = addslashes($datos['palabra']);
    }

    for ($x=0; $x<= $this->numeroFilas; $x++){
        $palabras[$x] = trim(strtolower($palabras[$x]));
    }
    $i=1;

    foreach($palabras as $palabra){
        $frecuencia[$palabra]=0;
        $array = explode(chr(32),$texto);
        for( $j = 0; $j < count($array); $j++){
            if($palabra==$array[$j])
                $frecuencia[$palabra]=$frecuencia[$palabra]+1;
        }

        if($frecuencia[$palabra]!=0)
            $linea=$linea.$i.":".$frecuencia[$palabra]." ";
        $i++;
    }

    return $linea;
}
```

En la cual le llega como parámetros el texto que se desea transformar y el sentido que tiene este texto. Las palabras asociadas al texto son almacenadas en una fila la que tiene el siguiente formato:

1 1:6 2:2 3:6 4:1 11:4 26:1 28:4 29:3 37:1 42:1 50:1 51:3 63:6 71:3 75:1 85:2 99:1 101:1 109:1 111:3 112:1 116:7

El cual me dice que el texto es de clase 1 y luego comienza a describir las frecuencias de cada una de las palabras, pudiendo ver en la segunda columna un 1:6 lo que me dice que la palabra numero uno esta repetida seis veces dentro del texto.

6 CONCLUSIONES

Debido al auge de la web 2.0 y al uso masivo de Internet para comunicarse, expresar opiniones y buscar información, entre otros, es que se hace necesario la investigación para clasificar esta información en formato texto de modo automático y entregársela al usuario etiquetada, ahorrando costos y tiempo valioso.

Como uno de los objetivos de este Trabajo de Título, era trabajar con un set de datos utilizado en otros estudios, para tener la posibilidad de comparar resultados y proponer posibles mejoras. Se ha elegido trabajar con comentarios sobre películas, todos textos en ingles, los cuales han sido etiquetados con una escala de 1 a 10 (siendo 1 muy negativo y 10 muy positivo), el cual se ha podido inferir si un texto es positivo o negativo.

En comentarios hechos por usuarios se encuentra una mayor cantidad de términos distintos por lo cual se puede utilizar Stemming para acortar muchas palabras que están en otros tiempos verbales y significan lo mismo. En el caso de este estudio no afecta en mayor medida su cambio, registrándose una variación del 1% con respecto a trabajar los textos sin Stemming. Además en el pre procesado se puede aplicar Stop Words, que consiste en eliminar todas los artículos y preposiciones, que son uniones utilizadas en las oraciones, al utilizar esta técnica se ha llegado a mejorar la precisión en un 1%.

Los textos han sido ordenados en forma vectorial, utilizando la frecuencia de cada una de las palabras dentro del texto para ser procesado por LibSVM, programa utilizado para trabajar con Máquinas de Soporte Vectorial el cual es de uso libre. Este formato ha sido el mismo utilizado tanto para Máquinas de Soporte Vectorial como para Bayes Ingenuo.

Como objetivo general se ha planteado la utilización de máquinas de aprendizaje, más específicamente Máquinas de Soporte Vectorial y Bayes Ingenuo en la clasificación de estos textos, en el cual se ha realizado un estudio profundo de sus distintas técnicas para obtener una mejor precisión en los resultados. En las pruebas realizadas a Bayes Ingenuo se llego a un 72,2% de precisión, con la utilización de Stemming y Stop Words. Mucho mejores fueron los resultados obtenidos por Máquinas de Soporte Vectorial, el cual se llego a una precisión de un 85% con la utilización de solo Stop Words y solo Stemming.

El desarrollo del software ha logrado ir analizando paso a paso los procedimientos por el cual deben pasar cada uno de los textos para ser clasificados, desde la extracción desde internet, pre-procesado con la frecuencia de palabras de cada texto, utilización de una de las dos técnicas elegidas y la elección de los distintos parámetros, dependiendo el algoritmo usado.

Si bien los resultados podrían apoyar los procesos de toma de decisiones a partir del análisis de grandes cantidades de información, en formato de texto, se abre el desafío de plantear técnicas que mejoren la precisión obtenida a través de selección de características, por ejemplo.

ANEXO A: RESULTADOS DE PRUEBAS

Las siguientes cuatro tablas fueron creadas sin aplicar ninguna técnica de limpieza en los textos [17]. Construyendo un vector de 29.054 palabras.

Tabla A1. Kernel Sigmoid, 660 textos de entrenamiento – 100 textos de prueba

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0.001	55%	**84%**	73%	61%	57%	53%	52%
0.01	55%	**84%**	73%	61%	57%	53%	52%
0.1	55%	83%	71%	60%	57%	52%	52%
1	55%	75%	50%	52%	52%	52%	53%
10	55%	65%	51%	51%	51%	50%	50%
20	55%	67%	51%	51%	51%	50%	50%
30	55%	66%	51%	51%	51%	50%	50%
40	55%	62%	51%	51%	51%	50%	50%
50	55%	65%	51%	51%	51%	50%	50%
60	55%	65%	51%	51%	51%	50%	50%
70	57%	65%	51%	51%	51%	50%	50%
80	60%	65%	51%	51%	51%	50%	50%
90	65%	65%	51%	51%	51%	50%	50%
100	66%	65%	51%	51%	51%	50%	50%
1000	82%	65%	51%	51%	51%	50%	50%

Tabla A2. Kernel Radial Basis Function, 660 textos de entrenamiento – 100 textos de prueba

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0	50%	50%	50%	50%	50%	50%	50%
0.001	55%	51%	50%	50%	50%	50%	50%
0.01	55%	51%	50%	50%	50%	50%	50%
0.1	55%	51%	50%	50%	50%	50%	50%
1	55%	57%	51%	51%	50%	50%	50%
10	55%	56%	51%	51%	50%	50%	50%
20	55%	56%	51%	51%	50%	50%	50%
30	55%	56%	51%	51%	50%	50%	50%
40	57%	56%	51%	51%	50%	50%	50%
50	66%	56%	51%	51%	50%	50%	50%
60	72%	56%	51%	51%	50%	50%	50%
70	77%	56%	51%	51%	50%	50%	50%
80	79%	56%	51%	51%	50%	50%	50%
90	82%	56%	51%	51%	50%	50%	50%
100	**83%**	56%	51%	51%	50%	50%	50%
1000	**83%**	56%	51%	51%	51%	50%	50%

Tabla A3. Kernel Polynomial, 660 textos de entrenamiento – 100 textos de prueba

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0	50%	50%	50%	50%	50%	50%	50%
0.001	50%	51%	51%	51%	58%	58%	58%
0.01	58%	51%	51%	58%	58%	58%	58%
0.1	58%	51%	58%	58%	58%	58%	58%
1	58%	51%	58%	58%	58%	58%	58%
10	58%	51%	58%	58%	58%	58%	58%
20	58%	51%	58%	58%	58%	58%	58%
30	58%	51%	58%	58%	58%	58%	58%
40	58%	51%	58%	58%	58%	58%	58%
50	58%	51%	58%	58%	58%	58%	58%
60	58%	51%	58%	58%	58%	58%	58%
70	58%	51%	58%	58%	58%	58%	58%
80	58%	51%	58%	58%	58%	58%	58%
90	58%	51%	58%	58%	58%	58%	58%
100	58%	51%	58%	58%	58%	58%	58%
1000	58%	51%	58%	58%	58%	58%	58%

Tabla A4. Kernel Linear, 660 textos de entrenamiento – 100 textos de prueba

Costo	Accuracy
0	58%
0.001	58%
0.01	55%
0.1	83%
1	83%
10	83%
20	83%
30	83%
40	83%
50	83%
60	83%
70	83%
80	83%
90	83%
100	83%
1000	83%

Las siguientes cuatro tablas fueron creadas aplicando Stop Words a los textos [17]. Construyendo un vector de 28.489 palabras.

Tabla A5. Kernel Sigmoid, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stop Words

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0.001	52%	78%	84%	81%	79%	73%	73%
0.01	52%	78%	84%	81%	79%	73%	73%
0.1	52%	80%	82%	**85%**	78%	74%	76%
1	52%	75%	68%	66%	58%	59%	52%
10	52%	72%	63%	61%	60%	59%	52%
20	52%	69%	64%	62%	62%	57%	52%
30	52%	74%	69%	59%	58%	58%	52%
40	52%	81%	73%	62%	58%	57%	52%
50	52%	80%	65%	62%	59%	57%	53%
60	52%	71%	67%	65%	56%	57%	53%
70	54%	70%	69%	65%	58%	57%	53%
80	56%	75%	67%	59%	58%	57%	53%
90	62%	69%	74%	61%	55%	57%	53%
100	66%	67%	74%	62%	54%	57%	53%
1000	81%	78%	70%	61%	62%	56%	53%

Tabla A6. Kernel Radial Basis Function, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stop Words

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0	53%	53%	53%	53%	53%	53%	53%
0.001	53%	51%	50%	50%	50%	50%	50%
0.01	53%	51%	50%	50%	50%	50%	50%
0.1	53%	51%	50%	50%	50%	50%	50%
1	53%	56%	51%	51%	50%	50%	50%
10	53%	57%	51%	51%	50%	50%	50%
20	53%	57%	51%	51%	50%	50%	50%
30	53%	57%	51%	51%	50%	50%	50%
40	54%	57%	51%	51%	50%	50%	50%
50	65%	57%	51%	51%	50%	50%	50%
60	71%	57%	51%	51%	50%	50%	50%
70	75%	57%	51%	51%	50%	50%	50%
80	78%	57%	51%	51%	50%	50%	50%
90	79%	57%	51%	51%	50%	50%	50%
100	80%	57%	51%	51%	50%	50%	50%
1000	**81%**	57%	51%	51%	51%	50%	50%

Tabla A7. Kernel Polynomial, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stop Words

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0	50%	50%	50%	50%	50%	50%	50%
0.001	50%	51%	50%	50%	51%	59%	59%
0.01	59%	51%	50%	59%	59%	59%	59%
0.1	59%	51%	50%	59%	59%	59%	59%
1	59%	51%	59%	59%	59%	59%	59%
10	59%	51%	59%	59%	59%	59%	59%
20	59%	51%	59%	59%	59%	59%	59%
30	59%	51%	59%	59%	59%	59%	59%
40	59%	51%	59%	59%	59%	59%	59%
50	59%	51%	59%	59%	59%	59%	59%
60	59%	51%	59%	59%	59%	59%	59%
70	59%	51%	59%	59%	59%	59%	59%
80	59%	51%	59%	59%	59%	59%	59%
90	59%	51%	59%	59%	59%	59%	59%
100	59%	51%	59%	59%	59%	59%	59%
1000	59%	51%	59%	59%	59%	59%	59%

Tabla A8. Kernel Linear, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stop Words

Costo	Accuracy
0	59%
0.001	59%
0.01	52%
0.1	81%
1	81%
10	81%
20	81%
30	81%
40	81%
50	81%
60	81%
70	81%
80	81%
90	81%
100	81%
1000	81%

Las siguientes cuatro tablas fueron creadas aplicando Stemming a los textos [17]. Construyendo un vector de 21.341 palabras.

Tabla A9. Kernel Sigmoid, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stemming

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0.001	56%	**85%**	67%	58%	53%	51%	51%
0.01	56%	**85%**	67%	58%	53%	51%	51%
0.1	56%	**85%**	73%	63%	59%	51%	51%
1	56%	70%	50%	50%	50%	49%	49%
10	56%	65%	48%	50%	50%	49%	48%
20	56%	63%	50%	50%	50%	49%	49%
30	56%	67%	50%	50%	50%	49%	49%
40	56%	62%	50%	50%	50%	49%	49%
50	57%	62%	50%	50%	50%	49%	49%
60	59%	63%	50%	50%	50%	49%	49%
70	61%	66%	50%	50%	50%	49%	49%
80	65%	66%	50%	50%	50%	49%	49%
90	68%	65%	50%	50%	50%	49%	49%
100	69%	65%	50%	50%	50%	49%	49%
1000	82%	63%	50%	50%	50%	49%	49%

Tabla A10. Kernel Radial Basis Function, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stemming

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0	49%	49%	49%	49%	49%	49%	49%
0.001	56%	51%	50%	50%	50%	50%	50%
0.01	56%	51%	50%	50%	50%	50%	50%
0.1	56%	51%	50%	50%	50%	50%	50%
1	56%	57%	51%	51%	51%	50%	50%
10	56%	59%	51%	51%	51%	50%	50%
20	56%	59%	51%	51%	51%	50%	50%
30	59%	59%	51%	51%	51%	50%	50%
40	62%	59%	51%	51%	51%	50%	50%
50	69%	59%	51%	51%	51%	50%	50%
60	76%	59%	51%	51%	51%	50%	50%
70	78%	59%	51%	51%	51%	50%	50%
80	79%	59%	51%	51%	51%	50%	50%
90	82%	59%	51%	51%	51%	50%	50%
100	**83%**	59%	51%	51%	51%	50%	50%
1000	**83%**	59%	51%	51%	51%	51%	50%

Tabla A11. Kernel Polynomial, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stemming

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0	50%	50%	50%	50%	50%	50%	50%
0.001	50%	51%	51%	55%	55%	64%	64%
0.01	64%	51%	55%	66%	64%	64%	64%
0.1	64%	51%	55%	64%	64%	64%	64%
1	64%	51%	64%	64%	64%	64%	64%
10	64%	51%	64%	64%	64%	64%	64%
20	64%	51%	64%	64%	64%	64%	64%
30	64%	51%	64%	64%	64%	64%	64%
40	64%	51%	64%	64%	64%	64%	64%
50	64%	51%	64%	64%	64%	64%	64%
60	64%	51%	64%	64%	64%	64%	64%
70	64%	51%	64%	64%	64%	64%	64%
80	64%	51%	64%	64%	64%	64%	64%
90	64%	51%	64%	64%	64%	64%	64%
100	64%	51%	64%	64%	64%	64%	64%
1000	64%	51%	64%	64%	64%	64%	64%

Tabla A12. Kernel Linear, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stemming

Costo	Accuracy
0	64%
0.001	64%
0.01	56%
0.1	82%
1	83%
10	83%
20	83%
30	83%
40	83%
50	83%
60	83%
70	83%
80	83%
90	83%
100	83%
1000	83%

Las siguientes cuatro tablas fueron creadas aplicando Stop Words más Stemming a los textos [17]. Construyendo un vector de 20.999 palabras.

Tabla A13. Kernel Sigmoid, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stop Words más Stemming

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0.001	55%	76%	82%	83%	78%	76%	74%
0.01	55%	76%	82%	83%	78%	76%	74%
0.1	55%	77%	**84%**	**84%**	79%	77%	76%
1	55%	75%	68%	60%	59%	52%	52%
10	55%	77%	63%	59%	55%	52%	51%
20	55%	75%	66%	61%	56%	52%	50%
30	55%	81%	65%	58%	55%	52%	50%
40	55%	67%	64%	60%	54%	52%	51%
50	55%	71%	68%	57%	53%	52%	50%
60	57%	72%	64%	58%	53%	52%	50%
70	60%	67%	65%	63%	53%	52%	50%
80	66%	67%	69%	62%	53%	52%	50%
90	67%	75%	69%	61%	53%	52%	50%
100	68%	75%	65%	59%	53%	52%	50%
1000	83%	76%	68%	63%	55%	52%	50%

Tabla A14. Kernel Radial Basis Function, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stop Words más Stemming

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0	50%	50%	50%	50%	50%	50%	50%
0.001	55%	50%	50%	50%	50%	50%	50%
0.01	55%	50%	50%	50%	50%	50%	50%
0.1	55%	50%	50%	50%	50%	50%	50%
1	55%	58%	51%	51%	51%	50%	50%
10	55%	57%	51%	51%	51%	50%	50%
20	55%	57%	51%	51%	51%	50%	50%
30	57%	57%	51%	51%	51%	50%	50%
40	64%	57%	51%	51%	51%	50%	50%
50	68%	57%	51%	51%	51%	50%	50%
60	76%	57%	51%	51%	51%	50%	50%
70	76%	57%	51%	51%	51%	50%	50%
80	77%	57%	51%	51%	51%	50%	50%
90	77%	57%	51%	51%	51%	50%	50%
100	79%	57%	51%	51%	51%	50%	50%
1000	**81%**	57%	51%	51%	51%	51%	50%

Tabla A15. Kernel Polynomial, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stop Words más Stemming

Costo/Gamma	0	0.1	0.3	0.5	0.7	0.9	1
0	50%	50%	50%	50%	50%	50%	50%
0.001	50%	50%	50%	51%	51%	65%	51%
0.01	51%	50%	51%	58%	51%	51%	51%
0.1	51%	50%	51%	51%	51%	51%	51%
1	51%	50%	51%	51%	51%	51%	51%
10	51%	50%	51%	51%	51%	51%	51%
20	51%	50%	51%	51%	51%	51%	51%
30	51%	50%	51%	51%	51%	51%	51%
40	51%	50%	51%	51%	51%	51%	51%
50	51%	50%	51%	51%	51%	51%	51%
60	51%	50%	51%	51%	51%	51%	51%
70	51%	50%	51%	51%	51%	51%	51%
80	51%	50%	51%	51%	51%	51%	51%
90	51%	50%	51%	51%	51%	51%	51%
100	51%	50%	51%	51%	51%	51%	51%
1000	51%	50%	51%	51%	51%	51%	51%

Tabla A16. Kernel Linear, 660 textos de entrenamiento – 100 textos de prueba, aplicando Stop Words más Stemming

Costo	Accuracy
0	51%
0.001	51%
0.01	55%
0.1	80%
1	81%
10	81%
20	81%
30	81%
40	81%
50	81%
60	81%
70	81%
80	81%
90	81%
100	81%
1000	81%

Resultados de pruebas utilizando Naïve Bayes

La siguiente tabla fue creada utilizando textos de comentarios de películas [17].

Tabla A17. Naïve Bayes, 660 textos de entrenamiento – 100 textos de prueba

Pre Procesado	Accuracy
Ninguno	71,7%
Stop Words + Stemming	72,2%

REFERENCIAS

[1] Andrea Esuli and Fabrizio Sebastiani "Determining Term Subjectivity and Term Orientation for Opinion Mining", 2006.

[2] Andrea Esuli and Fabrizio Sebastiani "SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining", 2006.

[3] Argamon, Bloom, Esuli, Sebastiani "Automatically Determining Attitude Type and Force for Sentiment Analysis", 2007.

[4] Peter R.R White, "Un recorrido por la teoría de la valoración".

[5] G. Salton and C. Buckley. Term-Weighting Approaches in Automatic Retrieval. *Information Processing and Management*, 24(5), pp. 513-525, 1988.

[6] S. E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Filtering Tracks. In Proc. Of the Seventh Text Retrieval Conference (TREC-7), pp. 253-264, 1999.

[7] R. Agrawal, J. R. Bayardo, and R. Srikant. Athena: Mining-Based Interactive Management of Text Databases. In Proc. Of Extending Database Technology (EDBT '00), pp 365-379, 2000.

[8]J. Ponte, and W. B. Croft. A Language Modeling Approach to Information Retrieval. In Proc. Of the Annual Intl. ACM SIGIR Conf. On Research and Development in Information Retrieval (SIGIR '98), pp. 275-281, 1998.

[9] S Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by Latent Semantinc Analysis. Journal of the American Society for Information Science, 41, pp.391-407, 1990.

[10] Bing Liu, "Opinion Mining and Search", Google Pittsburh, 2006.

[11] Ethem Alpaydin, "Introduction to Machine Learning", 2004.

[12] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machine. Cambridge University Press, 2000.

[13] T. G. Dietterich and G. Bakiri. Solving Multiclass Learning Problems via Error-Correcting Output codes, J. of Artificial Intelligence Research, 2. Pp. 263-286. 1995.

[14] Constantino Malagón, "Clasificadores bayesianos. El Algoritmo Naïva Bayes", 2003.

[15] A. McCallum, and K. Nigam. A Comparison of Event Models for Naive Bayes Text Classification. In Proc. Of the AAAI-98 Workshop on Learning for Text Categorization. 1998.

[16] A Singhal. Modern Information Retrieval: A Brief Overview. IEEE Data Engineering Bulletin 24(4), pp. 35-43, 2001.

[17] Data obtenida de Internet Movie Database (IMDb), archivada por rec.arts.movies.reviews y obtenida de http://www.cs.cornell.edu/people/pabo/movie-review-data/