PROTOTIPO PARA LA CLASIFICACIÓN SEMI-SUPERVISADA DE COMENTARIOS QUE SE GENERAN EN LAS REDES SOCIALES, SOBRE PRODUCTOS

JAIRO ANDRÉS VALENCIA GUTIÉRREZ
200859212
jaanvagu@gmail.com

UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
INGENIERÍA DE SISTEMAS
TULUÁ 2013

PROTOTIPO PARA LA CLASIFICACIÓN SEMI-SUPERVISADA DE COMENTARIOS QUE SE GENERAN EN LAS REDES SOCIALES, SOBRE PRODUCTOS

JAIRO ANDRÉS VALENCIA GUTIÉRREZ
200859212
jaanvagu@gmail.com

Director:

EDGAR MOLINA ROJAS
Ingeniero de Sistemas
edgarmolina2@gmail.com

Asesor:

DAVID ALEJANDRO PRZYBILLA
Ingeniero de Sistemas
dav.alejandro@gmail.com

UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
INGENIERÍA DE SISTEMAS
TULUÁ 2013

A mis padres, a mi hermano, y a Salomé. En general a todos los que me ayudaron y creyeron en mí a lo largo de estos cinco años. A los que lo esperaban. Y a Dios por hacerlo posible.

Le agradezco a Dios por haberme concedido el inmenso privilegio de estudiar Ingeniería de Sistemas y por todas las bendiciones que me ha otorgado a lo largo de la vida. También quiero agradecerles a mis padres por su apoyo incondicional y su gran esfuerzo para permitirme estudiar. A mi hermano por ser un motorcito gigantesco. Agradezco también a mi familia; abuelos, tíos, tías, primos y demás, por creer en mí, por motivarme y por sentirse orgullosos. A mi tío Carlos Eduardo, por siempre estar ahí, por su especial respaldo. A mi primo Luis Felipe. A Esperanza, por alentarme siempre. A mi novia Salomé, por su inconmensurable amor, por hacer parte de este proceso y por sus invaluables consejos. A mis amigos, compañeros y demás personas cercanas. Desde luego, gracias a David, mi director, por su gran asesoría y su acompañamiento desinteresado durante la realización de este trabajo de grado. Por último, a mis profesores de jardín, de primaria, de secundaria y, por supuesto, a los de la Universidad del Valle, por transmitirme su conocimiento, y contribuir de gran manera a hacer de mí lo que soy ahora.

Nota de aceptación

_____

_____

_____

_____

_____

Presidente del Jurado

_____

Jurado 1

_____

Jurado 2

Tuluá, Diciembre de 2013

Índice general

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

En la actualidad las empresas utilizan gran cantidad de información, siendo un componente fundamental para las mismas [Pl99]. Las fuentes de datos son múltiples, y debido al crecimiento tecnológico global, enfáticamente en las ciencias de la computación, las redes sociales se han convertido en un recurso de gran importancia para las organizaciones [ELGF07].

La información que se encuentra en las redes sociales sobre compañías en particular, no sólo es importante para ellas, sino también indispensable para las que usan dicha información como insumo para realizar análisis de mercado, imagen, preferencias, etc [BSH09]. Por ejemplo, en Bogotá hay una empresa llamada *Meridean*[1] que realiza estudios de Social Media para otras corporaciones a partir de información extraída de las redes sociales.

El presente trabajo de grado, propone como solución, la construcción de un prototipo de clasificador semi-supervisado de texto, que ayudará a automatizar (evitar labores manuales costosas y demoradas) el proceso de clasificación de información en la compañía. A partir de características[2] (features) de los datos (comentarios de social media), el prototipo entrega los mismos, clasificados por categorías. Estas anotaciones obedecen a parámetros establecidos por *Meridean*. Para verificar la aplicabilidad del sistema, se realizaron pruebas de rendimiento al clasificador, las cuales se detallan en este documento.

Meridean recopila información (comentarios) proveniente de las redes sociales, sobre otras empresas, para después procesarla y realizar estudios de marketing. Dispone de un sistema para clasificar sus datos, sin embargo, no es eficiente, tanto así, que se ven obligados a usar personal en tareas de clasificación manual. Por ejemplo, si un comentario trata sobre la cámara del nuevo Sony Xperia, la etiqueta asignada es "cámara".

El procesamiento del lenguaje natural (PLN o NLP por sus siglas en inglés, Natural Language Processing) y el aprendizaje de máquina (Machine Learning) usados como técnicas de clasificación de textos cortos en español, son temas sin explorar que tienen información compleja y esto lo convierte en un problema difícil de tratar; por lo tanto los resultados obtenidos en este trabajo son aproximaciones a lo que sería una solución completa. El tipo de texto de los comentarios difiere del texto tradicional, ya que éste es un texto corto, informal y con mucho ruído. La aproximación a la solución se basa en una técnica de clasificación semi-supervisada, la cual fue escogida por sus antecedentes y por la compatibilidad con el presente proyecto.

El prototipo bajo ciertas condiciones alcanza una precisión máxima del 68 %, y en promedio su efectividad es del 40 %. A pesar de no alcanzar un mínimo del 50 %, los resultados son buenos teniendo en cuenta lo que representa en comparación con otros desarrollos de la misma línea.

[1] Sitio Web Oficial http://merideangroup.com

[2] Hace referencia a las propiedades que contiene un texto, por ejemplo la cantidad de palabras. De aquí en adelante se añadirá "(features)" a la palabra "características" cuando se refiera a este concepto, así se evitarán ambigüedades.

Se realizó una labor conjunta entre la empresa, como proveedora de información e insumos en general, el director, el asesor y el desarrollador del proyecto, siendo un trabajo de caracter investigativo. A continuación se describirán las actividades realizadas, la forma en que se desarrollaron los objetivos, los resultados obtenidos, las conclusiones y el trabajo futuro.

1.1. Descripción del Problema

Meridean, como fue mencionado anteriormente, es una empresa que realiza monitoreo, análisis e interpretación de Social Media como insumo para proponer estrategias orientadas a los objetivos de negocio de las empresas. Ofrecen servicios como: Social Media Metrics, Social Media Analytics, Análisis de Tendencias y Análisis de Reputación de Marca. Han tenido clientes como: *Coca Cola, El Tiempo, ESPN, DANONE, Santander y Telefónica.* Y tienen alianzas con: *Alterian*[3], *Astrolabio*[4], *Socialatom*[5], entre otras.

Sus fuentes de información las obtiene al extraer miles de comentarios generados en las redes sociales, los cuales se relacionan con productos, empresas y servicios. Su principal tarea consiste en realizar Opinion Mining sobre estos datos, por lo tanto, un paso inicial y fundamental para cumplir su objetivo es clasificar por categorías los comentarios. A continuación se muestran ejemplos de información sin clasificar y posteriormente clasificada sobre la compañía *Kotex*[6].

Lista de Comentarios
"que kotex Unika esta padrisima,huelen riquisimo, estan suavesitas y super cool!!!"
"que son unas toallas nuevas!!!! pero que no son como las demas, son muy comodas y ademas con un olor rico, asi pueden evitar malos aromas despues =)"
"te recomiendo kotex unica para mi siempre me a sido muy util y comoda aparte de que su precio es nuy accesible."
"hola amiga denis t recomiendo k pruebas kotex unika estan a un super precio y tienen un aroma lindo."

Tabla 1.1: Fragmento de comentarios sobre Kotex.

Como se muestra en los ejemplos, inicialmente se dispone de una lista de comentarios sobre una compañía en particular, y se busca clasificar como se observa en la Tabla 1.2. El tema o particularidad que se relaciona en un comentario (para el caso de ejemplo: Comodidad, Tamaño y Precio), se llamará de aquí en adelante "Etiqueta". La empresa tiene un sistema que permite anotar sobre los comentarios, es decir, establece etiquetas pertinentes al contenido de un fragmento de texto como se muestra en la Tabla 1.2. El cual, en cierta medida, y sin ser realmente eficaz reconoce distintivos relacionados en ellos. Al utilizar una herramienta poco competente para dicho proceso, se ven obligados a usar personal para etiquetar manualmente los comentarios. La compañía necesita indispensablemente mejorar el rendimiento en el procedimiento automático de etiquetado, y en la medida de lo posible, eliminar las labores manuales.

Una de sus opciones contempla la posibilidad de utilizar un sistema que realice las anotaciones automáticamente, a partir de unos cuantos comentarios anotados, de tal manera que se anote por ejemplo el 10 % manualmente, y que el sistema etiquete el 90 % restante de los datos. Esta alternativa fue abordada en el presente trabajo, se construyó un clasificador semi-supervisado, que toma un porcentaje de la información previamente etiquetada de una lista de comentarios, se entrena, y finalmente

[3]Sitio Web http://www.alterian.com
[4]Sitio Web http://www.astrolabio.com.co
[5]Sitio Web http://socialatomgroup.com
[6]Kotex es una marca de productos de higiene femenina. http://www.kotexevolution.com/

Lista de Comentarios	Etiquetas		
	Comodidad	**Tamaño**	**Precio**
"que kotex Unika esta padrisima,huelen riquisimo, estan suavesitas y super cool!!!"	x		
"que son unas toallas nuevas!!!! pero que no son como las demas, son muy comodas y ademas con un olor rico, asi pueden evitar malos aromas despues =)"	x		
"te recomiendo kotex unica para mi siempre me a sido muy util y comoda aparte de que su precio es nuy accesible."			x
"hola amiga denis t recomiendo k pruebas kotex unika tienen el largo perfecto."		x	

Tabla 1.2: Comentarios sobre Kotex clasificados.

predice las etiquetas correspondientes para el remanente de la lista. Cabe anotar que las técnicas de clasificación semi-supervisadas arrojan mejores resultados cuando se cuenta con una cantidad reducida de datos anotados.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un prototipo de Clasificador semi-supervisado para etiquetar comentarios generados en las redes sociales, sobre productos.

1.2.2. Objetivos Específicos

- OE1. Identificar las características (features) que se deben extraer de cada comentario, para posteriormente procesarlas en el clasificador.
- OE2. Seleccionar una técnica de clasificación semi-supervisada que se ajuste a las características específicas del problema de Meridean.
- OE3. Implementar el prototipo de clasificador semi-supervisado, utilizando la técnica seleccionada.
- OE4. Evaluar el rendimiento del prototipo, utilizando los indicadores precisión y recall, con base en los datos anotados de la compañía Meridean.

1.3. Organización del Documento

El documento está estructurado en Capítulos, Secciones, Subsecciones y Subsubsecciones. El presente capítulo introduce mediante un breve resumen las características del proyecto, describe el problema y presenta los objetivos.

El segundo capítulo corresponde al marco referencial, en el cual se definen los conceptos incluidos en el trabajo de grado que no son comunes para el lector, posteriormente se explican las teorías sobre procesamiento de lenguaje natural inmersas en la investigación, tales como: el problema de la

clasificación, los tipos de aprendizaje de máquina, los indicadores que se utilizaron para evaluar el rendimiento del prototipo, entre otros. También se describen las técnicas de clasificación referenciadas subsiguientemente. Además, en la última parte del capítulo se expone el marco de antecedentes sobre la clasificación de textos cortos.

Los capítulos tres, cuatro y cinco abarcan el cuerpo del trabajo. El tercer capítulo explica el modelo, la construccion, el funcionamiento y las particularidades del clasificador, tales como: características (features) extraídas de los comentarios, el uso de la técnica de clasificación semi-supervisada elegida y ajustada a las necesidades del desarrollo, y en general el diseño del clasificador.

En el cuarto capítulo se detallan los resultados de la implementación del prototipo, es decir, los sistemas de clasificación. En el capítulo cinco se presentan los ciclos de pruebas realizadas al sistema y se discute sobre los resultados. Finalmente en el capítulo seis se exponen las conclusiones y el trabajo futuro.

Capítulo 2

Marco de Referencia

2.1. Marco Conceptual

2.1.1. Comentario

El comentario es un ejercicio orientado a plasmar por escrito o de forma oral todas las claves que permiten la comprensión plena de un documento o un escrito en general. El comentarista se sitúa, a través de dicho ejercicio, como intermediario entre el documento y un público imaginado. Se trata de un ejercicio que fomenta todas las capacidades intelectuales necesarias para enfrentarse a cualquier tipo de documentación: la lectura comprensiva, el análisis, la crítica, la capacidad de relacionar y contextualizar lo leído, la expresión oral o escrita [Vil09]. En cuanto al proyecto desarrollado, el comentario (mensaje de social media) es un texto de aproximadamente 200 caracteres que expresa una opinión sobre un producto, servicio o una marca.

2.1.2. Estudio de Marketing

Consiste en una iniciativa empresarial con el fin de hacerse una idea sobre la viabilidad comercial de una actividad económica, el estudio de mercado consta de tres grandes análisis importantes [Wik13a]:

- Análisis del consumidor
- Análisis de la competencia
- Estrategia

2.1.3. Etiqueta

Una etiqueta o tag es una palabra clave asignada a un dato almacenado en un repositorio. Las etiquetas son en consecuencia un tipo de metadato, pues proporcionan información que describe el dato (una imagen digital, un clip de vídeo o cualquier otro tipo de archivo informático) y que facilita su recuperación.

La diferencia entre las etiquetas y las palabras clave tradicionales es que las etiquetas son elegidas de manera informal y personal por los usuarios del repositorio. A diferencia de otros sistemas de clasificación, en los sistemas basados en etiquetas no es necesario que exista un esquema de clasificación previo (por ejemplo un tesauro) como base para la clasificación. En los sitios web que permiten etiquetar sus datos, la colección de etiquetas se llama folcsonomía.

Ejemplo: Una página web con una base de datos de imágenes que incluya el sistema de etiquetado, podría tener su contenido marcado con varias etiquetas descriptoras tales como ”pez”, ”anfibio”, ”azul”, ”rojo”... Quienes visiten dicha página probablemente podrán comprender con mayor facilidad el propósito de la misma interpretando su lista de etiquetas. Los sitios web suelen mostrar las etiquetas

en una lista en la propia página, de tal modo que cada etiqueta enlaza a una página índice que enumera todos los datos (en este caso imágenes) marcados con esa etiqueta. Un sistema como éste permite, en nuestro ejemplo, encontrar rápidamente todas las imágenes de peces. Además, si el sitio incorpora un buscador de etiquetas, el visitante podría encontrar todas las páginas que utilizan una combinación de etiquetas, como ”pez” y ”rojo” [Wik13b].

2.1.4. Meridean

Agencia de Monitoreo e Investigación de espacios de interacción espontánea como insumo estratégico. [Mer13]. Es la empresa para la cual se desarrolla el prototipo como propuesta de solución a su problema de clasificación. A continuación se muestran algunos de los trabajos realizados por la compañía:

- Estudio: reputación de los operadores celulares en Colombia (`http://merideangroup.com/operadores/`).
- Tendencias en consumo de información: principales medios de Colombia (`http://merideangroup.com/tendencias-medios/`).
- Estudio: un mes de Twitter en Colombia (`http://merideangroup.com/un-mes-de-twitter-en-colombia/`).

2.1.5. Opinion Mining

Aplicación de procesamiento del lenguaje natural, la lingüística computacional y análisis de textos para identificar y extraer información que permite determinar la actitud de un orador o un escritor con respecto a algún tema [Wik13c].

La minería de opiniones (Opinion Mining) es una disciplina reciente, que mezcla recuperación de información y lingüística computacional, no enfocada al tema tratado en un texto sino a la opinión que expresa [PL08]. Está compuesta, a grandes rasgos, por tres tareas principales que se encargan de identificaciones automáticas:

- Detección de Polaridad, lo cual es identificar si un comentario es *positivo* o *negativo.* En algunos casos se busca obtener un *valor numérico* en un rango determinado que indica polaridad.
- Identificación del *tema* tratado en un comentario.
- Identificación de si un comentario es una *opinión* o es un *hecho.*

2.1.6. Preprocesamiento de Texto

Previamente al proceso de clasificación es necesario limpiar los datos, esta acción será llamada *Preprocesamiento*, la cual corresponde a técnicas de análisis de datos que permiten mejorar la calidad de un conjunto de datos de modo que los métodos de extracción de conocimiento puedan obtener mayor y mejor información.

Una de las tareas previas más importantes para el etiquetado robusto del lenguaje natural, es la correcta segmentación de los textos, esta fase, que puede involucrar procesos mucho más complejos que la simple identificación de las diferentes frases y de cada uno de sus componentes individuales, es a menudo obviada en muchos de los desarrollos actuales [GRF01].

A continuación se enlistan algunas técnicas o procedimientos inmersos en preprocesamiento de texto [GB00]:

- Conversión de formatos diversos a texto plano.
- Identificar y separar los tokens presentes en el texto, de manera que cada palabra ortográfica individual y cada signo de puntuación constituyan un token[7] diferente, por ejemplo “apple” es un token, sin embargo “coca cola” y “Estados Unidos” serían un token cada uno.

[7] Elemento individual o componente léxico.

- Lematizar[8] las palabras, lo que correspondería a estandarizar cada una de ellas, por ejemplo convertir todas las variaciones del verbo "gustar", justamente en esa palabra (lema); entonces si se encuentra "gusta", "gustó", "gustaría", etc, todas serían tratadas como "gustar".
- Eliminar ruido (signos de puntuación, números, acentos, caracteres especiales, frases o expresiones onomatopéyicas como: "jaja", "jiji", "ah", "oh", etc.).
- De acuerdo al contexto contractua, se debe eliminar cierta información, como por ejemplo articulos o adjetivos; no obstante en este proyecto se requiere eliminar artículos, y adverbios (que no cumplan la función de adjetivos).
- Compactación de los separadores redundantes que existen en el texto (eliminar múltiples espacios, espacios a inicio de frase, etc.).

2.1.7. Redes Sociales

Una red es una forma abstracta de visualizar una serie de sistemas, y en general, casi todos los sistemas complejos. Las redes sociales son también redes complejas, aunque usan una terminología ligeramente diferente: los nodos son agentes, porque hacen algo, mientras que las aristas o arcos expresan habitualmente, una relación social tal como conoce-a, es-amigo-de, o han-comido-spaghetis-juntos [MG06].

En general, una red social es un conjunto bien delimitado de actores –individuos-, grupos, organizaciones, comunidades, sociedades globales, etc. Vinculados unos a otros a través de una relación o un conjunto de relaciones sociales [Loz96]. Este concepto es de relevancia para el proyecto porque es la fuente de alimentación del clasificador.

2.1.8. Social Media

No existe una definición concreta para este término, a continuación se mencionarán algunas denotaciones [Dir13]:

- Son contenidos creados y compartidos por individuos en internet, utilizando para ello plataformas web que permiten al usuario publicar sus propias imágenes, vídeos y textos y compartirlos con toda la red o con un grupo reducido de usuarios. *Affilorama.*
- Son instrumentos online de comunicación de masas, que incluyen blogs, microblogs, redes sociales y podcasts. *Answers.com*
- Los social media son plataformas online que suministran contenido al usuario y permiten que éste participe también de alguna manera en la creación y desarrollo de dicho contenido. *BlackBox Social Media.*
- Los social media son una forma de compartir información generada por el usuario y de interactuar online utilizando para ello tecnologías de internet. *Briancee.*

2.1.9. Bag of Words

En las ciencias de la computación han surgido diversas técnicas de extracción de datos, un modelo reconocido es el enfoque "Bag of Words" que en su traducción literal es una bolsa de palabras. Dicha técnica extrae todas las palabras de conjunto de textos (escritos, artículo, en este caso comentarios) y genera un vector n-dimensional de palabras (contiene todas las palabras que existan en el conjunto de textos). Posteriormente genera vectores de frecuencia para cada una de las entradas que formaron el vector incial, cuyo resultado corresponde a la obtención de caracteríasticas del texto. Se aclara que en el uso de esta técnica el orden de las palabras no tiene relevancia. Ejemplo:

[8]El lema de una palabra es la forma de la palabra que nos encontraríamos en un diccionario tradicional; forma singular para sustantivos, masculino singular para adjetivos e infinitivo para verbos. Por ejemplo el lema de *encontrado* es *encontrar*.

- Comentario 1: “me encantó el nuevo sabor”
- Comentario 2: “ya lo probé”
- **Vector n-dimensional incial**: [me, encantó, el, nuevo, sabor, ya, lo, probé]
- Vector frecuencias **comentario 1**: [1, 1, 1, 1, 0, 0, 0]
- Vector frecuencias **comentario 2**: [0, 0, 0, 0, 1, 1, 1]

En el área de recuperación de información, es importante identificar la relevancia de las palabras y las expresiones, por ejemplo, una búsqueda en Google[9] arroja los resultados más relevantes de acuerdo a una evaluación, lo que generalmente obedece al peso de las palabras. A continuación se describen algunos tecnicismos relacionados con este concepto [Ram03]:

- **Term Frequency (TF)**

  Es la cantidad de veces que aparece una palabra en un documento, dividido entre el número total de palabras contenidas en él.

$$TF = \frac{apariciones\ de\ un\ término}{número\ total\ de\ palabras}$$

- **Inverse Document Frequency (IDF)**

  Es el número total de documentos de una colección, dividido entre el número de documentos que contienen una palabra específica, posteriormente se calcula el logaritmo de la operación anterior.

$$IDF = log\left(\frac{número\ documentos\ colección}{1 + número\ documentos\ contienen\ palabra}\right)$$

  Se acostumbra sumar 1 al denominador para evitar la división por cero en caso de que la palabra no aparezca ni una vez.

- **TF-IDF**

$$TF - IDF = TF \times IDF$$

- **Número de ocurrencia (TO)**

  Es la frecuencia absoluta de los términos de un documento.

- **Ocurrencia binaria (BTO)**

  A un término se le asigna el valor 1 si aparece en el documento, y 0 si no aparece.

2.1.10. Codificación ASCII

El acrónimo viene del inglés *American Standard Code for Information Interchange* (en español, Código Estándar Estadounidense para el Intercambio de Información). Es una representación númerica de caracteres como “a”, “-”, “?” (en números decimales, el número correspondiente al caracter “a” es 97). Se usa en las computadoras debido a que éstas sólo entienden números [RC96]. Sin embargo, en el ámbito de este proyecto se utiliza con fines de identificación de caracteres para filtrado de los mismos.

Representación Numérica	Caracter
97	a
98	b
99	c
100	d

Tabla 2.1: Ejemplo de representación de caracteres en código ASCII.

[9] http://www.google.com

2.2. Marco Teórico

2.2.1. Aprendizaje de Máquina

También es conocido como "Aprendizaje Automático" o "Machine Learning" (en inglés). Su objetivo es conseguir que una máquina (habitualmente una computadora) sea capaz de utilizar datos o experiencias pasadas para resolver un problema que se le plantee. El ordenador puede realizar, de una forma adecuada y automática, un aprendizaje que le lleva a ser capaz de solucionar, problemas que requieren ciertas habilidades más allá de la mera capacidad de cálculo. Para ello debemos ser capaces de indicarle de dónde debe aprender, cuál es el objetivo a cumplir, y qué tipo de resultados esperamos que nos ofrezca [Sie06].

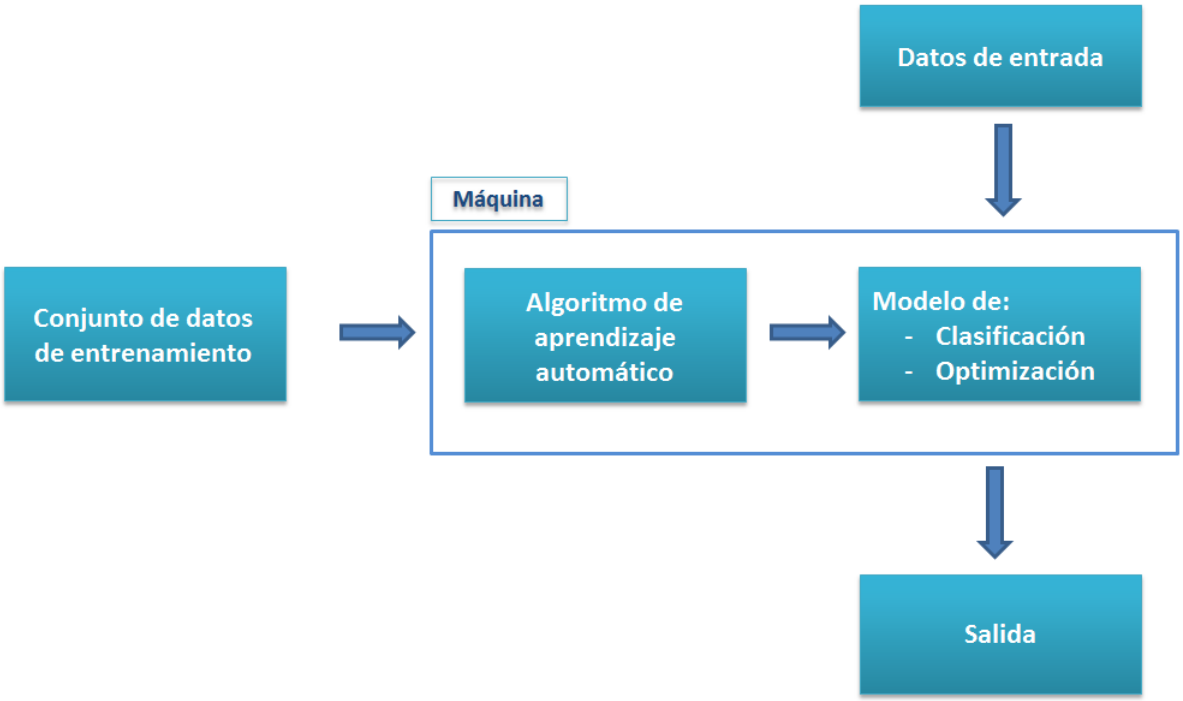


Figura 2.1: Esquema de aprendizaje.

Una máquina aprende de una experiencia E con respecto a una clase de tareas T y medida de desempeño P, si su desempeño en las tareas en T, de acuerdo con la medida P, mejora con la experiencia E [Mit97].

Ejemplo:

- Tarea T: clasificar imágenes.
- Medida de desempeño P: porcentaje de imágenes clasificadas correctamente.
- Experiencia de entrenamiento E: conjunto de imágenes clasificadas manualmente.

Actualmente existen numerosas aplicaciones funcionando satisfactoriamente, incluyendo métodos de optimización del comportamiento de un robot en el desarrollo de las tareas que debe realizar, sistemas que analizan históricos de ventas para predecir las compras de un determinado cliente, aplicaciones de reconocimiento de caras, reconocimiento de voz, extracción de información de datos bioinformáticos para detectar los genes que más influyen en determinadas enfermedades, clasificación de texto, etc.

2.2.2. Problema de la Clasificación

En un sentido estricto, clasificar es ordenar por clases. No obstante, la clasificación de texto es tradicionalmente una tarea de aprendizaje supervisado (ver Subsección 2.2.5), sin embargo para entrenar un clasificador se necesitan conjuntos de datos de entrenamiento etiquetados, y éstos, generalmente son difíciles, costosos y lentos de obtener, ya que se requieren experimentados anotadores humanos para la realización de esta actividad [Zhu05b].

Definición:

- Determinar la asignación de un valor $a_{ij} \epsilon \{0, 1\}$ para cada entrada de una matriz de decisión (Tabla 2.2).
- Donde $C = \{c_1, c_2, ..., c_m\}$ representaría el conjunto de categorías predefinidas.
- $D = \{d_1, d_2, ..., d_n\}$ el conjunto de textos a ser clasificados.
- Y a_{ij} la decisión de clasificar d_j en la categoría c_i, de manera que si $a_{ij} = 1$ entonces el elemento d_j es clasificado en c_i y $a_{ij} = 0$ en caso contrario.

	$\mathbf{d_1}$	...	$\mathbf{d_j}$	...	$\mathbf{d_n}$
$\mathbf{c_1}$	a_{11}	...	a_{1j}	...	a_{1n}
...	...	...	...	...	...
$\mathbf{c_i}$	a_{i1}	...	a_{ij}	...	a_{in}
...	...	...	...	...	...
$\mathbf{c_m}$	a_{m1}	...	a_{mj}	...	a_{mn}

Tabla 2.2: Matriz de decisión.
(Tomada de [PS02])

Para conseguir conjuntos de datos de entrenamiento, se tienen problemas relacionados con:

- **Reconocimiento de voz:** La transcripción exacta de la expresión verbal a nivel fonético[10], consume mucho tiempo (400 veces más que la declaración del enunciado) y requiere experiencia en lingüística. La transcripción a nivel de palabra, consume aún más.
- **Categorización de texto:** Filtrar correo no deseado en los mensajes del correo electrónico, o etiquetar artículos como interesantes o no, tener que leer miles de documentos, es una tarea de enormes proporciones para las personas promedio.
- **Análisis Sintáctico[11]:** Para entrenar un analizador se necesitan pares de árboles, conocidos como treebanks. Éstos últimos toman mucho tiempo para ser construidos. Expertos tardaron años creando árboles para unas pocas miles de frases.
- **Videos de vigilancia:** Etiquetar manualmente personas, en grandes cantidades de imágenes de vigilancia, puede llevar mucho tiempo.
- **Predicción de la estructura de las proteínas:** Puede tomar meses de trabajo, en un laboratorio costoso, a un equipo de cristalógrafos expertos, identificar la estructura 3D de una sola proteína.

[10] Estudio de los sonidos físicos del habla humana.
[11] Es una de las partes de un compilador que transforma su entrada en un árbol de derivación. Por ejemplo, descomponer la fecha "15/04/13" en una estructura como: $día = 15$; $mes = abril$; $año = 2013$.

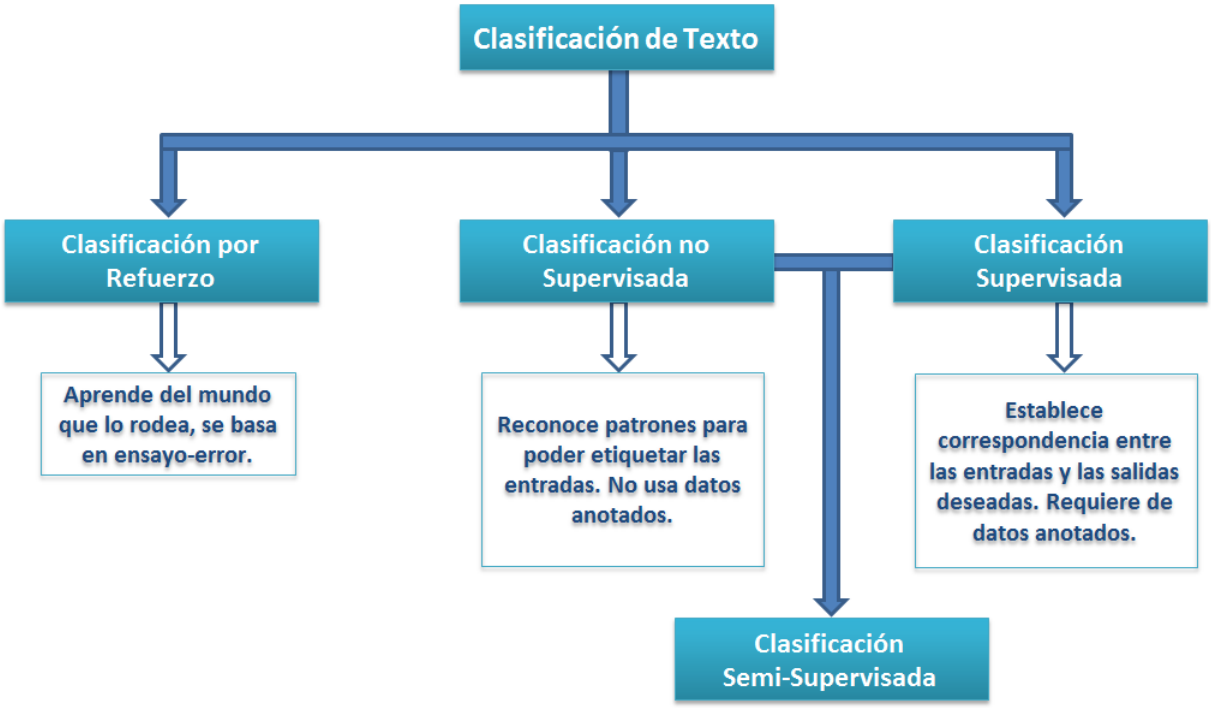


Figura 2.2: Estructura de la clasificación de texto.

Por otro lado, los datos no etiquetados, generalmente están disponibles en grandes cantidades, y su recopilación no es costosa. Las expresiones se pueden grabar de cualquier estación radial, los documentos de texto se encuentran disponibles en internet; las oraciones están por todas partes, las cámaras de vigilancia funcionan las 24 horas del día, las secuencias de ADN de las proteínas son fácilmente accesibles en bases de datos genéticas. El problema con los métodos tradicionales de clasificación, es que no pueden utilizar datos sin anotar para entrenarse. El campo del aprendizaje automático se ha dividido tradicionalmente en tres partes: aprendizaje por refuerzo, aprendizaje no supervisado y aprendizaje supervisado; adicionalmente sugió un híbrido, el aprendizaje semi-supervisado (Figura 2.2). [Zhu05a].

2.2.3. Aprendizaje por Refuerzo

Consiste en aprender qué acciones realizar, dado el estado actual del ambiente, con el objetivo de maximizar una recompensa, lo cual requiere de un mapeo de situaciones a acciones. El sistema que aprende debe descubrir por sí solo cuáles son las acciones que le dan a ganar más. En los casos más interesantes y difíciles, las acciones pueden afectar no sólo la recompensa inmediata sino también la siguiente situación, y así repercutir sobre las recompensas próximas. Estas dos características, *prueba-error* y *recompensas retrasadas*, son las más sobresalientes en este tipo de aprendizaje [SB98].

El aprendizaje se realiza en la mayor parte de los algoritmos mediante interacción entre el agente y su medio ambiente, como se muestra en la Figura 2.3. El agente debe explotar el conocimiento que actualmente tiene para obtener recompensas, pero también debe explorar para poder ejecutar mejores acciones en el futuro.

2.2.4. Aprendizaje no Supervisado

Este tipo de clasificación no requiere de datos anotados, es decir, maneja las variables de entrada sin tipificación, no posee conocimiento a priori. A groso modo, toma el conjunto de entrada, y agrupa los elementos comunes mediante técnicas estadísticas[12] [Wil02]. El sistema de aprendizaje observa un

[12]Inferencia bayesiana, distribución de probabilidades, prueba de χ^2, análisis de regresión, entre otras.

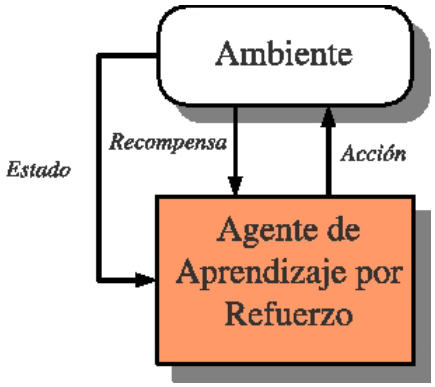


Figura 2.3: Arquitectura básica de un sistema de arendizaje por refuerzo.
(Tomada de [Sem12]).

conjunto de elementos no etiquetados, representados por sus características (features) $\{x_1, x_2, ..., x_n\}$. El objetivo es organizar los elementos asociándolos en grupos y detectando valores atípicos para determinar si un nuevo elemento x es significativamente diferente de los elementos vistos, para así, reducir su dimensionalidad preservando algunas propiedades relevantes [Zhu05b].

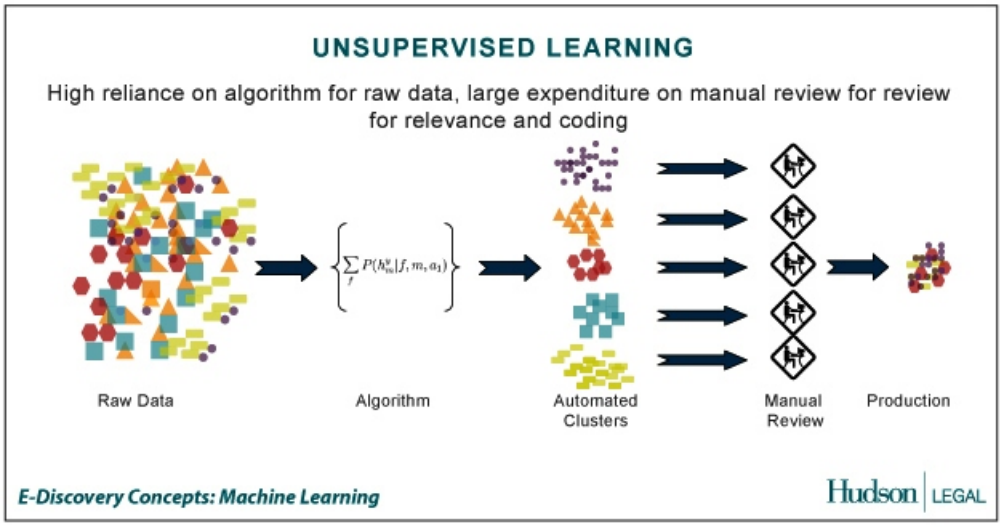


Figura 2.4: Estructura del aprendizaje no supervisado.
(Tomada de http://hudsonlegalblog.com).

Definición:

- $N = \{e_1, e_1, ..., e_n\}$ es un conjunto de elementos.
- $C = \{c_1, c_2, ..., c_n\}$ es una división de N en subconjuntos.

Donde cada subconjunto c_i $\{i = 1, 2, ..., n\}$ es llamado *cluster,* y C es llamada un *Clustering*[13].

El objetivo del Clustering, es encontrar divisiones (subconjuntos) de los elementos de N homogéneas y poco separadas. Los elementos de un mismo cluster deberían tener alta similitud, en cambio, baja con

[13]En este contexto, cluster es un subconjunto de elementos agrupados con características comúnes y Clustering es el conjunto global de elementos.

los elementos de otros clusters. En la práctica, la homogeneidad y la separación dependen del problema, sin embargo, usualmente la alta homogeneidad está ligada con baja separación y viceversa [Cra99].

2.2.5. Aprendizaje Supervisado

La clasificación supervisada es una técnica que emplea una función, la cual, a partir de un conocimiento general, arroja un resultado particular. El conocimiento se inyecta en forma de pares de datos, siendo el primer elemento una entrada y el segundo el dato de salida esperado. Cuando la función retorna un valor numérico, se conoce como regresión, mientras que cuando arroja una etiqueta de clase, se le denomina clasificación [Zhu05b].

En el marco de la clasificación de texto, el sistema de aprendizaje observa un conjunto de datos de entrenamiento conformado por pares característica/etiqueta; y su propósito es predecir una etiqueta para cualquier característica (features) nueva introducida como entrada [Møl93].

Definición:

- Sea la lista de pares $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ el conjunto de datos de entrenamiento, donde cada x_i $\{i = 1, 2, .., n\}$ representa el vector de características (features), y cada y_i $\{i = 1, 2, ..., n\}$ representa la etiqueta asociada al vector x_i.
- La función a aprender se describe como[14]:

$$y = Y(x)$$

- En particular, para todos los pares (x_i, y_i):

$$y_i = Y(x_i)$$

El objetivo del aprendizaje supervisado, es predecir el valor de y para instancias que no están en los datos de entrenamiento [Abn07].

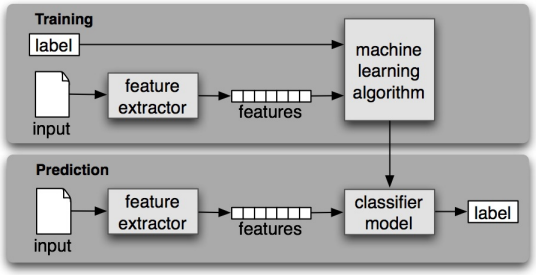


Figura 2.5: Modelo de clasificación supervisada.
(Tomada de http://nltk.googlecode.com/svn/trunk/doc/book/ch06.html).

2.2.6. Aprendizaje Semi-Supervisado

Antes de definir qué es, se aclara que, como se describió en la Subsección 2.2.5, el aprendizaje supervisado es aquel que usa sólo datos anotados para generalizar el concepto que debe predecir, en cambio el semi-supervisado usa estos últimos y además da un vistazo a la estructura de los datos no anotados para universalizar el concepto que debe predecir.

[14]Para un instancia x, la salida y es la etiqueta correspondiente.

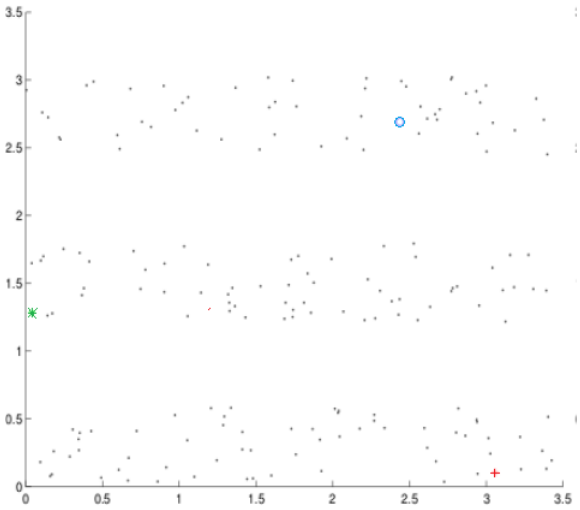


Figura 2.6: Datos de entrada con pocos datos anotados.

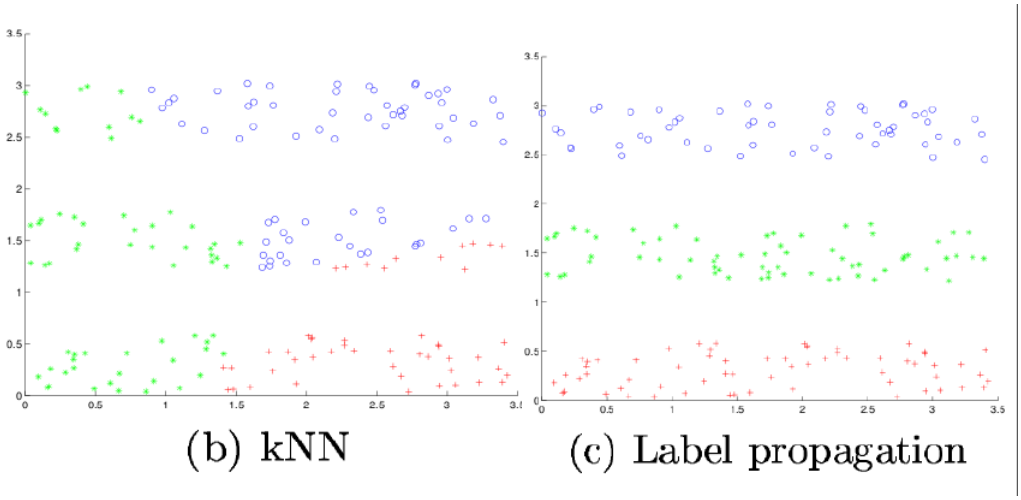


Figura 2.7: Resultados usando knn y LP.

En la anterior figura se puede observar como Label Propagation (técnica de aprendizaje semi-supervisado) es capaz de encontrar la estructura de los datos, en cambio kNN (técnica supervisada) falla en su intento, por tanto, teniendo en cuenta las características del proyecto, se justifica la elección de Label Propagation como herramienta de clasificación (a continuación se adiciona gráfica de comparación de rendimiento de SVM y LP con cantidades diferentes de datos anotados).

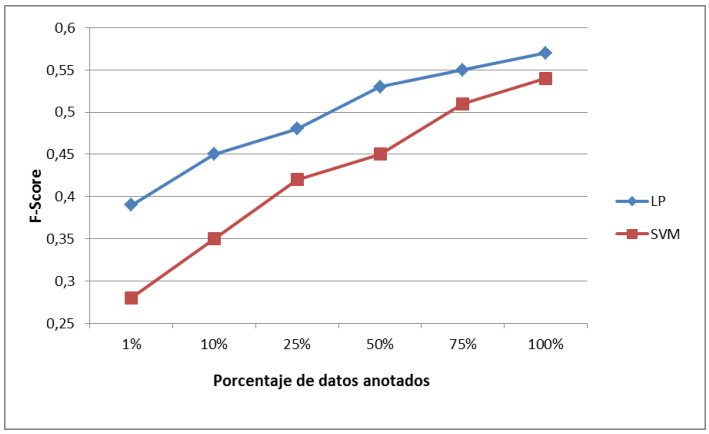


Figura 2.8: Comparación de rendimiento de SVM y LP con cantidades diferentes de datos anotados.

Ésta es una forma especial de clasificación, puesto que, mediante el uso de gran cantidad de datos no etiquetados, soluciona el problema de la costosa obtención de datos anotados, y consigue construir mejores clasificadores. Debido a que requiere menor empeño humano, este aprendizaje es de gran interés tanto en la teoría como en la práctica; es una técnica que promete una alta precisión anotando con menor esfuerzo [Zhu05a].

Definición: [Zhu05b, CSZ+06, Abn07]

- Dado un conjunto relativamente pequeño de datos etiquetados $\{(x, y)\}$ y un gran conjunto de datos no etiquetados $\{x\}$, se tiene un grupo de entrenamiento para un clasificador semi-supervisado. Donde x es una lista de conjuntos de caracteristicas (features) $\{x_1, x_2, ..., x_n\}$ y y es una lista de etiquetas $\{y_1, y_2, ..., y_n\}$ correspondientes a cada x_i.
- Entonces, sea la función a aprender:

$$f(x) = y$$

- La función f recibe un conjunto de características (features) x, y devuelve la etiqueta correspondiente y. La forma de hallar y varía de acuerdo a la técnica utilizada, entonces se tiene que:

$$f(x) = L$$

- Donde L (*Loss Function*) es una función que varía de acuerdo a la técnica empleada. El propósito es encontrar una función f, tal que maximice el valor arrojado por L.

2.2.7. Indicadores Precision y Recall

En el marco de la clasificación, para medir la eficacia de un sistema, se usan indicadores de rendimiento. Ellos son datos en series temporales que reflejan y registran cambios por medio de un número significativo de dimensiones relevantes, a través de los cuales se juzgará la eficacia y eficiencia de un sistema para alcanzar unos objetivos [NR97].

Los indicadores Precision y Recall son los más comunes en el campo de reconocimiento de patrones (también es usado el F-Score), la forma de usarlos varía dependiendo del ambiente donde se requieran. Para la clasificación de texto por categorías se tiene que [MGT03]:

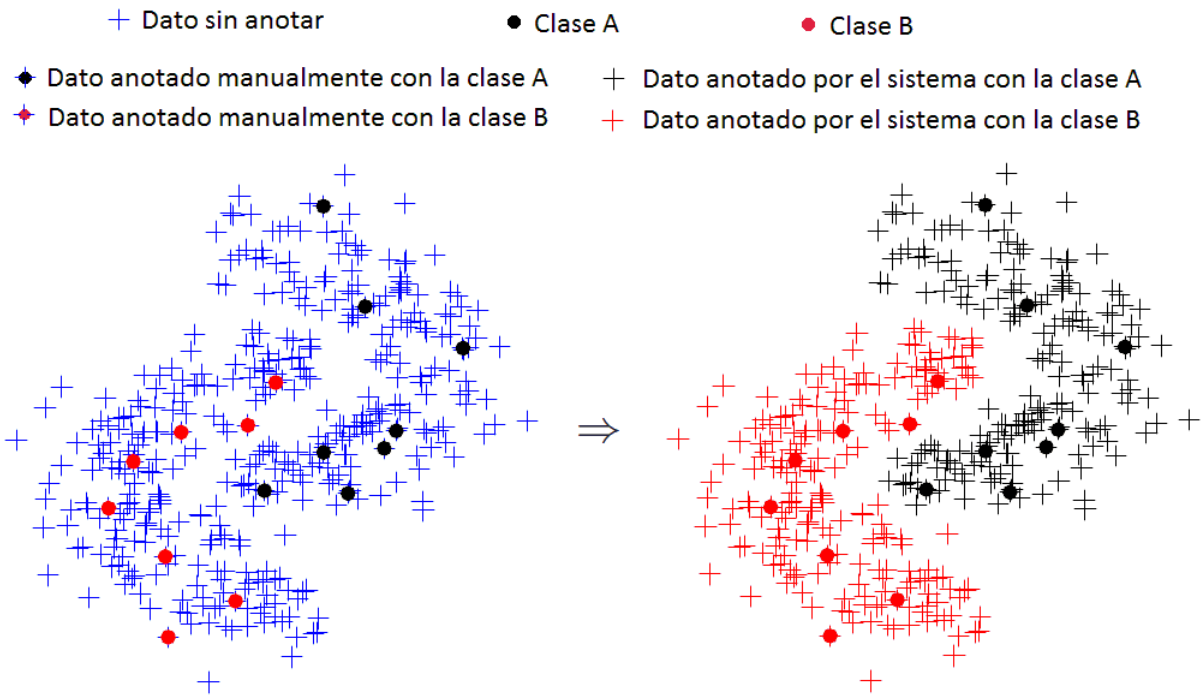


Figura 2.9: Clasificación semi-supervisada.
(Adaptada de [Bac12]).

- Indicador Precision: Se define como la cantidad de predicciones correctas -de una etiqueta PCE- hechas por el sistema, sobre el número total de predicciones PTE -de dicha etiqueta- hechas por el sistema.

$$Precision = \frac{PCE}{PTE}$$

- Indicador Recall: Se define como la cantidad de predicciones correctas -de una etiqueta PCE- hechas por el sistema, sobre el número total de predicciones PT que debía haber realizado el sistema.

$$Recall = \frac{PCE}{PT}$$

- Indicador F-Score:

$$F - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

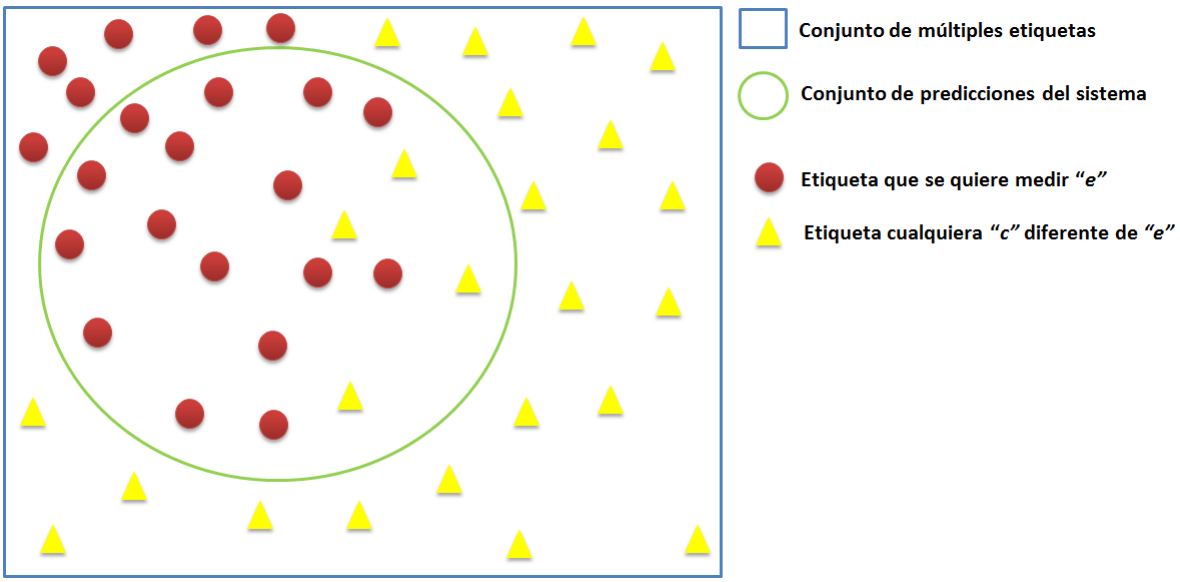


Figura 2.10: Ejemplo sobre clasificación por etiquetas.

Ejemplo:

Se requiere calcular el indicador precision y el recall para una etiqueta particular e. Entonces según lo ilustrado en la Figura 2.9 las variables del problema son:

- $PCE = 16 \rightarrow$ Predicciones de e hechas por el sistema.
- $PTE = 20 \rightarrow$ Etiquetas totales predichas por el sistema.
- $PT = 22 \rightarrow$ Cantidad total de predicciones de e que debió realizar el sistema.

Por lo tanto:

$$Precision = \frac{16}{20} = 0,8$$

$$Recall = \frac{16}{22} = 0,7272$$

2.2.8. Técnicas de Clasificación

Las técnicas descritas en esta sección no son más que algoritmos. Existen múltiples métodos que se enfocan en la clasificación de textos, sin embargo, aquí sólo se retrataran los que fueron considerados durante el desarrollo del proyecto [WF05].

2.2.8.1. Clasificador Bayesiano ingenuo (Naive Bayes)

En minería de datos y probabilidad, es un clasificador probabilístico fundamentado en el teorema de Bayes. Dado un ejemplo x representado por k valores, se requiere encontrar la hipótesis más probable que describa a ese ejemplo. Si la descripción de ese ejemplo viene dada por los valores $< a_1, a_2, ..., a_n >$, la hipótesis más probable será aquella que cumpla [Luq03]:

$$v_{MAP} = argmax_{v_j \epsilon V} \frac{P(a_1, ..., a_n | v_j)}{P(a_1, ..., a_n)} = argmax_{v_j \epsilon V} P(a_1, ..., a_n | v_j) p(v_j)$$

$P(v_j)$ se estima 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. 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 pertenecen, es justamente el producto de las probabilidades de cada valor por separado: $P(a_1, ..., a_n|v_j) = \prod_i P(a_i|v_j)$ [PLV02].

2.2.8.2. Máxima Entropía (Maximum Entropy)

En estadística y teoría de la información, Maximun Entropy es una distribución de probabilidad cuya entropía es al menos tan grande como la de todos los demás miembros de una clase específica de las distribuciones. De acuerdo con el principio de máxima entropía, si no se sabe nada acerca de una distribución excepto que pertenece a cierta clase, entonces la distribución con la mayor entropía debe ser elegida como el valor predeterminado.

Nigam et al. (1999) muestra que a veces, pero no siempre, supera Naive Bayes. Se estima $P(c|d)$ de la siguiente forma:

$$P_{ME}(c|d) := \frac{1}{Z(d)} exp\left(\sum_i \lambda_{i,c} F_{i,c}(d,c)\right)$$

Donde $Z(d)$ es una función de normalización. $F_{i,c}$ es una función *caracteristica/etiqueta* [PLV02].

2.2.8.3. Support Vector Machine

SVM (Support Vector Machine) fue utilizado por primera vez en la Clasificación de Texto en 1998 por T. Joachims [Joa99]. En términos geométricos, se puede ver como el intento de encontrar un espacio n-dimensional, que permita separar los ejemplos positivos de entrenamiento de los negativos, permitiendo especificar el margen más amplio posible. [VMLC05]. En la última década se ha convertido en una de las técnicas más utilizadas debido a los buenos resultados que se han obtenido. Está basada en la representación de los documentos en un modelo de espacio vectorial, donde se asume que los documentos de cada clase se agrupan en regiones separables del espacio de representación. En base a ello, trata de buscar un hiperplano que separe cada clase, maximizando la distancia entre los documentos y el propio hiperplano, lo que se denomina margen (Ver Figura 2.10). Este hiperplano se define mediante la siguiente función:

$$f(x) = w \cdot x + b$$

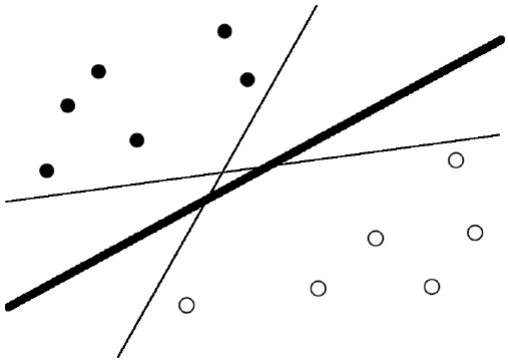


Figura 2.11: Ejemplo de máximización del margen con SVM, donde la línea más gruesa sería la escogida por el sistema.

La optimización de esta función supondría tener en cuenta todos los valores posibles para w y b, para después quedarse con aquellos que maximicen los márgenes. Esto resulta muy difícil de optimizar, por lo que en la práctica se utiliza la siguiente función de optimización equivalente (Ver Figura 2.11):

$$min\frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i^d$$

$$\text{Sujeto a: } y_i(w \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0$$

donde C es el parametro de penalización y ξ_i es la distancia entre el hiperplano y el documento i.

De esta manera únicamente se resuelven problemas linealmente separables, por lo que en muchos casos se requiere de la utilización de una función de kernel para la redimensión del espacio. Así, el nuevo espacio obtenido resultará linealmente separable. Posteriormente, la redimensión se deshace, de modo que el hiperplano encontrado será transformado al espacio original, constituyendo la función de clasificación.

Es importante destacar que esta función únicamente puede resolver problemas binarios y de forma supervisada [ZFM09].

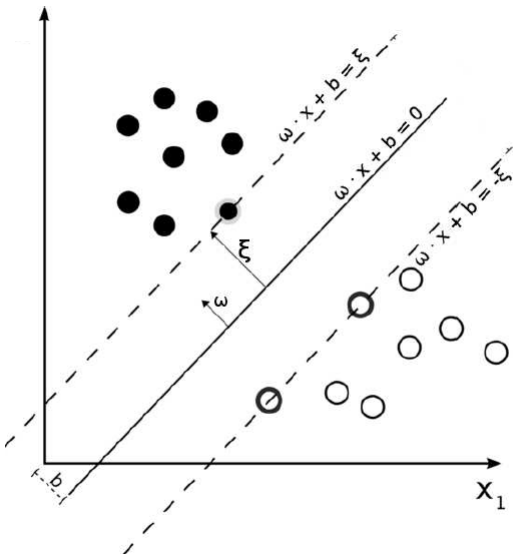


Figura 2.12: Representación gráfica de la función de clasificación.

Apartándose de las definiciones formales, un ejemplo de clasificación usando SVM se vería como a continuación (es necesario tener en cuenta que SVM sólo es capaz de indicar si un conjunto de datos pertenece a una etiqueta de clase o no, es decir, solo establece si un objeto es negro o no lo es, no podría asegurar que es blanco):

- Se requiere identificar si un recipiente contiene Coca Cola en su interior. Para ello se establecen las siguientes variables, que serán caracteristicas suministradas al clasificador SVM.
  - Color
  - Estado
  - Sabor

- Entonces, se eligen los ejemplos positivos (aquellas características que indican que el contenido SÍ es Coca Cola) y negativos (cuando es valor de las variables no cumplen con las condiciones de ser Coca Cola) de entrenamiento. Si un líquido es de color negro, se asigna el número uno a dicha variable, si el estado es líquido, igualmente se asigna el número uno, si el sabor es dulce se procede igual; para los valores diferentes a los mencionados, se asigna el número cero.

Ejemplo	Color	Estado	Sabor	Positivo	Negativo
Ejemplo 1	1	1	1	SÍ	NO
Ejemplo 2	1	1	1	SÍ	NO
Ejemplo 3	1	0	1	NO	SÍ
Ejemplo 4	0	0	1	NO	SÍ

- Después de entrenar el clasificador, es posible ejecutar pruebas de clasificación, es decir, se construye un conjunto de caracteríticas, se le pasa a SVM y este indicará si es Coca Cola o no, según como fue entrenado.

Prueba	Color	Estado	Sabor	¿Es Coca Cola?
Datos de prueba 1	1	1	1	SÍ
Datos de prueba 1	0	0	0	NO

2.2.8.4. Label Propagation

Es un algoritmo de clasificación semi-supervisado que propaga soft-labels, es decir, da un estimado de probabilidad para cada una de las etiquetas de propagación, donde dichas labels (soft probabilities) se refieren a priors, es decir a una distribución previa del sistema. Por ejemplo si se conoce que en Colombia hay más que usan iPhone que Samsung, se podría inyectar ese tipo de conocimiento en los soft labels.

LP Comparte similitudes con kNN (K nearest neighbours)[15], sin embargo usa tanto datos anotados como no-anotados durante el proceso de entrenamiento. Este método de aprendizaje basado en grafos, fue propuesto por Zhu [Zhu05b]. Este enfoque mapea los datos a una representación en un grafo, en el cual, cada arco conectará dos nodos si y solo si, éstos son similares; los pesos de los arcos son directamente proporcionales a la similitud de los nodos incidentes. En el contexto del presente trabajo, cada nodo del grafo corresponde al vector de características (features) de un comentario, donde el peso de los arcos está dado por la similitud coseno entre los vectores. La representación final es un grafo completo.

La etiqueta de un nodo se propaga a los nodos vecinos a través sus pesos ponderados, si un arco tiene un peso alto, la etiqueta viajará fácilmente a través del grafo [CJTN06]. El algoritmo itera hasta la convergencia, en cada paso empuja las etiquetas de los datos etiquetados a través de los arcos de altos pesos. El supuesto básico es que los nodos que pertenezcan a una determinada clase, conectarán con los nodos anotados que corresponden a la misma. En cada iteración las etiquetas de los datos anotados permanecen amarradas a ellos. Esta propagación se realiza mediante la medición de una matriz de transición T, la cual indica la probabilidad de propagar una etiqueta de una instancia, y se mide a partir de los pesos de los arcos del grafo.

Siendo $T_{i\,j}$ la probabilidad de propagación de la etiqueta de la instancia i a la instancia j, entonces se tiene que:

$$T_{i\,j} = P(i \to j) = \frac{w_{i\,j}}{\sum_{k=1}^{n} w_{k\,j}}$$

Algoritmo

[15]Es un método de clasificación supervisada que sirve para estimar la función de densidad $F(x/C_j)$ de las predictoras x por cada clase C_j.

Se define $Y_{i\,j}$ como la probabilidad (soft-probability) de etiquetar la instancia i con la etiqueta j. Donde Y se divide entre:

Y_U : Las etiquetas (soft-labels) para los datos no-anotados
Y_L : Las etiquetas (soft-labels) para los datos anotados

Esto significa que Y_L es dado al principio del problema, y el objetivo es encontrar los valores correctos para Y_U.

- **Paso 1 - Inicio:** Inicialmente se asocian las etiquetas para Y_L, y se asignan etiquetas al azar para Y_U. En [Zhu05b] se evidencia que la inicialización de los valores para Y_U es intrascendente.
- **Paso 2 - Propagación:** Se establecen etiquetas para los nodos vecinos. Siendo t la iteración actual, entonces Y^{t+1} es definida como:

$$Y^{t+1} = TY^t$$

- **Paso 3 - Asignación de datos anotados:** Se amarran las etiquetas para Y_L.
- **Paso 4 - Repetición:** Se repite el proceso desde el paso 2 hasta que converge.
- **Paso 5 - Enfoque del etiquetado:** Una vez hay convergencia, se debe elegir una etiqueta para cada nodo del grafo. Existen múltiples enfoques para la selección, el más simple sería tomar aquella con la probabilidad más alta. Éste último es el que se utiliza en el proyecto actual.

2.2.9. Tecnologías Complementarias

2.2.9.1. SPARQL

SPARQL es un acrónimo recursivo de SPARQL Protocol and RDF Query Language. Con SPARQL los desarrolladores y usuarios finales pueden representar y utilizar los resultados obtenidos en las consultas a través de una gran variedad de información como: datos personales, redes sociales y metadatos sobre información. Al igual que sucede con SQL, es necesario distinguir entre el lenguaje de consulta y el motor para el almacenamiento y recuperación de los datos. Por este motivo, existen múltiples implementaciones de SPARQL, generalmente ligadas a entornos de desarrollo y plataformas tecnológicas. SPARQL contiene las capacidades para la consulta de los patrones obligatorios y opcionales de grafo, junto con conjunciones y disyunciones [PHS+08]. También soporta la ampliación o restricciones del ámbito de las consultas indicando los grafos sobre los que se opera. Los resultados de las consultas SPARQL pueden ser conjuntos de resultados o grafos RDF.

SPARQL tiene cuatro formas de consulta. Estas formas usan las soluciones de la coincidencia de patrones para formar conjuntos de resultados o grafos RDF. Estas formas son las siguientes:

- **SELECT**: Devuelve tuplas que incluyen todas o un subconjunto de las variables que coinciden con un patrón de consulta.
- **CONSTRUCT**: Devuelve un grafo RDF construido al sustituir las variables en un conjunto de plantillas de tripleta.
- **ASK**: Devuelve un valor booleano que indica si el patrón de consulta existe o no.
- **DESCRIBE**: Devuelve un único grafo RDF con datos sobre recursos.

2.2.9.2. DBpedia

El proyecto DBpedia es una herramienta para extraer información estructurada de Wikipedia y hacer esta información accesible en la Web, es una ontología construida a partir de los infoboxes de Wikipedia. Actualmente la base de conocimientos DBpedia describe más de 2,6 millones de entidades. Para cada una de estas entidades, DBpedia define un único identificador global con una descripción RDF de la entidad, incluyendo definiciones en 30 idiomas, relaciones con otros recursos, clasificaciones de cuatro jerarquías conceptuales, diversos hechos, así como enlaces a nivel de datos a otros datos de la Web y fuentes que describen la entidad.

Actualmente, un número cada vez mayor de editores de datos han comenzado a establecer enlaces de datos a nivel de recursos DBpedia , haciendo de DBpedia un concentrador de interconexión para la Web emergente. En la actualidad, la Red de fuentes de datos interrelacionadas alrededor de DBpedia proporciona aproximadamente 4,7 millones de piezas de dominios de información, y dominios tales como información geográfica, personas, empresas, música, medicamentos, libros y publicaciones científicas [BLK+09].

Está conformado por dos etapas:

- Spot : mediante modelos de lenguaje es capaz de establecer que sub-cadenas dentro de un texto son entidades.
- Desambiaguación: elije la entidad/concepto adecuado para las sub-cadenas calculadas en la etapa anterior, por ejemplo: pueden existir dos “Obamas”, entonces de acuerdo al contexto infiere si es Obama el político u otro Obama que podría ser deportista.

2.2.9.3. DBpediaSpotlight

DBpedia Spotlight es una herramienta para la anotación automática de recursos DBpedia mencionados en textos, proporcionando una solución para vincular las fuentes de información no estructurada a la nube a través de Linked Open Data. Reconoce cuando se han mencionado nombres de conceptos o entidades (por ejemplo, ”Michael Jordan”), y posteriormente, relaciona con estos nombres a los identificadores únicos (por ejemplo DBpedia: Michael_I._Jordan, la máquina de aprendizaje de profesor o DBpedia: Michael_Jordan, el jugador de baloncesto). También puede ser utilizada para el reconocimiento de entidades, extracción de frases clave, etiquetado, etc. La anotación de texto tiene el potencial de mejorar una amplia gama de aplicaciones, incluyendo la búsqueda y navegación por facetas. Mediante la conexión de documentos de texto con DBpedia, el sistema permite una variedad de casos de uso interesantes. Por ejemplo, la ontología se puede utilizar como conocimiento de fondo para mostrar la información complementaria en páginas web o para mejorar las tareas de recuperación de información. Por último, siguiendo los enlaces de DBpedia en otras fuentes de datos, la nube Linked Open Data se acerca más a la Web de Documentos [MJGSB11].

2.2.9.4. WordNet

WordNet es una ontología del idioma Inglés. Agrupa palabras en inglés en conjuntos de sinónimos llamados synsets, proporcionando definiciones cortas y generales, y almacena las relaciones semánticas entre los conjuntos de sinónimos. Su propósito es doble: producir una combinación de diccionario y tesauro cuyo uso sea más intuitivo, y soportar análisis automático de texto y a aplicaciones de Inteligencia Artificial.

WordNet se ha venido desarrollando desde los años 80 bajo la dirección del psicolingüista George Miller en la Universidad de Princeton. La unidad básica en la que se estructura WordNet es el synset o conjunto de sinónimos, el cual se considera representativo de un concepto lexicalizado. Así, en WordNet, las relaciones se establecen fundamentalmente entre conceptos, no entre palabras, asumiéndose que un concepto viene definido por el conjunto de formas léxicas que, en un contexto apropiado, sirven para representarlo en el lenguaje. Tal asunción implica la de una noción débil de sinonimia, la sinonimia

en contexto, una versión laxa de la noción tradicional de sinonimia -atribuida a Leibniz- según la cual dos unidades léxicas son sinónimas si la sustitución de una por la otra no produce en ningún caso alteración del valor de verdad de la proposición en la que aparecen. Por dicha razón, en WordNet se apuesta por la sinonimia en contexto como solución pragmática y realista que, aunque poco purista y, como veremos, siempre sujeta a matices e interpretaciones, permite afrontar la tarea de representar y tratar computacionalmente el conocimiento léxico-semántico de una lengua[16].

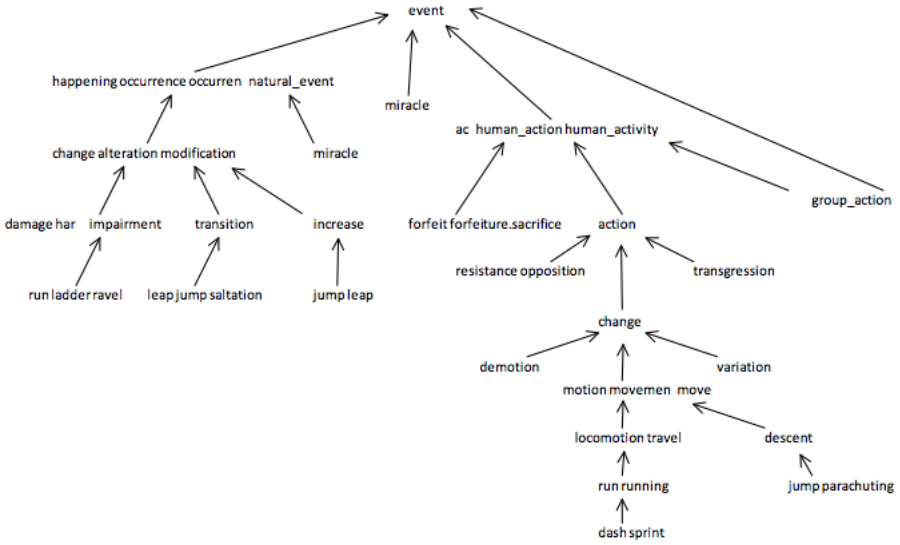


Figura 2.13: Representación de un synset.

[16]Tomado de http://elies.rediris.es/elies8/cap3-1.html

2.3. Marco de Antecedentes

El problema de la clasificación de textos cortos (a lo sumo 200 caracteres), y además informales por el hecho de originarse en las redes sociales, no ha sido muy abordado, caso contrario ocurre en textos largos y formales; como la categorización de artículos [BB63, MM03], filtros de spam[17] para correo electrónico [APK+00], identificación de entidades en documentos [FKMM05, NS07], identificación del tema principal en documentos [Guz98], entre otros. En ese orden de ideas, por ser aún un tema de investigación, son pocos los casos que se relacionan en esta sección. El agrupamiento automático de comentarios de opinión, no tiene un procedimiento definido a seguir, además, la mayoría de las investigaciones realizadas están basadas en textos en inglés. En algunos aspectos es posible hallar similitudes entre el tratamiento de textos en inglés y textos en español, sin embargo, la manera de abordar el problema de la clasificación difiere en gran medida dependiendo del idioma.

Referenciando estrictamente la clasificación de textos cortos en español, es necesario anotar que en la literatura no se encuentran casos específicos. Se ha trabajado sobre selección de términos índice [SPR05], sobre categorización semántica [RER10] y sobre detección de sentimientos (positivo, negativo) [yLNCyPMyAS12], pero no directamente sobre la agrupación por etiquetas.

2.3.1. Preprocesamiento de texto

La etapa inicial en el proceso de construcción de un clasificador, es el preprocesamiento (limpieza de los datos), en [PS02], un clasificador estomatológico[18], se divide en cuatro filtros esta fase: inicialmente se separa en frases el texto de acuerdo a signos de puntuación y conjunciones; posteriormente se eliminan las palabras no relevantes; después se eliminan las frases que no contienen palabras que existan en el diccionario de términos estomatológicos; finalmente se generan dos conjuntos, uno con palabras y otro con frases. Este desarrollo ofrece una ventaja al utilizar un diccionario con términos del dominio estomatológico en el proceso de limpieza, sin embargo no tiene en cuenta las irregularidades que podría tener el texto, por ejemplo, palabras incompletas o mal escritas. En [MCM+11], se hicieron pruebas de clasificación usando en el preprocesado las técnicas: $TF-IDF$, TF, TO, BTO; obteniendo como resultado que $TF-IDF$ ofrece el mejor rendimiento.

2.3.2. Técnicas de clasificación de texto

Existen múltiples algoritmos o técnicas para clasificar textos, su rendimiento está asociado a diversos factores, el tipo de texto (largo o corto, formal, científico), el ambiente de ejecución, las características (features) dadas, el uso, el tipo de usuario, entre otros. En [MCM+11] usan y comparan dos técnicas, SVM[19] y Naive Bayes[20], concluyendo que SVM es la mejor opción, basándose en los resultados obtenidos. También en [PS02] utilizan un enfoque supervisado con el que obtienen buenos resultados ajustando algunos parámetros.

Otro caso [PLV02], establece diferencias entre: Naive Bayes, Maximum Entropy y SVM, nuevamente mostrando que SVM es la técnica de mejor comportamiento. En este último caso, se utilizaron como línea base para identificar sentimientos, dos listas de palabras propuestas por personas, estando la primera conformada por expresiones *positivas* y la segunda por *negativas;* a pesar de esto, se determinó que en algunas ocasiones es necesario construir un clasificador mucho más sofisticado, por ejemplo: "La película es mala, es muy aburrida para el público general, sin embargo a mí me gustó." Aunque este comentario posee varias palabras *negativas*, es *positivo*. Finalmente en [SdB00], donde

[17] Correos no solicitados, no deseados, o de remitentes no conocidos, habitualmente de tipo publicitario.

[18] Entiéndase también como Odontología. Es una rama de la Medicina que se encarga del diagnóstico, tratamiento y prevención de las enfermedades del aparato estomatognático, que incluye los dientes, el periodonto , la articulación temporomandibular y el sistema neuromuscular. Y todas las estructuras de la cavidad oral como la lengua, el paladar, la mucosa oral, las glándulas salivales y otras estructuras anatómicas implicadas como los labios, las amígdalas, y la orofaringe.

[19] Support vector machine.

[20] Clasificador bayesiano ingenuo.

proponen como solución, tener un conocimiento semántico a priori de las palabras, SVM como en los anteriores, resulta ser el algoritmo con mejor rendimiento.

2.3.3. Resultados obtenidos

En [MBR12], se propone el desarrollo de una herramienta que realiza opinion mining sobre social media. Aunque tradicionalmente se ha usado aprendizaje automático para estas tareas, en dicho artículo, se desarrolló un enfoque modular basado en reglas, que lleva a cabo un análisis lingüístico superficial y se apoya en una serie de sub-componentes lingüísticos para generar la polaridad de la opinión. Para la evaluación de la herramienta se anotó manualmente un pequeño corpus[21] de 20 comentarios (que contienen sentimientos) de facebook[22] (en inglés) sobre la crisis financiera en Grecia. El sistema identificó oraciones con sentimientos con un 86 % de Precisión y un 71 % de Recall, y de estas oraciones identificadas correctamente, se obtuvo un 66 % de precisión en determinación de polaridad (positivo o negativo). El porcentaje de precisión no fue muy alto, sin embargo se destaca que los componentes como negaciones y sarcasmo aún requieren más trabajo. Por otro lado, el reconocimiento de nombres de entidades tuvo una precisión del 92 % y 69 % de Recall.

Daniel Preotiuc-Pietro en [PPSC+12], describe un framework[23] de código abierto, eficiente para el procesamiento de texto en tiempo real, originado en las redes sociales. Dicho framework soporta la inclusión de módulos extras para extender su funcionalidad. Al igual que la herramienta mencionada en el párrafo anterior, se encarga de procesar textos cortos (La información para las pruebas fue tomada de twitter[24]). La evaluación del sistema implementado arrojó los siguientes resultados:

Tiempo	Local	Hadoop	Total en Twitter
1 hora	0,51	7,6	10

Tabla 2.3: Número de tweets (en millones) creados y analizados en una hora.
(Tomada de [PPSC+12])

- **Detección del lenguaje:** La precisión fue del 89,3 % usando 2000 tweets. Esto muestra un buen comportamiento en la identificación.
- **Procesamiento en tiempo real:** La Tabla 2.3, muestra el comportamiento del procesamiento de tweets local y en hadoop. Ambos experimentos realizados el 10 de octubre de 2010. La herramienta local corre sobre un único núcleo y el hadoop corre en 6 máquinas, 84 núcleos virtuales y 42 físicos. Los resultados muestran que el hadoop, a pesar de ser relativamente pequeño, es capaz de procesar los tweets, casi que a medida que se van creando. Y un dato importante, es que puede analizar el 10 % de los tweets generados en una hora, en menos 10 minutos.

Por otro lado, en [SFD+10], se llevaron a cabo experimentos con la aplicación de la técnica Naïve Bayes en WEKA3 utilizando cross validation[25]. Los resultados obtenidos son aceptables, se afirma que la clasificación multi-etiqueta podria mejorar el rendimiento de la implementación. La siguiente figura describe el rendimiento del clasificador.

[21] Conjunto extenso de datos ordenados y clasificados, sobre un tópico, que sirve como base de una investigación.
[22] Red social disponible en: https://www.facebook.com/.
[23] Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.
[24] Red social disponible en: http://twitter.com. Una publicación o actualización en el estado de un usuario de Twitter se llama "Tweet".
[25] Técnica para estimar el rendimiento de un modelo predictivo.

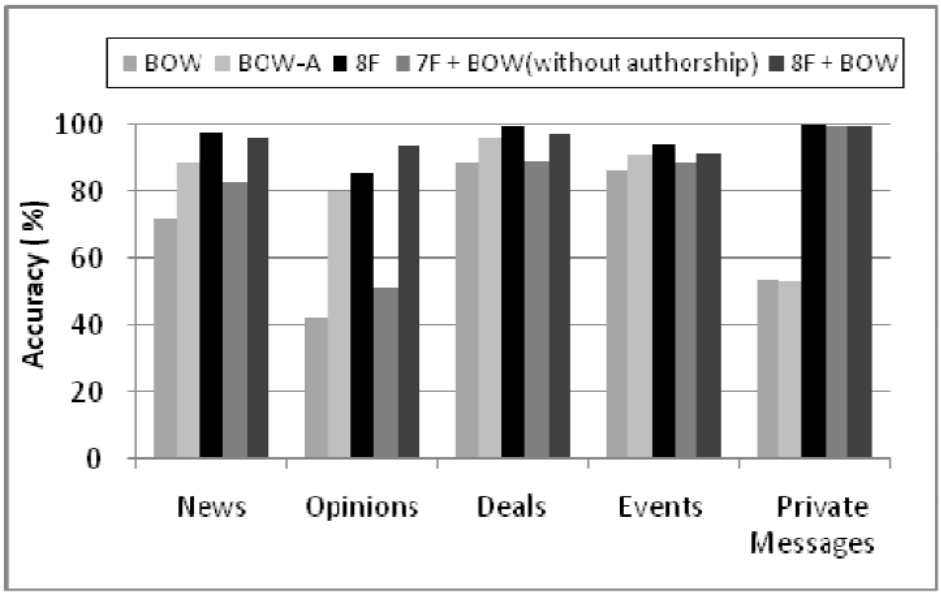


Tabla 2.4: Rendimiento del clasificador [SFD+10].

Capítulo 3

Diseño y Construcción del Sistema de Clasificación

En este capítulo se presenta el diseño que se propone como solución al problema. El capítulo está dividido en tres secciones. En la primera sección, se presenta la metodología de desarrollo utilizada en el trabajo. En la segunda sección se describen sus principales características. En la tercera, se muestran cada una de sus fases y los entregables de éstas.

3.1. Metodología SCRUM

Se escogió la metodología de desarrollo SCRUM, ya que al ser una metodología ágil se adapta a las características del proyecto, teniendo en cuenta que era necesario utilizar tecnologías nuevas y en lo que concierne a la capacidad de adaptación al cambio, SCRUM es ideal. Además de la entrega de avances continua en plazos breves de tiempo y de mantener un contacto permanente entre el cliente y el equipo de desarrollo, que en este caso sólo lo conformaba una persona. No obstante, siendo XP (eXtreme Programing) una metodología también ágil y de fácil acople al proyecto, no fue escogida porque una de las premisas de XP plantea que la codificación se debe ejecutar en un equipo conformado por dos personas, asimismo, los entregables en cuanto a documentación son relativamente pobres de acuerdo a las necesidades del presente proyecto.

Como complemento a la etapa de planeación descrita en SCRUM, se decidió anexar la especificación de requerimientos. Una descripción más detallada de SCRUM y los resultados de sus etapas se muestran en la siguiente sección.

3.1.1. Descripción de la Metología SCRUM

SCRUM es una de las metodologías de mayor propagación, junto con otras como RUP y XP. Es iterativa e incremental, enfatiza prácticas y valores del Project Management por sobre las demás disciplinas del desarrollo. Al principio del proyecto se define el Product Backlog, que contiene los requerimientos funcionales y no funcionales con los que deberá contar el sistema. Éstos se especifican de acuerdo a las convenciones propias de cada organización, aunque usualmente se utiliza un artefacto de alto nivel, cercano al usuario (como las historias de usuario) para describirlos. El Product Backlog se define durante reuniones con los interesados en el proyecto (Stakeholders). A partir de ahí se definen las iteraciones, conocidas como Sprint en la juerga de SCRUM, en la que se irá construyendo la aplicación evolutivamente. Cada Sprint tendrá su propio Product Backlog que será un subconjunto del Product Backlog definido inicialmente con los requerimientos a ser construidos en el Sprint correspondiente [RJ00].

3.2. Etapas de SCRUM

SCRUM describe el proceso de desarrollo en 3 etapas: planeación, desarrollo y liberación.

3.2.1. Planeación

La fase de planeación encierra la estimación y la arquitectura. En la planeación se tiene como artefacto principal el Product Backlog. En este se presentan los requisitos del sistema en un lenguaje no técnico, descritos en historias de usuario generalmente. Estos requisitos son priorizados por valor de negocio y revisados en intervalos regulares dentro del proceso de desarrollo. En lo que respecta al presente trabajo de grado, se definieron pequeños módulos internos de acuerdo a las necesidades identificadas en los requerimientos. Los requisitos del Product Backlog fueron priorizados de acuerdo a la complejidad que implicaba su desarrollo. Los requisitos más complejos contemplan una prioridad menor y los requisitos más básicos contemplan una prioridad mayor. Esta priorización permitió que las iteraciones que abarcaban requisitos básicos soportaran las iteraciones de requisitos más complejos.

3.2.1.1. Especificación de Requerimientos

Especificación de Requerimientos ISO-9000	**Documento:** ER-001	**Revisión:** 001
Especificación de Requerimientos		Fecha: 05/02/2013
Requerimiento	**Descripción**	
R1.1	El sistema debe recibir un archivo CSV e identificar si corresponde a un archivo del formato establecido por *Meridean* de tipo Lista de Comentarios.	
R1.2	El sistema debe procesar un archivo CSV identificado como correcto y normalizar los comentarios contenidos en él.	
R1.3	El sistema debe limpiar el texto de los comentarios normalizados.	
R1.4	El sistema a partir de un porcentaje de datos anotados, debe clasificar por categorías (etiquetas) los comentarios normalizados.	
R1.5	El sistema debe exportar un archivo XLS con los comentarios y sus etiquetas asignadas.	

Tabla 3.1: Requerimientos.

3.2.1.2. Producto Backlog

Pieza fundamental de SCRUM, se identifica cada característica básica a implementar y la prioridad que oscila entre 1 y 10.

Producto Backlog		**Documento**: PB-001	**Revisión**: 001	**Fecha**: 05/02/2013
ID	**Título**	**Descripción**	**Prueba de Aceptación**	**Prioridad**
H01	Como usario quiero que el sistema lea archivos que contienen comentarios y verifique su formato.	El sistema lee un archivo con comentarios.	El sistema identifica si el archivo es correcto o incorrecto.	8
H02	Como usuario quiero que el sistema limpie los comentarios que lea del archivo.	El sistema limpia el ruido que pueda haber en los comentarios.	El sistema elimina los emoticones y caracteres especiales de los comentarios.	7
H03	Como usuario quiero que el sistema, a partir de un porcentaje de datos anotados, etiquete los comentarios del archivo no anotados.	El sistema se entrena con los datos anotados y propaga la etiquetas a los no anotados.	El sistema asigna etiquetas para todos los comentarios no anotados.	10
H04	Como usuario quiero que el sistema exporte un archivo XLS con los comentarios y sus respectivas etiquetas asiganadas.	El sistema exporta un archivo XLS con los comentarios etiquetados.	El sistema almacena en disco local un archivo XLS con lo requerido.	8

Tabla 3.2: Producto Backlog.

3.3. Modelo del Clasificador

En esta sección se describen los diseños del clasificador implementados en las diferentes iteraciones. Por el carácter explorativo-investigativo del proyecto, desde el inicio no se tenía estimado un modelo único, ni se precisaba el alcance del mismo.

3.3.1. Diagramas de Flujo

El primer modelo se limitaba a cumplir con las características básicas de un clasificador, en el cual se desarrollan las etapas: *Preprocesamiento* → *Entrenamiento* → *Clasificación*. Además sólo se tenía previsto usar Label Propagation.

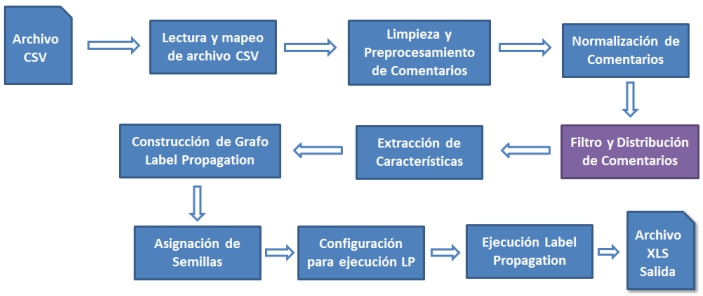


Figura 3.1: Digrama de Flujo Clasificador 1.0.

En el segundo y definitivo modelo, se añadieron las etapas de, evaluación de rendimiento (fase crítica e indispensable), extensión de comentarios y uso de SVM (Support Vector Machine) como técnica de clasificación paralela a Lable Propagation.

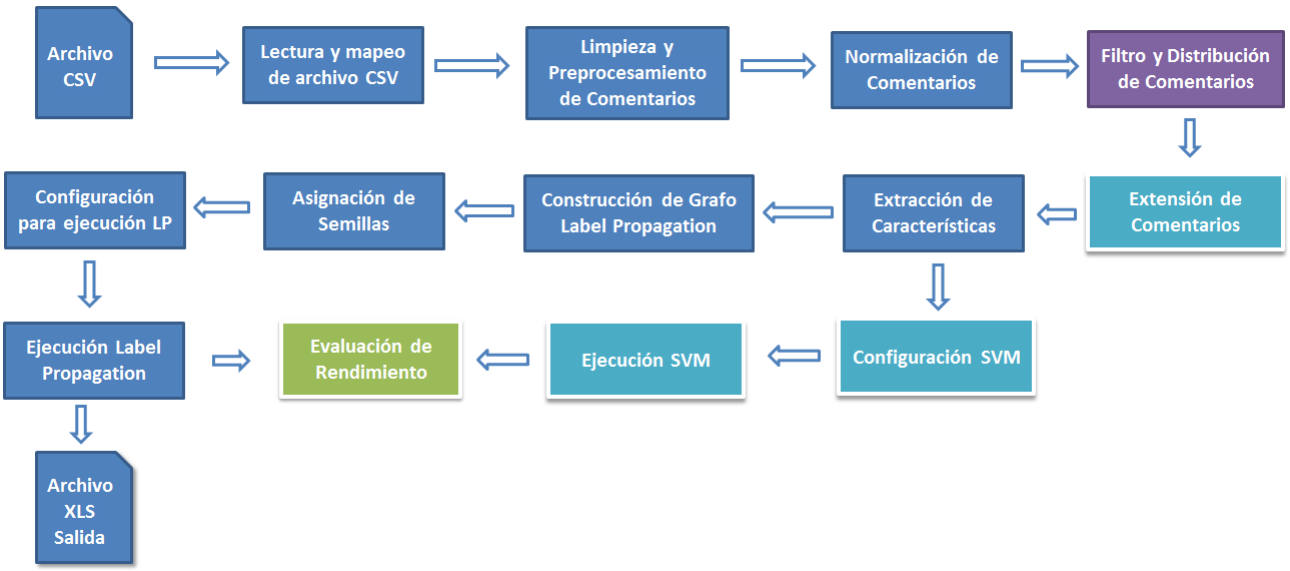


Figura 3.2: Digrama de Flujo Clasificador 2.0.

3.3.2. Descripción de Etapas

3.3.2.1. Lectura y Mapeo de archivo CSV

El primer paso corresponde a la lectura de un archivo CSV suministrado por *Meridean* que contiene una lista de comentarios (incluye información adicional como el autor, la fuente y en algunos casos, las etiquetas. Más adelante se desarrolla esta temática). Y posteriormente se realiza la transformación de texto plano a una estructura propia denominada Comentario.

3.3.2.2. Limpieza y Preprocesamiento de Comentarios

Antes de ejecutar los procesos directamente relacionados con la clasificación, es necesario realizar un preprocesamiento de los comentarios, para intentar dentro de lo posible, trabajar con información consistente, es decir, con comentarios que sólo contengan datos (palabras) críticas.

3.3.2.3. Normalización de Comentarios

En esta etapa, los comentarios son convetidos a listas de palabras, es decir, un comentario está conformado por múltiples palabras que forman un texto, dicho texto es separado atómicamente para convertirlo en un Comentario Normalizado.

3.3.2.4. Filtro y Distribución de Comentarios

El filtrado y la distribución, corresponden a una fase propia del presente trabajo. Debió incluirse debido a que era necesario identificar los casos que servirían de prueba para el clasificador. Por ejemplo, sólo se utilizan los comentarios anotados directamente por *Meridean* para probar la efectividad del sistema comparado con etiquetado manual.

3.3.2.5. Extensión de Comentarios

Es común enriquecer los términos de un texto por ejemplo, incluyendo sus sinónimos. Esta idea se emplea en diversos contextos. La expansión habrá de apoyarse en una fuente que disponga los términos relacionados para cada término. Se dispone de ricas fuentes de información léxica, como WordNet, estas son de caracter general y no abarcan dominios especializados. Se emplearon dos técnicas de extensión, WordNet y DBpedia. Si consideramos que los términos representan a cada texto, entonces los términos asociados representarán de una manera más rica a los textos.

La idea principal (en el contexto del presente proyecto) detrás de la extensión, es crear dimensiones (posiciones en el vector de palabras) comunes dado que los vectores están llenos de ceros (este concepto se entiende en las secciones posteriores).

3.3.2.6. Extracción de Características

Es este un pilar del proyecto, corresponde al primer objetivo específico y se encarga de particularizar cada comentario. Dicho de otro modo, corresponde a una descripción no-verbal ni escrita de un texto, por ejemplo, la cantidad de palabras, o la cantidad de letras. No debe confundirse con la extracción de palabras clave.

3.3.2.7. Uso de Label Propagation y SVM

Finalmente se decidió usar e implementar dos técnicas de clasificación, SVM y Label Propagation. A pesar de que representó un reto, contribuyó notoriamente en el proceso de análisis de resultados. Label Propagation fue elegida como la técnica de clasificación a implementar por su característica de ser semi-supervisada, lo cual sólo requería que un porcentaje relativamente pequeño de datos fuera anotado; esto teniendo en cuenta lo costosa que resulta la anotación manual en cuanto a tiempo y a recursos, que no sólo implica asignar una etiqueta, sino realizar todo el proceso de análisis y comparación con el conjunto de datos. Esta es la razón principal por la cual se elegió atacar el problema con Laber Propagation.

Finalmente, aunque no se encontraba dentro del alcance del proyecto, se implementó una versión sencilla del clasificador SVM, la cual fue usada para realizar comparaciones de comportamiento y rendimiento con respecto a Label Propagation, y teniendo en cuenta la naturaleza de los datos.

3.3.2.8. Evaluación de Rendimiento

El desarrollo del objetivo específico final, constituyó la construcción de todo un módulo (por así decirlo), en él se capturan los resultados del clasificador y se obtienen los indicadores de rendimiento.

3.3.3. Diagrama de Arquitectura

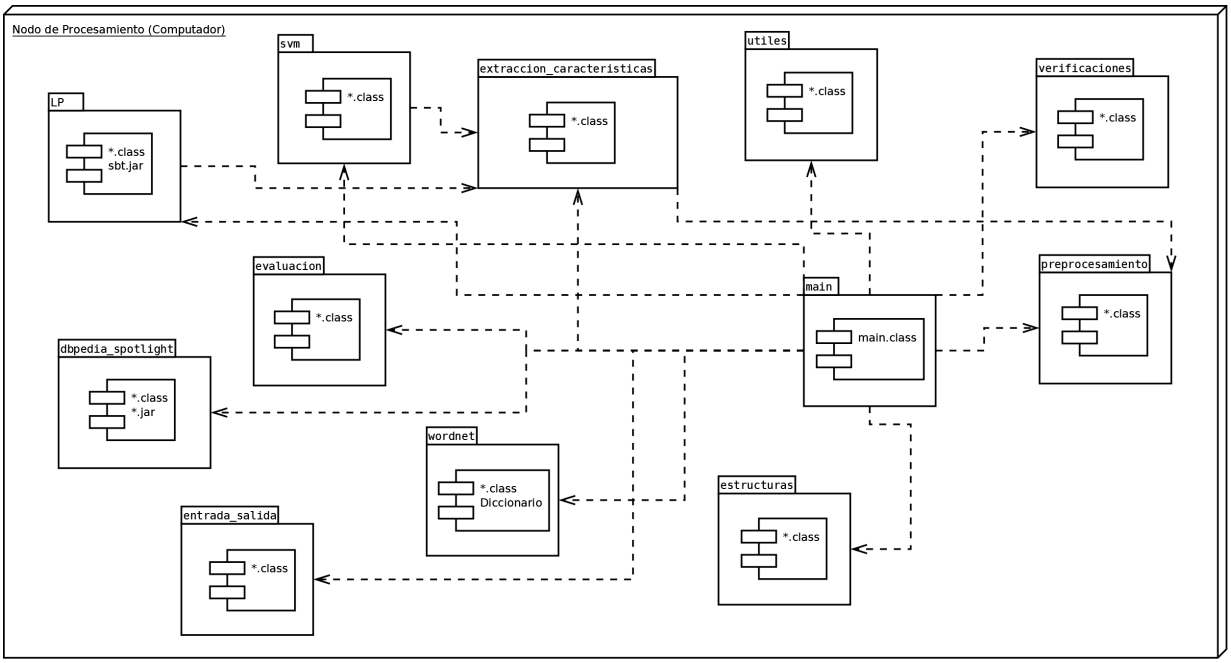


Figura 3.3: Digrama de Arquitectura

3.4. Construcción del Clasificador

Las especificaciones presentadas en esta sección corresponden únicamente a detalles de codificación. De antemano se anota que no se mencionarán pormenores de instalación, configuración y puesta en marcha, para ello referirse al Anexo A (Configuración y Puesta en Marcha). Adicionalmente, cabe aclarar que no se construyó una interfaz gráfica de usuario, debido a que el clasificador no lo ameritaba por ser una herramienta de ejecución única y lineal.

Sprint 1: Limpieza y preprocesamiento		**Documento**: SP-001	**Revisión**: 001	**Fecha**: 15/03/2013
ID	**Título**	**Prioridad**	**Duración**	**Responsable**
H01	Leer y mapear archivo CSV, construir limpieza, lematización y normalización.	9	5 semanas	Equipo de desarrollo

Tabla 3.3: Sprint 1: Limpieza y Preprocesamiento.

3.4.1. Lectura archivo CSV

Se construyó la clase *LeerArchivoCSV.java* para efectuar la lectura de un archivo de entrada en formato CSV el cual puede pertenecer a una de las siguientes estructuras:

- Estructura A:

Nombre	Mensaje	Campaña	Comunidad	Otros
DOLLY JAZMIN CARVAJAL ESPEJO	proteger· por 72 horas gracias a su acciÛn Fresh Dry, adem·s no tiene perfume lo cual permitir· que uses tus fragancias favoritas sin que el desodorante interfiera	x		
DOLLY JAZMIN CARVAJAL ESPEJO	proteger· por 72 horas gracias a su acciÛn Fresh Dry, adem·s no tiene perfume lo cual permitir· que uses tus fragancias favoritas sin que el desodorante interfiera	x		
DOLLY JAZMIN CARVAJAL ESPEJO	proteger· por 72 horas gracias a su acciÛn Fresh Dry, adem·s no tiene perfume lo cual permitir· que uses tus fragancias favoritas sin que el desodorante interfiera en estas. Te lo recomiendo. Adem·s, puedes hacer parte de la Comunidad Talk y		x	
DOLLY JAZMIN CARVAJAL ESPEJO	proteger· por 72 horas gracias a su acciÛn Fresh Dry, adem·s no tiene perfume lo cual permitir· que uses tus fragancias favoritas sin que el desodorante interfiera en estas. Te lo recomiendo. Adem·s, puedes hacer parte de la Comunidad Talk y			x

Figura 3.4: Estructura A - Archivo CSV

- Estructura B:

Autor	Mensaje	Fuente	Categoría 1
Adrian Contreras	:D woooo	facebook	otros
MRo Miguel	¡Pásenla suave!	Facebook	otros
Emile Contreras	A que horas se termiina el concurso! Marte	facebook	campaña
Comunidad Talk México	Adrian Contreras gracias por comprender, se hace un poco largo, acabamos	facebook	tecnologia
Comunidad Talk México	Adrian Contreras la sorpresa es q habrá tablas del Piso28 edición limitada	facebook	campaña
Le Poi Zix	aweeeeeeeeeeeeeeeebo sino no abria dmr mas ke meresidos	facebook	campaña
Shərəzada Adəzərəhs	Aww que lindoooos. Quería unos para mi bbezoteeeee :S	facebook	diseño
Ozcar Cruz	bambiii! chidooo! Iml :)	facebook	otros
Jorge Lupus	bien apretadito el kiñas jaja	Facebook	otros
Marte Haziel Guevara El	Buena idea mi carnal Marco Zet!!! ...	facebook	otros
Oscar Castañeda	bueno comunidad talk les contare mi historia son un chico de 18 años mi historia comenzó hace 3 años yo desde muy pequeño deseaba patinar	Facebook	historias
Comunidad Talk México	Cada 15 días se cuentan puntos desde qué inicia hasta cada campaña, los puntos son por calidad no por cantidad :)	Facebook	campaña
Miguel Rod Sb	Cm no esta mi nombre :'(	facebook	otros
Edwin Strempler	De echo a eso voy a salir a grabar hoy :D	facebook	otros

Figura 3.5: Estructura B - Archivo CSV

Además de la lectura, la clase se encarga de mapear cada una de las filas a un comentario de la clase *Comentario.java.* De cada comentarios se captura: el autor, el mensaje, y las etiquetas asociadas, sin embargo, sólo se trabaja con una de ellas, la primera en ser leída.

3.4.2. Limpieza y Preprocesamiento

Para limpiar los comentarios se creó la clase *Preprocesamiento.java* la cual se encarga de:

- Convertir a minúsculas cada una de las letras que componen un comentario.
- Elimina los acentos (tildes).
- Elimina las palabras ruido de las redes sociales, por ejempo: #LlegoDiciembre o @usuario.
- Eliminar las URL's.
- Eliminar las onomatopeyas, por ejemplo: "jaja", ""jeje", "ohh". En este punto fue necesario crear secuencias de identificación de patrones, para que se eliminara tanto "jaja" como "jjaaja".
- Eliminar caracteres diferentes a letras. Se filtran caracteres por código ASCII, es decir, se eliminan todos los caracteres que no esten en el abecedario del español y que sean diferentes a el caracter espacio.
- Eliminar las palabras vacías o stopwords[26], que para este caso fueron recopiladas tomando como base varias listas de la web.
- Eliminar espacios en blanco adicionales. Corresponden a aquellos que están al incio o final de una palabra y que no son el separadar entre la palabra anterior/siguiente y la actual.

[26]Es el nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc. que son filtradas antes o después del procesamiento de datos en lenguaje natural (texto).

Adicionalmente se construyó la clase *Lematizar.java* con la que se lematizan[27] cada una de las palabras resultantes que arrobajaba la ejecución del preprocesamiento. Para conseguir esta tarea, se utilizó una herramienta llamada *TreeTagger.*[28]

El paso posterior a la limpieza es la normalización de los comentarios, que en otras palabras corresponde a la transformación de los mensajes de cada comentario (o del resultado de la limpieza) en una lista de palabras, la cual es gestionada a través de *ComentarioNormalizado.java.*

Sprint 2: Distribución y Extensión de comentarios		**Documento**: SP-002	**Revisión**: 001	**Fecha**: 02/05/2013
ID	**Título**	**Prioridad**	**Duración**	**Responsable**
H01	Parametrizar la distribución de los paquetes, extender usando WordNet y usando DBPedia.	7	4 semanas	Equipo de desarrollo

Tabla 3.4: Sprint 2: Distribución y Extensión de comentarios.

3.4.3. Extensión de Comentarios

Se traduce en inyección de conocimiento, aunque a un nivel simple, no es más que un eriquecimiento del texto. Para conseguirlo, se crearon las clases *ListaPalabrasRelevantes.java* y *ExtenderComentarios.java.* Las cuales se encargan de identificar aquellas palabras que son relevantes en el contexto del mensaje, es decir: adjetivos, adverbios y sustantivos, para posteriormente ejecutar el proceso de extensión.
Herramientas usadas para extender:

- WordNet (GestionarWordNet.java)
  - Toma una palabra (en este caso de las que hayan sido identificadas como relevantes) y obtiene sus synsets, que dicho de otro modo, los synsets son sinónimos. Se procesa palabra por palabra, comentario por comentario.
- DBpedia (*GestionarDBPediaSpotlight.java* y *ConsultaSPARQL.java*)
  - La ejecución de DBpedia, sobre todo de DBpedia Spotlight es costosa y engorrosa. En este caso, se toman todas las palabras relevantes de un paquete de comentarios, éstas se escriben en un archivo plano, y sobre este archivo se ejecuta DBpediaSpotlight. El resultado es una lista de enlaces que corresponden a conceptos de la ontologia DBpedia. Por ejemplo: Palabras: “sony”, “iphone”, “pantalla”; Enlances: [http://dbpedia.org/page/sony], [http://dbpedia.org/page/apple]. [http://dbpedia.org/page/monitor] respectivamente.
  - Paso seguido, se realiza una consulta SPARQL sobre el concepto identificado y se obtienen de él los términos asociados a la categoría a la que pertenece dicho concepto. Por ejemplo: Concepto: Sony [http://dbpedia.org/page/sony]; Resultado: [Televisores, consolas, celulares...].

Ejemplos de extensión:

- Comentario original 1: “me encantan los animales domésticos”.

[27] El lema de una palabra es la palabra que nos encontraríamos como entrada en un diccionario tradicional: singular para sustantivos, masculino singular para adjetivos, infinitivo para verbos.

[28] *TreeTagger es una herramienta para la anotación de texto con la parte de discurso y la información lema. Fue desarrollado por Helmut Schmid en el proyecto de cooperación técnica en el Instituto de Lingüística Computacional de la Universidad de Stuttgart. http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/*

- Comentario original 2: “yo adoro los perros”.
- Comentario 1 extendido: “me encantan los animales domésticos **perro gato carnívoros amigables**”.
- Comentario 2 extendido: “yo adoro los perros **domésticos carnívoros**”.

Posteriormente las palabras de extensión (en negrita) son incluídas como nuevas dimensiones en el vector de palabras (bolsa de palabras) del paquete de prueba.

Es preciso aclarar que tanto WordNet como DBpedia son herramientas que sólo funcionan en inglés, por ende fue necesario construir un diccionario de palabras que tradujera de español a inglés los comentarios originales, y de esa manera usar los comentarios traducidos. Adicionalmente se construyó uno de inglés a español para traducir las palabras en inglés arrojadas por las herramientas. Ejemplo:

- Comentario original: “me encantan los animales domésticos”.
- Traducciones: [love, animals, domestic] → DBpedia y WordNet
- Resultado DBpedia y WordNet: [dog, cat, carnivorous, friendly] →[perro, gato, carnívoros, amigables]
- Comentario extendido: “me encantan los animales domésticos **perro gato carnívoros amigables**”.

3.4.4. Distribución de Comentarios

Por ser el clasificador un prototipo, era necesario trabajar sólo con datos anotadas para verificar qué tan efectivo era respecto a una muestra categorizada manualmente. Se construye la clase *DistribuirDatos.java,* donde se ofrecen diferentes tipos de distribución (cantidad de comentarios, cantidad de etiquetas por paquete, forma de distribución de etiquetas, entre otras).

Sprint 3: Extracción, clasificación y evaluación		**Documento**: SP-003	**Revisión**: 001	**Fecha**: 22/06/2013
ID	**Título**	**Prioridad**	**Duración**	**Responsable**
H01	Extracción de caracteristicas de comentarios, uso de herramientas de clasificación para propagación de etiquetas y evaluación de rendimiento del clasificador.	10	16 semanas	Equipo de desarrollo

Tabla 3.5: Sprint 3: Extracción, clasificación y evaluación.

3.4.5. Extracción de Caracteristicas

Este punto hace referencia al primer objetivo específico del presente trabajo, por tanto se indica que la caracterísca (feature) o las caracteríscas que se extrajeron, ellas corresponden al modelo conocido como “Bag of Words” (Bolsa de Palabras). El cual manifiesta que se debe tener una bolsa colectiva de todas las diferentes palabras contenidas en un paquete de prueba. Ejemplo:

- Comentario 1: hola amigos, me gusta jugar viajar
- Comentario 2: eyy donde estan?

- Bolsa de Palabras: [hola, amigos, me, gusta, jugar, viajar, eyy, donde, estan].

Posteriormente se generan los vectores de características para cada comentario, lo cual implica contar la frecuencia de cada palabra y asignar el valor obtenido en la posición donde se ecuentra la palabra en la bolsa, pero en un nuevo vector. Ejemplo:

- Bolsa de Palabras: [hola, amigos, me, gusta, jugar, viajar, eyy, donde, estan].
- Vector Comentario 1: [1, 1, 1, 1, 1, 1, 0, 0, 0]
- Vector Comentario 2: [0, 0, 0, 0, 0, 0, 1, 1, 1]

Se crearon las clases *GestionarVectorPalabras.java* y *GestionarDistancias.java*.

3.4.6. Uso de Label Propagation y SVM

Los vectores descritos en la subsección anterior, son los insumos para estas herramientas (internamente algoritmos) de clasificación.

- Label Propagation (*GestionarLabelPropagation.java* y *GestionarSemillasLP.java*)
  - Recibe la similitud (se usó la similitud coseno vectorial) entre cada par de comentarios. Por ejemplo, si se tienen 3 comentarios: N1, N2, N3; será necesario calcular las similitudes entre:
    - N1 - N2
    - N1 - N3
    - N2 - N3
  - Además recibe una lista con los comentarios anotados que van a servir para entrenar el clasificador.
- SVM (*SVM.java*)
  - Recibe cada uno de los vectores de características, es decir, por cada comentario se le debe pasar un vector. Además se debe indicar si pertenece o no a la categoría para la cual se está entrenando, por ejemplo:
    - N1 [0, 0, 0] - NO PERTENCE
    - N2 [1, 1, 0] - SÍ PERTENECE
    - N3 [1, 0, 1] - SÍ PERTENECE
  - En ese punto, después de recibir dichos parámetros, el clasificador se entrena y está en capacidad de predecir por ejemplo que el vector [0, 0, 1] no pertenece, en cambio el [1, 1, 1] sí pertenece.

Ajustando las condiciones de clasificación al caso de la empresa Meridean, un ejemplo de selección de ejemplos positivos y negativos se vería como a continuación:

Se requiere obtener un conjunto de ejemplos de entrenamiento sobre la etiqueta “Precio” a partir de una lista de comentarios

Comentario	**Tipo de ejemplo para la etiqueta “Precio”**
Su tamaño es excelente, son perfectos.	Ejemplo negativo
Me encantan porque son súper económicos.	Ejemplo positivo
Son baratos y buenos.	Ejemplo positivo
Que bueno que tengan un año de garantía.	Ejemplo negativo

3.4.7. Evaluación del Clasificador

Se construyó la clase *GestionarIndicadores.java* quien se encarga de leer el resultado arrojado por LP o SVM y traducirlo en tablas que representan los indicadores Precisión, Recall y Fscore, adicionalmente establece mendiante tablas, cuánto fueron los aciertos y cuántos los desatinos en valores reales.

3.4.8. Export de Archivo XLS

Al finalizar el proceso de clasificación, el prototipo genera un archivo XLS con los comentarios y sus respectivas etiquetas. Se asume que siempre ingresa un archivo con un porcentaje pequeño (10 % o 20 %) etiquetado y el resto debe generarlo el clasificador, sin embargo, por cuestiones de pruebas, siempre se trabaja con datos etiquetados en su totalidad (100 %), sólo que se simula la propagación automática sobre un porcentaje para comparar la anotación automática del prototipo contra la anotación manual.

3.4.9. Objetos JAVA

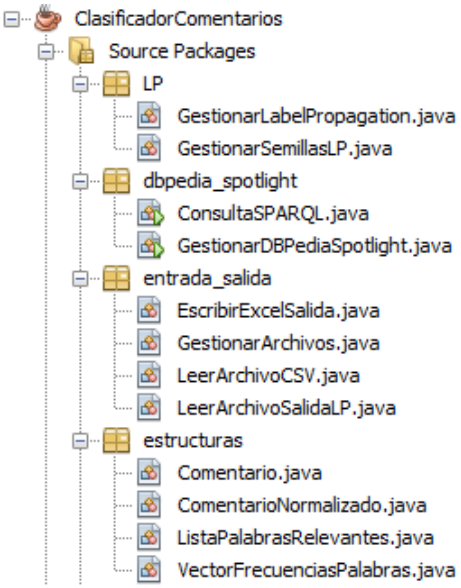


Figura 3.6: Diagrama de Objetos parte 1.

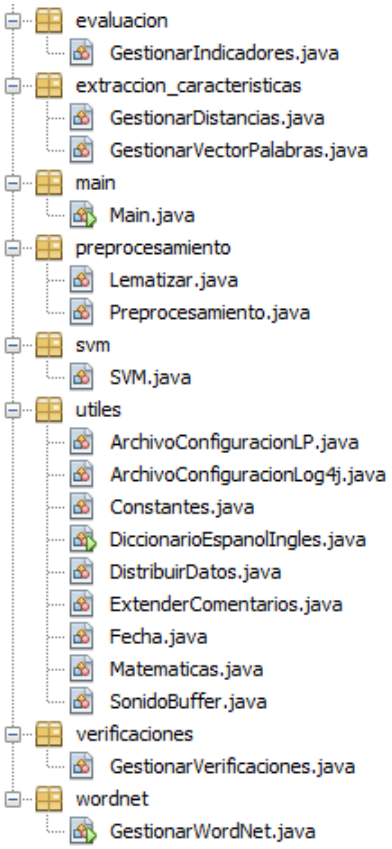


Figura 3.7: Diagrma de Objetos parte 2.

Capítulo 4

Sistemas de Clasificación

En este capítulo se detallarán los sistemas de clasificación obtenidos durante las iteraciones. Los primeros son los más ingenuos respecto a todas las etapas de procesamiento, y a medida que avanzan se hacen notorias las mejoras y la evolución de un sistema respecto al anterior. En esta división no se muestran pruebas ni resultados, únicamente se referencian los detalles de cada fase de los sistemas.

Es indispensable relacionar este capítulo con el siguiente, en el cual se describen las pruebas realizadas a los sistemas y los resultados obtenidos, para así comprender los cambios y la evolución de los sistemas de clasificación. Asimismo, para interpretar la dependencia de los cambios relacionada con las características de los paquetes de comentarios (paquetes de prueba) es necesario tener en cuenta el siguiente capítulo, debido a que al inicio del mismo se encuentra la caracterización de dichos paquetes.

4.1. Sistema v0.0

Corresponde al primer prototipo desarrollado. En este sistema se experimenta por primera vez con la herramienta Junto - Label Propagation[28]. Ver Tabla 4.1.

Característica	**Descripción**
Preprocesamiento	Limpieza sencilla. Los caracteres especiales se buscan y eliminan uno a uno.
Técnica de Clasificación	Label Propagation ▪ 10 Iteraciones ▪ Distancia euclidiana para calcular diferencia entre comentarios ▪ 10 % de comentarios anotados
Paquetes Usados	Un paquete usado: Kotex.
Distribución de Comentarios	Distribución Uniforme ▪ 10 Etiquetas del paquete
Extensión de Comentarios	No aplica.

Tabla 4.1: Descripción Sistema de Clasificación v0.0

[28]El cálculo de la diferencia entre dos comentarios se realizaba con la distancia euclideana, lo cual era erroneo, ya que el indicador de diferencia entre un comentario y otro era de similitud, es decir, entre mayor más similares, y con esta distancia se asumía lo contrario.

4.2. Sistema v0.1

Presenta dos cambios fundamentales, la mejora en el preprocesamiento y la corrección del cálculo de diferencias entre dos comentarios. Ver Tabla 4.2. A la nueva diferencia se le llamó "Distancia Euclidiana Invertida", la cual se convierte en una función de similitud y se calcula como se muestra a continuación:

Dada una distancia euclidiana actual d, y un umbral u que corresponde a la distancia euclidiana de mayor magnitud entre dos comentarios de un paquete, la distancia euclidiana invertida di se define como:

$$di = u - d$$

Característica	**Descripción**
Preprocesamiento	Limpieza mejorada. Se filtran caracteres por código ASCII para no tener en cuenta los diferentes a letras y espacios en blanco.
Técnica de Clasificación	Label Propagation ▪ 10 Iteraciones ▪ Distancia euclidiana invertida ▪ 10 % de comentarios anotados
Paquetes Usados	Dos paquetes usados: Kotex y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ 10 Etiquetas por paquete
Extensión de Comentarios	No aplica.

Tabla 4.2: Descripción Sistema de Clasificación v0.1

4.3. Sistema v1.0

Bajo la premisa de ensayo y error se determina que la mejor configuración para Label Propagation es 1 iteración. Además, se descarta el uso de la distancia euclidiana, en cambio se implementa la Similitud Coseno. Ver Tabla 4.3.

4.4. Sistema v1.1.0

En esta versión se evidencian cambios significativos. Además se agrega SVM (Support Vector Machine) como nueva técnica de clasificación, la cual fue implementada para realizar pruebas en paralelo junto a Label Propagation. En el sistema anterior se ejecutaban pruebas usando 4 paquetes, ahora en éste se consideran 8 paquetes. Es preciso anotar que se halló un problema en la lectura del archivo CSV y esto trajo como consecuencia la exclusión de una de las etiquetas de los siguientes paquetes: Balance, Coca Cola, Dog Chow y Don Julio[29]. Ver Tabla 4.4.

4.5. Sistema v1.1.1

Se registra sólo un cambio relevante, la selección de las etiquetas que se incluirán para realizar las pruebas. En versiones anteriores se tomaban las etiquetas en orden de aparición, ahora se ordenan

[29]Cuando se leían las etiquetas del archivo CSV se pasaba por alto la primera etiqueta. Si eran 10 etiquetas las que hacían parte de la categorización del paquete, sólo se leian 9.

Característica	**Descripción**
Preprocesamiento	Se añade la identificación y eliminación de patrones como: “jaja”, “jeje” y sus diferentes combinaciones.
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados
Paquetes Usados	Cuatro paquetes: Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ 10 Etiquetas por paquete
Extensión de Comentarios	No aplica.

Tabla 4.3: Descripción Sistema de Clasificación v1.0

Característica	**Descripción**
Preprocesamiento	Semejante al de la versión anterior. Se añaden algunas palabras vacías (Stop words) a la lista de las mismas.
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Proporcional ▪ Todas las etiquetas de cada paquete
Extensión de Comentarios	No aplica.

Tabla 4.4: Descripción Sistema de Clasificación v1.1.0

Característica	Descripción
Preprocesamiento	Igual que en la versión 1.1.0
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ 10 Etiquetas máximo por paquete. Las de mayor aparición.
Extensión de Comentarios	No aplica.

Tabla 4.5: Descripción Sistema de Clasificación v1.1.1

descendentemente por cantidad de apariciones, es decir, si la etiqueta E_1 aparece 10 veces y la etiqueta E_2 aparece 8 veces, la prioridad de inclusión la tiene E_1. Ver Tabla 4.5.

Este sistema tiene la mejor configuración comparado con sus antecesores, y es tomado como Sistema de Clasificación Línea Base. A partir del mismo se mide la evolución real de los clasificadores. El problema de la lectura del archivo CSV queda solucionado de aquí en adelante.

4.6. Sistema v1.2

Se implementa por primera vez la extensión de comentarios en pro de mejorar el rendimiento del prototipo. El sistema es un reflejo de la línea base trazada en la versión 1.1.1. Se probó la extensión con WordNet obteniendo mejoras parciales. Ver Tabla 4.6.

4.7. Sistema v1.3

Usando nuevamente la línea base del Sistema v1.1.1 se extienden los comentarios con DBpedia para inglés, La configuración para las variables se obtuvo a través de ensayo y error, identificando que los mejores resultados se conseguían con $confidence = 0{,}0$; y $support = 0{,}0$. Los casos de éxito son mínimos para este sistema. Ver Tabla 4.7.

4.8. Sistema v1.4

Para esta versión se extendieron los comentarios con DBpedia y WordNet. En las demás etapas se mantuvieron estables las caracteristicas. Ver Tabla 4.8.

Característica	Descripción
Preprocesamiento	Pequeñas mejoras en la limpieza de las palabras vacías (Stop words).
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ 10 Etiquetas máximo por paquete. Las de mayor aparición
Extensión de Comentarios	WordNet ▪ Se añaden sólo synsets de sustantivos ▪ Peso igual a 1 para las palabras de extensión

Tabla 4.6: Descripción Sistema de Clasificación v1.2

4.9. Sistema v2.0

En el sistema anterior se cumplía con el alcance del trabajo de grado, la implementación era fiel al diseño descrito en el Capítulo 3, y los resultados aunque despreciables según el alcance del proyecto, representaban particularidades aceptables. Sin embargo, con el fin de alcanzar una mayor precisión y además reconocer las causas de los resultados anteriores, se contruyó una nueva línea base.

Se materializó un cambio fundamental, la revisión exhaustiva de los paquetes de prueba (uno a uno) y por ende la identificación de etiquetas ruido[30]. Estas etiquetas fueron descartadas en esta versión para probar su influencia en el rendimiendo del prototipo. Ver Tabla 4.9.

4.10. Sistema v2.1

Siendo el Sistema 2.0 la nueva línea base, podría decirse que esta versión es un reflejo del Sistema 1.2. Ver Tabla 4.10.

4.11. Sistema v2.2

Cuando inició el desarrollo de los sistemas, la herramienta DBpediaSpotlight sólo funcionaba para el idioma *inglés*; en el andar, evolucionó permitiendo la desambiguación para distintos idiomas incluyendo *español*. Entonces, se incluyó en el sistema actual el uso de DBpediaSpotlight en español, siendo éste

[30]La descripción y el informe de identificación se encuentran en el siguiente capítulo.

Característica	Descripción
Preprocesamiento	Igual que en 1.2
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ 10 Etiquetas máximo por paquete. Las de mayor aparición
Extensión de Comentarios	DBpedia ▪ Peso igual a 1 para las palabras de extensión

Tabla 4.7: Descripción Sistema de Clasificación v1.3

el segundo aspecto relevante de la línea base 2.0 después de la eliminación de las etiquetas ruido. Ver Tabla 4.11.

4.12. Sistema v2.3

Al igual que en 2.2, se extiende con DBpedia, pero usando el desambiguador para textos en inglés. Ver Tabla 4.12.

4.13. Sistema v2.4

Permanecen estables las particularidades de sistemas anteriores, se extiende con WordNet y con DBpedia pero con la desambiguación en inglés debido a que los resultados obtenidos en las pruebas realizadas al Sistema 2.2 no fueron los esperados.

Es este el Sistema de Clasificación final, en él se obtiene el mejor rendimiento histórico y se dan por finalizadas las pruebas. El desarrollo del prototipo también finaliza. Ver Tabla 4.13.

Característica	**Descripción**
Preprocesamiento	Igual que en 1.2
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ 10 Etiquetas máximo por paquete. Las de mayor aparición
Extensión de Comentarios	DBpedia ▪ Peso igual a 1 para las palabras de extensión WordNet ▪ Se añaden sólo synsets de sustantivos ▪ Peso igual a 1 para las palabras de extensión

Tabla 4.8: Descripción Sistema de Clasificación v1.4

Característica	**Descripción**
Preprocesamiento	Igual que en 1.2
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ Todas las etiquetas de cada paquete ▪ Excluidas las etiquetas ruido de cada paquete
Extensión de Comentarios	No aplica.

Tabla 4.9: Descripción Sistema de Clasificación v2.0

Característica	**Descripción**
Preprocesamiento	Igual que en 1.2
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ Todas las etiquetas de cada paquete ▪ Excluidas las etiquetas ruido de cada paquete
Extensión de Comentarios	WordNet ▪ Se añaden sólo synsets de sustantivos ▪ Peso igual a 1 para las palabras de extensión

Tabla 4.10: Descripción Sistema de Clasificación v2.1

Característica	**Descripción**
Preprocesamiento	Igual que en 1.2
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ Todas las etiquetas de cada paquete ▪ Excluidas las etiquetas ruido de cada paquete
Extensión de Comentarios	DBpedia ▪ Peso igual a 1 para las palabras de extensión ▪ Pruebas con DBpediaSpotlight para español.

Tabla 4.11: Descripción Sistema de Clasificación v2.2

Característica	**Descripción**
Preprocesamiento	Igual que en 1.2
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ Todas las etiquetas de cada paquete ▪ Excluidas las etiquetas ruido de cada paquete
Extensión de Comentarios	DBpedia ▪ Peso igual a 1 para las palabras de extensión ▪ Pruebas con DBpediaSpotlight para inglés.

Tabla 4.12: Descripción Sistema de Clasificación v2.3

Característica	**Descripción**
Preprocesamiento	Igual que en 1.2
Técnica de Clasificación	Label Propagation ▪ 1 Iteración ▪ Similitud Coseno ▪ 20 % de comentarios anotados SVM ▪ Usado como clasificador lineal ▪ 20 % de comentarios anotados
Paquetes Usados	Ocho paquetes: Balance, Coca Cola, Dog Chow, Don Julio, Kotex, Nike, Sony y Top Terra.
Distribución de Comentarios	Distribución Uniforme ▪ Todas las etiquetas de cada paquete ▪ Excluidas las etiquetas ruido de cada paquete
Extensión de Comentarios	DBpedia ▪ Peso igual a 1 para las palabras de extensión ▪ Pruebas con DBpediaSpotlight para inglés. WordNet ▪ Se añaden sólo synsets de sustantivos ▪ Peso igual a 1 para las palabras de extensión

Tabla 4.13: Descripción Sistema de Clasificación v2.4

Capítulo 5

Pruebas y Resultados

En el presente capítulo, inicialmente se describen las características de los paquetes de datos de prueba, además de reseñar brevemente los temas tratados en cada uno. Posteriormente, como las pruebas están directamente ligadas a los Sistemas de Clasificación, se detallan los resultados obtenidos algunas versiones del prototipo.

5.1. Características de los Paquetes de Prueba

En el capítulo introductorio se indicó que el desarrollo del prototipo intentaba dar solución a un problema de una empresa llamada *Meridean,* en ese orden de ideas, los paquetes que se usaron para ejecutar las pruebas fueron proporcionados por dicha compañía.

Nombre	Reseña	Imagen
Balance	Balance es una de las marcas con mayor tradición del mercado colombiano de desodorantes antitranspirantes. Es la marca que mas unidades vende en todo el país.	
Coca Cola	Coca-Cola, es una gaseosa efervescente vendida en tiendas, restaurantes y máquinas expendedoras en más de 200 países. Es producido por The Coca-Cola Company.	
Dog Chow	Dog Chow es una marca de comida para perros fabricada por Nestlé Purina Petcare. La marca se ofrece en todo el mundo en numerosas fórmulas, incluyendo una para los perros jóvenes, llamado "Puppy Chow", una para los perros de edad avanzada, entre otras.	
Don Julio	Don Julio es una marca de tequila producido en México. Es la octava marca más grande en los Estados Unidos por el volumen y la undécima más grande de México. Es destilada, fabricada, embotellada, vendida y distribuida por Tequila Don Julio.	
Kotex	Kotex es una marca de productos de higiene femenina. Kotex es propiedad y está gestionado por Kimberly-Clark, una empresa presente en más de 80 países.	
Nike	Nike, es una empresa multinacional estadounidense conocida por su marca de ropa, calzado y otros artículos de deporte.	
Sony	Es una de las empresas más grandes del mundo, de origen japonés y uno de los fabricantes líder en la electrónica de consumo, el audio y el vídeo profesional, los videojuegos y las tecnologías de la información y la comunicación.	
Top Terra	Es una marca de productos de aseo, los cuales son fabricados en Colombia por Detergentes Ltda.	

Tabla 5.1: Breve reseña de los paquetes de prueba

5.1.1. Particularidades de los paquetes de datos originales

Particularidades extraídas de los paquetes tal y como fueron enviados por *Meridean.*

Particularidades	**Balance**	**Coca Cola**	**Dog Chow**	**Don Julio**	**Kotex**	**Nike**	**Sony**	**Top Terra**
Cant Comentarios	3923	22017	2054	1172	6502	5835	2377	5566
Cantidad Etiquetas diferentes	10	17	10	13	15	13	17	10
Comentarios repetidos	1074	1053	709	350	1761	1044	664	1470
Comentarios vacíos	34	129	78	8	40	137	18	20
Sin etiqueta	1264	16418	0	0	112	0	0	0
Comentarios con palabras repetidas en exceso	29	4	45	4	8	39	7	3
Comentarios útiles	1431	5246	1164	604	4581	4615	1639	4053

Tabla 5.2: Particularidades paquetes originales

5.1.2. Particularidades de los paquetes de datos útiles

Un paquete útil NO tiene comentarios vacíos (sin palabras)[29], todos los comentarios contenidos en él tienen etiqueta, todos son diferentes, y la frecuencia mínima por etiqueta es de 35 apariciones. No tienen comentarios en los que alguna de sus palabras se repita excesivamente[30].

Particularidades	**Balance**	**Coca Cola**	**Dog Chow**	**Don Julio**	**Kotex**	**Nike**	**Sony**	**Top Terra**
Cant Comentarios	1431	5246	1164	604	4581	4615	1639	4053
Cantidad Etiquetas diferentes	8	15	7	7	13	13	13	9

Tabla 5.3: Particularidades paquetes útiles

5.1.3. Particularidades de los paquetes de datos sin etiquetas ruido

Cada uno de los paquetes tenía al menos una etiqueta considerada "*Etiqueta Ruido*", la cual es aquella que no tiene un patrón definido, es decir, no categoriza determinantemente un comentario dentro de una división. Por ejemplo la etiqueta "otros" en varios casos fue considerada etiqueta ruido debido a que la clasificación en esta categoría es muy subjetiva y ambigua.

Particularidades	**Balance**	**Coca Cola**	**Dog Chow**	**Don Julio**	**Kotex**	**Nike**	**Sony**	**Top Terra**
Cant Comentarios	1355	4104	536	352	2115	1837	1073	3460
Cantidad Etiquetas diferentes	7	12	5	4	8	8	9	8

Tabla 5.4: Particularidades paquetes sin etiquetas ruido

[29]Un comentario podría quedar vacío después del preprocesamiento si por ejemplo solo tiene un emoticón en su contenido: ":)".

[30]Por ejemplo: "Hola, me gusto mucho mucho mucho mucho mucho mucho".

5.1.3.1. Etiquetas ruido identificadas

Etiquetas ruido

Etiquetas Balance	Cantidad
campaña	382
organico	355
protección	263
aroma	111
frescura	111
otros	76
seguridad	67
composición	66

Etiquetas Coca Cola	Cantidad
campaña	1004
momentos de consumo	987
hidratación	889
componentes	673
beneficios	356
recetas	268
burbujas	266
orgánico	209
presentaciones	137
energía	133
intención de consumo	99
comunidad talk	75
otros	63
referencias externas	45
rechazo al consumo	42

Etiquetas Dog Chow	Cantidad
campaña	545
nutrición	194
desarrollo	100
intención de uso	100
organico	99
otros	83
presentaciónes	43

Etiquetas Don Julio	Cantidad
campaña	116
otros	110
orgánico	101
momentos	97
efectos	72
sabor	67
comunidad	41

Etiquetas Kotex	Cantidad
mensajes de campaña	863
comodidad	711
otros	660
intencion de uso	551
producto genérico	376
seguridad	299
comunidad talk	208
empaque	193
intencion de consumo	193
presentaciones	184
experiencia de uso	135
componentes	111
aroma	97

Etiquetas Nike	Cantidad
generico	1013
campaña	951
otros	669
tecnologia	457
emociones	330
comodidad	283
diseño	225
resistencia	171
intencioncompra	156
skaters	133
historias	97
control	82
respaldomarca	48

Etiquetas Sony	Cantidad
otros	275
campaña	273
diseño	206
cámara	147
producto	125
comunidad talk	116
aplicaciones	104
pantalla	87
fotografía	74
sistema operativo	74
intención de compra	68
redes sociales	50
conectividad	40

Etiquetas Top	Cantidad
valor ecológico	975
aroma	941
intención uso	665
otros	593
componentes	358
cuidado ropa	246
empaque	174
publicidad	66
precio	35

5.2. Comparativo de Rendimiento de los Sistemas de Clasificación

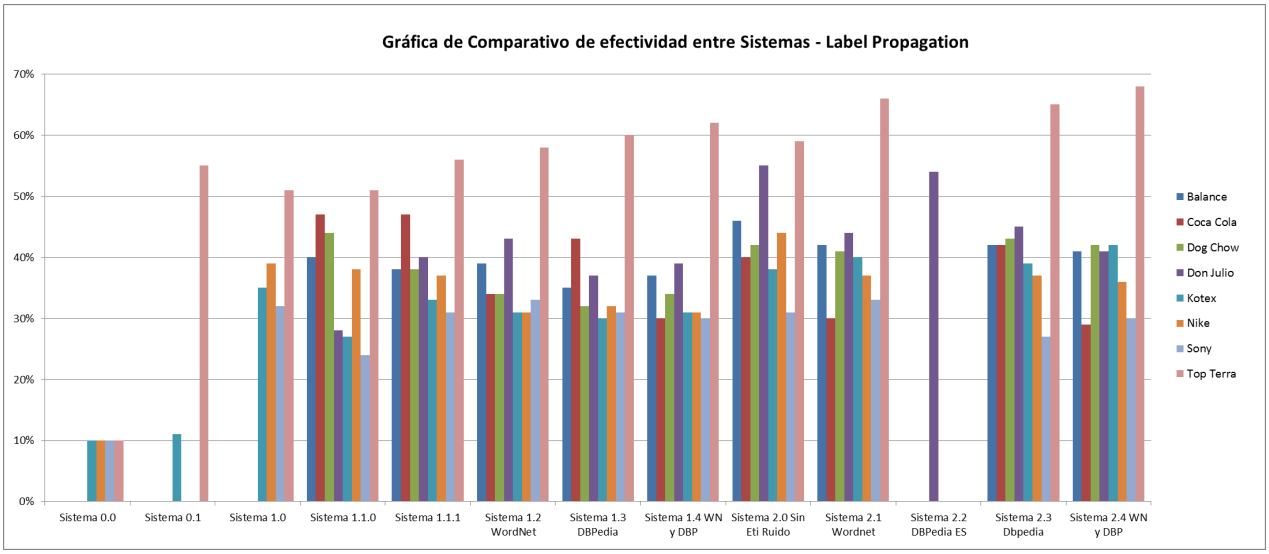


Figura 5.1: Rendimiento de Label Propagation en los Sistemas de Clasificación.

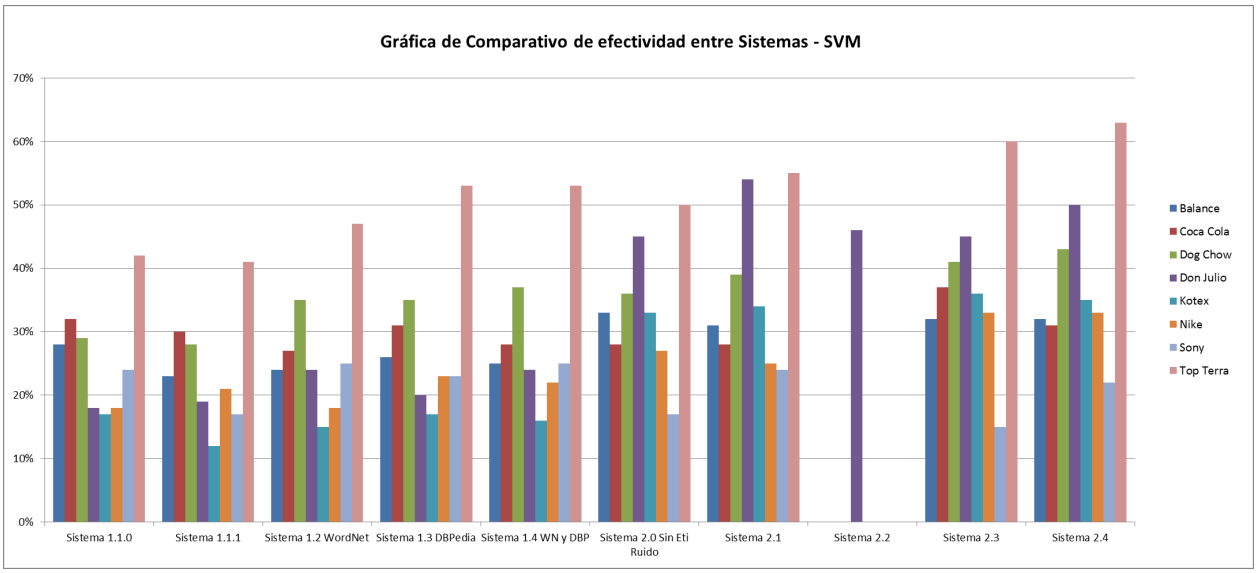


Figura 5.2: Rendimiento de SVM en los Sistemas de Clasificación.

5.3. Histórico de Pruebas y Resultados

A continuación se presentan las pruebas realizadas y los resultados obtenidos en algunos de los sistemas de clasificación. Cada reporte incluye una tabla que describe la información general[31] de la prueba ejecutada y una gráfica de los indicadores precisión, recall y fscore. Las gráficas indican el paquete, la versión del sistema y el algoritmo utilizado (LP: Label Propagation, SVM: Support Vector Machine).

No se incluyen en esta sección todos los reportes obtenidos como resultado generado por las pruebas (en los ANEXOS se pueden encontrar todos los reportes, sistema por sistema, LP y SVM). Sólo se incluyen la primera línea base, las mejoras derivadas a partir de la misma y las mejoras de la segunda línea base. Por otro lado, tampoco se incorporan los indicadores obtenidos en las pruebas con SVM ya que su rendimiento es inferior al de Label Propagation y su funcionamiento binario no se adapta al objetivo del prototipo.

5.3.1. Rendimiento Sistema v0.0

En el Sistema v0.0 no se generó reporte de rendimiento, debido a su baja efectividad y además porque al ser un prototipo ingenuo, el resultado obtenido correspondió a la asignación de una etiqueta a todos los comentarios, es decir, el paquete (100 comentarios) contenía 10 etiquetas distribuidas uniformemente (cada una correspondía a 10 comentarios) y el clasificador asigna una de las 10 para los 100 comentarios.

5.3.2. Rendimiento Sistema v0.1

Paquete Kotex	
Cant Comentarios	100
Porcentaje de datos anotados	10 %
Porcentaje de efectividad	11 %

Tabla 5.5: Prueba Kotex Sistema v0.1

Clasificador menos ingenuo que el 0.0, sin embargo con una baja efectividad.

[31] La efectividad de un sistema corresponde al porcentaje de aciertos, es decir, la cantidad de etiquetas puestas correctamente sobre el total.

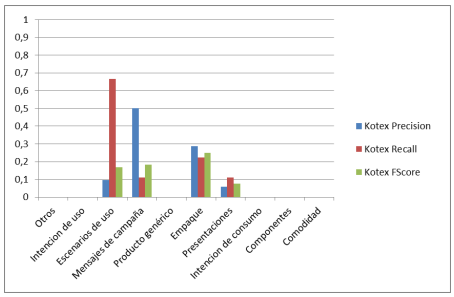


Figura 5.3: Indicadores LP Kotex 0.1

Paquete Top Terra	
Cant Comentarios	100
Porcentaje de datos anotados	10 %
Porcentaje de efectividad	55 %

Tabla 5.6: PruebaTop Terra Sistema v0.1

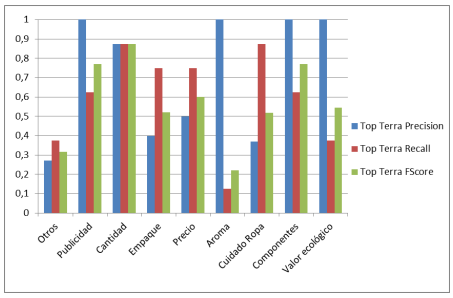


Figura 5.4: Indicadores LP Top Terra 0.1

5.3.3. Rendimiento Sistema v1.1.1

Primera línea base del sistema.

Paquete Balance	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	38 %

Tabla 5.7: Prueba Balance Sistema v1.1.1

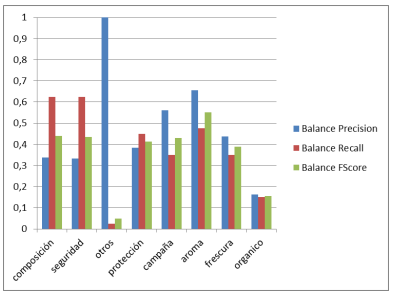


Figura 5.5: Indicadores LP Balance v1.1.1

Paquete Coca Cola	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	47 %

Tabla 5.8: Prueba Coca Cola Sistema v1.1.1

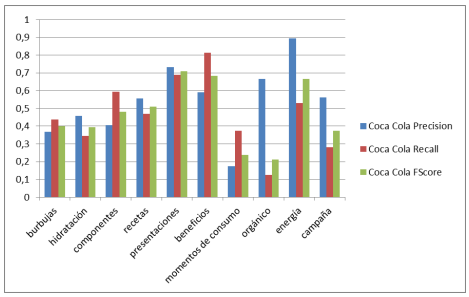


Figura 5.6: Indicadores LP Coca Cola v1.1.1

Paquete Dog Chow	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	38 %

Tabla 5.9: Prueba Dog Chow Sistema v1.1.1

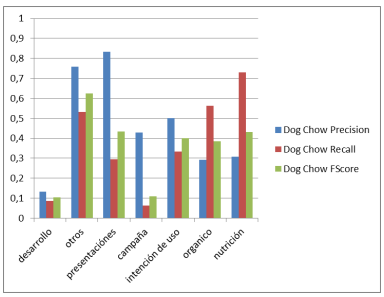


Figura 5.7: Indicadores LP Dog Chow v1.1.1

Paquete Don Julio	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	40 %

Tabla 5.10: Prueba Don Julio Sistema v1.1.1

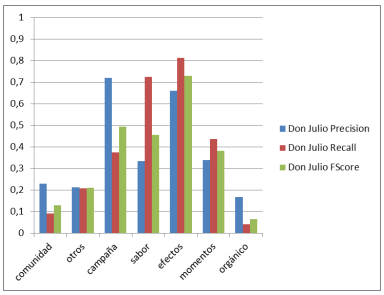


Figura 5.8: Indicadores LP Don Julio v1.1.1

Paquete Kotex	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	33 %

Tabla 5.11: Prueba Kotex Sistema v1.1.1

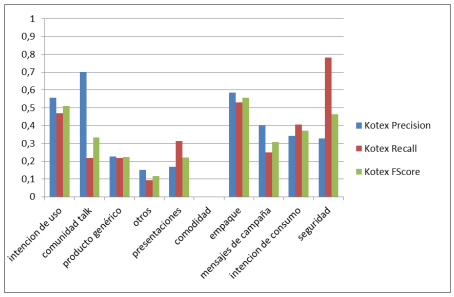


Figura 5.9: Indicadores LP Kotex v1.1.1

Paquete Nike	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	37 %

Tabla 5.12: Prueba Nike Sistema v1.1.1

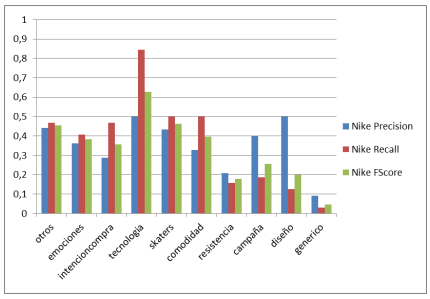


Figura 5.10: Indicadores LP Nike v1.1.1

Paquete Sony	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	31 %

Tabla 5.13: Prueba Sony Sistema v1.1.1

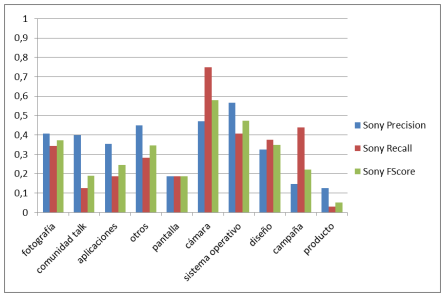


Figura 5.11: Indicadores LP Sony v1.1.1

Paquete Top Terra	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	56 %

Tabla 5.14: Prueba Top Terra Sistema v1.1.1

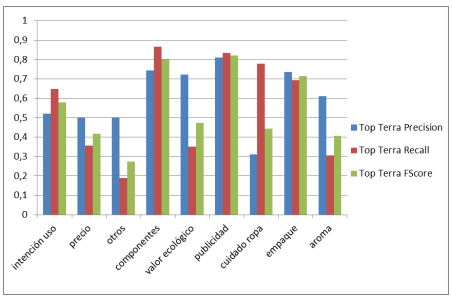


Figura 5.12: Indicadores LP Top Terra v1.1.1

5.3.4. Rendimiento Sistema v1.2

Paquete Balance	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	39 %

Tabla 5.15: Prueba Balance Sistema v1.2

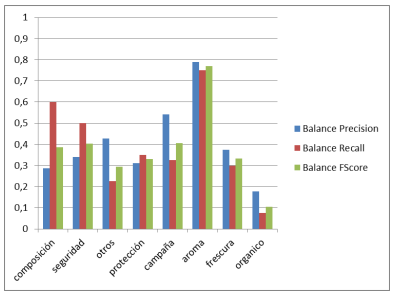


Figura 5.13: Indicadores LP Balance v1.2

Paquete Don Julio	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	43 %

Tabla 5.16: Prueba Don Julio Sistema v1.2

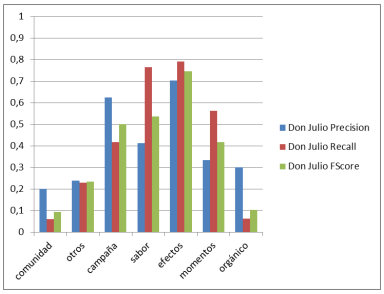


Figura 5.14: Indicadores LP Don Julio v1.2

Paquete Sony	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	33 %

Tabla 5.17: Prueba Sony Sistema v1.2

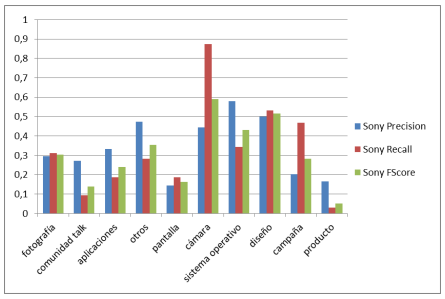


Figura 5.15: Indicadores LP Sony v1.2

Paquete Top Terra	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	58 %

Tabla 5.18: Prueba Top Terra Sistema v1.2

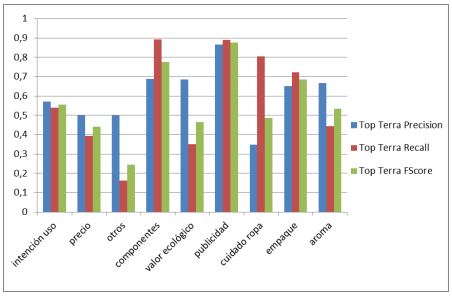


Figura 5.16: Indicadores LP Top Terra v1.2

5.3.5. Rendimiento Sistema v1.3

Paquete Top Terra	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	60 %

Tabla 5.19: Prueba Top Terra Sistema v1.3

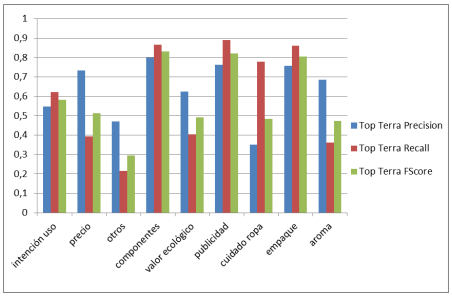


Figura 5.17: Indicadores LP Top Terra v1.3

5.3.6. Rendimiento Sistema v1.4

Paquete Top Terra	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	62 %

Tabla 5.20: Prueba Top Terra Sistema v1.4

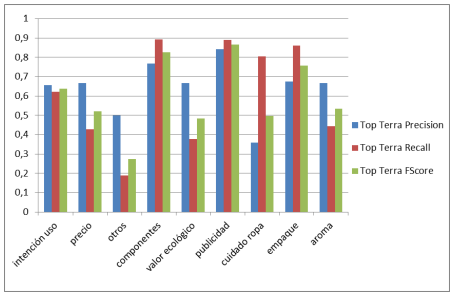


Figura 5.18: Indicadores LP Top Terra v1.4

5.3.7. Rendimiento Sistema 2.1

Paquete Kotex	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	40 %

Tabla 5.21: Prueba Kotex Sistema v2.1

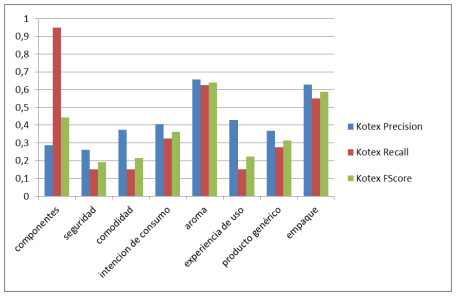


Figura 5.19: Indicadores LP Kotex v2.1

Paquete Sony	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	33 %

Tabla 5.22: Prueba Sony Sistema v2.1

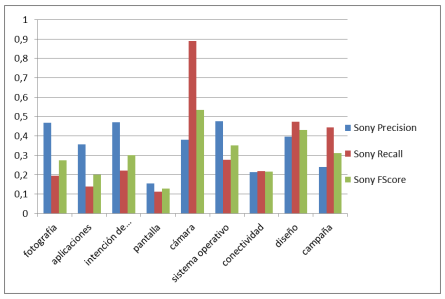


Figura 5.20: Indicadores LP Sony v2.1

Paquete Top Terra	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	66 %

Tabla 5.23: Prueba Top Terra Sistema v2.1

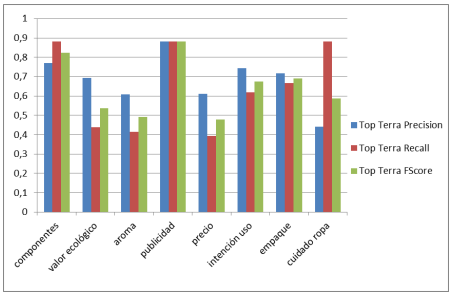


Figura 5.21: Indicadores LP Top Terra v2.1

5.3.8. Rendimiento Sistema v2.2

Paquete Don Julio	
Cant Comentarios	352
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	54 %

Tabla 5.24: Prueba Don Julio Sistema v2.2

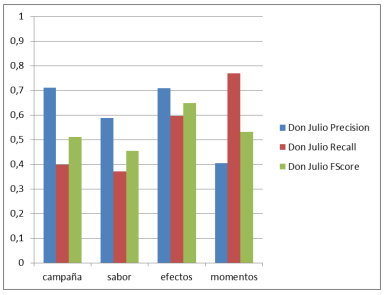


Figura 5.22: Indicadores LP Don Julio v2.2

5.3.9. Rendimiento Sistema v2.3

Paquete Coca Cola	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	42 %

Tabla 5.25: Prueba Coca Cola Sistema v2.3

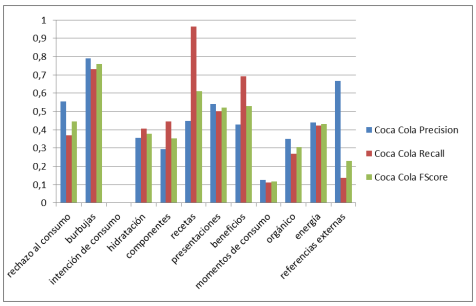


Figura 5.23: Indicadores LP Coca Cola v2.3

Paquete Dog Chow	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	43 %

Tabla 5.26: Prueba Dog Chow Sistema v2.3

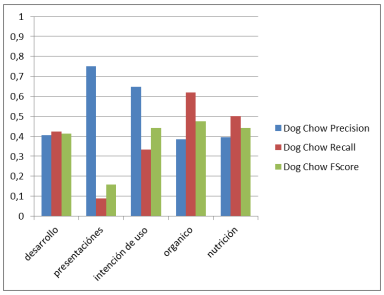


Figura 5.24: Indicadores LP Dog Chow v2.3

Paquete Kotex	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	39 %

Tabla 5.27: Prueba Kotex Sistema v2.3

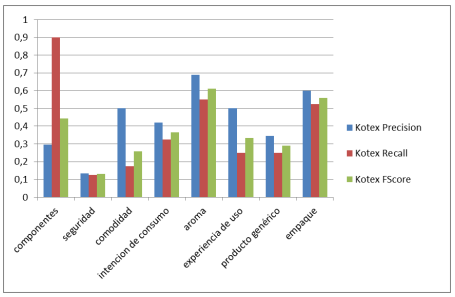


Figura 5.25: Indicadores LP Kotex v2.3

Paquete Top Terra	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	65 %

Tabla 5.28: Prueba Top Terra Sistema v2.3

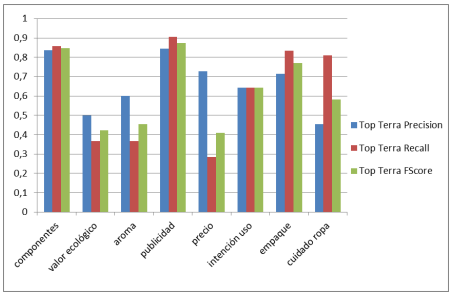


Figura 5.26: Indicadores LP Top Terra v2.3

5.3.10. Rendimiento Sistema v2.4

Paquete Kotex	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	42 %

Tabla 5.29: Prueba Kotex Sistema v2.4

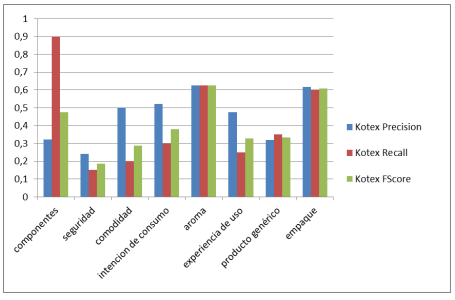


Figura 5.27: Indicadores LP Kotex v2.4

Paquete Top Terra	
Cant Comentarios	400
Porcentaje de datos anotados	20 %
Porcentaje de efectividad	68 %

Tabla 5.30: Prueba Top Terra Sistema v2.4

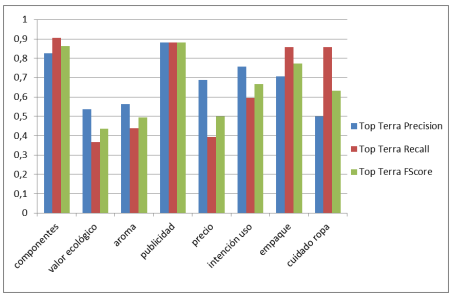


Figura 5.28: Indicadores LP Top Terra v2.4

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

A pesar de que los paquetes/archivos de prueba presentan una estructura similar visualmente, y hasta comparten etiquetas de clasificación (en más de uno aparece la etiqueta otros), el contenido, los temas tratados y los criterios de categorización son diferentes. En ese orden de ideas cabe mencionar que es más complicado identificar si un comentario de Sony habla de la cámara fotográfica del Xperia (referencia de un teléfono celular), que si un comentario de Top Terra habla de aroma.

Las ambigüedades son de mayor proporción y complejidad en algunos paquetes que en otros. Teniendo en cuenta esto, se puede inferir que la baja efectividad en los primeros sistemas (sobre todo en Sony) obedece a que se necesita un sistema que soporte una tarea compleja de desambiguación, sin embargo, por otro lado, el comportamiento de las pruebas sobre top terra, por ejemplo, fue siempre bueno, esto por sus temas relativamente sencillos y sus caracterizaciones marcadas.

La efectividad reducida alcanzada por el clasificador de comentarios obedece a diversos factores que se presentan a continuación:

- Resultó muy difícil evitar que se filtraran algunos comentarios casi idénticos (que diferían en dos o tres palabras), por el contrario, otros fueron difíciles de clasificar porque sólo tenían una palabra.
- Debido a que la eficacia en los resultados es directamente proporcional a la cantidad de datos de entrenamiento, los resultados con datos pequeños son significativamente menores.
- La inyección de conocimiento se vio limitada a causa de la herramienta DBPedia Spotligth, pues ésta cambiaba constantemente sin previo aviso y la ejecución cada vez difería de la anterior; por esto la versión que se presenta finalmente, no cuenta con ésta. Sería necesario adicionar recursos lingüísticos además de los dos mencionados anteriormente, buscando que ofrezcan un valor agregado y que sean lo suficientemente confiables como para no arriesgarse a perder o distorsionar el contenido crítico de cada comentario. Esto con el fin de obtener mejor resultados en cuanto a efectividad.
- A pesar de que se identificaron etiquetas de baja presición (aquellas que aparentemente eran muy difíciles de predecir) y se realizaron pruebas sin tenerlas en cuenta, se concluye que no es un escenario válido y que los resultados no son determinantes, ya que el lugar de esas supuestas etiquetas de baja precisón, lo ocuparon otras en las pruebas realizadas.
- Se presentó gran dificultad en la búsqueda de buenos “spell checker” (correctores ortográficos), sobre todo para textos cortos, informales y de temas indefinidos en el idioma español.

Entre otras cosas, se infiere que uno de los diferenciadores críticos entre los sistemas de clasificación es la cantidad de comentarios usados en las pruebas, no obstante en las versiones 1.2, 1.3 y 1.4 ocurre algo peculiar, y es que, aún relacionando la misma cantidad de datos de prueba y teniendo como premisa ser el siguiente mejor que el anterior, se observa que la extensión utilizando WordNet presenta el mejor resultado de los tres; lo cual ocurre debido a que el uso de DBpedia no está optimizado, (además de que se usa en inglés, traduciendo los comentarios) lo que genera es ruido en vez de enriquecer y desambiguar los comentarios.

Si se superpusieran las gráficas de Label Propagation y SVM que muestran la comparación de efectividad de los sistemas, se observaría que SVM presenta un comportamiento similar a Label Propagation, tanto así que en algunos casos es más efectivo. Sin embargo es necesario tener en cuenta que estos resultados obedecen a predicciones binarias, es decir, SVM sólo es capaz de decir si un comentario pertenece a una categoría dada o no.

Por ejemplo: se entrena SVM con 5 comentarios, donde C1, C2 y C3 pertenecen a la categoría "Aroma", pero C4 y C5 no pertenecen. Posteriormente se le pasa un comentario C6 de prueba y SVM indicará si pertenece a "Aroma" o no. Como se aprecia, los resultados de SVM hacen referencia a pruebas individuales para cada etiqueta de cada paquete, en cambio Label Propagation es capaz de asignar diferentes etiquetas a diferentes comentarios. Con lo anterior se descarta el uso y la supuesta competitiva efectividad de SVM.

A lo largo del proyecto se evidenció que Label Propagation, o en otras palabras, la clasificación semi-supervisada es la opción más viable (dar un vistazo a los datos no anotados, ayuda a encontrar estructuras de datos cuando los datos anotados son pocos), y se demostró que un sistema supervisado no es capaz de arrojar resultados superiores a LP en cuanto a rendimiento y bajo la conidición de trabajar con pocos datos anotados.

También es de destacar que los resultados obtenidos con el prototipo superan la elección aleatoria de categorías, y por tanto con varias mejoras sería probable que la empresa *Meridean* pueda automatizar la anotación de algunos de sus paquetes de información, teniendo en cuenta que deben presentar una estructura similar a la de los paquetes que obtuvieron los mejores resultados, como *Top Terra.*

6.2. Trabajo Futuro

Se podría decir que en cada etapa del desarrollo del clasificador quedaron varias cosas por mejorar (eso no quiere decir que el proyecto no haya dado ningún resultado, por el contrario, se realizó un gran trabajo) debido a que, para la envergadura del trabajo, el tiempo no fue suficiente y que, en algunos aspectos, se carecía del conocimiento necesario. Algunas de las carencias que este trabajo presenta, pueden convertirse, incluso, en objeto de un futuro trabajo de grado.

En lo concerniente a la etapa de preprocesamiento y lematización podría decirse que es una de las etapas más críticas junto a la extracción de características, y por ende una en la que hay mucho por hacer, como se especificó en el anteproyecto del presente trabajo y en algunas secciones anteriores, el clasificador realiza un preprocesamiento ingenuo (usando algoritmos propios y limitándose a lo estrictamente necesario) al cual le hace falta garantizar qué palabras mal escritas no van a generar ruido, y qué URL's incompletas u onomatopeyas imprevistas no van a propiciar ambigüedades en el momento de la clasificación. Sería necesario adaptar un corpus en español que apoye la tarea de corrección de palabras mal escritas o incompletas (un Spell checker), adicionalmente se requeriría una lista de palabras vacías mejor elaborada y más completa. Y como complemento final un lematizador más sofisticado y con corpus adaptables a contextos.

La extracción de características es la base fundamental del clasificador, ya que del resultado de la misma se alimenta el algoritmo o técnica de clasificación. Nuevamente, en un sentido estricto y exigente, el prototipo se queda corto ya que podría complementarse la extracción con técnicas sofisticadas

que asignen pesos a las palabras y así tener criterios diferenciadores y relevantes a la hora de clasificar.

En cuanto al núcleo de la clasificación, las técnicas semi-supervisadas, podrían implementarse diversas variaciones a las existentes (Label Propagation y SVM) y también ejecutar pruebas con algunas nuevas que aporten beneficios al problema de la categorización; por ejemplo deep learning, que es un conjunto de algoritmos de aprendizaje automático que intentan aprender en múltiples niveles, que corresponden a diferentes niveles de abstracción. Suele utilizarse en redes neuronales artificiales. Los niveles de estos modelos estadísticos corresponden a distintos niveles de conceptos, donde los conceptos de más alto nivel se definen a partir de los de nivel más bajo, y los mismos conceptos de nivel más bajo pueden ayudar a definir muchos conceptos de nivel superior [Ben09].

Por último y por supuesto importante, la inyección de conocimiento o extensión de comentarios. Como se observa a lo largo del proyecto, los resultados de las pruebas realizadas con DBpedia y WordNet no son los esperadas, es pertienente hacer una reevaluación de su uso y de sus características, y entonces de ese modo ajustarlas e incluirlas en un futuro prototipo. Además, se podría añadir un tesauro y/o un corpus de conocimiento contextual.

Referencias

[Abn07] Steven Abney. *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, 1st edition, 2007.

[APK+00] Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Georgios Sakkis, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. *arXiv preprint cs/0009009*, 2000.

[Bac12] Francis Bach. Learning with submodular functions. 2012.

[BB63] Harold Borko and Myrna Bernick. Automatic document classification. *J. ACM*, 10(2):151–162, April 1963.

[Ben09] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[BLK+09] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.

[BSH09] Michael J. Brzozowski, Thomas Sandholm, and Tad Hogg. Effects of feedback and peer pressure on contributions to enterprise social media. In *Proceedings of the ACM 2009 international conference on Supporting group work*, GROUP '09, pages 61–70, New York, NY, USA, 2009. ACM.

[CJTN06] Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 129–136. Association for Computational Linguistics, 2006.

[Cra99] *Commun. ACM*, 42(11), 1999.

[CSZ+06] Olivier Chapelle, Bernhard Schölkopf, Alexander Zien, et al. *Semi-supervised learning*, volume 2. MIT press Cambridge, 2006.

[Dir13] Marketing Directo. 50 definiciones de social media, 2013. [Internet; consultado 14-enero-2013].

[ELGF07] Kate Ehrlich, Ching-Yung Lin, and Vicky Griffiths-Fisher. Searching for experts in the enterprise: combining text and social network analysis. In *Proceedings of the 2007 international ACM conference on Supporting group work*, GROUP '07, pages 117–126, New York, NY, USA, 2007. ACM.

[FKMM05] Oscar Ferrández, Zornitsa Kozareva, Andrés Montoyo, and Rafael Muñoz. Nerua: sistema de detección y clasificación de entidades utilizando aprendizaje automático. *Procesamiento del Lenguaje Natural*, 35(37-44):94, 2005.

[GB00] Eduardo Gasca and Ricardo Barandela. Influencia del preprocesamiento de la muestra de entrenamiento en el poder de generalización del perceptron multicapa. In *6th Brazilian Symposium on Neural Networks*, 2000.

[GRF01] Jorge Graña Gil, Fco Mario Barcala Rodríguez, and Jesús Vilares Ferro. Etiquetación robusta del lenguaje natural: preprocesamiento y segmentación. *Procesamiento de Lenguaje Natural*, 27, 2001.

[Guz98] Adolfo Guzmán. Finding the main themes in a spanish document. *Expert Systems with Applications*, 14(1):139–148, 1998.

[Joa99] Thorsten Joachims. Making large scale svm learning practical. 1999.

[Loz96] Carlos Lozares. La teoría de redes sociales. *Papers*, 48(48):103–126, 1996.

[Luq03] Constantino Malagón Luque. Clasificadores bayesianos. el algoritmo naive bayes. *Universidad Nebrija*, 2003.

[MBR12] Diana Maynard, Kalina Bontcheva, and Dominic Rout. Challenges in developing opinion mining tools for social media. *Proceedings of@ NLP can u tag# user_generated_content*, 2012.

[MCM+11] Eugenio Martínez Cámara, María Teresa Martín, et al. Técnicas de clasificación de opiniones aplicadas a un corpus en español. 2011.

[Mer13] Meridean. Meridean — investigación, monitoreo, netnografía e insights en social media, 2013. [Internet; consultado 10-marzo-2013].

[MG06] Juan Julián Merelo Guervós. Redes sociales: una introducción. 2006.

[MGT03] I. Dan Melamed, Ryan Green, and Joseph P. Turian. Precision and recall of machine translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers - Volume 2*, NAACL-Short '03, pages 61–63, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[Mit97] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[MJGSB11] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

[MM03] Julia Sevilla Muñoz and Manuel Sevilla Muñoz. Una clasificación del texto científico-técnico desde un enfoque multidireccional. *Language Design: Journal of Theoretical and Experimental Linguistics*, (5):19–38, 2003.

[Møl93] Martin Fodslette Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533, 1993.

[NR97] Nigel NOR RIS. Evaluación, economía e indicadores de rendimiento. *Revista Electrónica HEURESIS. Universidad de Cádiz (www. uca. HEURESIS. es)*, 1997.

[NS07] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[PHS+08] Eric Prud Hommeaux, Andy Seaborne, et al. Sparql query language for rdf. *W3C recommendation*, 15, 2008.

[Pl99] Antonio Paños Álvarez. Reflexiones sobre el papel de la información como recurso competitivo de la empresa. *Anales de documentación, vol. 2, 1999*, 2:21–38, 1999.

[PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[PLV02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[PPSC+12] Daniel Preotiuc-Pietro, Sina Samangooei, Trevor Cohn, Nicholas Gibbins, and Mahesan Niranjan. Trendminer: An architecture for real time analysis of social media text. In *Proceedings of the workshop on Real-Time Analysis and Mining of Social Streams*, 2012.

[PS02] JI Peláez and P Sánchez. Un clasificador de texto por aprendizaje. *Inteligencia Artificial*, 6(15), 2002.

[Ram03] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[RC96] Gary S. Robinson and Carl Cargill. History and impact of computer standards. *Computer*, 29(10):79–85, 1996.

[RER10] María V Rosas, Marcelo L Errecalde, and Paolo Rosso. Un análisis comparativo de estrategias para la categorización semántica de textos cortos. *Procesamiento del lenguaje natural*, 44:11–18, 2010.

[RJ00] Linda Rising and Norman S Janoff. The scrum software development process for small teams. *Software, IEEE*, 17(4):26–32, 2000.

[SB98] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. Cambridge Univ Press, 1998.

[SdB00] Georges Siolas and Florence d'Alché Buc. Support vector machines based on a semantic kernel for text categorization. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 5, pages 205–209. IEEE, 2000.

[Sem12] José M. Sempere. Aprendizaje por refuerzo. 2012.

[SFD+10] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 841–842. ACM, 2010.

[Sie06] Basilio Sierra. *Aprendizaje Automático: conceptos básicos y avanzados*. 2006.

[SPR05] Héctor Jiménez Salazar, David Pinto, and Paolo Rosso. Uso del punto de transición en la selección de términos índice para agrupamiento de textos cortos. *Procesamiento del Lenguaje Natural*, 35, 2005.

[Vil09] José Mario Horcas Villarreal. Definición y evolución del comentario de texto. *Contribuciones a las Ciencias Sociales*, (2009-03), 2009.

[VMLC05] M Teresa Martín Valdivia, Antonio J Ortiz Martos, Luis Alfonso Ureña López, and Miguel Angel García Cumbreras. Detección automática de spam utilizando regresión logística bayesiana. *Procesamiento del Lenguaje Natural*, 35, 2005.

[WF05] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[Wik13a] Wikipedia. Estudio de mercado — wikipedia, la enciclopedia libre, 2013. [Internet; descargado 2-febrero-2013].

[Wik13b] Wikipedia. Etiqueta (metadato) — wikipedia, la enciclopedia libre, 2013. [Internet; descargado 2-abril-2013].

[Wik13c] Wikipedia. Sentiment analysis — wikipedia, the free encyclopedia, 2013. [Online; accessed 2-April-2013].

[Wil02] Rand R Wilcox. *Applying contemporary statistical techniques*. Academic Press, 2002.

[yLNCyPMyAS12] Antonio Fernández Anta y Luis Núnez Chiroque y Philippe Morere y Agustín Santos. Sentiment analysis and topic detection of spanish tweets: A comparative study of of nlp techniques. *Procesamiento del Lenguaje Natural*, 50(0), 2012.

[ZFM09] Arkaitz Zubiaga, Vıctor Fresno, and Raquel Martınez. Comparativa de aproximaciones a svm semisupervisado multiclase para clasificación de páginas web. *María Teresa Vicente-Díez, Paloma Martínez, Ángel Martínez-González*, 42:63–70, 2009.

[Zhu05a] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

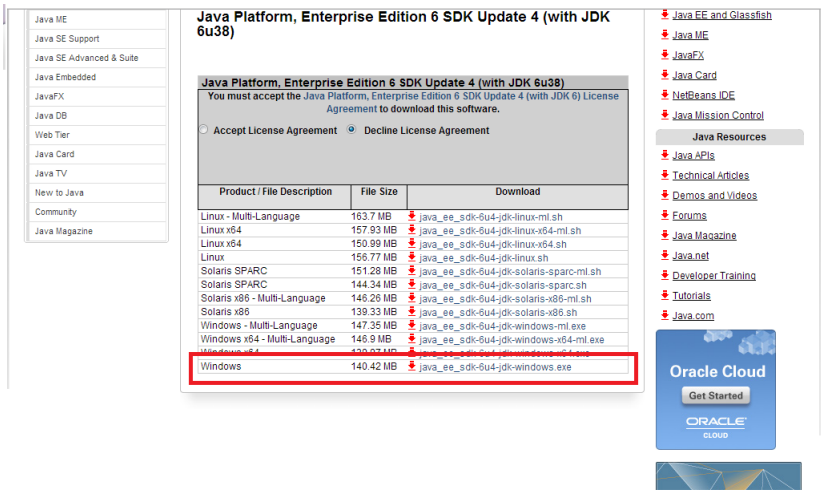
[Zhu05b] Xiaojin Zhu. *Semi-supervised learning with graphs*. PhD thesis, Pittsburgh, PA, USA, 2005. AAI3179046.

ANEXO A: Configuración y Puesta en Marcha del Clasificador

Requisitos de Configuración

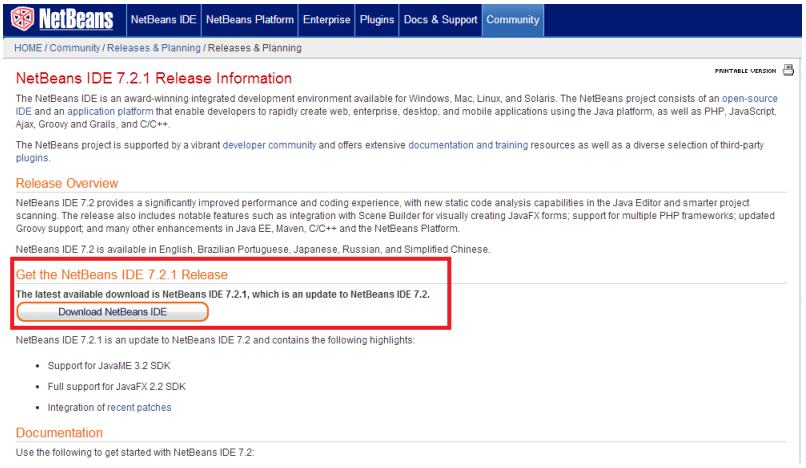
- Microsoft Windows XP o superior (Vista, 7, 8).
- Archivo CSV con comentarios
  - El encabezado debe tener una estructura similar a la de los archivos de prueba entregados en el disco óptico.
- Debe existir conexión a internet.
- Debe tenerse instalado Java JDK 1.6.

  http://www.oracle.com/technetwork/java/javaee/downloads/java-ee-sdk-6u3-jdk-6u29-downloads-523388.html



- Debe tenerse instalado Maven (Recomendable el que viene empaquetado con NetBeans 7.2.1).

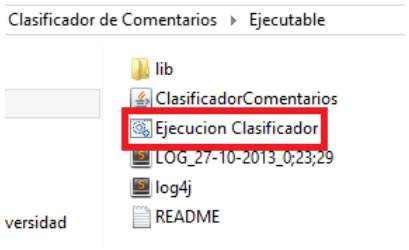
  https://netbeans.org/community/releases/72/



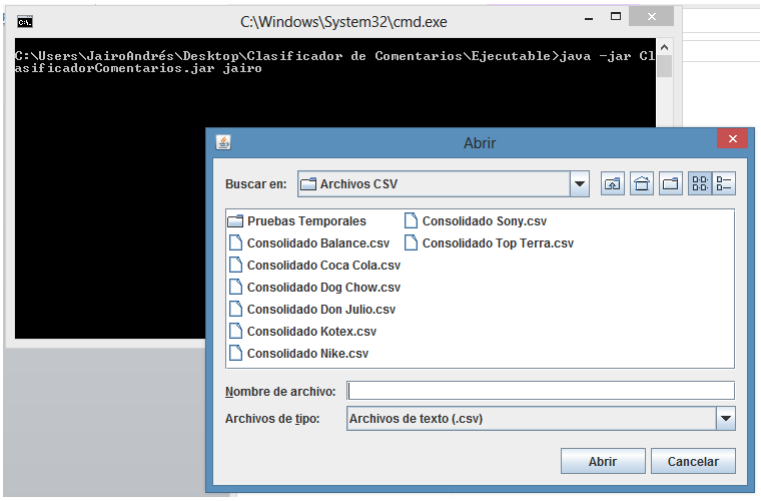
- Debe tenerse instalado Scala

Instrucciones de Puesta en Marcha

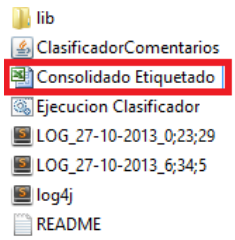
1. Tener el path de java y de maven. http://www.java.com/es/download/help/path.xml
2. Crear variable de entorno: JAVA_HOME= directorio de instalación. 2.Crearvariabledeentorno: JAVA_HOME=directoriodeinstalaci\penalty\@M\hskip\z@skip\unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{o\global\mathchardef\accent@spacefactor\spacefactor}\accent19o\egroup\spacefactor\accent@spacefactor\penalty\@M\hskip\z@skip\setbox\@tempboxa\hbox{o\global\mathchardef\accent@spacefactor\spacefactor}\spacefactor\accent@spacefactorn.
3. Copiar la carpeta Label Propagation a C:/
4. Incluir en el path el archivo sbt.bat que se encuentra en la ruta /Label Propagation
5. Ejecutar archivo: "Ejecucion Clasificador.bat" que se encuentra en la ruta /Ejecutable



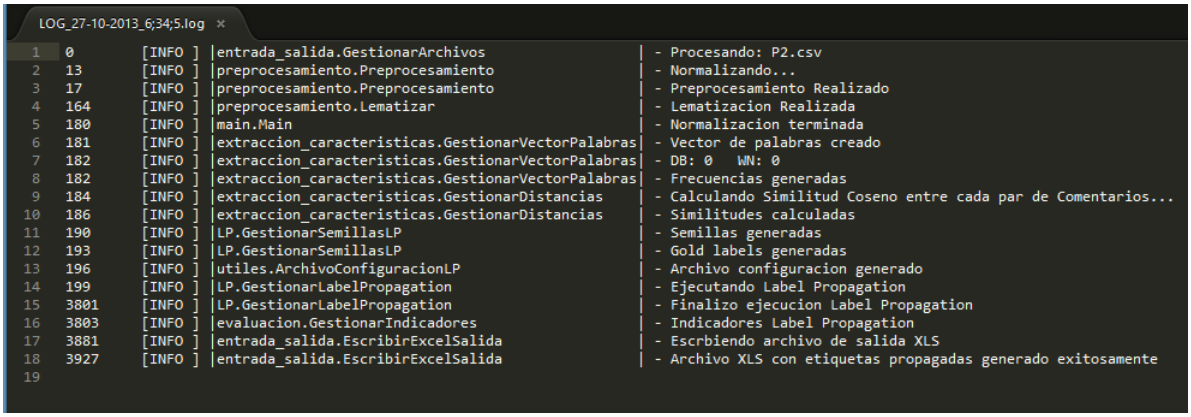
6. Se desplegará una consola y una ventana solicitando la ruta de un archivo de prueba CSV.



7. Finalmente la herramienta arroja un archivo XLS con los comentarios etiquetados por Label Propagation.



8. La herramienta genera un LOG de ejecución para visualizar los estados del proceso del calsificador.



LOG_27-10-2013_6;34;5.log

```
0       [INFO ] |entrada_salida.GestionarArchivos                 | - Procesando: P2.csv
13      [INFO ] |preprocesamiento.Preprocesamiento                 | - Normalizando...
17      [INFO ] |preprocesamiento.Preprocesamiento                 | - Preprocesamiento Realizado
164     [INFO ] |preprocesamiento.Lematizar                        | - Lematizacion Realizada
180     [INFO ] |main.Main                                         | - Normalizacion terminada
181     [INFO ] |extraccion_caracteristicas.GestionarVectorPalabras| - Vector de palabras creado
182     [INFO ] |extraccion_caracteristicas.GestionarVectorPalabras| - DB: 0   WN: 0
182     [INFO ] |extraccion_caracteristicas.GestionarVectorPalabras| - Frecuencias generadas
184     [INFO ] |extraccion_caracteristicas.GestionarDistancias    | - Calculando Similitud Coseno entre cada par de Comentarios...
186     [INFO ] |extraccion_caracteristicas.GestionarDistancias    | - Similitudes calculadas
190     [INFO ] |LP.GestionarSemillasLP                            | - Semillas generadas
193     [INFO ] |LP.GestionarSemillasLP                            | - Gold labels generadas
196     [INFO ] |utiles.ArchivoConfiguracionLP                     | - Archivo configuracion generado
199     [INFO ] |LP.GestionarLabelPropagation                      | - Ejecutando Label Propagation
3801    [INFO ] |LP.GestionarLabelPropagation                      | - Finalizo ejecucion Label Propagation
3803    [INFO ] |evaluacion.GestionarIndicadores                   | - Indicadores Label Propagation
3881    [INFO ] |entrada_salida.EscribirExcelSalida                | - Escrbiendo archivo de salida XLS
3927    [INFO ] |entrada_salida.EscribirExcelSalida                | - Archivo XLS con etiquetas propagadas generado exitosamente
```