BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Combinación de clasificadores para el análisis de sentimientos

TESIS DE MAESTRÍA

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

Monserrat Ramírez García

ASESOR:

Dra. Maya Carrillo Ruiz

Dr. Abraham Sánchez López

PUEBLA, PUE. Febrero, 2015

Resumen

El análisis de sentimientos es un campo de estudio que analiza las opiniones, evaluaciones y/o actitudes de las personas hacia entidades (productos, servicios, individuos). Actualmente la cantidad de información que expresa opinión, se ha incrementado, principalmente por la utilización de redes sociales, encuestas en línea, foros, microblogs, correos electrónicos, etc. Lo que ha generado interés en el análisis de sentimientos en diversas áreas de conocimiento como ciencias políticas, económicas y sociales. La industria ha mostrado interés en ésta área como un medio para conocer la opinión de los consumidores con respecto a los productos y servicios. De igual manera las organizaciones políticas han visto en el análisis de sentimientos, una herramienta valiosa para conocer la opinión de sus votantes.

En este trabajo, buscando mejorar el desempeño de la tarea de análisis de sentimientos utilizando el lenguaje español, se propone una arquitectura para fusionar clasificadores y mejorar el desempeño de los clasificadores base. Para este propósito se utilizó un corpus en español de 2625 opiniones. Mismo que fue preprocesado y representado como bigramas, bolsa de palabras con pesado *tf-idf*, etiquetado POS y una representación basada en la teoría de la valoración.

En la arquitectura propuesta se utilizan tres clasificadores base: máquina de soporte vectorial, naive bayes y arboles de decisión. Que fueron fusionados mediante tres métodos de combinación de clasificadores: mayoría de votos, cascada y ventanas.

Los resultados obtenidos muestran una mejora en medida F de hasta el 18.13 % en este corpus, con respecto a los resultados de los clasificadores base. Y del 12.52 % con respecto a los reportados por Eugenio Martínez, et al. [1], que utilizan el mismo corpus.

Adicionalmente se utilizaron dos corpus en inglés de 2000 y 10662 opiniones, con el objetivo de comprobar la validez de la arquitectura propuesta. Para estos corpus la mejora fue mínima del 0.13 % para el corpus de 2000 opiniones y del 0.04 % para el de 10662 opiniones, debido principalmente a los buenos resultados arrojados por los clasificadores base.

A pesar de los resultados favorables, que se han obtenidos con la fusión de clasificadores, en áreas como el procesamiento de imágenes y procesamiento de voz, para validar su eficacia en el análisis de sentimientos de textos en español, se requiere realizar pruebas adicionales.

Agradecimientos

En primer lugar deseo expresar un especial agradecimiento a mi asesora de tesis, la Dra. Maya Carrillo Ruiz, por la dedicación y orientación, por compartirme sus conocimientos y experiencia para la realización de este trabajo, y por toda la paciencia, atención y disposición brindada a mi persona, para la culminación de mi tesis. Muchas gracias Dra. Maya, realmente apreció bastante todo el apoyo, que me brindó.

También deseo expresar un sincero agradecimiento al Dr. Abraham Sánchez López, quien ha sido una pieza clave en mi formación de licenciatura, posgrado y ahora contribuyendo como co-asesor de mi tesis de maestría, muchas gracias por todo el apoyo que me ha brindado, en este trabajo y a lo largo de mi formación, por los regaños y palabras de aliento que me ha dado cuando ha sido necesario. Muchas gracias Doc. Abraham.

Quiero agradecer también a mi familia, que siempre me ha apoyado, principalmente a mi abuelita Francisca que ha sido una segunda madre para mi, a mis padres María Elena y Fernando que siempre han creído en mi y a mis hermanos Fernando y Brandon, quienes han sido un motivo para seguir adelante, para poder apoyarlos y ser un ejemplo para ellos. Muchas gracias a todos.

Por último y no menos importante, muchas gracias a mi querido esposo Héctor, quien siempre me brindó su amor, su apoyo, su comprensión, sus consejos y su paciencia, para que pudiera lograr esta meta. Muchas gracias Amor, por todo tu apoyo incondicional, este triunfo es de los dos.

Índice

Índice de Figuras

Índice de Tablas

Dedicada con mucho cariño a mi esposo y a mi familia...

Capítulo 1

Introducción

Hoy en día es muy común encontrar en redes sociales, blogs, microblogs, páginas web, entre otras, información u opiniones de los usuarios que expresan su punto de vista en internet acerca de algo. Dicho fenómeno ha generado interés por el análisis de sentimientos (AS), una área del procesamiento de lenguaje natural (NLP) que se encarga de identificar opiniones relacionadas con un objeto. [2] El interés proviene, de que un factor determinante para la toma de decisión de las personas es precisamente la opinión, por ejemplo cuando compramos algún producto en internet, queremos conocer la opinión de los demás acerca del producto que deseamos adquirir. De la misma manera, las empresas tienen como objetivo encontrar indicadores que contribuyan a cubrir las necesidades de sus clientes, mejorando sus productos y servicios, lanzando al mercado productos que con base en las opiniones adquiridas, prefieran o soliciten sus clientes, y estar atentos de su posicionamiento con respecto a la competencia. En el ambiente político es importante conocer la opinión de las personalidades públicas, elegir la propaganda idónea según las preferencias u opiniones de la gente u otro claro ejemplo son los cibernautas, los cuales buscan elegir un producto mejor valorado por los demás usuarios en internet. [3].

La utilidad de evaluar la opinión pública usando análisis de sentimiento sobre opiniones digitales permite la sustitución de los medios tradicionales como las encuestas y estudios de campo.

Dada la importancia del análisis de sentimientos, establecida en los párrafos anteriores, en esta investigación se explora la Combinación de Clasificadores para realizar Análisis de Sentimientos.

1.1. Objetivos

1.1.1. Objetivo General

- Mejorar la precisión de la tarea de análisis de sentimiento, producida por varios clasificadores base, definiendo una arquitectura para combinar los resultados de los mismos

1.1.2. Objetivos Particulares

- Analizar diferentes Métodos de Análisis de Sentimientos.
- Probar con diferentes clasificadores base y determinar cuales se comportan de mejor manera para el problema planteado.
- Identificar una arquitectura, que permita combinar clasificadores, aplicable al análisis de sentimientos.

La estructura del documento se describe a continuación:

1.2. Estructura de la tesis

La estructura del documento se presenta de la siguiente manera: en el capítulo 2 se presentan los fundamentos teóricos para la realización de este trabajo, el capítulo 3 contiene los trabaos relacionados con respecto a las áreas abordabas, el análisis de sentimientos y la combinación de clasificadores, en el capítulo 4, se presenta la arquitectura de combinación de clasificadores propuesta, posteriormente en el capítulo 5 se muestran los experimentos y resultados, en el capítulo 6 se presentan las conclusiones y finalmente la bibliografía utilizada.

Capítulo 2

Análisis de Sentimientos

En este capítulo se describen todos los conceptos contextuales, abordados a lo largo de la investigación, se presentan los fundamentos teóricos para la realización de la tesis en el área de análisis de sentimientos, representación de los textos, clasificadores base, y combinación de clasificadores.

2.1. Análisis de Sentimientos

Las opiniones son fundamentales para casi todas las actividades humanas, porque son importantes factores de influencia en nuestros comportamientos.

El análisis de sentimientos (AS), también llamado minería de opinión es un campo de estudio que analiza las opiniones, sentimientos, evaluaciones, actitudes de las personas hacia entidades como productos, servicios, organizaciones, individuos, cuestiones, eventos, tópicos y sus atributos. Dicha área tiene un amplio rango de aplicaciones casi en todos los dominios, la industria es la más interesada en la proliferación de aplicaciones comerciales, además sin mencionar que por primera vez en la historia, se cuenta con una gran cantidad de información en los medios de comunicación social y la web en general, lo cual ha originado que el análisis de sentimientos sea el centro de investigación de los medios sociales. La investigación de AS no solo ha tenido un gran impacto en el Procesamiento de Lenguaje Natural *(NLP)* , si no también ha tenido un profundo impacto en las ciencias de la administración: ciencias políticas económicas y sociales.

El análisis de sentimientos se conoce también como clasificación de sentimientos, o *clasificación de sentimiento a nivel documento*, ya que se considera todo el documento como una unidad. El problema se define de la siguiente manera:

Dado un conjunto de documentos de texto de evaluación D que contienen opiniones (o sentimientos) acerca de objetos, se pretende extraer atributos y componentes de objetos que han sido comentados en cada documento d en D y determinar si los comentarios son positivos, negativos o neutros [4].

Consta de dos pasos: Identificación y Clasificación [5].

- Identificación: Se identifican las partes con enunciados subjetivos en el texto (opiniones).
- Clasificación : Se realiza la clasificación de las opiniones: Positiva, Neutra y Negativa.

En la figura 2.1 se muestra el proceso del análisis de sentimientos que consta de 3 fases, la obtención de las opiniones, la identificación de subjetividad y la clasificación de las opiniones.

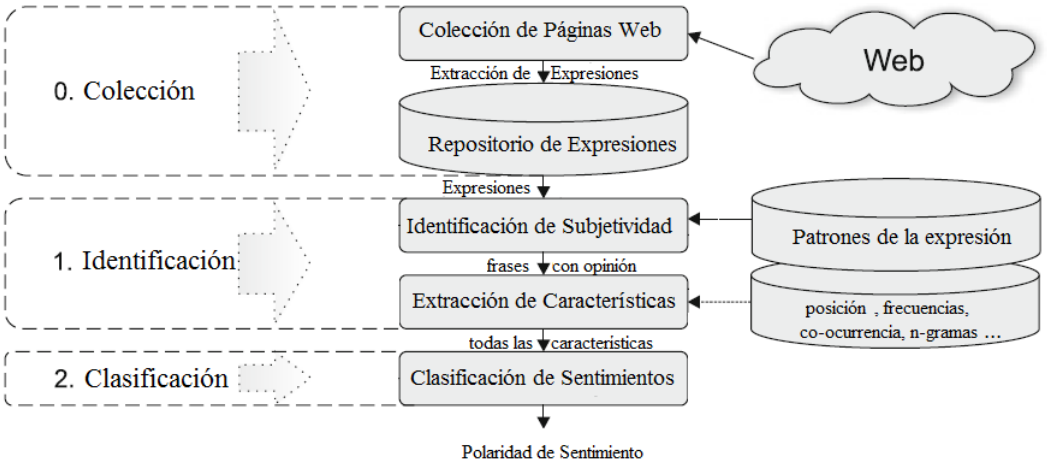

FIGURA 2.1: Arquitectura de Análisis de Sentimientos

Para este problema existen dos formulaciones basadas en los tipos de valores que toma s, donde s es el conjunto de características, si s toma valores categóricos como positivo, negativo, entonces es un problema de clasificación. Si s toma valores numéricos o puntuaciones dentro de un rango, entonces es un problema de regresión. [6]

Las opiniones y los sentimientos tienen una característica importante a diferencia de la información factual, son subjetivos, por lo que se necesita analizar las opiniones de varias personas y no solo de una. El tamaño influye en la dificultad de las opiniones, pues es un factor determinante para lograr una alta precisión en el análisis de sentimiento. Por ejemplo las opiniones obtenidas de twitter que son de a lo mas 140 caracteres, las

hace opiniones más centradas y más enfocadas con poca información irrelevante, siendo más fácil de analizarlas que por ejemplo, una opinión en un foro en donde las personas interactúan una con la otra y la dimensión de la opinión dependerá del dominio del tema. Las opiniones sobre productos y servicios son generalmente más fáciles de analizar. Discusiones sociales y políticas son mucho más difíciles debido al tema complejo y el sentimiento, expresiones, sarcasmos e ironías.

Los indicadores más importantes de sentimientos, son las palabras que expresan sentimiento, llamadas palabras de opinión (opinión words). Estas son palabras que comúnmente son usadas para expresar sentimientos positivos o negativos. Por ejemplo, *bueno*, *maravilloso* y *estupendo* son palabras que expresan sentimiento positivo, en cambio *malo*, *peor* y *terrible* son ejemplos de palabras que expresan sentimiento negativo, a dichas palabras se les conoce comúnmente como lexicón de opiniones (sentiment lexicon o opinion lexicon). A pesar de que las palabras y frases con sentimiento son muy importantes para el análisis de sentimientos no son suficientes para obtener éxito, la tarea es mucho más compleja, es decir que el lexicón de opiniones es necesario pero no suficiente para el AS. A continuación se describen algunas situaciones que hacen de AS un problema complejo.

- Una palabra que expresa un sentimiento negativo o positivo puede tener orientaciones opuestas, según el contexto de la oración. Por ejemplo *"sencillo"* usualmente indica un sentimiento positivo como en la oración "El método es sencillo de entender", y en otra oración como "El evento fue muy sencillo para la ocasión" tiene un sentimiento negativo.

- Una oración que contiene una palabra considerada como expresión de sentimiento, puede no expresar un sentimiento. Dicho fenómeno ocurre en oraciones interrogativas y condicionales. Por ejemplo, "¿Qué marca de cámara es buena?" y "Si encuentro una cámara buena, la compraré.", en donde ambas oraciones contienen la palabra *"buena"* y en ningún caso se expresa algún sentimiento positivo o negativo. Sin embargo no todas las oraciones interrogativas y condicionales no expresan algún sentimiento, como en la oración "¿Alguien sabe como reparar este terrible impresora?" y "Si buscas un carro bueno, la opción es hyundai."

- Oraciones Sarcásticas como "Que fantástico carro! Dejo de funcionar a los dos días.", estas oraciones no son muy comunes en opiniones de productos o servicios, pero son muy comunes en discusiones políticas.

- Existen oraciones que no contienen palabras que expresan algún sentimiento, es decir oraciones objetivas que expresan información factual y sin embargo son oraciones con sentimiento negativo o positivo. Un ejemplo es "Esta lavadora utiliza

bastante agua" lo cual expresa una opinión negativa acerca de la lavadora. Otra oración es "Después de dormir dos días en el colchón, se le ha formado un valle en medio", que también es una opinión negativa acerca del colchón.

- Las opiniones spam han llegado a ser el mayor problema, ya que cualquier persona tiene acceso a la web y es libre de expresar una opinión sin tener la necesidad de identificarse, lo que ha originado consecuencias indeseables, puesto que personas con identidades ocultas e intenciones maliciosas y haciéndose pasar por público en general realicen publicaciones falsas para promover o bien desacreditar algún producto, servicio, organización o individuos sin ser descubiertas sus verdaderas intenciones. Dichos individuos son llamados escritores de falsas opiniones (opinión spammers) [25].

2.1.1. Opinión

Para definir el concepto de opinión, se presenta un ejemplo:
Publicado por: John Smith Date:10-09-2011 "(1) Compré una cámara Canon G12 hace seis meses. (2) Simplemente me encanta. (3) La calidad de las imágenes es impresionante. (4) La duración de la batería es larga. (5) Sin embargo, mi esposa piensa que es muy pesada para ella."

La pregunta es: ¿Qué extraer de la opinión?

Del ejemplo anterior podemos detectar que contiene frases con un sentimiento positivo y negativo acerca de la cámara G12. La oración (2) expresa una opinión positiva acerca de la cámara, la oración (3) expresa una opinión positiva de la calidad de las imágenes, la oración (4) expresa una opinión positiva de la duración de la batería, y la expresión (5) es una opinión negativa acerca del peso de la cámara.

Una opinión consiste de dos componentes clave: un objeto g y un sentimiento s del objeto. *(g,s)* En donde g puede ser una entidad o aspecto de la entidad acerca de la opinión expresada y s es un sentimiento positivo, negativo, neutro o una puntuación que expresa la fuerza o intensidad del sentimiento. Positivo, negativo y neutro son llamados sentimientos (u opiniones) orientaciones o polaridades. Por ejemplo la oración (3) podría descomponerse de la siguiente manera:

(Cannon-G12, Calidad-imágen)

Otro aspecto importante es que la opinión habla acerca de dos personas, quien son llamadas fuentes de opinión (opinión sources) y emisor de la opinión (opinión holder). Y finalmente también la fecha es importante ya que frecuentemente se quieren conocer las opiniones a lo largo del tiempo y como van cambiando.

A continuación se da la definición formal de opinión:

Una opinión es una cuádrupla

$$(g, s, h, t) \tag{2.1}$$

Donde g es la opinión, s es el sentimiento de la opinión, h es la persona que expresa la opinión, y t es el tiempo o la fecha en que se expresa la opinión.

Una entidad e es un producto, servicio, tópico, persona, organización o evento. Se describe con un par *e: (T,W)*, donde T es una jerarquía de partes, subpartes, etc. Y W es un conjunto de atributos de *e*.

Por ejemplo continuando con el ejemplo de la cámara, un modelo de cámara en particular es una entidad y sus atributos son la calidad de la imagen, el tamaño, peso y su conjunto de partes son el lente, visor y la batería. La batería a su vez también tiene un conjunto de atributos, la vida de la batería y el peso de la misma. De lo anterior, puede concluirse que una entidad se define como una descomposición jerárquica de sus partes, donde la raíz es la entidad. Las jerarquías de dos niveles se pueden simplificar utilizando términos llamados aspectos o características para denotar partes y atributos.

Después de la descomposición antes mencionada se puede redefinir la opinión como:

Una opinión es una quíntupla

$$(e_i, a_{ij}, s_{ijkl}, h_k, t_l) \tag{2.2}$$

Donde e_i es el nombre de la entidad, a_{ij} es un aspecto de e_i, ,s_{ijkl}, es el sentimiento en el aspecto a_{ij} de la entidad e_i, ,h_k es el autor de la opinión, y t_l es el tiempo en el cual es expresada la opinión por h_k. Es decir una opinión está dada por s_{ijkl}, que está dada por un autor h_k acerca de aspectos a_{ij} de una entidad e_i en un tiempo t_l.

Ahora se definirá el concepto de modelo de entidad, un modelo de documentos de opinión y el objetivo de la minería, también nombrados aspectos basados en minería de opinión

Modelo de entidad

Una entidad e_i es representada por un conjunto de aspectos, A_i =$\{a_{i1}$,a_{i2},...,$a_{in}\}$. Cada aspecto a_{ij} ϵ A_i de la entidad puede ser expresado por algún conjunto finito de expresiones de aspectos AE_{ij} = { ae_{ij1},ae_{ij2},...,$ae_{ijm}\}$.

Por ejemplo.

Sean A_i =$\{a_{calidad}$,a_{peso},...,$a_{color}\}$, los aspectos de una cámara, donde cámara es una entidad.

2.1.2. Modelo de documentos de opinión

Un documento de opinión d, contiene opiniones sobre un conjunto de entidades $\{e_1$,e_2,...,$e_r\}$ expresadas por un conjunto de autores de opinión $\{h_1$,h_2,...,$h_p\}$. La opinión de cada entidad e_i expresa la entidad misma y su subconjunto de aspectos.

Ahora bien el objetivo de la minería de opinión es:
Dada una colección de documentos de opinión D, descubrir las opiniones o quíntuplas $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ en D.

Para la realización de este objetivo son necesarias las siguientes tareas:

- Extracción de entidades y agrupamiento
- Extracción de aspectos y agrupamiento
- Determinación del autor de la opinión y el tiempo de la opinión.
- Generación de la quíntupla

La dificultad de la minería de opinión, radica en que todas las tareas anteriores son problemáticas recientemente abordadas y por tanto no resueltas, además también hay problemáticas en la sintaxis usada puesto que una frase puede no mencionar explícitamente algunas piezas que están implícitas como pronombre, convenciones y contexto.

Una vez definidos los conceptos importantes en AS, a continuación se presentan algunas características utilizadas para la representación de documentos en el *AS*.

2.2. Representaciones textuales

Para realizar el trabajo presentado, 4 representaciones son de interés, N-gramas, etiquetado POS, teoría de la valoración y tf-idf o bolsa de palabras.

2.2.1. N-gramas

Es una representación tradicional en recuperación de la información, que consiste de palabras individuales (unigramas), o conjuntos de palabras (n-gramas), con sus frecuencias asociadas. En algunos casos podemos representar mejor un concepto mediante la unión de n palabras que se encuentran adyacentes al término principal, lo que se le conoce como *n-gramas*. La importancia de esta representación radica en que la posición de las palabras es considerada, puesto que el significado de una palabra, no tiene sentido sin las palabras adyacentes que le acompañan en cualquier texto, por lo que la posición de una palabra afecta potencialmente en el sentido del significado de la oración, es decir el sentimiento o la subjetividad dentro de una unidad textual.

Para el trabajo realizado se utiliza n-gramas de tamaño n = 2, es decir, bigramas.

Un bigrama o digrama, es un caso especial del n-grama, es un grupo de dos letras, dos sílabas, o dos palabras. Los bigramas son utilizados comúnmente como base para el simple análisis estadístico de texto. Se utilizan en uno de los más exitosos modelos de lenguaje para el reconocimiento de voz [7].

2.2.2. Partes de la oración (POS)

Una técnica de representación muy utilizada se basa en las reglas lingüísticas, donde las palabras y frases son categorizadas como sustantivos, verbos, adjetivos y adverbios. De acuerdo con Turney, son características gramaticales que tienen la capacidad de expresar subjetividad [2]. Existen investigaciones enfocadas principalmente en adjetivos y adverbios, como en el trabajo reportado por Farah Benamara et al [8], en donde expone que las expresiones de una opinión dependen principalmente de algunas palabras, por ejemplo, la palabra *"bueno"* es utilizada comúnmente para una opinión positiva, y la palabra *"malo"*, para algo negativo, dichas palabras son identificadas como adjetivos en términos lingüísticos.

En general los adjetivos son importantes indicadores en una opinión, son considerados características especiales, sin embargo no significa que otras partes de la oración no contribuyan a la expresión de sentimientos. Existen trabajos en donde los sustantivos, verbos, adverbios y sustantivos subjetivos también han tenido buenos resultados [9].

2.2.3. TF-IDF (term frequency-inverse document frequency)

Es un esquema de ponderación de términos comúnmente utilizado para representar documentos de texto como vectores, que se ajusta al modelo denominado bolsa de palabras, donde cada documento es representado como serie de palabras sin orden. Se trata de una medida estadística de cuán importante es una palabra para un documento en un corpus. Dicha técnica es utilizada para hacer ranking u ordenaciones de los resultados de búsqueda, generación de resúmenes de texto, agrupación y clasificación de documentos, identificación de la autoría de algún texto, recomendación de documentos, etc.

Cálculo del TF

Un término t_j que aparece muchas veces en un documento d_i es más importante que otro que aparece pocas.

$$tf_{ij} = \frac{(n_{ij})}{\sum_{i=1}^{N} n_{ij}} = \frac{(n_{ij})}{|d_i|} \tag{2.3}$$

Donde n_{ij} es el número de veces que aparece el término t_j en el documento d_i y $\sum_{i=1}^{N}$ es la sumatoria del número de veces que aparece el término t_j en todos los documentos.

Calculo del IDF

Un término t_j que aparece en pocos documentos, discrimina mejor que uno que aparece en muchos.

$$idf_j = log\left(\frac{N}{n_j}\right) \tag{2.4}$$

Donde N es el número total de documentos, y n_j es el número de documentos que contiene el término t_j.

Representación Final del documento

Cada elemento queda representado como un vector de características d_j:

$$d_j = (d_{j1}, ..., d_{jn}) \tag{2.5}$$

donde, d_{ij}=tf_{ij}*idf_{ij}

Es decir finalmente se seleccionan n términos con los valores más altos en todos los documentos.

2.2.4. Teoría de la valoración utilizando reglas sintácticas

La teoría de la valoración propuesta por Peter R.R White [10], se ocupa de los recursos lingüísticos por medio de los cuales las personas expresan alguna opinión. Particularmente del lenguaje (expresiones lingüísticas), la valoración, la actitud y la emoción del conjunto de recursos que explícitamente posicionan de manera interpersonal las propuestas y proposiciones textuales. Es decir trabaja con los significados de las palabras que hacen variar o modificar los términos del compromiso del hablante en sus emisiones, es decir, que modifican lo que está en juego en la relación interpersonal.

Dicha técnica fue implementada por Víctor Morales en su trabajo [11], que haciendo uso de un diccionario de aptitud, el cual contiene características de la teoría de la valoración *(juicio, apreciación y afecto*), utilizan sintagmas adverbiales con el fin de que dichas reglas obtengan una valor más preciso acerca del significado de cada palabra. El objetivo es contabilizar los valores de positivo, negativo, juicio, apreciación y afecto, que están presentes en una opinión cualquiera, como si se tratase de una bolsa de palabras ponderada, sin embargo las reglas sintácticas juegan un papel primordial en este proceso, ya que dependiendo del tipo de regla, los valores pueden aumentar, disminuir, o intercambiarse, afectando de esa manera los valores finales asignados al sentimiento de cada opinión.

2.3. Clasificación

Primero se discute el problema de clasificación, para predecir la categoría de clase. Para ello es importante definir el concepto de aprendizaje automático:

Los algoritmos de aprendizaje automático son métodos que dado un conjunto de ejemplos de entrenamiento infieren un modelo de las categorías en las que se agrupan los datos, de tal forma que se pueda asignar a nuevos ejemplos una o más categorías de manera automática mediante analogía de patrones en dicho modelo.

Las técnicas más utilizadas para la determinación de las clases, son los métodos de clasificación de documentos que llevan a cabo el aprendizaje supervisado, el cual cuenta con una clase que contiene las etiquetas de las instancias, y también las técnicas de aprendizaje no supervisado, o agrupamiento, que no cuentan con información adicional a los datos, como es el caso de el aprendizaje supervisado, la diferencia entre ambos se muestra en la figura 2.2, en donde se puede observar, que el aprendizaje supervisado realiza un proceso de entrenamiento y prueba, en el cual existe una partición de los datos para cada proceso respectivamente, una vez realizado el proceso de entrenamiento,

el algoritmo se prueba y se obtiene un resultado, el cual es comparado con la clase etiquetada, si es alcanzado el resultado adecuado según la métrica definida, entonces el proceso termina, si no lo es, entonces se repite el proceso de entrenamiento, hasta obtener un resultado satisfactorio. En el caso de el aprendizaje no supervisado, no se cuenta con información adicional a los datos, Por lo que la condición de paro en este caso, será hasta que las instancias estén suficientemente separadas y los grupos formados sean diferenciables.

Para el análisis de sentimientos se utilizan las técnica de aprendizaje supervisado.

El Análisis de sentimientos *(AS)*, como ya se menciono, es usualmente formulado como un problema de clasificación de texto en el cual son consideradas dos clases: positiva y negativa. Por lo general la clase neutra no es utilizada.

Para esta tarea de clasificación, se han aplicado métodos de aprendizaje supervisado como Máquina de Soporte Vectorial *(SVM)*, Naive Bayes *(NB)*. El primer trabajo realizado que tomo éste enfoque de clasificación fue realizado en [12], en el cual se clasificaron opiniones de películas considerando dos clases, usando unigramas (bolsa de palabras) como características y los dos clasificadores antes mencionados.

Para la realización del análisis de sentimiento, primero se necesita representar los textos de manera computacional para su análisis. Para construir la representación, es importante considerar que las palabras relacionadas en cada texto a analizar son las características principales, y es aquí donde se presenta la primera dificultad del problema, puesto que la clave para poder obtener resultados satisfactorios en el Análisis de Sentimientos, es la ingeniería de selección del conjunto de características efectivas [13].

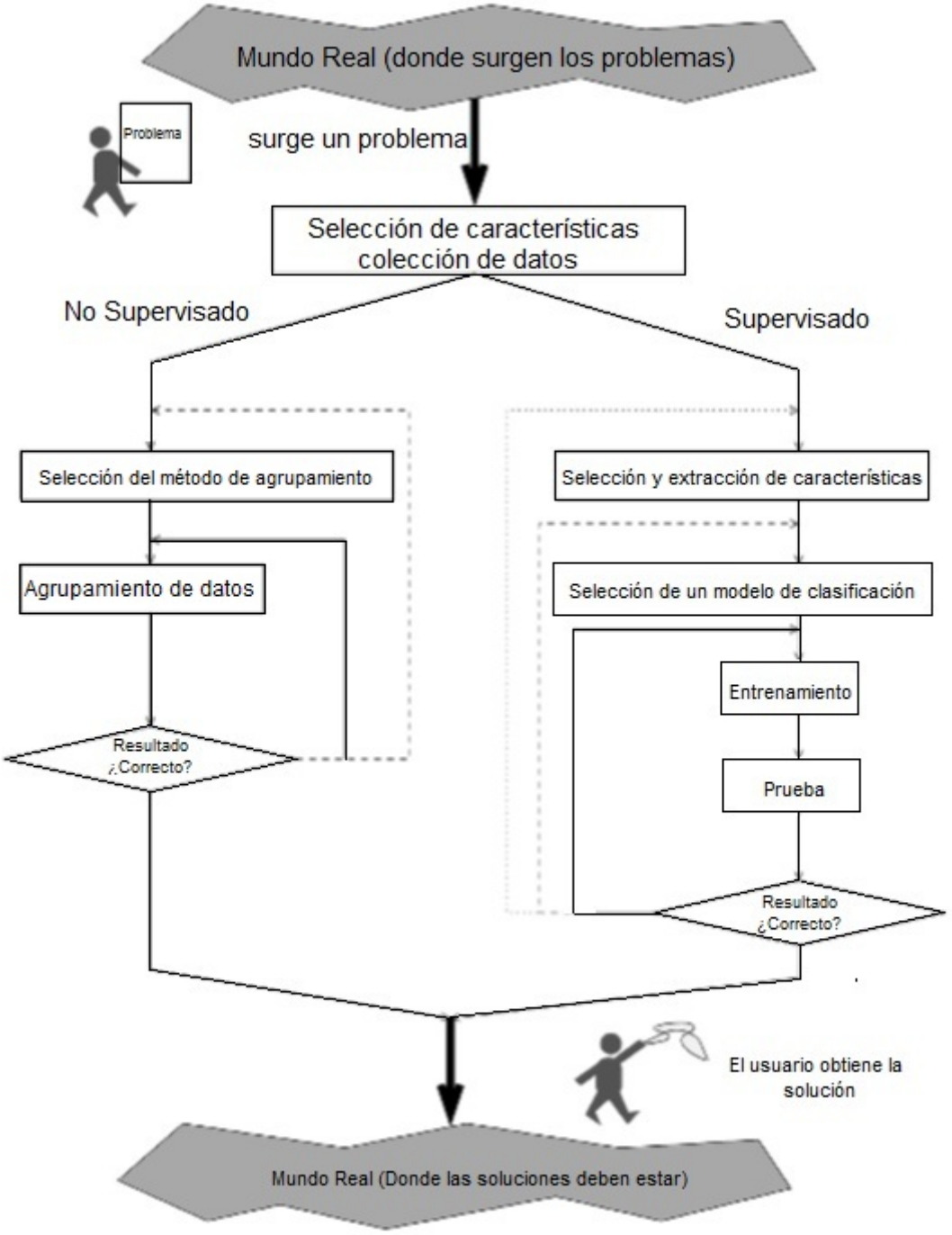

FIGURA 2.2: Aprendizaje Supervisado, y no supervisado

2.4. Combinación de Clasificadores

La idea de un ensamble de clasificadores, es combinar un conjunto de clasificadores para resolver una tarea en conjunto, en donde el objetivo principal es combinar las salidas de los clasificadores base, para generar una salida en donde sean considerados todos los clasificadores y dicha salida sea mejor que la obtenida por cualquier clasificador base.

Un ensamble de clasificadores es un grupo de clasificadores quienes individualmente toman decisiones que son fusionadas de alguna manera, para finalmente obtener una decisión por consenso.

La selección de los clasificadores base se puede realizar de dos maneras: estático o dinámico. En el enfoque estático, se aplica el mismo subconjunto de clasificadores base que se seleccionan para todas las muestras de prueba, en el enfoque dinámico la selección se realiza para cada nueva instancia individualmente.

Posterior a la obtención de la colección de los clasificadores base, el siguiente paso es combinar las salidas en orden de obtener una decisión final, en esta fase, debe ser considerado principalmente el tipo de información que se va a combinar y qué método de combinación se va a aplicar.

Otro aspecto importante son las salidas de los clasificadores, diferentes métodos de combinación utilizan diferentes tipos de salidas de clasificadores base, por ejemplo una etiqueta de clase o una distribución de probabilidad. Un enfoque alternativo es utilizar predicciones como un conjunto de atributos para formar una función de combinación en términos de meta-aprendizaje [14]. En años pasados, estudios experimentales realizados por la comunidad de machine learning, mostraron que la combinación de las salidas de múltiples clasificadores reducen la generalización del error. Los métodos de ensamble son muy efectivos, debido principalmente a que varios tipos de clasificadores tienen sesgos inductivos, y provocan que la diversidad de los clasificadores utilizados reduzca el error de la varianza, sin incrementar el error bias [15].

La combinación de clasificadores y por lo tanto la creación de ensamble de clasificadores fue propuesto para mejorar los resultados obtenidos por los clasificadores base. La llave para producir un ensamble exitoso, es elegir los métodos de clasificación apropiados y seleccionar los clasificadores base indicados para el problema planteado.

2.5. Definiciones importantes

Un clasificador es una función

$$D : R_n \rightarrow \Omega \tag{2.6}$$

En el "modelo canónico de un clasificador" [16], se consideran un conjunto de c funciones discriminantes $G = g_1(x), ..., g_c(x)$,

$$g_i : R^n \rightarrow R \tag{2.7}$$

$$i=1,...,c$$

Cada uno produciendo un puntaje para la clase respectiva. Por lo general, x está etiquetado en la clase con la puntuación más alta. Esta elección de etiquetado se denomina la *regla de máxima afiliación*, la cual se describe a continuación:

$$D(x) = w_{i*} \epsilon \Omega \leftrightarrow g_{i*}(x) = min_{i=1...c}\{g_i(x)\} \tag{2.8}$$

Donde w_i son las características de cada instancia, Ω es el dominio del conjunto de características. g_i son las funciones discriminantes de cada clase o etiqueta. La ecuación 2.8, obtiene la clase de la función que minimiza su valor, es decir, la instancia x se asignará a la clase que minimize el valor de g_i. Los empates se rompen al azar, es decir, x se asignan al azar a una de las clases.

2.6. Clasificadores base

Una vez teniendo las representaciones del corpus podemos clasificar las instancias, para lo cual es importante conocer la definición de clasificación.

La Clasificación es la tarea de predecir una variable discreta *"y"* usando un conjunto de características x_1, x_2, ..., x_n como variables independientes. Para realizar el entrenamiento del clasificador se necesita una función hipótesis h de una colección de ejemplos de entrenamiento. Dicha colección tiene la forma *(X,Y)* y usualmente se refiere a un conjunto de datos *(dataset)*. Cada entrada del conjunto de datos es una tupla *(x,y)*, donde x es el conjunto de características y y es la clase o etiqueta la cual es una variable discreta con c posibles categorías. Cuando los resultados posibles son restringidos a valores binarios, $y_i \ \epsilon \ +1, -1, \ \forall$ i $\epsilon \ 1, ..., N$ [17].

En la figura 2.3 se muestra la partición del conjunto de datos en n particiones, que son entrada del algoritmo de aprendizaje, el cual utiliza una función hipótesis para realizar la clasificación.

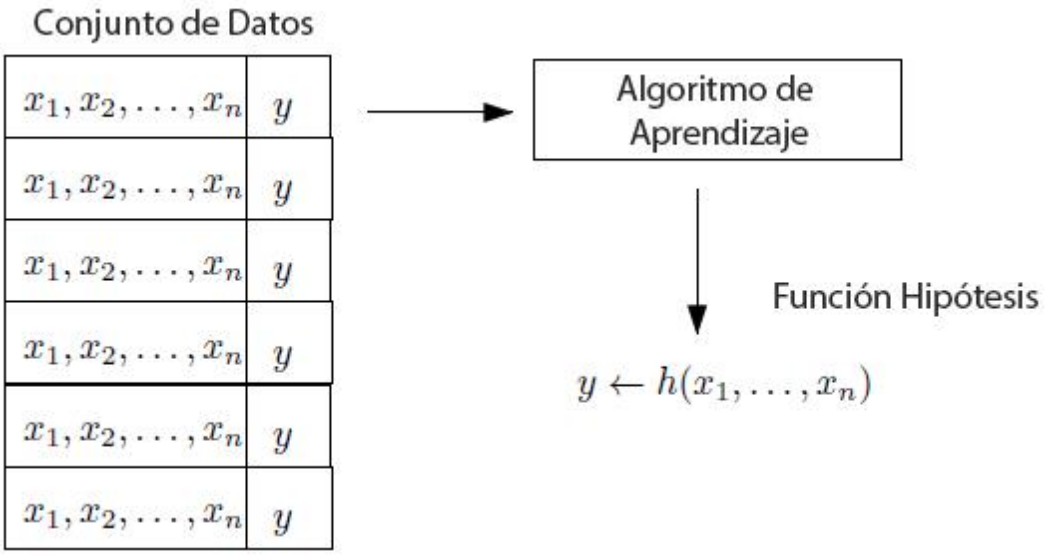

FIGURA 2.3: Partición de los datos y función hipótesis del algoritmo de aprendizaje.

Se utilizan 3 algoritmos de aprendizaje supervisado, Máquina de Soporte Vectorial, Naive Bayes y Árboles de Decisión. Cada uno de estos algoritmos se describe brevemente a continuación:

2.6.1. Naive Bayes

Es un clasificador probabilístico que aplica el Teorema de Bayes para estimar la probabilidad posterior *P(y | x)* de la clase y dada la variable x

$$P(y|x) = \frac{P(y|x)P(y)}{P(x)} \tag{2.9}$$

Naive Bayes se centra en las probabilidades *P(x|y)* que se refieren a la verosimilitud y representan la probabilidad de observar el valor x, dado el valor de clase y. Debido a esto Naive Nayes es considerado un *clasificador generativo.*

De la ecuación 2.9 podemos observar que el denominador *P(x)* es constante para algún valor de y, por lo que no es necesario calcularlo en orden de tomar una predicción. Por lo tanto podemos usar la siguiente aproximación:

$$P(y|x) \ P(y|x)P(y) \tag{2.10}$$

El valor *P(y)* se refiere a la probabilidad anterior y puede ser estimada directamente por los datos. Sin embargo *P(x|y)* depende de la distribución conjunta de x dado y . Y dado que x es una variable aleatoria multivariable, *P(x|y)* es muy caro de estimar.

De acuerdo a la regla de la cadena, la distribución conjunta de *P(x|y)* puede ser expresada de la siguiente manera:

$$P(x_1, ..., x_n|y) = P(x_1|y)P(x_2|x_1, y)...P(x_n|x_{n-1}, .., x_2, x_1, y) \tag{2.11}$$

A manera de evitar la costosa estimación de *P(x|y)*, el clasificador considera una fuerte suposición, que todos los pares de características x_i y x_j son independientes para cada evidencia de y dada. De esta manera se tiene *P(x_i|x_j,y)=P(x_i |y)* para algún par *i,j* ε *[1,n].* Por lo tanto la función de verosimilitud puede ser representada de acuerdo a la siguiente expresión:

$$P(x|y) = P(x_1|y)P(x_2|y)...P(x_n|y) = \prod_{i=1}^{n} P(x_i|y) \tag{2.12}$$

De esta manera las probabilidades *P(x_i|y)* pueden ser estimadas directamente de los datos.

2.6.2. Máquina de Soporte Vectorial

La máquina de Soporte Vectorial SVM es un clasificador binario discriminante, dirigido a encontrar el hiperplano óptimo ($w^T * x + b$) que separa los dos posibles valores de la variable etiquetada $y\ \varepsilon\ \{+1,-1\}$, de acuerdo al espacio de características representado por x. El hiperplano óptimo es aquel que maximiza el margen entre las instancias positivas y negativas en el conjunto de datos de entrenamiento formado por N observaciones. La tarea de aprendizaje de una SVM se formaliza con el siguiente problema de optimización:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \tag{2.13}$$

$$\text{sujeto a}\ \ y_i(w^T x_i + b) \geq 1 - \xi_i\ \forall i \varepsilon \{1, ...N\}$$

$$\xi_i \geq 0,\ \forall i \varepsilon \{1, .., N\}$$

El objetivo del problema se enfoca en dos aspectos, el primero, obtener el máximo margen en el hiperplano y minimizar el error $\sum_i^N xi_i$. El parámetro C se refiere al parámetro suave de regularización de margen y controla la sensibilidad de la SVM para los posibles valores atípicos.

También es posible hacer que las SVM encuentren patrones no lineales, de manera eficiente usando el kernel *trick*. Una función $\phi(x)$ que realiza el mapeo del espacio de características x usando un espacio dimensional alto conocido como el *espacio de Hilbert*, donde el producto punto $\phi(x)\phi(x')$ es conocido como la función kernel *K(x,x')*. De esta manera, el hiperplano es calculado en un espacio de dimensión alta $w^T\phi(x) + b$. La formulación dual de las SVM permite reemplazar cada producto punto por una función kernel como se muestra en la siguiente expresión:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \tag{2.14}$$

$$\text{sujeto a}\ \ \alpha_i \geq 0,\ \forall i \varepsilon \{1, .., N\},\ \sum_{1=1}^{N} \alpha_i y_i = 0$$

Donde los parámetros $\alpha_i,\ i\ \varepsilon \{1, ..., N\}$ corresponde a los *multiplicadores de Lagrange* del problema de optimización. Una vez que los parámetros se determinaron, es posible clasificar nuevas observaciones x_j de acuerdo a la siguiente expresión.

$$sign\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x_j) + b\right) \tag{2.15}$$

2.6.3. Árboles de Decisión

Un árbol de decisión describe un conjunto de reglas organizadas de forma jerárquica, que implementan una estructura de decisión. Se compone de hojas y nodos. Una hoja registra una respuesta *(clase)* y un nodo especifica algunas condiciones de las pruebas que se llevarán a cabo en un valor único, rasgo de una instancia, con una rama y sub-árbol para cada posible resultado de la prueba. Para un determinado vector, se toma una decisión partiendo de la raíz de un árbol, y se mueve a través del árbol determinado por el resultado de una prueba de estado en cada nodo, hasta que se encuentra una hoja [18]. El proceso de construcción de un árbol de decisión es una partición recursiva de un conjunto de entrenamiento.

A continuación se listan algunas de sus características

- Si todos los objetos son distinguibles, entonces podemos construir un clasificador árbol con error de resubstitution cero. Este hecho permite que el árbol sea capaz de memorizar los datos de entrenamiento, para que pequeñas alteraciones pudieran conducir a un clasificador árbol estructurado de manera diferente. La inestabilidad puede ser una ventaja más que un inconveniente cuando se consideran los conjuntos de clasificadores
- Los clasificadores de árboles son intuitivos porque el proceso de decisión puede ser rastreado como una secuencia de decisiones simples.
- Para dicho método son adecuadas las características cuantitativas y cualitativas, Con un pequeño número de categorías son especialmente útiles porque la decisión puede ser fácilmente diversificada. Para características cuantitativas, un punto de división tiene que ser encontrado para transformar la función en datos categóricos. Los árboles de decisión no se basan en un concepto de distancia en el espacio de características. Son principalmente útiles cuando se tienen características categóricas o mixtas. Esta es la razón pos la cual los árboles de decisión se consideran como métodos no métricos para la clasificación.

2.7. Ensamble de clasificadores

La primera etapa para construir un ensamble de clasificadores involucra el proceso de generación de una colección de clasificadores base, un enfoque es aplicar N diferentes métodos de aprendizaje, con un solo conjunto de datos de entrenamiento, para obtener N diferentes modelos de clasificación [19].

Otro enfoque es crear N diferentes particiones de los datos de entrenamiento y emplear un solo algoritmo de aprendizaje con cada partición [20]. El principal problema en este enfoque es la conversión del conjunto de datos originales en una colección de diferentes conjuntos de datos de entrenamiento. En algunas técnicas, el conjunto de datos original está dividido en N subconjuntos seleccionados aleatoriamente. Otros trabajos involucran la manipulación de los datos de acuerdo a la distribución de los mismos.

Dado el potencial uso del ensamble de clasificadores, existen algunos factores que deben ser diferenciados entre los varios métodos de ensamble. Los principales factores se listan a continuación:

1. Relación inter-clasificadores. ¿Cómo cada clasificador afecta a otros clasificadores? Los métodos de ensamble pueden ser divididos en dos principales tipos: secuenciales y concurrentes.

2. Método de combinación. La estrategia de combinar los clasificadores generados por un algoritmo de inducción. El combinador simple determina la salida exclusivamente a partir de las salidas de los inductores individuales.

3. Generador de diversidad. Con el objetivo de realizar un ensamble eficiente, debe existir diversidad entre los clasificadores involucrados. La diversidad puede ser obtenida a través de diferentes presentaciones de entrada de datos, como en bagging, variaciones en el diseño de aprendizaje, aplicando una sanción a las salidas para fomentar la diversidad.

A continuación se describen las técnicas más populares de ensamble de clasificadores.

2.7.1. Cascada

También conocida como generalización de cascada, es una arquitectura para combinar clasificadores como se muestra en la figura 2.4, puede presentar n niveles, sin embargo normalmente presenta dos niveles, en donde el nivel 1 es entrenado con el conjunto de datos original, el nivel 2 con un conjunto de datos aumentado, el cual contiene las características del conjunto de datos original junto con la salida del clasificador del nivel 1. La salida del clasificador del nivel 1 es un vector que contiene la distribución de probabilidad condicional $(p_1, ..., p_c)$, donde c es el número de clases del conjunto de datos original, y p_i es la estimación de probabilidad calculada por el clasificador del nivel 1, de que la instancia pertenezca a la clase i.

El entrenamiento del clasificador del nivel 2 es influenciado por el clasificador del nivel anterior, debido a que considera su salida obtenida, derivando un esquema global sobre-entrenado. Sin embargo, en cascada se reduce este problema debido a dos razones: en cada nivel se utiliza un clasificador de diferente naturaleza al otro y además el clasificador del nivel 2 no se entrena únicamente con la salida del clasificador de nivel 1, sino que además tiene en cuenta las características originales.

Cascada combina dos clasificadores, seleccionando aquel clasificador que obtenga un error bajo de bias, y otro con valor de varianza baja también, para conseguir uno nuevo que tenga valores bajos en ambas medidas. En el trabajo realizado en [21] se ocupa el clasificador con error de varianza baja en el nivel 1 y el clasificador con error de bias baja en el nivel 2, debido a que seleccionando métodos con bajo bias en el nivel superior, es posible ajustarse a áreas de decisión más complejas, teniendo en cuenta las superficies estables, trazadas por el clasificador o los clasificadores de nivel inferior. En [21], se realiza una validación experimental utilizando 26 conjuntos de datos del repositorio de la universidad de California, Irvine *(University of California, Irvine UCI)* [22], que da soporte a realizar el ensamble de clasificadores de ésta manera.

En la figura 2.4, se muestra el proceso de Cascada a dos niveles, en donde el clasificador del primer nivel (árbol), tiene como entrada los datos originales del problema y el clasificador del segundo nivel (SVM), los datos de entrada son los datos originales, pero también se agrega la salida del clasificador del primer nivel.

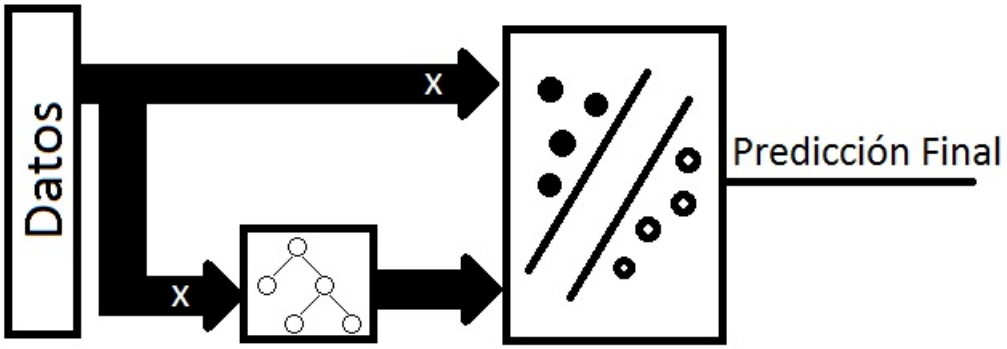

FIGURA 2.4: Estructura de Cascada de 2 niveles

2.7.2. Mayoría de votos

Es un método simple de combinación de clasificadores base, en el cual todos los clasificadores incluidos proveen un voto a alguna de las clases, el método realiza la sumatoria de dichos votos y la clase que recibe más votos es seleccionada como la decisión final. Dicho método es representado por la siguiente ecuación:

$$x \to w \; if \; w = \arg\max_{w \epsilon \theta} \sum_{i=1}^{T} 1(C_i(x) = w) \tag{2.16}$$

$$1(C_i(x) = w) = \begin{cases} 1 & \text{si } C_i(x) = w \\ 0 & en \; otro \; caso \end{cases} \tag{2.17}$$

x es una instancia, θ es el conjunto de etiquetas de clase, w es la clase asignada para la instancia x y C_1,...,C_T son los clasificadores base.

Este método a pesar de ser tan simple ha sido ampliamente utilizado en diferentes áreas, un ejemplo de ello se describe en [23] donde es utilizado junto con la transformada de *"Haar"* para mejorar la efectividad de los sistemas de autenticación basados en reconocimiento de iris. Otro trabajo más donde también se utiliza el esquema de mayoría de votos se describe en [24], en el cual dicho método se utiliza para detectar canales cubiertos maliciosamente dentro de una red a través de un túnel DNS.

Así como el esquema de mayoría de votos ha tenido éxito en otras áreas de conocimiento, aplicó este método al problema planteado.

2.7.3. Ventanas

El método de *Ventanas* es una técnica general, que tiene por objetivo mejorar la eficiencia de los métodos de aprendizaje o clasificadores utilizados, mediante la identificación de un subconjunto adecuado de instancias de entrenamiento. Dicho método se lleva a cabo mediante el uso de un procedimiento de submuestreo.

El método funciona de la siguiente manera: Se selecciona un subconjunto aleatorio de instancias para el entrenamiento de un clasificador *(una ventana)*, el resto de instancias son utilizadas para los datos de prueba, si la precisión obtenida del clasificador es insuficiente, las instancias de prueba clasificadas erróneamente se eliminan de las instancias de prueba y se añaden al conjunto de instancias para el entrenamiento en la siguiente iteración. En 1993 [25] Quinlan propone dos formas diferentes de la formación de una ventana: en la primera, la ventana actual se extiende hasta un límite especificado. En la segunda, se identifican varios casos *clave* en la ventana actual y el resto son reemplazados. Así, el tamaño de la ventana se mantiene constante. El proceso continúa hasta que se obtiene una precisión suficiente.

El método de ventanas también ha sido estudiado por Fürnkranz en 1997, [26] en donde se muestra que para este tipo de algoritmo, se pueden presentar mejoras significativas en la eficiencia, solo en dominios libres de ruido. En dicho trabajo se propone un versión de ventanas en donde se elimina de los datos de entrenamiento todos los casos que han sido clasificados correctamente, y se agregan todos los casos que han sido clasificados erróneamente. La eliminación de instancias desde la ventana mantiene su pequeño tamaño y por lo tanto disminuye el tiempo de ejecución.

En conclusión, en ambos casos el método de ventanas construye una secuencia de clasificadores para obtener una muestra final. Es importante mencionar que ventanas no combina clasificadores, su tarea radica en mejorar el resultado de un clasificador.

Una vez que se han explicado los métodos de clasificación y arquitecturas utilizadas, es importante conocer las métricas que permitirán evaluar el resultado obtenido por los mismos. A continuación se presentan las métricas de evaluación utilizadas.

2.7.4. Métricas de evaluación

Para realizar la evaluación de los métodos de clasificación aplicados sobre un *dataset*, se describen a continuación las métricas utilizadas.

Dado que el problema planteado esta formulado como un problema de clasificación binaria, se definen los términos utilizados por las métricas en la tabla 2.1.

	y=+1	y=-1
c(x)=+1	TP	FP
c(x)=-1	FN	TN

CUADRO 2.1: Combinación de clasificaciones

Donde c es la clasificación asignada al valor x, y y es la clase correcta de la instancia, TP son las instancias clasificadas correctamente como positivas, *FP*, son las instancias clasificadas erróneamente como positivas y de la misma manera para las instancias negativas, *FN*, son las instancias clasificadas erróneamente como positivas y *TN* son las clasificadas correctamente como negativas. Ahora teniendo las salidas antes descritas los siguientes criterios de evaluación pueden ser utilizados.

Precisión. Es la fracción de observaciones clasificadas correctamente como positivas, sobre todas las observaciones clasificadas como positivas.

$$Precision = \frac{TP}{TP + FP} \tag{2.18}$$

Recuerdo. Es la fracción de observaciones clasificadas correctamente como positivas, sobre todas las observaciones positivas.

$$Recuerdo = \frac{TP}{TP + FN} \tag{2.19}$$

Medida F. Es el significado armónico entre precisión y recuerdo

$$MedidaF = \frac{(1 + \beta^2)(2 * Presicion * Recuerdo)}{(\beta^2 * Precision) + Recuerdo} \tag{2.20}$$

Las medidas de evaluación son promediadas por todas las submuestras, asegurando que todas las observaciones fueron usadas para entrenamiento y prueba.

Capítulo 3

Trabajos relacionados

En esta sección se presentan trabajos de dos áreas importantes para la presente investigación: Análisis de Sentimientos y Combinación de Clasificadores.

Como primera parte se citan los trabajos reportados en el área de AS (Liu, Sentiment Analysis and Opinion Mining, 2012) [27], (Khan, 2009), [12], (Liu, Web Data Mining: Exploring Hyperlinks, Contents and Usage Data, 2011) [28], etc.

Los primeros trabajos en el área fueron desarrollados a partir del año 2000. Se trabajó desde el inicio con enfoques de diccionarios: estadísticos y semánticos. El enfoque basado en aprendizaje automático, fue una herramienta utilizada en las investigaciones de ese tiempo, la popularidad actual de este enfoque para la minería de opinión se origina en el trabajo publicado en el 2002 "Thumbs up?" por Pang y Lee donde utilizan tres métodos de clasificación: Naive Bayes(NB), Máxima Entropía(ME) y Maquinas de Soporte Vectorial (SVM), los tres algoritmos con base a una elección aleatoria, obtuvieron alrededor del 80 % de la precisión, siendo SVM el mejor. Posteriormente Dave junto con su equipo en 2003 basados en el trabajo de Pang y Lee enfatizan la selección de características, y utilizan una técnica de Laplaciano suavizado y con ello se mejora al 87 % de precisión (para un dataset Particular), sin embargo SVM siempre dio resultados cercanos a este porcentaje. En 2004 Pang y Lee utilizan identificación de subjetividad como un paso de pre procesamiento con el fin de mejorar la precisión del NB.

El análisis de opinión es un problema similar a la tarea de asignación de calificación considerando valores escalares, "estrellas". A pesar de que el método de SVM daba buenos resultados en clasificaciones binarias, la nueva aproximación, exigía soluciones más sofisticadas. Pang y Lee haciendo frente a este nuevo problema en 2005, realizan un estudio llamado "Seeing Stars", que propone utilizar SVM en múltiples clases, es decir una contra todas, y Regresión de Maquinas de Soporte Vectorial (SVR) combinándolo

con un etiquetado numérico. Los resultados demostraron que la combinación de las SVM con algún otro método de clasificación no supervisado obtiene una mejor precisión. En un trabajo posterior realizado en 2006 por Matt Thomas, Bo Pang, and Lillian Lee [29], también son estudiados algunos otros enfoques con SVM que coinciden con los resultados de los trabajos anteriores.

El desempeño de los métodos de aprendizaje automático es altamente dependiente de la calidad y de la cantidad de datos de entrenamiento. En el artículo publicado en 2006 titulado "Seeing Stars When There Are Not Many Stars", Goldberg y Zhu propusieron una técnica de aprendizaje semi supervisado operando en un gráfico de datos etiquetados y sin etiquetar. Los autores representan documentos con un gráfico, donde los vértices corresponden a los documentos, y los bordes se dibujan entre documentos similares utilizando una medida de la distancia calculada directamente de las características del documento. A pesar de que su enfoque exhibió un mejor rendimiento que RVS, los autores mencionan que es sensible a la elección de la medida de similitud, y no es capaz de beneficiarse de la utilización de los datos etiquetados adicionales.

En 2005 surgió un nuevo problema, la clasificación de sentimiento contextual, ahora se requería de algoritmos no solo operando en el nivel de la oración, sino que también en el análisis del contexto de cada frase. En función de esto, Shimada y Endo en 2008 proponen analizar calificaciones de nivel de producto-características, nombrando su trabajo como "Seeing Several Stars", donde demostraron que la regresión de vectores de soporte (SVR), a pesar de ser menos preciso que SVM (Support Vector Machine), produce etiquetas de salida que son más cercanas a las reales. Esta evidencia también apoya la afirmación de Pang y Lee que con el uso de una función gradual en RVS los objetos similares reciben necesariamente etiquetas similares.

En 2007 Osherenko y André demuestran que es posible utilizar sólo un pequeño conjunto de las palabras (diccionario) más afectivas como características, casi sin ninguna degradación en el rendimiento del clasificador. Sin embargo, el uso directo de los valores de sentimiento de tales diccionarios ha mostrado poco o incluso ningún aumento de la precisión. Los estudios por lo general utilizan frecuencias de palabras. Por ejemplo, en 2007 Devitt y Ahmad identifican palabras de soporte de sentimiento en un documento mediante la herramienta SentiWord –Net. Especificando un poco más, la Aproximación de Diccionario, se basa en un diccionario pre construido que contiene polaridades de opinión de las palabras, como: Inquirer2, WordNet-Affect 4 o la SentiWordNet, que es el diccionario más popular hoy en día. Fahrni y Klenner en 2008; Tsytsarau en 2010. Missen y Boughanem en 2009, utilizan las puntuaciones de polaridad directa, proporcionando un valor de sentimiento en una escala continua. La mayoría de los métodos de los diccionarios agregan los valores de polaridad de una sentencia o documento, y calculan

el sentimiento resultante usando algoritmos simples basados en reglas como en el trabajo de Zhu en 2009. Herramientas más sofisticadas, como el Analizador de Sentimiento introducido por Yi et al. en 2003 o el enfoque lingüístico en 2009 por Thet et al., extraen con precisión los sentimientos de algunos temas de destino mediante métodos avanzados que aprovechan las características específicas de dominio, así como las estructuras de frases de opinión y etiquetas. Fahrni y Klenner en 2008, proponen para obtener polaridades, utilizar la co-ocurrencia de los adjetivos en un corpus. En este caso, la capacidad de adaptación se consigue a través de la construcción de un diccionario de corpus-específico. En cuanto al problema de la falta de disponibilidad de algunas palabras, el método de corpus estadístico propone superar los resultados mediante el uso de corpus suficientemente grandes. Podemos identificar la polaridad de una palabra mediante el estudio de la frecuencia con la que esta palabra aparece en un gran corpus dado de textos. Si la palabra se presenta con mayor frecuencia entre los textos positivos o negativos, entonces tiene una polaridad positiva o negativa respectivamente. Igualdad de frecuencias indican palabras neutras. También es interesante mencionar, que en el trabajo de Ku et al. en 2006, 2007 que las aplicaciones que trabajan con la lengua china son capaces de reconocer la polaridad incluso para decir que el mensaje está oculto, gracias al hecho de que los caracteres fonéticos determinan el sentido de la palabra. Turney et al. Proponen obtener la frecuencia de co-ocurrencia de términos basándose en las estadísticas del buscador web AltaVista. En 2005 extendiéndose sobre este trabajo, Chaovalit y Zhou utilizan el motor de búsqueda de Google para determinar la co-ocurrencia de las palabras, lo que aumenta la precisión. Read y Carroll en 2009 extendieron aún más este enfoque, empleando espacios semánticos y similitudes distributivas como métodos alternativos débilmente supervisados. Un estudio detallado sobre la construcción de diccionarios de este tipo fue realizada por Taboada en 2006, quien mencionó algunos de los problemas que se producen debido a la indisponibilidad del modificador cercano o parecido o la no persistencia de la producción del motor de búsqueda. Ben He et al. (2008), haciendo uso de métodos estadísticos en el cálculo de la polaridad de opinión, propone utilizar un diccionario de opinión junto con métodos de Recuperación de Información (IR) con el objetivo de recuperar las opiniones de blogs.

El enfoque semántico proporciona valores de sentimiento directamente (al igual que el enfoque estadístico), excepto que se basa en principios diferentes para calcular la similitud entre las palabras. El principio fundamental de todos los enfoques en esta categoría es que semanticamente palabras parecidas deben recibir valores de sentimiento similar. WordNet proporciona diferentes tipos de relaciones semánticas entre las palabras. La posibilidad de eliminar la ambigüedad de sentidos de palabras usando WordNet puede servir como una manera de incluir el contexto de estas palabras en la tarea de análisis de la opinión. Kamps et al. Propusieron utilizar la distancia relativa de la ruta más

corta de la relación "sinónimo", con lo que se obtiene una precisión de (70 %), con un diccionario dado. Otra forma popular de utilizar WordNet es obtener una lista de palabras de sentimiento expandiendo de forma iterativa el conjunto inicial de sinónimos y antónimos. La polaridad de sentimiento de una palabra desconocida, se determina por el recuento relativo de sus sinónimos: positivos y negativos. De lo contrario, las palabras desconocidas, pueden ser descartadas. Sin embargo es importante tener cuidado con la diferencia entre el sinónimo y la palabra original, como es señalado por Godbole et al. en 2007, sólo debemos tener en cuenta los caminos que van a través de las palabras de la misma polaridad inicial. Con el crecimiento de las redes sociales, el análisis de opinión se ha extendido a los microblogs. Existen aplicaciones, como el análisis de mensajes en microblogs en Twitter, los cuales son capaces de adaptar el modelo utilizado a la evolución de los datos durante el análisis. Recientemente, Tumasjan et al. en 2010 demostró que los sentimientos de mensajes de Twitter se correlacionan con las preferencias políticas, e incluso Bollen et al. en 2010 también demostraron mejorar la predicción del mercado de valores, mediante el análisis de microblogs. Trabajos recientes han identificado varias diferencias entre la minería de opinión en microblogs en comparación con los análisis de la opinión convencional de documentos. La principal diferencia es la disponibilidad de sentimiento o estado de ánimo en los mensajes en microblogs, que proporciona una buena fuente de datos de entrenamiento para los clasificadores. Pak y Paroubek (2010) realizaron un análisis estadístico de las características lingüísticas de los mensajes de Twitter, en su trabajo reportan patrones interesantes para la clasificación de AS, así también demuestran que debe ser con un clasificador NB, considerando palabras negadas y con características representadas en n-gramas principalmente bigramas, se logra una buena precisión (aunque, a expensas de bajo recuerdo), se plantea que ésta contribución puede ser útil para aplicaciones de recuperación de información. Bermingham y Smeaton (2010) compararon el desempeño de clasificadores SVM y NB multinomiales (NBM) en datos de microblog y demostraron que en la mayoría de los casos, estos clasificadores dan mejores resultados en opiniones de longitud corta, los mensajes de microblogs son ricos en opiniones. Bifet y Frank en 2010 estudiaron el problema de usar un clasificador adaptable a los datos de correo electrónico. Propusieron utilizar el método de descenso de gradiente estocástico (DGS) con el cuál obtuvieron una precisión más pequeña, pero comparable a la de NBM (67,41 % frente a 73,81 %).

Un clasificador basado en la opinión es el propuesto por Turney. Dicho clasificador decide el carácter positivo o negativo de un documento en base a la orientación semántica de los términos que aparecen en el mismo, pues son el factor dominante para la clasificación de sentimientos. Los patrones sintácticos están compuestos en base a etiquetas POS. A continuación se presenta el algoritmo de Turney [2] que consta de tres pasos:

	Primera palabra	Segunda Palabra	Tercera palabra
1	Adjetivos,	Sustantivos	cualquiera
2	Adverbios, Adjetivos	Adjetivos	No sustantivos singulares ni plurales.
3	Adjetivos	Adjetivos	No sustantivos singulares ni plurales.
4	Sustantivos Singular	Adjetivos	No sustantivos singulares ni plurales.
5	Sustantivos Plural	Verbos	Cualquiera

CUADRO 3.1: Etiquetas POS

Paso 1. Dos palabras consecutivas son extraídas si sus etiquetas POS corresponden a alguno de los patrones de la tabla 3.1 Por ejemplo, el patrón 2 significa que dos palabras consecutivas se extraen si la primera palabra es un adverbio, la segunda palabra es un adjetivo y la tercera palabra no es un sustantivo. Como en el ejemplo *"Este piano produce sonidos hermosos"*, *"sonidos hermosos"* satisface el primer patrón.

Paso 2. Se estima la orientación del sentimiento (SO) de las frases extraídas mediante la información mutua puntual llamada medida PMI *(Point-wise Mutual Information (PMI))*.

$$PMI(term_1, term_2) = log_2\left(\frac{(Pr(term_1 \& term_2))}{Pr(term_1)Pr(term_2)}\right) \tag{3.1}$$

PMI mide el grado de dependencia estadística entre dos términos, $Pr(term_1 \& term_2)$ es la probabilidad de co-ocurrencia del termino 1 y el termino 2 y $Pr(term_1)$ y $Pr(term_2)$ es la probabilidad de ocurrencia simultanea de los dos términos si son estadísticamente independientes. La orientación de sentimiento *(SO)* de una frase se calcula en función de su asociación con las palabras de referencia positiva y la palabra de referencia negativa.

$$SO(phrase) = PMI(phrase, "excellent") - PMI(phase, "poor") \tag{3.2}$$

Las probabilidades son calculadas mediante la emisión de consultas, en donde para cada búsqueda se suele dar un número de documentos relevantes, el cual es el número de éxitos.

En el trabajo presentado por Turney en 2002, el motor de búsqueda de Alta Vista fue utilizado por que tiene un operador *"NEAR"*, para limitar la búsqueda a documentos que contienen las palabras a menos de diez palabras de uno al otro en cualquier orden.

Dados los éxitos de consulta, el número de aciertos obtenidos se pueden calcular como:

$$SO(phrase) = log_2 \left(\frac{hits(phrase NEAR "excellent") hits("poor")}{hits(phrase NEAR "poor") hits("excellent")} \right) \tag{3.3}$$

Paso 3. Dada una opinión, el algoritmo calcula el promedio SO de todas las frases en la opinión, y clasifica las opiniones como positivas si el promedio de SO es positivo y negativo en otro caso. Otro enfoque no supervisado, es el método basado en léxico, el cual usa un diccionario de palabras de sentimiento con tamaño y orientaciones asociadas y que incorpora intensificación y negación para calcular la puntuación de sentimiento para cada documento [30].

Como segunda parte del estado del arte, se revisará el área de combinación de clasificadores, citando de manera breve, algunas aportaciones importantes (Rokach, 2005), (Kuncheva, 2004), (Jurek, 2013). En ésta área se construye un ensamble de clasificadores, ésto involucra un proceso de selección de diferentes clasificadores base como primera etapa. Dos son los enfoques principales para este proceso.

- Utilizar un sólo algoritmo de aprendizaje y diferentes conjuntos de entrenamiento, en donde el objetivo principal es la conversión del conjunto de datos original para obtener una colección de diferentes conjuntos de datos de entrenamiento. Algunas técnicas dividen el conjunto de datos de forma aleatoria, o con la manipulación de la distribución de datos.
- Cada clasificador base es entrenado con el mismo conjunto de datos y utilizando diferente algoritmo de aprendizaje, como resultado se obtienen diferentes modelos de clasificación.

A continuación se presentan algunos trabajos correspondientes al primer enfoque: En 1996 Breiman propone el método de Bagging. Bagging es la técnica más popular para obtener diferentes conjuntos de datos de entrenamiento. Bagging se relaciona con el enfoque donde el conjunto de entrenamiento es elegido aleatoriamente k veces con reemplazamiento (técnicas bootstrap) de un conjunto de datos original. La ventaja principal de esta técnica es la independencia de los miembros del ensamble por lo tanto pueden ser entrenados en paralelo, lo cual reduce tiempo. La desventaja es que los subconjuntos de entrenamiento generados de forma aleatoria con reemplazamiento no son totalmente independientes. Skurichina y Duin, en 1998 con su método Nice bagging, modifican el método de Bagging, seleccionando los clasificadores base de mejor rendimiento. El modelo final no mejora el rendimiento de los clasificadores base, pero es más estable. En 1999 el método Wagging es propuesto por Bauer y Kohavi, el cual trabaja con el algoritmo de Bagging modificando la distribución de los datos en entrenamiento agregando

el "ruido Gaussiano". Con ello se obtuvo resultados con mayor diversidad. En 2007 se propone el algoritmo de Bagging con SVM por Wang y Lin donde buscan la generación de las mejores clases en cada muestra agregada, con ello se mejora el rendimiento de las SVM simples. En 2005 Zhou y Yu con su algoritmo BagInRand, definen una nueva métrica aleatoria modificando cada iteración, con ello aumenta la diversidad entre los clasificadores y mejora el rendimiento del algoritmo de los K vecinos cercanos (kNN).

En 2009 Gan y Xiao siguiendo con kNN realizan un muestreo de los datos de entrenamiento con una técnica que mejora la generación de clusters en kNN. Siguiendo el mismo enfoque muchos otros trabajos han sido publicados. A continuación se presentan algunos otros métodos de boosting y propuestas de intentos de mejora. En 1999 Freund y Schapire, proponen AdaBoost el cual entrena cada clasificador base con diferente distribución de datos, los resultados son una mayor diversidad entre los clasificadores base. Dicho método resulto exitoso aplicado en modelos inestables. MultiBoosting fue propuesto por Webb en el 2000, donde se agrega ruido Gaussiano a cada peso, con lo cual se logra una mayor diversidad entre los clasificadores base y supera a los algoritmos de bagging aplicados con árboles de decisión. En el mismo año Domingo y Watanabe proponen MadaBoost en el que se limita el peso de cada instancia a su probabilidad inicial, con ello se reduce el problema de sobreajuste y supera el método AdaBoost. Posteriormente otros métodos utilizando SVM fueron propuestos como AdaBoostSVM que ajusta los parámetros del kernel en cada ciclo para obtener un promedio preciso de clasificadores, con esta aportación se reduce el problema de sobreajuste. En 2007 Vezhnevets & Barinova proponen eliminar muestras confusas, eliminando instancias que no son clasificadas correctamente por un modelo bayesiano perfecto, dicha aportación también disminuye el sobreajuste. En 2010 se propone el algoritmo Bs-kNN en el cual cada iteración del modelo es construida con diferentes conjuntos de características, con ello se incrementa la diversidad entre los clasificadores base y mejora significativamente el rendimiento del kNN simple. Otro enfoque para generar una colección de clasificadores base con el mismo conjunto de datos de entrenamiento es mediante la aplicación de diferentes subconjuntos de características. A continuación se citan algunos de los métodos basados en este enfoque. En 2001 Breiman presenta "Random forest" generando un número grande de árboles individuales, seleccionando variables aleatorias en cada nodo, lo que aumenta la diversidad y supera al método de bagging decisión trees y boosting. En 2003 Bryll et al. Proponen "Attribute bagging" en el cual en cada iteración es tomado un conjunto de características de forma aleatoria, dicho método logra un mejor rendimiento del clasificador final y es más estable comparado con el algoritmo simple de bagging.

Una vez obtenida la colección de clasificadores base, el segundo paso en el proceso de construcción del ensamble es combinar los resultados obtenidos. Existe una técnica que combina todos los clasificadores individuales considerados. Mientras que otra selecciona

un subconjunto óptimo de modelos utilizados. Existen diferentes métodos para realizar dicha tarea, la forma más sencilla es con métodos de ponderación, en donde los votos proporcionados por todos los clasificadores se cuentan y la clase que recibe el mayor número de votos, es seleccionada como decisión final. Los métodos probabilísticos, son muy populares y con un esquema efectivo de combinación basado en el Teorema de Bayes, en donde el objetivo es asignar Z patrones dentro de las w_j clases y maximizar las probabilidades. Otro enfoque es el basado en razonamiento evidencial, donde las salidas de todos los clasificadores base son modeladas como distribuciones de probabilidad para todas las clases consideradas y luego son tratadas como piezas de evidencia, usualmente estas técnicas son aplicadas a clasificadores base entrenados con diferentes métodos de aprendizaje. La combinación de las decisiones de clasificadores individuales es basada en la selección del subconjunto óptimo de clasificadores, lo que resalta, que no todos los modelos generados contribuyen al proceso de toma de decisión, sin embargo solo el grupo seleccionado puede obtener el mejor rendimiento posible. La selección del ensamble puede mejorar el rendimiento final en términos de precisión y eficiencia. Dicho enfoque permite optimizar los costos computacionales, reduciendo el número de clasificadores base, pues se ha probado que se pueden obtener mejores resultados en comparación con el conjunto original. Las técnicas de selección son divididas en dos categorías: selección estática y selección dinámica. En la primera técnica se nomina un subconjunto de modelos que son seleccionados una sola vez al inicio y es fijo para todas las muestras de prueba. En la selección dinámica dicho proceso se realiza para cada instancia nueva individualmente, en función de sus características. El problema clave para ambas técnicas es el criterio de selección. El criterio más popular es la diversidad de medidas y el rendimiento individual y combinado.

A continuación se presentan algunos métodos estáticos para la selección de clasificadores base. En 2005 se propone aplicar Algoritmos Genéticos GA como herramienta de optimización, utilizando un criterio de selección con base al rendimiento del conjunto final y a la diversidad entre los clasificadores. “Greedy approach” propuesto en 2008 por Abdelazeem, en el cual se seleccionan los N mejores clasificadores, eliminando o añadiendo modelos específicos para maximizar la mejora del rendimiento, el criterio de selección se basa en la precisión de los clasificadores individuales. En el mismo año Shi y Lv, proponen ASDM que se enfocan en una aplicación de selección de atributos para obtener diversos clasificadores base, donde el criterio de selección es la diversidad entre los clasificadores base y el rendimiento del conjunto final. Zhiqiang y Balaji en 2007 presentan el método EMO y ESW, el primero se enfoca en una aplicación de programación lineal para encontrar los modelos más eficientes seleccionando los clasificadores individuales con mejor precisión, el segundo también se enfoca en aplicación de programación lineal con la diferencia que es para calcular los pesos de los clasificadores

base. En 2011 Diao y Shen proponen "FRFS" enfocado en una aplicación de selección de características difusas rígidas para seleccionar grupos de clasificadores base siendo el criterio de selección la independencia de los clasificadores. En el mismo año Pillai et al. Proponen HSM que selecciona diferentes subconjuntos de clasificadores para cada clase, y busca el mejor rendimiento del conjunto final. Finalmente se citan algunos de los métodos dinámicos para selección de clasificadores base. En 1997 Woods et al., propone DCS-LA que selecciona los clasificadores con la más alta precisión en la región pequeña cercana a los patrones de prueba, siendo el criterio de selección la mejor precisión local de los clasificadores base. En 2007 Ko et al., propone KNORA que selecciona grupos de clasificadores que clasifican correctamente los K vecinos cercanos de los patrones de prueba, buscando seleccionar también aquella arquitectura con la mejor precisión local de los clasificadores. Xiao and He en 2009 seleccionan un conjunto de clasificadores base con una función de aptitud que combina la precisión y la diversidad del ensamble, el criterio de selección es la diversidad entre los clasificadores en la región local. Batista et al., en 2011 propone dos métodos OP-ELIMINAT que selecciona métodos de clasificación que clasifican correctamente los k vecinos cercanos de los patrones de prueba y OP-UNION que para cada vecino de los patrones de prueba, selecciona un número k de clasificadores, que clasifiquen correctamente. El criterio de selección para ambas propuestas es la precisión local de los clasificadores base.

Algunos de los trabajos relacionados con otras áreas de conocimientos, son los siguientes:

En 2009, S. Zibri Boujelbene et al. [31] realizan una combinación de clasificadores, para la identificación de hablantes (voz), utilizando tres clasificadores: SVM, perceptrón multicapa, máquinas de soporte vectorial y árboles de decisión. Los resultados muestran que el sistema desarrollado mejora los resultados obtenidos por los clasificadores base.

Otro trabajo, es el presentado por Lijun Dai y Chuang Liu en 2010 [32], el cual busca mejorar el rendimiento de la clasificación de coberturas de una aplicación de imágenes de sensores remotos. En este trabajo se propone una arquitectura combinando seis métodos de clasificación: máxima verosimilitud, máquinas de soporte vectorial, redes neuronales artificiales, mapeo de ángulos espectrales, distancia mínima y arboles de decisión.

Capítulo 4

Arquitectura propuesta

En este capítulo se presenta la descripción de la arquitectura del sistema generado, las herramientas utilizadas, y finalmente se detalla la arquitectura propuesta.

4.1. Arquitectura del sistema

Para realizar la tarea del análisis de sentimientos, es necesario realizar primero el preprocesamiento de los textos, la representación de ellos y finalmente la clasificación de los mismos, con el fin de obtener un modelo de clasificación. En la figura 4.1 se muestra la arquitectura del sistema realizado, para el análisis de sentimientos.

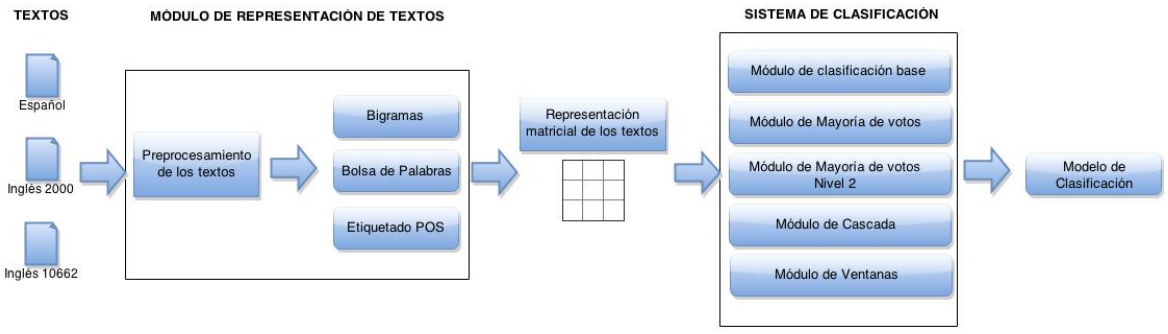

FIGURA 4.1: Arquitectura del sistema construido

A continuación se describe cada módulo de la arquitectura

4.1.1. Módulo de representación de textos

Como se puede observar en la figura anterior, la entrada al sistema, es un conjunto de corpus que son documentos de texto, los cuales contienen las opiniones a analizar en lenguaje natural. La función del módulo de representación de textos es trasladar los textos en lenguaje natural a una representación que sea entendible por la computadora.

- **Submódulo preprocesamiento de los textos**

Existe un problema antes de realizar el proceso de representación de textos.

Las opiniones que serán analizadas son escritas por usuarios comunes, por lo que pueden contener espacios adicionales, palabras mal escritas, símbolos alfanuméricos, palabras sin significado, que fueron escritas por error, etc. Para realizar un análisis correcto y reducir el tiempo de ejecución en el mismo, es necesario quitar estos casos de símbolos y palabras, por lo cual se construye este submódulo de preprocesamiento de los textos.

Una vez refinando el corpus, se realiza la generación de vocabulario utilizado en cada uno de los corpus, el cual es necesario para la representación de los textos.

Posteriormente se calculan frecuencias de las palabras para poder eliminar del vocabulario aquellas palabras que tienen un valor bajo de frecuencias o un valor muy alto, estas palabras no son de utilidad para realizar el análisis ya que o bien aparecen muy pocas veces en todo el corpus y no son lo suficientemente significativas, o bien son palabras que aparecen en todas las opiniones como los artículos, conjunciones, preposiciones, etc. lo cual hacen difícil la tarea de diferenciar las opiniones, además de que son palabras que no son subjetivas, lo cual no contribuye a encontrar la polaridad de la opinión. Por lo tanto al descartar estás palabras no influye en los resultados y reducen el tiempo de ejecución al reducir el tamaño el vocabulario, ya que las representaciones son en base a éste.

A continuación se explica de manera breve el proceso llevado a cabo para la realización de las representaciones de texto utilizadas.

- **Módulo Bigramas**

En este módulo, se implementa la representación de los textos en base a bigramas, el proceso realizado es el siguiente:

1. Dado el vocabulario del corpus, se realiza la concatenación de todas las parejas de palabras adyacentes en cada opinión, lo cual genera un nuevo diccionario de pares de palabras, es decir bigramas, al diccionario generado le llamamos "*diccionario de bigramas*".
2. Se calcula la frecuencia de cada bigrama del diccionario, en todo el corpus.
3. Se realiza el truncamiento de los bigramas definiendo un valor alto x y un valor bajo y, los bigramas de frecuencia mayor a x y menor a y, son eliminados del diccionario de bigramas.
4. Finalmente se realiza la representación matricial de bigramas, en donde las filas representan cada opinión y las columnas representan cada bigrama en el diccionario de bigramas. El valor colocado en dicha representación es la frecuencia de cada bigrama en cada opinión.

La salida de este módulo es la representación matricial con bigramas del corpus introducido, la cual es de tamaño m = número de opiniones y n = número de bigramas en el diccionario de bigramas, donde m son las filas y n las columnas.

- **Bolsa de palabras**

El módulo de bolsa de palabras, implementa la representación mediante el cálculo del esquema de ponderación *tf-idf*, el proceso se describe a continuación:

- Dado el vocabulario generado del corpus, para cada palabra en el vocabulario, se calcula la frecuencia *tf*, la cual ya ha sido definida en los fundamentos teóricos, por ejemplo, el cálculo del *tf* de una palabra dada, dentro de una opinión, es el número de veces que aparece dicha palabra en la opinión, divida por el número de veces que aparece la misma palabra, pero en todo el corpus. Con esto obtenemos una matriz la cual es de m = número de opiniones y n = número de palabras en el vocabulario, donde m es el número de opiniones y n es el número de palabras en el vocabulario.

- La segunda parte consiste de calcular el *idf*, el cual se calcula para cada palabra en el vocabulario, se realiza el cálculo del logaritmo natural de *N*, que es el número total de opiniones, dividido entre el número de opiniones que contienen una palabra dada del vocabulario. La salida obtenida en este caso es de m = número de palabras en el vocabulario y $n = 1$ ya que dicho cálculo se realiza solo una vez por cada palabra.

- Finalmente se realiza la representación matricial de los textos mediante la multiplicación de los resultados obtenidos en *tf* e *idf*, es decir *tfidf = tf*idf*. La salida es una representación matricial de tamaño m = número de opiniones, y $n = 1$, donde m son las filas y n las columnas.

- **Módulo de Etiquetado POS**

La implementación realizada por el módulo de representación del texto con POS, se muestra a continuación:

1. Se elige la herramienta de etiquetado a utilizar
2. Se introduce al etiquetador POS cada opinión dentro del corpus, sin tomar en cuenta el vocabulario de las palabras del corpus antes generado, se realiza de esta manera debido a que para el etiquetador es importante el contexto y el orden de las palabras, pues por ejemplo a las palabras homónimas, en cuestión del etiquetado POS, si se les asigna diferente peso o valor, sin embargo en el diccionario que se genera en la subsección preprocesamiento de textos, las palabras homónimas no pueden ser consideradas como diferentes.
3. Una vez teniendo las opiniones etiquetadas, solo se consideran aquellas palabras etiquetadas como adjetivos y adverbios, pues como antes ya se ha mencionado, estas palabras son las que contribuyen a obtener la polaridad de la opinión.
4. Posteriormente se realiza un diccionario de las palabras con etiquetas adverbios y adjetivos.
5. Una vez obtenido el diccionario realizado en el paso anterior, se realiza la representación matricial de las opiniones, colocando la frecuencia de las palabras de cada opinión, definiendo como columnas los diferentes casos en los cuales la etiqueta pertenece a un adverbio o adjetivo, por ejemplo una columna es adjetivos calificativos, otra es adjetivos diminutivos
6. Finalmente la representación de etiquetado POS, es una matriz de m = número de opiniones y n = es el número de las diferentes etiquetas de adverbios y adjetivos.

Las herramientas utilizadas para el etiquetado POS, son las siguientes.

1. *Freeling 3.0.* la cual es una librería de código abierto para el procesamiento multilíngue automático, que proporciona servicios de análisis lingüístico para varios idiomas, [33].
2. *POS Tagger Stanford* es un etiquetador gramatical, ampliamente utilizado en el idioma inglés, [34].

Los método de representación de textos, originan como salidas, documentos que contienen la representación matricial de cada uno ellos, la cuales son la entrada a el sistema de clasificación. A continuación se describen los módulos de clasificación, implementados en Matlab 2014b.

- **Módulo de clasificación base**

En este módulo se realiza la clasificación de los métodos SVM, arboles y Naive Nayes.

- **Módulo de mayoría de votos**

Este módulo aplica el método de mayoría de votos tomando los resultados de los clasificadores base, de cada una de las representación del corpus.

- **Módulo de mayoría de votos mejores**

En este módulo se aplica nuevamente el método de mayoría de votos, pero ahora considerando las clases obtenidas por el clasificador que obtuvo en sus resultados la mejor medida F, de cada representación, es decir se esta combinando las clases o resultados con las diferentes representaciones.

- **Módulo de Cascada**

El módulo de Cascada recibe como entrada la salida obtenida por mayoría de votos mejores. Posteriormente incorpora la entrada al conjunto de datos originales del problema, y realiza la clasificación tomando de forma automática el mejor clasificador en medida F, considerando representaciones textuales y clasificadores.

- **Módulo de Ventanas**

Dado que los resultados obtenidos por cascada tienen una mejora, el método de ventanas toma como entrada la salida proporcionada por Cascada, selecciona automáticamente el método de clasificación base mejor en medida F, respecto a representaciones y clasificadores, y define un valor N para el número máximo de iteraciones. Teniendo estos parámetros, realiza la clasificación N veces, en cada iteración el algoritmo selecciona las instancias clasificadas de manera errónea y las agrega al conjunto de datos de entrenamiento, intercambiando instancias, hasta que el valor de N se cumpla.

4.2. Arquitectura Propuesta

La arquitectura propuesta para el corpus en español, se muestra en la figura 4.2, y consiste de tres niveles que a se describen a continuación:

1. En la primera fase, se realiza la clasificación de los textos con los clasificadores base, para cada representación del corpus, y se aplica el método de combinación de clasificadores: mayoría de votos, la cual se realiza en dos niveles.
   a) En la segunda fase se aplica el módulo de mayoría de votos.
   b) En el segundo nivel, se aplica el módulo de mayoría de votos mejores.
2. El segundo nivel incluye el método de Cascada, definido en el módulo de cascada.
3. En el tercer y último nivel, se incorpora el método de Ventanas, el cual tiene como objetivo, mejorar los resultados ya obtenidos.Los resultados obtenidos son los resultados finales de la arquitectura.

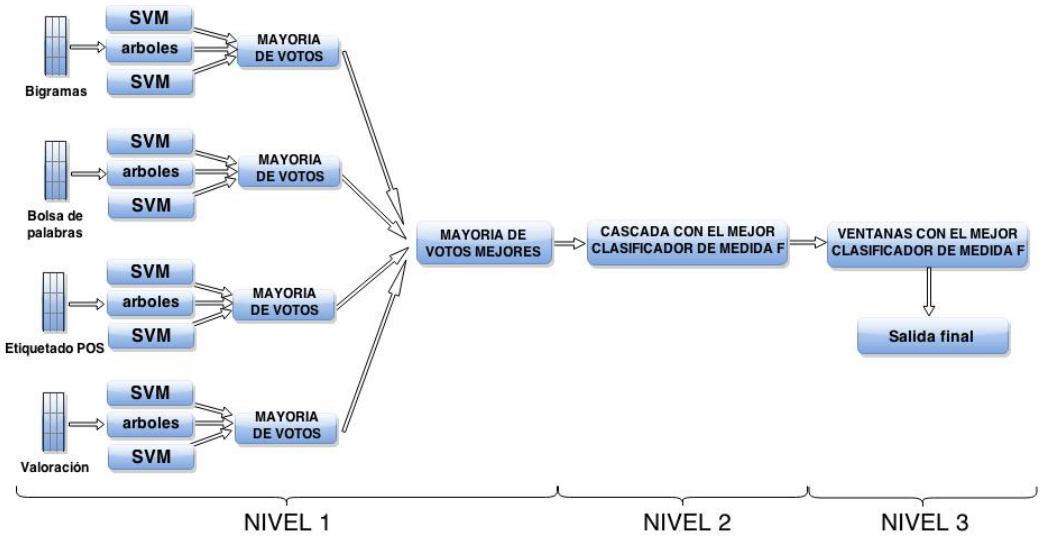

FIGURA 4.2: Arquitectura propuesta

Capítulo 5

Experimentos y Resultados

En este capítulo se presentan los experimentos realizados, los métodos de clasificación utilizados son: máquina de soporte vectorial (SVM), arboles de decisión y naive bayes. Y los métodos de combinación de clasificadores: son mayoría de votos, cascada y ventanas.

5.1. Corpus utilizados

Para la realización de los experimentos se utilizó un corpus en español y para la validación de la arquitectura propuesta se consideraron dos corpus en inglés. A continuación se describen cada uno de ellos.

- **Corpus Español**

  Es un corpus de opiniones de películas de cine, creado en [35], con 3878 críticas que contienen una puntuación asignada del 1 al 5 donde 1 es la más negativa y 5 es la más positiva, del cual se tomaron 2625 críticas (1351 positivas, 1274 negativas) no incluyendo las criticas neutras es decir con puntuación 3.

- **Corpus en idioma Inglés con 2000 opiniones**

  El segundo corpus es de opiniones de cine en inglés, con 1000 opiniones negativas y 1000 opiniones positivas, extraídas de la página *Rotten Tomatoes*, [36]

- **Corpus en idioma Inglés con 10662**

  Es un corpus de críticas de cine con 10662 opiniones, 5331 positivas y 5331 negativas, extraídas también de la página de *Rotten Tomatoes*.

Corpus	Vocabulario completo	Vocabulario Truncado
Español	57713	19876 (34.43 %)
Inglés 2000	34692	13536 (39.01 %)
Inglés 10662	33207	15824 (47.65 %)

CUADRO 5.1: Cardinalidad de vocabularios.

5.2. Aplicaciones desarrolladas

- Se realizó una aplicación en Microsoft Visual Studio 2012, para implementar el módulo de representación de textos.
- Se contruyó también una aplicación en Matlab 2014b, para la implementación del sistema de clasificación.

5.3. Pre Procesamiento de los datos

Una vez elegidos los corpus y antes de realizar el Análisis de Sentimientos, primero se debe realizar un pre procesamiento de los datos, ya que los corpus fueron construidos a partir de opiniones introducidas por usuarios comunes de la web y no por críticos especializados, lo cual dificulta la tarea del procesamiento de los datos, pues el corpus puede contener palabras vacías, faltas de ortografía, incoherencias, palabras incompletas, etc.

A continuación se lista las acciones realizadas en el pre procesamiento de los datos, para los 3 corpus utilizados.

1. Eliminación de símbolos no alfanuméricos.
    - Eliminación de símbolos de puntuación.
    - Eliminación de números.
    - Eliminación de palabras vacías.

En la tabla 5.1, se muestran la cardinalidad de los vocabularios utilizados a cada corpus, después de realizar el preprocesamiento de los textos.

5.4. Condiciones de ejecución

Los experimentos realizados fueron variando el porcentaje de datos de entrenamiento y de prueba, con 80 % - 20 % y 60 %- 40 % respectivamente. Los clasificadores utilizados son SVM, árboles de decisión y Naive Bayes.

Las representaciones utilizadas para los 3 corpus, son bolsa de palabras, bigramas, etiquetas POS considerando adjetivos y adverbios, y para el corpus en español también se utilizó la representación de la teoría de la valoración.

El clasificador SVM utiliza el kernel lineal.

Es importante mencionar que inicialmente también se consideró utilizar el algoritmo de clasificación *kNN*, sin embargo los resultados obtenidos eran bajos, motivo por el cual se utilizó otro algoritmo de clasificación, arboles de decisión.

5.5. Experimentos

En esta sección se presentan los experimentos realizados, mostrando los resultados obtenidos con el corpus en español, en el cual esta enfocado este trabajo y posteriormente se muestran los resultados obtenidos por los corpus en inglés con el objetivo de validar la arquitectura propuesta.

5.5.1. Nivel 1. Experimentos con clasificadores base y Mayoría de votos

Para realizar los experimentos primero se eligieron los datos de entrenamiento y prueba de manera aleatoria, posteriormente se utilizó validación cruzada a 10 pliegues.

Se aplicó el primer nivel de la arquitectura al corpus en español con los experimentos 60 %-40 %, y 80 %-20 %, realizando primero la clasificación con los clasificadores base, una vez obtenidos los resultados de los clasificadores base, se utilizó el método más sencillo de combinación de clasificadores: *Mayoría de votos*. Mayoría de votos combina las salidas o clases obtenidas por los 3 clasificadores realizando el conteo de votos, para obtener una salida que será la decisión final de este nivel. En las tablas 5.2 y 5.3, se muestran los resultados obtenidos por los clasificadores base de todos los corpus y en las tablas 5.4 y 5.5, se muestra los resultados obtenidos por el primer nivel de la arquitectura, para el corpus en español.

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
		SVM	0.824138	0.821306	0.822739
		Árboles	0.646429	0.621993	0.633995
	Bigramas	Naive Bayes	0.563953	1	0.72121
		SVM	0.683824	0.319588	0.435617
		Árboles	0.557823	0.563574	0.560704
Español	Bolsa de palabras	Naive Bayes	0.655462	0.536082	0.589812
		SVM	0.580205	0.584192	0.582212
		Árboles	0.586614	0.512027	0.546809
	POS	Naive Bayes	0.579882	0.67354	0.623231
		SVM	0.75	0.597938	0.665412
		Árboles	0.658915	0.584192	0.619328
	Valoración	Naive Bayes	0.733945	0.549828	0.628704
		SVM	0.846154	0.858537	0.85232
		Árboles	1	1	1
	Bigramas	Naive Bayes	0.508861	0.980488	0.67002
		SVM	1	1	1
		arboles	1	1	1
Inglés 2000	Bolsa de palabras	Naive Bayes	0.990338	1	0.995166
		SVM	0.995098	0.990244	0.992685
		Árboles	1	1	1
	POS	Naive Bayes	0.990196	0.985366	0.987795
		SVM	0.989681	0.970561	0.980048
		Árboles	1	1	1
	Bigramas	Naive Bayes	0.50985	1	0.675385
		SVM	1	1	1
		arboles	1	1	1
Inglés 10662	Bolsa de palabras	Naive Bayes	0.998163	1	0.999101
		SVM	0.998162	0.99908	0.998641
		Árboles	1	1	1
	POS	Naive Bayes	0.998111	0.972401	0.985109

Cuadro 5.2: Clasificadores base 80 %-20 %.

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Español	Bigramas	SVM	0.715736	0.776147	0.744738
		Árboles	0.613508	0.6	0.606699
		Naive Bayes	0.730769	0.499083	0.577069
	Bolsa de palabras	SVM	0.647482	0.330275	0.437444
		Árboles	0.536101	0.544954	0.540511
		Naive Bayes	0.605691	0.546789	0.574755
	POS	SVM	0.578348	0.372477	0.453145
		Árboles	0.575875	0.543119	0.559038
		Naive Bayes	0.571942	0.291743	0.386411
	Valoración	SVM	0.693182	0.559633	0.619309
		Árboles	0.598148	0.592661	0.595412
		Naive Bayes	0.666667	0.502752	0.573242
Inglés 2000	Bigramas	SVM	0.785219	0.862944	0.822269
		Árboles	0.796721	1	0.886881
		Naive Bayes	0.491206	0.992386	0.657163
	Bolsa de palabras	SVM	1	0.997462	0.998749
		Árboles	1	0.995885	0.997958
		Naive Bayes	0.994924	0.994924	0.994944
	POS	SVM	1	0.994924	0.997475
		Árboles	1	0.99177	0.995888
		Naive Bayes	1	0.92132	0.959069
Inglés 10662	Bigramas	SVM	0.987648	0.966078	0.976764
		Árboles	0.987934	0.975065	0.981477
		Naive Bayes	0.50469	1	0.670843
	Bolsa de palabras	SVM	0.999628	0.999628	0.999834
		Árboles	0.998883	0.998511	0.998717
		Naive Bayes	1	0.998141	0.99909
	POS	SVM	0.999071	1	0.999556
		Árboles	0.988231	1	0.994101
		Naive Bayes	0.999039	0.966543	0.982542

Cuadro 5.3: Clasificadores base 60 %-40 %

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Español	Bigramas	SVM	0.7157	**0.7761**	**0.7447**
		Árboles	0.61350	0.6000	0.6067
		Naive Bayes	0.7308	0.4991	0.5771
		Mayoría de votos	**0.7757**	0.5266	0.6273
	Bolsa de palabras	SVM	**0.6475**	0.3303	0.4374
		Árboles	0.5361	0.5450	0.5405
		Naive Bayes	0.605691	**0.546789**	**0.574755**
		Mayoría de votos	0.635697	0.477064	0.545093
	POS	SVM	0.578348	0.372477	0.453145
		Árboles	0.575875	**0.543119**	**0.559038**
		Naive Bayes	0.571942	0.291743	0.386411
		Mayoría de votos	**0.59816**	0.357798	0.447781
	Valoración	SVM	**0.693182**	0.559633	**0.619309**
		Árboles	0.598148	**0.592661**	0.595412
		Naive Bayes	0.666667	0.502752	0.573242
		Mayoría de votos	0.681093	0.548624	0.607744
		Mayoría de votos Mejores	**0.7323**	**0.7027**	**0.7172**

CUADRO 5.4: Resultados obtenidos para el corpus en español, con el primer nivel de la arquitectura propuesta, en los experimentos 60 %-40 %

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Español	Bigramas	SVM	**0.8241**	0.8213	**0.8227**
		Árboles	0.6464	0.6220	0.6340
		Naive Bayes	0.5640	**1**	0.7212
		Mayoría de votos	0.6872	0.9210	0.7871
	Bolsa de palabras	SVM	**0.6838**	0.3196	0.4356
		Árboles	0.5578	**0.5636**	0.5607
		Naive Bayes	0.6555	0.5361	**0.5898**
		Mayoría de votos	0.6550	0.4502	0.5336
	POS	SVM	0.5802	0.5842	0.5822
		Árboles	0.5866	0.5120	0.5468
		Naive Bayes	0.5799	**0.6735**	**0.6232**
		Mayoría de votos	**0.5922**	0.6289	0.6100
	Valoración	SVM	**0.7500**	**0.5979**	**0.6654**
		Árboles	0.6589	0.5842	0.6193
		Naive Bayes	0.7339	0.5498	0.6287
		Mayoría de votos	0.7489	0.5842	0.6564
		Mayoría de votos Mejores	**0.9159**	**0.8920**	**0.9038**

CUADRO 5.5: Resultados obtenidos para el corpus en español, con el primer nivel de la arquitectura propuesta, en los experimentos 80 %-20 %

Los resultados obtenidos en el nivel uno, fueron mejores solo con los experimentos 80 %-20 % en precisión y medida F, en los demás casos no superó los resultados obtenidos. En los experimentos 60 %-40 % los resultados no fueron mejorados en ninguna medida.

Con el propósito de evaluar la arquitectura, se realizan los experimentos para los corpus en inglés, aplicando también el primer nivel de la arquitectura. Los resultados se muestran a continuación, en la tablas 5.6 y 5.7

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Inglés 2000	Bigramas	SVM	0.785219	0.862944	0.822269
		Árboles	0.796721	**1**	0.886881
		Naive Bayes	0.491206	0.992386	0.657163
		Mayoría de votos	**0.8107**	**1**	**0.8955**
	Bolsa de palabras	SVM	**1**	**0.9975**	**0.9987**
		Árboles	**1**	0.9959	0.9980
		Naive Bayes	0.9949	0.9949	0.9949
		Mayoría de votos	**1**	**0.9975**	**0.9987**
	POS	SVM	**1**	**0.9949**	**0.9975**
		Árboles	**1**	0.9918	0.9959
		Naive Bayes	**1**	0.9213	0.9591
		Mayoría de votos	**1**	**0.9949**	**0.9975**
		Mayoría de votos Mejores	**1**	**1**	**1**
Inglés 10662	Bigramas	SVM	0.9876	0.9661	0.9768
		Árboles	0.9879	0.9751	0.9815
		Naive Bayes	0.5047	**1**	0.6708
		Mayoría de votos	**0.9881**	**1**	**0.9940**
	Bolsa de palabras	SVM	0.9996	0.9996	0.9998
		Árboles	0.9989	0.9985	0.9987
		Naive Bayes	**1**	0.9981	0.9991
		Mayoría de votos	**1**	**1**	**1**
	POS	SVM	**0.9991**	**1**	**0.9996**
		Árboles	0.9882	**1**	0.9941
		Naive Bayes	0.9990	0.9665	0.9825
		Mayoría de votos	**0.9991**	**1**	**0.9996**
		Mayoría de votos Mejores	**1**	**1**	**1**

Cuadro 5.6: Resultados obtenidos para los corpus en inglés, con el primer nivel de la arquitectura propuesta, en los experimentos 60 %-40 %

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Inglés 2000	Bigramas	SVM	0.8462	0.8585	0.8523
		Árboles	**1**	**1**	**1**
		Naive Bayes	0.5089	0.9805	0.6700
		Mayoría de votos	0.8638	0.9902	0.9227
	Bolsa de palabras	SVM	**1**	**1**	**1**
		arboles	**1**	**1**	**1**
		Naive Bayes	0.9903	1	0.9952
		Mayoría de votos	**1**	**1**	**1**
	POS	SVM	0.9951	0.9902	0.9927
		Árboles	**1**	**1**	**1**
		Naive Bayes	0.9902	0.9854	0.9878
		Mayoría de votos	0.9951	**1**	0.9976
		Mayoría de votos Mejores	**1**	**1**	**1**
Inglés 10662	Bigramas	SVM	0.9897	0.9706	0.9800
		Árboles	**1**	**1**	**1**
		Naive Bayes	0.5099	1	0.6754
		Mayoría de votos	0.9900	**1**	0.9950
	Bolsa de palabras	SVM	**1**	**1**	**1**
		Árboles	**1**	**1**	**1**
		Naive Bayes	0.9982	**1**	0.9991
		Mayoría de votos	**1**	**1**	**1**
	POS	SVM	0.9981	0.9991	0.9986
		Árboles	**1**	**1**	**1**
		Naive Bayes	0.9981	0.9724	0.9851
		Mayoría de votos	0.9982	0.9991	**1**
		Mayoría de votos Mejores	**1**	**1**	**1**

Cuadro 5.7: Resultados obtenidos para los corpus en inglés, con el primer nivel de la arquitectura propuesta, en los experimentos 80 %-20 %

Los resultados obtenidos mejoraron en ambos corpus en inglés, bastó con solo el primer nivel de la arquitectura propuesta, para obtener el 100 % en precisión, recuerdo y medida F, obteniendo una clasificación del 100 % correcta, en los experimentos del 60 %-40 % y 80 %-20 %.

Dados los resultados obtenidos para el corpus en español, se aplica el segundo nivel de la arquitectura propuesta: Cascada.

5.5.2. Nivel 2. Cascada

Con el objetivo de mejorar los resultados obtenidos en el corpus en español, se realizaron experimentos con el método de cascada, variando el número de clasificadores utilizados, con 2, 3 y 4 respectivamente, sin embargo se observó un comportamiento general en todos los casos: con 2 clasificadores el resultado mejoraba considerablemente, con respecto al obtenido con los clasificadores base, con 3 clasificadores, se esperaba que mejorara aún mas el resultado, sin embargo no fue así, los resultados decrecieron. No se obtuvo mejora alguna con 3 y 4 clasificadores. Por lo tanto se utilizó una cascada de nivel 2.

El procedimiento realizado para incluir cascada en los experimentos se describe a continuación:

Se aplica el método de cascada tomando como entrada la salida del nivel 1, se realiza la clasificación, con el clasificador que el sistema de manera automática detecta como mejor clasificador en medida F, respecto a representaciones textuales y clasificadores. Dicho proceso obtiene una salida que será la entrada al siguiente nivel de la arquitectura.

El resultado obtenido por cascada en los experimentos 60 %-40 %, se muestran en la tabla 5.8.

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Español	Bigramas	Cascada	**0.8346**	**0.8360**	**0.8353**

CUADRO 5.8: Resultados obtenidos con el nivel 2 de la arquitectura, para el corpus en español 60 %-40 %

Para comprobar que el sistema haya obtenido el mejor resultado se muestra en la tabla 5.9, los resultados obtenidos con todas las representaciones.

El resultado obtenido con los experimentos 80 %-20 %, se muestra en la tabla 5.10.

En la tabla 5.11 se muestran los resultados obtenidos por todas las representaciones para los experimentos 80 %-20 %.

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Español		Mayoría de votos Mejores	**0.7323**	**0.7028**	**0.7172**
	Bigramas	SVM	0.7157	0.7761	0.7447
		Cascada	**0.8346**	**0.8360**	**0.8353**
	Bolsa de palabras	Naive Bayes	0.6057	0.5468	0.5748
		Cascada	**0.7647**	**0.7860**	**0.7752**
	POS	Árboles	0.5759	0.5431	0.5590
		Cascada	0.6352	0.6581	0.6464
	Valoración	SVM	0.6932	0.5597	0.6193
		Cascada	**0.7575**	**0.7506**	**0.7541**

Cuadro 5.9: Resultados obtenidos con todas las representaciones en el nivel 2 de la arquitectura, para el corpus en español 60 %-40 %

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Español	Valoración	Cascada	**0.9828**	**0.9385**	**0.9602**

Cuadro 5.10: Resultado obtenido con el nivel 2 de la arquitectura, para el corpus en español, 80 %-20 %

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Español		Mayoría de votos Mejores	**0.9159**	**0.8920**	**0.9038**
	Bigramas	SVM	0.8241	0.8213	0.8227
		Cascada	**0.9252**	**0.9340**	**0.9296**
	Bolsa de palabras	Naive Bayes	0.6555	0.5361	0.5898
		Cascada	**0.9240**	**1**	**0.9592**
	POS	Naive Bayes	0.5799	**0.6735**	**0.6232**
		Cascada	**0.9204**	0.6198	0.5750
	Valoración	SVM	0.7500	0.5979	0.6654
		Cascada	**0.9828**	**0.9385**	**0.9602**

Cuadro 5.11: Resultados obtenidos con todas las representaciones en el nivel 2 de la arquitectura, para el corpus en español, 80 %-20 %

Los resultados obtenidos, aplicando el método cascada al corpus en español para ambos experimentos 60 %-40 % y 80 %-20 %, mejoraron considerablemente los resultados, pues todas las representaciones a excepción de POS, superaron en precisión, recuerdo y medida F a los resultados obtenidos por los clasificadores base. Con la representación POS, los resultados no favorecieron del todo, pues supero a los demás clasificadores en precisión, sin embargo en recuerdo y medida F, fue superado.

Los resultados para el corpus en español han sido mejorados hasta en un 16.9 % en este nivel, sin embargo no se ha logrado obtener una clasificación del 100 %, por lo que, con el objetivo de mejorar aún más los resultados obtenidos, se requiere aplicar el nivel 3 de la arquitectura al corpus en español.

Se aplicó el nivel 2 de la arquitectura a los corpus en inglés, con el objetivo de observar

el comportamiento, dado que en este corpus ya se ha obtenido el 100 % de clasificación correcta. Los resultados se muestran en las tablas 5.12 y 5.13.

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Inglés 2000	Bigramas	SVM	0.785219	0.862944	0.822269
		Árboles	0.796721	1	0.886881
		Naive Bayes	0.491206	0.992386	0.657163
		mayoría de votos	0.8107	1	0.895475
		Cascada	**0.997561**	**1**	**0.998799**
Inglés 2000	Bolsa de palabras	SVM	1	0.997462	0.998749
		Árboles	1	0.995885	0.997958
		Naive Bayes	0.994924	0.994924	0.994944
		mayoría de votos	1	0.997462	0.998749
		Cascada	**0.997561**	**1**	**0.998799**
Inglés 2000	POS	SVM	1	0.994924	0.997475
		Árboles	1	0.99177	0.995888
		Naive Bayes	1	0.92132	0.959069
		mayoría de votos	1	0.994924	0.997475
		Cascada	**0.997561**	**1**	**0.998799**
		mayoría de votos Mejores	1	1	1
Inglés 10662	Bigramas	SVM	0.987648	0.966078	0.976764
		Árboles	0.987934	0.975065	0.981477
		Naive Bayes	0.50469	1	0.670843
		mayoría de votos	0.988062	1	0.994015
		Cascada	**0.999522**	**1**	**0.999781**
Inglés 10662	Bolsa de palabras	SVM	0.999628	0.999628	0.999834
		Árboles	0.998883	0.998511	0.998717
		Naive Bayes	1	0.998141	0.99909
		mayoría de votos	1	1	1
		Cascada	**0.999522**	**1**	**0.999781**
Inglés 10662	POS	SVM	0.999071	1	0.999556
		Árboles	0.988231	1	0.994101
		Naive Bayes	0.999039	0.966543	0.982542
		mayoría de votos	0.999071	1	0.999556
		Cascada	**0.999522**	**1**	**0.999781**
		mayoría de votos Mejores	1	1	1

Cuadro 5.12: Resultados obtenidos con el nivel 2 de la arquitectura, para los corpus en inglés, 60 %-40 %

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
		SVM	0.846154	0.858537	0.85232
		Árboles	1	1	1
	Bigramas	Naive Bayes	0.508861	0.980488	0.67002
		mayoría de votos	0.86383	0.990244	0.922747
		Cascada	**1**	**1**	**1**
		SVM	1	1	1
		arboles	1	1	1
Inglés 2000	Bolsa de palabras	Naive Bayes	0.990338	1	0.995166
		mayoría de votos	1	1	1
		Cascada	**1**	**1**	**1**
		SVM	0.995098	0.990244	0.992685
		Árboles	1	1	1
	POS	Naive Bayes	0.990196	0.985366	0.987795
		mayoría de votos	0.995146	1	0.997587
		Cascada	**1**	**1**	**1**
		mayoría de votos Mejores	1	1	1
		SVM	0.989681	0.970561	0.980048
		Árboles	1	1	1
	Bigramas	Naive Bayes	0.50985	1	0.675385
		mayoría de votos	0.989982	1	0.994986
		Cascada	**0.998162**	**0.99908**	**0.998641**
		SVM	1	1	1
		Árboles	1	1	1
Inglés 10662	Bolsa de palabras	Naive Bayes	0.998163	1	0.999101
		mayoría de votos	1	1	1
		Cascada	**1**	**0.824595**	**0.903886**
		SVM	0.998162	0.99908	0.998641
		Árboles	1	1	1
	POS	Naive Bayes	0.998111	0.972401	0.985109
		mayoría de votos	0.998162	0.99908	1
		Cascada	**0.998162**	**1**	**0.9991**
		mayoría de votos Mejores	1	1	1

Cuadro 5.13: Resultados obtenidos con el nivel 2 de la arquitectura, para los corpus en inglés 80 %-20 %

Como se puede observar en las tablas anteriores, los resultados obtenidos con el nivel dos para los corpus en inglés, conservan sus resultados.

5.5.3. Nivel 3. Ventanas

Finalmente la última técnica utilizada es *Ventanas*, la cual se incorpora al final de la arquitectura, con el objetivo de realizar un análisis de las instancias clasificadas erróneamente, para poder elegir el mejor clasificador según la representación.

Como ya se ha mencionado en la descripción del método, consiste en dados los datos de entrenamiento y prueba, se realiza la clasificación con el clasificador seleccionado automáticamente por el sistema y se obtiene una clase, de la cual las instancias clasificadas erróneamente, son seleccionadas y agregadas al conjunto de datos de entrenamiento, así mismo son seleccionadas aleatoriamente el mismo número de instancias del conjunto de datos de entrenamiento, para ser reemplazadas por las instancias erróneas y agregadas al conjunto de datos de prueba. Dicho procedimiento puede realizarse n veces.

A continuación se describe el procedimiento:

1. Se elige el valor del parámetro n, el cual se obtuvo en base a pruebas, realizando el algoritmo n veces hasta que una variable llamemosle *epsilon*, la cual corresponde a la mejora de la medida F, este dentro de un cierto intervalo, o bien si es que no hay una mejora considerable se toma el valor de un máximo número de iteraciones. El valor de n para la implementación de ventanas en los experimentos realizados es 4.
2. Se toma la salida del nivel anterior como entrada, y se selecciona el mejor clasificador en medida F, con respecto de representaciones textuales y clasificadores utilizados.

En tabla 5.14 se muestra el resultado obtenido en el corpus en español con los experimentos 60 %-40 %, para el nivel 3.

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Español	Valoración	Ventanas	**0.8632**	0.7047	**0.7760**

CUADRO 5.14: Resultado obtenido por el sistema en el nivel 3, para el corpus en español, experimentos 60 %-40 %

A continuación se presenta en la tabla 5.15, los resultados con cada representación, lo que nos permite visualizar que el sistema da como salida el mejor resultado en cuanto a medida F, en el nivel tres de la arquitectura.

[0.8mm] Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
[0.8mm]		Mayoría de votos Mejores	0.7323	0.7028	0.7172
Español	Bigramas	SVM	0.7157	0.7761	0.7447
		Cascada	**0.8346**	0.8360	**0.8353**
		Ventanas	0.8091	**0.8385**	0.8235
Español	Bolsa de palabras	Naive Bayes	0.6057	0.5468	0.5748
		Cascada	0.7647	**0.7860**	**0.7752**
		Ventanas	**0.9685**	0.6178	0.7544
	POS	Árboles	0.5758	0.5431	0.5590
		Cascada	0.6352	**0.6581**	**0.6464**
		Ventanas	**0.8946**	0.4773	0.6225
	Valoración	SVM	0.6932	0.5596	0.6193
		Cascada	0.7575	**0.7506**	0.7541
		Ventanas	**0.8632**	0.7047	**0.7760**

CUADRO 5.15: Resultados para todas las representaciones con ventanas, con el nivel 3, corpus en español y experimentos 60 %-40 %

El resultado obtenido por el sistema, en los experimentos 80 %-20 % se muestra en la tabla 5.16.

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Español	Bigramas	Ventanas	**0.9788**	**0.9652**	**0.9719**

CUADRO 5.16: Resultado obtenido por el sistema en el nivel 3, para el corpus en español, experimentos 80 %-20 %

Para corroborar que el sistema haya obtenido el mejor resultado, se muestra e la tabla 5.17, todas las representaciones.

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
		Mayoría de votos Mejores	0.9159	0.8920	0.9038
	Bigramas	SVM	0.8241	0.8213	0.8227
		Cascada	0.9252	0.9340	0.9296
		Ventanas	**0.9788**	**0.9652**	**0.9719**
Español	Bolsa de palabras	Naive Bayes	0.6555	0.5361	0.5898
		Cascada	0.9240	**1**	**0.9592**
		Ventanas	**0.9526**	0.9198	0.9359
	POS	Naive Bayes	0.5799	0.6735	0.6232
		Cascada	0.9204	0.6198	0.5750
		Ventanas	**0.9221**	**0.9457**	**0.9338**
	Valoración	SVM	0.7500	0.5979	0.6654
		Cascada	**0.9828**	**0.9385**	**0.9602**
		Ventanas	**0.9828**	**0.9385**	**0.9602**

CUADRO 5.17: Resultados para todas las representaciones con ventanas en el nivel 3, corpus en español y experimentos 80 %-20 %

Finalmente se observa que aplicando el último nivel de la arquitectura se logra obtener un porcentaje de hasta el 0.9719 en medida F, lo cual representa un porcentaje de mejora del 18.13 % sobre los resultados obtenidos por el mejor clasificador base.

Con el fin de observar el comportamiento del corpus en inglés, se presentan los resultados obtenidos en el nivel tres de la arquitectura, para estos corpus en las tablas 5.18 y 5.19.

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Inglés 2000	Bigramas	Árboles	0.796721	1	0.886881
		mayoría de votos	0.8107	1	0.895475
		Cascada	0.997561	1	0.998799
		Ventanas	**1**	**1**	**1**
Inglés 2000	Bolsa de palabras	SVM	1	0.997462	0.998749
		mayoría de votos	1	0.997462	0.998749
		Cascada	0.997561	1	0.998799
		Ventanas	**1**	**1**	**1**
Inglés 2000	POS	SVM	1	0.994924	0.997475
		mayoría de votos	1	0.994924	0.997475
		Cascada	0.997561	1	0.998799
		Ventanas	**1**	**1**	**1**
		mayoría de votos Mejores	1	1	1
Inglés 10662	Bigramas	Árboles	0.987934	0.975065	0.981477
		mayoría de votos	0.988062	1	0.994015
		Cascada	0.999522	1	0.999781
		Ventanas	**1**	**1**	**1**
Inglés 10662	Bolsa de palabras	SVM	0.999628	0.999628	0.999834
		mayoría de votos	1	1	1
		Cascada	0.999522	1	0.999781
		Ventanas	**1**	**1**	**1**
Inglés 10662	POS	SVM	0.999071	1	0.999556
		mayoría de votos	0.999071	1	0.999556
		Cascada	0.999522	1	0.999781
		Ventanas	**1**	**1**	**1**
		mayoría de votos Mejores	1	1	1

Cuadro 5.18: Resultados de los corpus en inglés para el nivel 3, 60 %-40 %

Corpus	Representación	Clasificador	Precisión	Recuerdo	Medida F
Inglés 2000	Bigramas	Árboles	**1**	**1**	**1**
		mayoría de votos	0.86383	0.990244	0.922747
		Cascada	**1**	**1**	**1**
		Ventanas	**1**	**1**	**1**
Inglés 2000	Bolsa de palabras	SVM	**1**	**1**	**1**
		mayoría de votos	0.994872	**1**	**1**
		Cascada	**1**	**1**	**1**
		Ventanas	**1**	**1**	**1**
Inglés 2000	POS	arboles	**1**	**1**	**1**
		mayoría de votos	0.994872	0.995146	**1**
		Cascada	**1**	**1**	**1**
		Ventanas	**1**	**1**	**1**
		mayoría de votos Mejores	**1**	**1**	**1**
Inglés 10662	Bigramas	Árboles	**1**	**1**	**1**
		mayoría de votos	0.989982	**1**	0.994986
		Cascada	0.998162	0.99908	0.998641
		Ventanas	**1**	**1**	**1**
Inglés 10662	Bolsa de palabras	arboles	**1**	**1**	**1**
		mayoría de votos	**1**	**1**	**1**
		Cascada	**1**	0.824595	0.903886
		Ventanas	**1**	**1**	**1**
Inglés 10662	POS	arboles	**1**	**1**	**1**
		mayoría de votos	0.998162	0.99908	0.998641
		Cascada	**1**	**1**	**1**
		Ventanas	**1**	**1**	**1**
		mayoría de votos Mejores	**1**	**1**	**1**

Cuadro 5.19: Resultados de los corpus en inglés para el nivel 3, 80 %-20 %

Dados los resultados obtenidos por los corpus en inglés, para el nivel 3, podemos observar que también se siguen manteniendo los resultados encontrados desde el nivel 1. Por lo cual para los corpus en inglés solo se propone utilizar el nivel uno de la arquitectura.

La arquitectura que se propone para el corpus en español, es realizar los tres niveles de la arquitectura, utilizando a partir del nivel 2 la representación de bigramas, y el clasificador SVM.

Capítulo 6

Conclusiones

Existen trabajos limitados de análisis de sentimientos para textos en español, en los que es notorio que la medida F alcanzada es mucho más baja que los métodos reportados para el idioma inglés.

Se han realizado varios experimentos con distintos métodos de clasificación y distintas formas de representación de los datos, los resultados obtenidos han sido muy diversos. Se ha podido distinguir que las características utilizadas han sido un factor determinante para obtener resultados satisfactorios, la selección de características es un aspecto muy importante puesto que el éxito de la clasificación depende de tomar las características o los atributos que mejor representen a los documentos, ya que de esto dependerá expresar la polaridad correcta de los documentos y obtener resultados satisfactorios.

Los resultados obtenidos con la arquitectura propuesta para los corpus en español e inglés, son superiores en precisión, recuerdo y medida F, con respecto a los obtenidos por los clasificadores base.

El resultado obtenido por el primer nivel de la arquitectura en el corpus en español, obtuvo en los experimentos 80 %-20 %, un porcentaje de mejora en medida F del 9.85 % con respecto del mejor clasificador base, y en los experimentos 60 %-40 %, no se obtuvo mejora alguna al respecto, por lo cual fueron necesarios los niveles dos y tres de la arquitectura. En el nivel dos, se obtiene un porcentaje de mejora de la medida F de hasta el 16.71 %, respecto a los clasificadores base. A pesar de la mejora obtenida en este nivel, no se logra obtener el 100 % de la clasificación correcta, por lo cual se aplica el nivel tres de la arquitectura, en donde se obtiene un porcentaje de mejora en medida F, de hasta el 18.13 % respecto al mejor clasificador base.

Aplicando la arquitectura a los corpus en inglés, desde el primer nivel se obtienen resultados del 100 % de clasificación correcta, sin embargo se aplicó la arquitectura en los niveles dos y tres, para poder validar que los resultados obtenidos en el nivel uno fueran conservados.

Bigramas resultó ser la representación de textos que permitió obtener mejores resultados, en la arquitectura.

SVM es el clasificador base que mostró mejor desempeño en la arquitectura propuesta.

La arquitectura propuesta obtuvo una mejora del 12.52 %, con respecto a un trabajo publicado en 2011, que utiliza el mismo corpus.

La arquitectura propuesta cumple con los objetivos planteados para esta investigación.

Referencias

[1] Eugenio Martínez-Cámara, Maite Teresa Martín-Valdivia, José M. Perea-Ortega, and Luis Alfonso Ureña López. Técnicas de clasificación de opiniones aplicadas a un corpus en español. *Procesamiento del Lenguaje Natural*, 47:163–170, 2011. URL http://dblp.uni-trier.de/db/journals/pdln/pdln47.html#Martinez-CamaraMPL11.

[2] Peter D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073153. URL http://dx.doi.org/10.3115/1073083.1073153.

[3] Bing Liu. Web usage mining. In *Web Data Mining*, Data-Centric Systems and Applications, pages 449–483. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-37881-5. doi: 10.1007/978-3-540-37882-2_12. URL http://dx.doi.org/10.1007/978-3-540-37882-2_12.

[4] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[5] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[6] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004. ISBN 0471210781.

[7] Michael John Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 184–191, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. doi: 10.3115/981863.981888. URL http://dx.doi.org/10.3115/981863.981888.

[8] F. Benamara, C. Cesarano, A. Picariello, D. Reforgiato, and V.S. Subrahmanian. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.

[9] Ellen Riloff, Janyce Wiebe, and Theresa Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 25–32, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119176.1119180. URL http://dx.doi.org/10.3115/1119176.1119180.

[10] Peter R. R. White. Appraisal outline. Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. URL www.grammatics.com/appraisal.

[11] V. M. Morales de Jesús. Utilización de expresiones de actitud para el análisis de sentimientos. Puebla, Puebla., 2014.

[12] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86, 2002.

[13] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012. doi: 10.2200/S00416ED1V01Y201204HLT016. URL http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016.

[14] Kai Ming Ting and Ian H. Witten. Stacked generalization: when does it work? In *in Procs. International Joint Conference on Artificial Intelligence*, pages 866–871. Morgan Kaufmann, 1997.

[15] Kagan Tumer and Joydeep Ghosh. Linear and order statistics combiners for pattern classification. *CoRR*, cs.NE/9905012, 1999. URL http://arxiv.org/abs/cs.NE/9905012.

[16] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. A Wiley Interscience Publication. Wiley, 1973.

[17] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. ISBN 0123748569, 9780123748560.

[18] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, 2001.

[19] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3):226–239, Mar 1998. ISSN 0162-8828. doi: 10.1109/34.667881.

[20] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, London, UK, UK, 2000. Springer-Verlag. ISBN 3-540-67704-6. URL http://dl.acm.org/citation.cfm?id=648054.743935.

[21] João Gama and Pavel Brazdil. Cascade generalization. *Machine Learning*, 41 (3):315–343, 2000. ISSN 0885-6125. doi: 10.1023/A:1007652114878. URL http://dx.doi.org/10.1023/A%3A1007652114878.

[22] Uci. machine learning repository. urlhttp://archive.ics.uci.edu/ml/, .

[23] V. Anitha and R.Leela Velusamy. Iris recognition systems with reduced storage and high accuracy using majority voting and haar transform. Advances in Intelligent Systems and Computing, pages 813–822. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-30110-0. doi: 10.1007/978-3-642-30111-7_78. URL http://dx.doi.org/10.1007/978-3-642-30111-7_78.

[24] Maurizio Aiello, Maurizio Mongelli, and Gianluca Papaleo. Supervised learning approaches with majority voting for dns tunneling detection. In *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14*, volume 299 of *Advances in Intelligent Systems and Computing*, pages 463–472. Springer International Publishing, 2014. ISBN 978-3-319-07994-3. doi: 10.1007/978-3-319-07995-0_46. URL http://dx.doi.org/10.1007/978-3-319-07995-0_46.

[25] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

[26] Johannes Fürnkranz. More efficient windowing. Technical Report OEFAI-TR-97-01, Austrian Research Institute for Artificial Intelligence, Wien, Austria, 1997. URL http://www.ke.informatik.tu-darmstadt.de/~juffi/publications/aaai-97.ps.gz.

[27] Bing Liu 0001 and Lei Zhang 0016. A survey of opinion mining and sentiment analysis. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 415–463. Springer, 2012. ISBN 978-1-4419-8462-3. URL http://dblp.uni-trier.de/db/books/collections/Mining2012.html#LiuZ12.

[28] Bing Liu, Bamshad Mobasher, and Olfa Nasraoui. Web usage mining. In *Web Data Mining*, Data-Centric Systems and Applications, pages 527–603. Springer Berlin

Heidelberg, 2011. ISBN 978-3-642-19459-7. doi: 10.1007/978-3-642-19460-3_12. URL http://dx.doi.org/10.1007/978-3-642-19460-3_12.

[29] Wenqian Shang, Houkuan Huang, Haibin Zhu, Yongmin Lin, Zhihai Wang, and Youli Qu. An improved knn algorithm – fuzzy knn. In Yue Hao, Jiming Liu, Yuping Wang, Yiu-ming Cheung, Hujun Yin, Licheng Jiao, Jianfeng Ma, and Yong-Chang Jiao, editors, *Computational Intelligence and Security*, volume 3801 of *Lecture Notes in Computer Science*, pages 741–746. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-30818-8. doi: 10.1007/11596448_109. URL http://dx.doi.org/10.1007/11596448_109.

[30] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Comput. Linguist.*, 37(2):267–307, June 2011. ISSN 0891-2017. doi: 10.1162/COLI_a_00049. URL http://dx.doi.org/10.1162/COLI_a_00049.

[31] Ali Zulfiqar, Aslam Muhammad, AnaMaria Martinez-Enriquez, and G. Escalada-Imaz. Text-independent speaker identification using vq-hmm model based multiple classifier system. In Grigori Sidorov, Arturo Hernández Aguirre, and CarlosAlberto Reyes García, editors, *Advances in Soft Computing*, volume 6438 of *Lecture Notes in Computer Science*, pages 116–125. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-16772-0. doi: 10.1007/978-3-642-16773-7_10. URL http://dx.doi.org/10.1007/978-3-642-16773-7_10.

[32] Lijun Dai and Chuang Liu. Multiple classifier combination for land cover classification of remote sensing image. In *Information Science and Engineering (ICISE), 2010 2nd International Conference on*, pages 3835–3839, Dec 2010. doi: 10.1109/ICISE.2010.5691420.

[33] Lluís Padró and Evgeny Stanilovsky. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*, Istanbul, Turkey, May 2012. ELRA.

[34] The stanford natural language processing group. urlhttp://nlp.stanford.edu/software/tagger.shtml, .

[35] Fermín L. Cruz, José A. Troyano, Fernando Enríquez, and F. Javier Ortega. Clasificación de documentos basada en la opinión: experimentos con un corpus de críticas de cine en español. 2008.

[36] Rotten tomatoes page. urlhttp://www.rottentomatoes.com/, .