SELECCIÓN DE CARACTERÍSTICAS MEDIANTE MUESTREO, TÉCNICAS DE INDUCCIÓN DE REGLAS Y ARREGLOS DE COBERTURA PARA DETECCIÓN DE POLARIDAD EN ANÁLISIS DE SENTIMIENTOS EN TWITTER

Universidad
del Cauca

JORGE ARMANDO VILLEGAS GONZÁLEZ

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE SISTEMAS
GRUPO DE I+D EN TECNOLOGÍAS DE LA INFORMACIÓN (GTI)
ÁREAS DE INVESTIGACIÓN: GESTIÓN DE LA INFORMACIÓN – MINERÍA DE TEXTOS Y PROCESAMIENTO DE LENGUAJE NATURAL
POPAYÁN
2018

SELECCIÓN DE CARACTERÍSTICAS MEDIANTE MUESTREO, TÉCNICAS DE INDUCCIÓN DE REGLAS Y ARREGLOS DE COBERTURA PARA DETECCIÓN DE POLARIDAD EN ANÁLISIS DE SENTIMIENTOS EN TWITTER

TESIS PRESENTADA A LA FACULTAD DE
INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES DE LA UNIVERSIDAD
DEL CAUCA PARA LA OBTENCIÓN DEL TÍTULO
DE

MAGÍSTER EN COMPUTACIÓN

DIRECTORES
DIRECTOR: PHD. CARLOS ALBERTO COBOS LOZADA
CO-DIRECTOR: PHD. MARTHA ELIANA MENDOZA BECERRA

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES
DEPARTAMENTO DE SISTEMAS
GRUPO DE I+D EN TECNOLOGÍAS DE LA INFORMACIÓN (GTI)
ÁREAS DE INVESTIGACIÓN: GESTIÓN DE LA INFORMACIÓN –
MINERÍA DE TEXTOS Y PROCESAMIENTO DE LENGUAJE
NATURAL
POPAYÁN
2018

Dedicatoria

A Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A Daniela Gonzalez, por ser mi consejera, quererme mucho, creer en mí y porque siempre me apoyaste.

A mis tíos: Yesid, Judit, Alfonso, Teresa, Herlendith, Nelly Villegas y mis abuelos: Belen y Gregorio por ser el modelo a seguir, por regalarme su amor y su atención en cada momento, por guiarme a través de mi vida, por creer en mí y brindarme su apoyo, por sembrar en mí tantos valores que hacen de mí hoy una gran persona.

A mis hermanas Leidy y Saduar por estar a mi lado compartiendo mis triunfos y fracasos y brindarme su apoyo incondicional.

A mis sobrinos y primos, para que vean en mí un ejemplo a seguir.

Todos mis amigos, Karol, Carlos Ramírez, Robin, Hernan, Pedro, Jose, Darwin, Nohemi, Victor, Eduardo, Arnol, Ariana, Jhonatan y Paola, por compartir los buenos y malos momentos.

A Mahuer, que en paz descanses amigo, gracias por toda la ayuda, creíste en mi cuando pocos lo hacían, gracias por comprarme el formulario para ingresar a la universidad cuando la escasez intentaba apoderarse de mis sueños.

Todos aquellos familiares y amigos que no recordé al momento de escribir esto. Ustedes saben quiénes son.

Agradecimientos

El presente trabajo de investigación fue realizado bajo la supervisión del Ph.D. Carlos Cobos y la Ph.D. Martha Mendoza, a quienes me gustaría expresar mi más profundo agradecimiento, por hacer posible la realización de este estudio. Además, de agradecer su paciencia, tiempo y dedicación que tuvieron para que esto saliera de manera exitosa.

Al profesor Enrique Herrera-Viedma, por su colaboración, asesoría y la oportunidad de realizar mi estancia en la Universidad de Granada- España, prestigiosa universidad y hermosa ciudad, que aportaron en mis muchas experiencias enriquecedoras para mi vida.

A la Universidad del Cauca y todos los docentes que con su tiempo y dedicación participaron en mi formación como Magister.

Resumen Estructurado

Una de las principales tareas en el Análisis de Sentimientos en Twitter es la detección de polaridad, que se resume en clasificar los 'tweets' en términos de los sentimientos, opiniones y actitudes que expresan. La detección de polaridad en Twitter mediante métodos de aprendizaje de máquina en general se ve afectada por el uso de características irrelevantes, redundantes, ruidosas o correlacionadas, máxime cuando se usa una representación de alta dimensionalidad en el conjunto de características.

Por lo anterior, se hace necesario de un método de selección de características que permita eliminar aquellas que hacen ineficiente el funcionamiento del algoritmo de clasificación. En este trabajo, se propone un método para la selección de las características basado en el concepto de bagging, con dos modificaciones importantes: i), el uso de arreglos de cobertura para soportar el proceso de definición del número de muestras bootstrap y las características a incluir en cada uno de ellos y ii) el uso del resultado de las técnicas de inducción de reglas (JRIP, C4.5, CART u otro) para generar la representación reducida de los tweets con las características seleccionadas.

Los resultados experimentales muestran que al usar el método propuesto se obtienen resultados similares o superiores a los obtenidos con la representación original (incluye un conjunto de 91 características usadas en trabajos relacionados con detección de polaridad en Twitter) y permite obtener modelos más sencillos y rápidos de procesar. Es así como se identifica un subconjunto de características que permiten soportar mejoras en las futuras propuestas de detección de polaridad en Twitter.

Palabras claves: Análisis de sentimientos, detección de polaridad, arreglos de cobertura, selección de características, Twitter.

Structured Abstract

One of the main tasks in analyzing sentiment on Twitter is polarity detection – i.e. the classification of 'tweets' in terms of feelings, opinions and attitudes expressed. Polarity detection on Twitter by means of machine learning methods is generally affected by the use of irrelevant, redundant, noisy or correlated features, especially when a high-dimensional representation is used in the feature set.

There is thus a need for a selection method that removes those features that render the classification algorithm inefficient. In this work, we propose a feature selection method based on the concept of bagging, with two important modifications: i) the use of covering arrays to support the process of building bootstrap samples and the characteristics to be included in each of them and ii) the use of the results of rule-induction techniques (JRIP, C4.5, CART or others) to generate the reduced representation of tweets with the features selected.

The experimental results show that on using the method proposed, we obtain similar or better results than those obtained with the original representation (this comprising a set of 91 features used in research related to polarity detection in Twitter), bringing the possibility of simpler and faster process models. A subset of features is thereby identified that can facilitate improvements in future polarity detection proposals on Twitter.

Keywords: Sentiment analysis, polarity detection, covering arrays, feature selection, Twitter.

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE ANEXOS

Anexo 1: Artículo “Feature Selection using Sampling with Replacement, Covering Arrays and Rule-Induction Techniques to aid Polarity Detection in Twitter Sentiment Analysis”.

Anexo 2: Implementación en Visual Studio .Net del método de selección de características mediante muestreo, técnicas de inducción de reglas y arreglos de cobertura para detección de polaridad en análisis de sentimientos en twitter en lenguaje c#.

Anexo 3: Comparación entre clasificación de características sin incluir incrustaciones, solo incrustaciones e incluyendo incrustaciones.

Anexo 4: Implementación del preprocesamiento.

Anexo 5: Vistas minables generadas del preprocesamiento.

Anexo 6: Covering Arrays.

Anexo 7: Framework de prueba.

Anexo 8: Resumen de experimentos.

Capítulo 1

1 Introducción

1.1 Definición del Problema

La Web 2.0 proporciona a las personas, la posibilidad de expresar y compartir libremente las opiniones sobre las diferentes actividades del día a día [1]. Debido a lo cual, los mensajes publicados en los sitios de redes sociales han ayudado a mejorar los negocios e influir en la opinión pública, afectando profundamente nuestras vidas sociales y políticas [2]. En la última década y media, las comunidades de investigación, la academia, la industria y los servicios públicos han estado trabajando de manera notable, en el análisis de sentimientos, también conocido como minería de opinión, para extraer y analizar el estado de ánimo del público y sus puntos de vista sobre diferentes temas [1].

El análisis de sentimientos se encarga de la detección, extracción y clasificación de las opiniones, sentimientos y actitudes en relación con diferentes temas, ayuda en la consecución de diversos objetivos, como la observación del ánimo del público en relación con el movimiento político, inteligencia de mercado [3], la medición de la satisfacción del cliente [4], la predicción de ventas de películas [5] y muchos más [2, 6].

Debido a su creciente popularidad, Twitter ha atraído el interés de muchos investigadores en el área de recuperación de información, procesamiento de lenguaje natural y análisis de sentimientos, generando una subárea de investigación denominada *'Análisis de Sentimientos en Twitter'* (AST) donde se aborda el problema de analizar los *'tweets'* (mensajes publicados en Twitter) en términos de los sentimientos que expresan. Un aspecto importante de Twitter es que la información que se publica con frecuencia contiene opiniones sobre productos, servicios, celebridades, eventos y en general diversos temas que son de interés de los usuarios de esta aplicación. Twitter es un nuevo dominio para el análisis de sentimientos y plantea unos desafíos únicos, entre ellos: la limitación del tamaño de los mensajes a 140 caracteres (o más recientemente a 280), el lenguaje informal (o que usa jergas) que esta restricción de tamaño de textos ha generado y el uso de emoticones para expresar y enfatizar sentimientos [7].

En cuanto a la detección de polaridad, se identifica como uno de los problemas más complejos para el análisis de sentimientos, debido a que palabras de sentimiento positivo o negativo pueden tener orientaciones o polaridades opuestas en diferentes contextos. Por orientación o polaridad, se dice que un sentimiento u opinión es positiva, negativa o neutral. Por ejemplo, "*suck*" suele indicar un sentimiento negativo, por ejemplo, *"This camera sucks"*, pero también puede implicar un sentimiento positivo, por ejemplo, *"This vacuum cleaner really sucks"*. Por lo tanto,

las orientaciones del sentimiento de las palabras pueden ser dependientes del dominio o del contexto de la oración [1, 2].

Un enfoque con el que se aborda la detección de polaridad, es el aprendizaje automático, el cual trata el problema como un problema de clasificación, y debido a la alta dimensionalidad del conjunto de características, requiere de un proceso previo de selección de características para obtener resultados más precisos o para hacer más sencillo el modelo de clasificación; para este fin es necesario extraer las características, medir su importancia representativa y de acuerdo a esa importancia, descartar las de menos importancia para el proceso de clasificación [1, 7, 8].

En este contexto, los problemas más importantes se relacionan con la apropiada selección de las características y el manejo de la alta dimensionalidad de los conjuntos de características, que son determinantes en el proceso de clasificación o de detección de polaridad de los sentimientos [7].

Para contribuir en el desarrollo de una solución óptima a estos problemas, se planteó la siguiente pregunta de investigación: ¿Cómo realizar un proceso de selección de características usando una técnica de muestreo (con remplazo o ponderado), arreglos de cobertura y técnicas de inducción de reglas (JRip, C4.5, CART), de manera que se mejore la precisión en la detección de polaridad en el análisis de sentimientos sobre Twitter usando algoritmos tradicionales de clasificación como Naïve Bayes, Random Forest, Multi Layer Perceptron y SVM?

1.2 Aportes del proyecto

Desde la perspectiva de investigación, las contribuciones del presente proyecto se centran en la generación de nuevo conocimiento para la comunidad académica y científica de procesamiento de lenguaje natural, que trata específicamente el tema de selección de características y detección de polaridad en Twitter. El aporte se centra en el uso de un enfoque nuevo y diferente (teniendo en cuenta que, hasta la presentación de los resultados de esta tesis, no se había publicado ningún reporte de investigación con el enfoque propuesto, en las bases de datos de ACM, IEEE, ScienceDirect y Springer Link). El nuevo conocimiento que se encontró es de dos tipos: exploratorio y descriptivo, respondiendo a las preguntas: ¿Es posible obtener mejoras de precisión con el enfoque propuesto, o no? y ¿Cuáles son las condiciones que hacen que el enfoque propuesto mejore la precisión del proceso de detección de polaridad en tweets?

Adicionalmente, el uso del método propuesto permite seleccionar la mejor lista de características de un dataset de AST, sin hacer búsqueda combinatoria exhaustiva, que posibilita a los algoritmos de clasificación tradicionales obtener iguales o mejores resultados en cuanto a precisión con modelos más sencillos, lo que permite reducir el trabajo de representación de los tweets. Además de las implicaciones teóricas (mejor precisión, representación más compacta y sencilla y menor tiempo de procesamiento), las implicaciones en las aplicaciones reales son muy

importantes, por ejemplo: tener mayor seguridad en los sentimientos de los clientes sobre un producto, saber qué características se deben almacenar y cuales no y disminuir el tiempo del análisis para la modificación o construcción de productos.

1.3 Objetivos

A continuación, se presentan los objetivos de la presente tesis como fueron aprobados por el Consejo de Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca.

1.3.1 Objetivo general

Proponer un método de selección de características basado en una técnica de muestreo, técnicas de inducción de reglas y arreglos de cobertura, buscando una mayor precisión en la detección de polaridad en el análisis de sentimientos en Twitter.

1.3.2 Objetivos específicos

- Definir la línea base del estado del arte de las técnicas utilizadas para la selección de características en la detección de polaridad en Twitter y organizar algoritmos y datos en un framework de prueba.

- Proponer un proceso de selección de características usando una técnica de inducción de reglas (C4.5, CART o RIPPER) que trabaje con esquemas de muestreo de filas (aleatorio o ponderado) y subconjuntos de características (columnas) seleccionados mediante arreglos de cobertura buscando mejorar la precisión en la detección de polaridad en el análisis de sentimientos en Twitter.

- Evaluar los resultados de diferentes clasificadores usando el método de selección de características propuesto a través de medidas clásicas como Precisión, Recuerdo, medida F, Tasa de Verdaderos Positivos, Tasa de Falsos Positivos, Tasa de Verdaderos Negativos, Tasa de Falsos Negativos y compararlos con la línea base previamente definida.

1.4 Resultados Obtenidos

Los resultados principales del desarrollo de la tesis son:

- Monografía del trabajo de grado, cuya estructura del documento se detalla en la siguiente sección.

- Código fuente del preprocesamiento necesario para los datasets, el cual obtiene la vista minable de las características estudiadas, 91 en total. Código implementado en lenguaje Python 2 en el entorno de desarrollo Anaconda.

- Código fuente del método propuesto para selección de características implementado en lenguaje C# y desarrollado en el entorno de desarrollo Microsoft Visual Studio 2015.

- Artículo que resume el desarrollo de la investigación y los resultados obtenidos. Aceptado en la conferencia denominada IBERAMIA'2018 - 16th Ibero-American Conference on Artificial Intelligence, Trujillo, Perú, noviembre 13-16 de 2018. Dicho evento publica sus memorias en un volumen de la serie Lecture Notes in Artificial Intelligence (LNCS/LNAI) de Springer categorizada por el PUBLINDEX de COLCIENCIAS como A2 en 2017, (Ver **Anexo 1**).

1.5 Organización del documento

El documento de la monografía está estructurado de la siguiente manera:

Capítulo 1: Introducción: Hace referencia al presente capitulo que introduce el tema de investigación, presenta la pregunta de investigación que originó el trabajo, los aportes al problema, también los objetivos (general y específicos) definidos en el anteproyecto, los resultados principales del desarrollo de la tesis y finalmente la organización de la monografía.

Capítulo 2. Presenta el marco teórico, que incluye los conceptos básicos necesarios para contextualizar la temática tratada en la tesis.

Capítulo 3. Presenta el estado del arte relacionado con la tesis, destacando las características que actualmente se han tenido en cuenta para el análisis de sentimientos, realizar procesos de selección de características y mejoras en la detección de polaridad en Twitter.

Capítulo 4. Presenta los resultados obtenidos en 3 competencias previas de evaluación de clasificación de polaridad en Twitter, también algunos algoritmos relevantes que se implementaron y se probaron con los datasets utilizados en la presente investigación y algunos sistemas encontrados para los cuales se consideró importante realizar una pequeña descripción que muestre la importancia de algunos de sus componentes, todo esto conforma el framework de prueba que permitió definir la línea base de la investigación.

Capítulo 5. Presenta el método propuesto, detallando cada uno de sus pasos: preprocesamiento y definición de características iniciales, muestreo de filas con reemplazo y de columnas basado en arreglos de cobertura e inducción de reglas y selección de características.

Capítulo 6. Este capítulo presenta el análisis de los resultados de la experimentación y se realiza la comparación con el estado el arte.

Capítulo 7. Presenta las conclusiones y el trabajo futuro que el grupo de investigación espera realizar en el corto plazo.

Capítulo 8. Presenta las referencias bibliográficas utilizadas en el presente trabajo.

Capítulo 2

2 Marco Teórico

En este apartado, se presentan los desafíos del análisis de sentimientos sobre Twitter y los enfoques utilizados para la solución de este problema, luego, se definen los arreglos de cobertura y se presentan algunos conceptos básicos de técnicas de inducción de reglas.

2.1 Desafíos del análisis de sentimientos en Twitter

El análisis de sentimientos en Twitter enfrenta varios desafíos entre los cuales se tienen [7]:

- Longitud del texto: Una de las características únicas de los tweets es su corta longitud, que puede tener un máximo de 140 caracteres (o más recientemente de 280) que ha motivado el uso de un lenguaje informal que se debe procesar con herramientas específicas para el área.

- Relevancia de temas: La mayor parte del trabajo que se realiza en AST pretende clasificar la orientación sentimental de un tweet sin tener en cuenta la relevancia del tema de un tweet, en muchos trabajos se considera la presencia de una palabra como un indicador de la relevancia tópica y en otros se consideran los 'hashtags' como un fuerte indicador de la relevancia del tweet hacia un tema específico. Debido a la corta longitud de los tweets, esos enfoques pueden ser parcialmente correctos, en la mayoría de los casos, el sentimiento se centrará en ese tema específico.

- Lenguaje Incorrecto: Debido a su tipo informal y la limitación de longitud, el lenguaje utilizado en Twitter es muy diferente del lenguaje utilizado en otros géneros de texto (web, blogs, noticias, etc.). Los tweets contienen peculiaridades textuales que incluyen mayúsculas enérgicas, alargamiento enfático, abreviaturas y el uso de argot y neologismos (palabra o expresión de nueva creación en una lengua).

- Poca densidad (Sparsity) de los datos: los tweets contienen mucho ruido debido al mal uso del inglés y errores ortográficos. La razón principal de la escasez de datos en Twitter es el hecho de que un gran porcentaje de los términos que aparecen en los tweets se producen menos de 10 veces en todo el corpus.

- Negación: La presencia de palabras negativas tiene un papel importante en la detección de la polaridad de un mensaje. Las negaciones pueden causar la

inversión de la polaridad de un mensaje (positivo se convierte en negativo o viceversa).

- Palabras vacías: Son palabras comunes que tienen bajo poder de discriminación (por ejemplo, artículos como el, la, los, las, verbos como es y pronombres relativos como quién), y generalmente se filtran antes de procesar el texto. Las listas típicas de palabras vacías no son adecuadas para Twitter e incluso pueden influir negativamente en el rendimiento del análisis de sentimientos sobre Twitter. Por ejemplo, la palabra "like" generalmente se considera una 'palabra vacía' en muchas aplicaciones de NLP (procesamiento de lenguaje natural), sin embargo, tiene un importante poder de discriminación en Twitter.

- Tokenización: Otro desafío relacionado con AST es la tokenización de las oraciones. En lugar de dividir los términos basado en espacios en blanco, es necesario un tokenizador específico para Twitter.

- Contenido multilingüe: Los tweets están escritos en una amplia variedad de idiomas, a veces mezclados incluso en el mismo mensaje. La dificultad para la detección del lenguaje aumenta como resultado de la corta longitud de los tweets.

- Contenido multimodal: En algunos casos, los tweets contienen imágenes o vídeos. El análisis de imagen y video puede proporcionar información útil para determinar quién es el titular de la opinión o extraer la entidad.

Estos desafíos son muy importantes y tienen que ser considerados para el análisis de sentimientos en Twitter ya que para su procesamiento se requiere pasos adicionales o modificaciones a los realizados en el procesamiento tradicional de textos, con el objetivo de manejar adecuadamente las características únicas de estos textos. Debido al alcance del presente trabajo no se tuvo en cuenta el contenido multimodal en los tweets, se empleó sólo el componente textual, la representación de los emoticones como texto y se trabajó solamente tweets de idioma Ingles.

2.2 Enfoques para análisis de sentimientos en Twitter

En la literatura de análisis de sentimientos en Twitter se pueden identificar cuatro diferentes enfoques:

- Aprendizaje automático, este enfoque emplea métodos de aprendizaje de máquina para construir un clasificador que puede detectar la polaridad de los tweets indicando el sentimiento. Algunos de estos clasificadores son: Naive Bayes, Support Vector Machines (SVM), Multinomial Naive Bayes, Regresión Logística, Random Forest, Multi Layer Perceptron y Campos Aleatorios Condicionales [2, 6, 7, 9-11].

- Basados en Léxico, utiliza una lista generada manual o automáticamente de términos positivos y negativos para detectar la polaridad del mensaje. Luego calcula la orientación de un tweet a partir de la orientación semántica de palabras o frases en el tweet. La principal ventaja de estos métodos es que no requieren datos de entrenamiento [2, 12].

- Híbridos, combinan métodos de aprendizaje automático y enfoques basados en el léxico [8, 13]. Por ejemplo en [14] los autores usan léxicos de dos palabras y datos no etiquetados, dividiendo estos léxicos de dos palabras en dos clases discretas, negativas y positivas. Con esto, crean pseudo documentos que abarcan todas las palabras del conjunto de léxicos elegidos. Luego calculan la similitud de cosenos entre los pseudo documentos y los documentos no etiquetados. Dependiendo de la medida de similitud, se asigna a estos nuevos documentos un sentimiento positivo o negativo. Este conjunto de datos se usa luego para entrenar un clasificador Naïve Bayes.

- Basados en Grafos, aunque los métodos de aprendizaje automático alcanzan un rendimiento aceptable en el análisis de sentimientos en Twitter, requieren un gran número de datos previamente clasificados que sirvan para el entrenamiento del clasificador. La propagación de etiquetas es un método que puede reducir la demanda de los datos anotados (clasificados). Para este fin, se ha utilizado el grafo social de Twitter bajo el supuesto de que las personas se influencian mutuamente [8].

2.3 Arreglos de Cobertura

Los arreglos de cobertura identificados como CA por sus siglas en inglés (Covering Arrays), son objetos combinatorios derivados de los arreglos ortogonales, que pueden ser usados y aplicados para diversos fines, entre ellos, el diseño de experimentos y la automatización de las pruebas funcionales de software. En pruebas de software se usan para generar el menor número de casos de prueba para cubrir todos los conjuntos de interacciones entre los parámetros de cada unidad lógica de software, llámese función, método o procedimiento [15]. Estos arreglos tienen cardinalidad mínima (reducen al mínimo el número de casos de prueba) y cobertura máxima (garantizan cubrir todas las combinaciones entre los parámetros de entrada basado en la fuerza de la interacción que se desee cubrir).

Un arreglo de cobertura denotado por CA (N; k, v, t), es una matriz *M* de *N* filas y *k* columnas, donde *N* es el número de experimentos o pruebas, *k* el número de factores o parámetros (del método, función o unidad de procesamiento a evaluar), *v* es el número de símbolos (valores posibles) por cada parámetro, denominado alfabeto y *t* es el grado de interacción entre los parámetros, denominado fuerza. Cada sub matriz de tamaño $N \times t$ contiene cada tupla de símbolos de tamaño *t (t-tupla)* al menos una vez [16]. Un arreglo de cobertura es óptimo, si contiene el mínimo número posible de filas [17] y se conoce como *CAN*.

En la **Tabla 1**, se muestra un ejemplo de un CA (9;4,3,2). La fuerza de este arreglo de cobertura es 2 (*t=2*), con 4 parámetros, factores o columnas (*k=4*) y un alfabeto de 3 (*v=3*) valores que está definido por los símbolos (0,1,2) dentro de cada celda.

0	0	0	0
0	1	1	1
0	2	2	2
1	0	1	2
1	1	2	0
1	2	0	1
2	0	2	1
2	1	0	2
2	2	1	0

Tabla 1. Ejemplo de CA (9;2,4,3)

Un diseño experimental completo o exhaustivo de 4 factores (k=4) con 3 posibles valores en cada celda, alfabeto de 3 (v=3) implica que la fuerza es igual al número de factores *(t=k)* y además que se deben cubrir v^t experimentos. En el ejemplo anterior serían 3^4 = 81 experimentos. En este caso, si se relaja la interacción a 2 (*t=2*) se reduce el número de casos de prueba a nueve (9) que representa una reducción del 90% de los casos de prueba. Nótese que en cada par de columnas aparecen las combinaciones (0,0),(0,1),(0,2),(1,0),(1,1),(1,2),(2,0),(2,1),(2,2) al menos una vez [16], lo que significa que dicha tabla si contiene un CA de fuerza 2.

Para ilustrar como se usan los covering arrays aplicado al diseño de pruebas de software, se considera el ejemplo de un Sistema Web que se espera probar con los parámetros de la **Tabla 2**.

	Navegador	**Sistema Operativo**	**BDMS**	**Conexión**
0	IE	Windows 7	MySQL	ISDN
1	Chrome	Ubuntu	Oracle	ADSL
2	Mozilla	Red Hat	SQL Server	Cable

Tabla 2. Parámetros y valores de un Sistema Web

El ejemplo involucra 4 factores o parámetros (Navegador, Sistema Operativo, BDMS o base de datos y Conexión) cada uno con tres posibles valores. Si se realizará una prueba exhaustiva, que involucraría una fuerza de 4 (t=4), deberían cubrirse 3^4 posibilidades, en otras palabras, 81 casos de prueba. Sin embargo, si la interacción se lleva a fuerza 2 (t=2), el número de posibles combinaciones se reduce a 9 casos de prueba, garantizando un amplio cubrimiento o cobertura en la prueba [18].

La **Tabla 1** muestra el CA (9;4,3,2) que se usa en la prueba del sistema web. Para hacer el mapeo entre el sistema Web y el CA, cada posible valor de cada parámetro de la **Tabla 2**, es etiquetado por el número de fila, para este caso 0,1 y 2. La **Tabla 3** muestra específicamente los casos de prueba o experimentos a realizar. Obsérvese que cada uno de los nueve experimentos es análogo a cada una de las filas del covering array.

IE	Windows 7	MySQL	ISDN
IE	Ubuntu	Oracle	ADSL
IE	Red Hat	SQL Server	Cable
Chrome	Windows 7	Oracle	Cable
Chrome	Ubuntu	SQL Server	ISDN
Chrome	Red Hat	MySQL	ADSL
Mozilla	Windows 7	SQL Server	ADSL
Mozilla	Ubuntu	MySQL	Cable
Mozilla	Red Hat	Oracle	ISDN

Tabla 3. Casos de prueba para el sistema Web

La construcción de un covering array óptimo es un problema de alta complejidad computacional debido al gran espacio de búsqueda que tiene que ser explorado para poder encontrar una solución con el menor número posible de filas. Pero contar con un CA óptimo (CAN) implica la reducción significativa en el número de pruebas a realizar, costo y tiempo durante la ejecución de las pruebas en los diferentes proyectos de software que lo utilicen.

2.4 Técnicas de Inducción de Reglas

Las reglas de decisión se forman con el objetivo de poder describir un sistema en el cual pueden ocurrir diferentes eventos, son fácilmente comprensibles y aplicables. Las regularidades ocultas en los datos se expresan frecuentemente en términos de reglas. Generalmente las reglas son expresiones de la forma [19]:

si (atributo − 1, valor − 1) y (atributo − 2, valor − 2) y ⋯ y (atributo − n, valor − n) entonces (decisión, valor).

Los datos a partir de los cuales las reglas se inducen se presentan generalmente como una tabla en la cual los casos (o ejemplos) están etiquetados, a esta tabla se le conoce como tabla de entrenamiento o datos de entrenamiento. Cada fila de la tabla es un ejemplo y las columnas son variables de entrada o atributos y hay una columna especial que registra la clase de cada ejemplo. El valor de decisión o clase se asigna a cada caso por parte de un experto en un trabajo que se hace mayormente de forma manual, lo que lo hace costoso de acuerdo con el tamaño de los datos. Los atributos o variables de entradas son independientes y la decisión (clase) es la variable dependiente [19].

Entre las técnicas de inducción de reglas se pueden destacar los arboles de decisión C4.5 y CART y el algoritmo de aprendizaje de reglas proposicionales RIPPER (Repeated Incremental Pruning Produce Error Reduction o JRip en Weka) que se describen a continuación:

El algoritmo C4.5 genera un árbol de decisión a partir de los datos de entrenamiento mediante particiones recursivas de estos, utilizando una métrica heurística conocida como ganancia de información, radio de ganancia (gain ratio) u otra para evaluar la calidad del criterio usado para la partición. El algoritmo considera todas las pruebas posibles que pueden dividir (partir) el conjunto de datos de entrenamiento y selecciona la prueba que le haya generado la mayor ganancia de información [20, 21]. Algunas características del algoritmo son las siguientes [20, 21]:

- Permite trabajar con valores continuos para los atributos, separando los posibles resultados en 2 ramas $A_i<=N$ y $A_i>N$ siendo N un umbral definido previamente.

- Los árboles son menos frondosos, ya que cada hoja cubre una distribución de clases no una clase en particular.

- Mediante "divide y vencerás" genera el árbol de decisión a partir del conjunto de datos de entrenamiento.

- El criterio de radio de ganancia (gain ratio), le permite evitar que las variables con mayor número de posibles valores salgan beneficiadas en la selección. Aunque se puede ejecutar con otros criterios como la ganancia de información.

- Es recursivo.

Del mismo modo que el algoritmo C4.5, el algoritmo CART entrega como resultado un árbol de decisión, pero a diferencia de C4.5 el árbol es binario, es decir, se tienen dos ramas por cada decisión [20, 22].

A diferencia de los dos anteriores, el algoritmo RIPPER no genera arboles de decisión sino reglas de decisión o reglas proposicionales. Se basa en reglas de asociación con reducción de error mediante poda (REP), una técnica muy común y efectiva que se encuentra en los algoritmos de árboles de decisión. En REP, los datos de entrenamiento se dividen en un conjunto creciente y un conjunto de poda. En primer lugar, se forma un conjunto de reglas inicial que supera el conjunto de crecimiento usando alguna métrica heurística. Este conjunto de reglas se simplifica repetidamente aplicando uno de un conjunto de operadores de poda tales como eliminar cualquier condición única o cualquier regla única. En cada etapa de simplificación, el operador de poda elegido es el que produce la mayor reducción de error en el conjunto de poda. La simplificación termina cuando se aplica cualquier operador de poda y aumenta el error en el conjunto de poda [20].

2.5 Bagging

El bagging busca aprovechar el conocimiento de varios expertos de un problema para la toma de decisiones. Lo anterior implica que se pregunta a varios expertos la clasificación de un nuevo registro y luego se combinan las respuestas para presentar una sola clasificación/predicción del nuevo registro. En bagging se eligen al azar varios conjuntos de datos de entrenamiento del mismo tamaño del dominio del problema provocando una disminución de la varianza y evitando el sobreajuste. Se utiliza técnicas de aprendizaje automático para construir un modelo para cada conjunto de datos. Ligeros cambios en los datos de entrenamiento pueden dar lugar fácilmente a producir predicciones correctas o incorrectas [23].

El Bagging (ver **Figura 1**) se puede resumir en los siguientes pasos [24]:

1. Dado un conjunto de entrenamiento, ***D***, de tamaño ***n***, bagging genera ***m*** nuevos conjuntos de entrenamiento, D_i, de tamaño ***n'***, **tomando al azar** elementos de ***D*** de manera **uniforme** y con **reemplazo**, por tanto, algunos elementos del conjunto original pueden aparecer repetidos en los nuevos conjuntos generados.

2. Si ***n'*** = ***n***, entonces para valores de ***n*** suficientemente grandes, se espera que cada D_i tenga una fracción de (1 - 1/e) (≈63.2%) elementos únicos de ***D***, y el resto son duplicados.

3. A partir de estos ***m*** nuevos conjuntos de entrenamiento se construyen ***m*** modelos de aprendizaje, y la respuesta final de la combinación se consigue por medio de votación de las ***m*** respuestas (en el caso de problemas de clasificación) o por la media de ellas (en el caso de problemas de regresión).

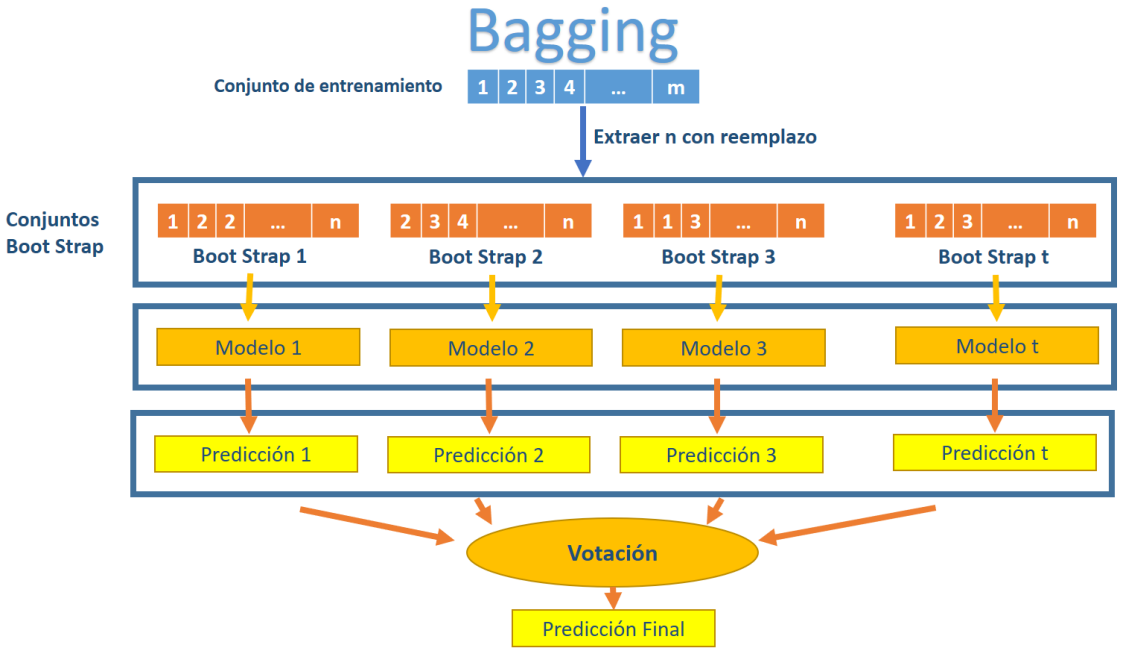


Figura 1. Descripción visual de bagging (Adaptado de [24])

2.6 Boosting

Boosting también es un esquema que busca aprovechar el conocimiento de varios expertos, pero a diferencia del bagging, no crea varias versiones aleatorias del conjunto de entrenamiento, sino que se trabaja siempre con el conjunto completo de entrada, y se manipulan los pesos de los datos para generar distintos modelos. La idea es que en cada iteración (cuando se crea un nuevo modelo) se incremente el peso de los objetos mal clasificados por el modelo en la iteración actual, por lo que en la construcción del próximo predictor o modelo estos objetos serán más importantes y será más probable clasificarlos bien (ver **Figura 2**) [23, 24].

Una de las muchas variantes de la idea de boosting se conoce como AdaBoost que consta de los siguientes pasos [24]:

1. Inicialmente a todos los datos del conjunto de entrenamiento se les asigna un peso idéntico, $w_i = 1/n$, donde n es el tamaño del conjunto de datos.

2. Se entrena el primer modelo usando el conjunto de entrenamiento.

3. Se calcula el error del modelo en el conjunto de entrenamiento, se cuentan cuántos objetos han sido mal clasificados y se identifican cuáles son.

4. Se incrementan los pesos en aquellos casos de entrenamiento en los que el modelo anterior ha dado resultados erróneos.

5. Se entrena un nuevo modelo usando el conjunto de pesos modificados.

6. Se repiten los pasos 3, 4 y 5 hasta el número de iteraciones (modelos) previamente fijados.

7. El modelo final se consigue por votación ponderada usando los pesos de todos los modelos.

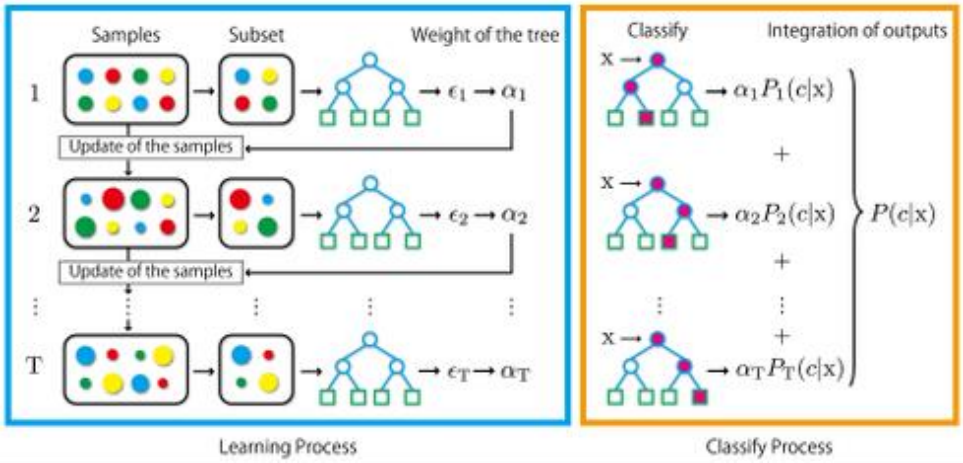


Figura 2. Descripción visual de boosting (Tomado de [24])

Capítulo 3

3 Estado del Arte

En esta sección se muestran las características que actualmente se han tenido en cuenta para realizar el proceso de selección de características y los trabajos más recientes y relevantes realizados en detección de polaridad en Twitter.

3.1 Selección de características en análisis de sentimientos en Twitter

Las características seleccionadas y su combinación juegan un papel importante para detectar el sentimiento de un texto. En Twitter, se puede identificar cuatro clases diferentes de características textuales: semánticas, sintácticas, estilísticas y características específicas de Twitter. Las características semánticas, sintácticas y estilísticas incluyen características bien conocidas y se han utilizado en la literatura existente de análisis de sentimientos de otros géneros tales como revisiones, blogs y foros [10]. Las características semánticas incluyen los términos que revelan el sentimiento negativo o positivo tomado generalmente de los léxicos de sentimientos o de conceptos semánticos de términos. Las características sintácticas que se aplican con frecuencia incluyen n-gramas y etiquetas de partes del discurso. Las características estilísticas se refieren al estilo de escritura utilizado en Twitter, mientras que la última clase incluye características que surgieron de las características únicas de los tweets como retweets o hashtags [1, 7]. A continuación, se muestra las características más comunes que se han utilizado para el análisis de sentimientos en Twitter [7]:

- Características semánticas: Las características semánticas más utilizadas son palabras de opinión, palabras de sentimiento, conceptos semánticos y negación. Las palabras de opinión se refieren a palabras o frases que se caracterizan como indicativas de opinión, mientras que las palabras de sentimiento son indicativas de sentimientos positivos o negativos. Las palabras y frases de opinión y sentimiento son de las características más utilizadas en análisis de sentimientos y pueden extraerse manual o semiautomáticamente de los léxicos de opinión y sentimiento, respectivamente. Algunos trabajos enfocados en análisis de sentimientos en Twitter utilizando características semánticas son los descritos en [25-30].

- Características sintácticas: incluyen los unigramas, bigramas, n-gramas, frecuencias de términos, partes-de-discurso, árboles de dependencia, y resolución de correferencia. Con el propósito de explorar el impacto de diferentes términos en el análisis del sentimiento, una serie de estudios asignan una puntuación de ponderación binaria (presencia/ausencia) a los términos, mientras

que otros usan un esquema de ponderación más avanzado considerando la frecuencia de los términos. En la literatura, un clasificador entrenado sólo con unigramas se utiliza con frecuencia como una línea base para comparación. Además de los términos, las etiquetas de partes-de-discurso (POS) (por ejemplo, sustantivos, verbos y adjetivos) son otra característica sintáctica que puede capturar información que indica opinión como en el caso de adjetivos relacionados con la opinión [31-34]. La resolución de la correferencia que ocurre cuando dos o más expresiones se refieren a la misma persona o cosa es una característica sintáctica adicional que se ha examinado para AST [35].

- Características estilísticas: Estas incluyen características que emergen del estilo de escritura no estándar que se utiliza en Twitter. Algunos ejemplos son emoticones, intensificadores (que se utilizan para aumentar el énfasis de lo que está escrito y que incluyen caracteres repetidos, alargamiento enfático y mayúsculas enfáticas), abreviaturas, términos de jerga y signos de puntuación (por ejemplo, signos de exclamación). Una característica importante es la presencia de emoticones, cuya utilidad ha sido ampliamente examinada en la literatura [34].

- Características específicas de Twitter: Teniendo en cuenta la presencia, ausencia o frecuencia en un tweet, se consideran los hashtags, retweets, respuestas, menciones, nombres de usuario, seguidores y URLs [33, 36].

La selección de características no es un proceso simple, se necesita un análisis exhaustivo para detectar las características más útiles para cada dominio. En [34], se propuso un modelo basado en características y se realizó un conjunto de experimentos para examinar la utilidad de varias características, incluyendo características POS y léxicas, concluyendo que la combinación más útil son las características POS y la polaridad de las palabras. Del mismo modo en [32] el estudio del impacto de diferentes características se centró principalmente en características semánticas y estilísticas, incluyendo emoticones, abreviaturas y la presencia de intensificadores, este estudio concluye que la combinación de la polaridad de los términos y los n-gramas logran el mejor desempeño.

El proceso de selección de características más típico consiste en aislar palabras y otras características tales como negaciones, emoticones, hastags e intensificadores y aplicar diferentes técnicas con el fin de identificar las más informativas. Para el uso de este enfoque se debe tener una estrategia para manipular la negación o detectar el sarcasmo, y tener en cuenta que si el número de características es muy grande, encontrar la mejor combinación de características no siempre es factible [7].

3.2 Detección de Polaridad en Twitter

Hay muchos trabajos relacionados con la detección de polaridad o análisis de sentimientos en Twitter, los cuales hacen uso de diversos métodos o técnicas para seleccionar las características más importantes que se deben tener en cuenta en la

clasificación, a continuación, se describen algunos de los más recientes y relevantes:

En 2015 [37], se desarrolló un método combinado de dos clasificadores de máxima entropía para el análisis de sentimientos en Twitter: uno para la subjetividad y el otro para la clasificación de la polaridad. Este método emplea formas superficiales[1], semántica y características de sentimiento. La complejidad de la clasificación de esta combinación de modelos lineales es lineal con respecto al número de características. El objetivo principal de este trabajo fue seleccionar un subconjunto compacto de características del total de características con el fin de reducir la complejidad computacional sin reducir la precisión de la clasificación. Las características seleccionadas mostraron una ganancia de rendimiento en la clasificación sobre la línea base de las características en los datasets de prueba CrowdScale y SemEval.

En 2016 [38], se abordó la categorización de tweets integrando dos aspectos fundamentales de un tweet, el contenido textual propio y su información estructural subyacente. Así, no sólo se analiza el contenido textual de los tweets, sino también la información estructural proporcionada por la relación entre tweets y usuarios, y se propone diferentes métodos para combinar de manera efectiva ambos tipos de modelos de características extraídos de las diferentes fuentes de conocimiento. Los resultados evidenciaron que este enfoque de integración de conocimiento es notablemente mejor que el modelo clásico basado en texto. Para este trabajo se definió un paso automático de selección de características para procesar el modelo de bolsa de palabras, utilizando un bosque de árboles muy aleatorizados con un alto número de estimadores. Este tipo de bosque es similar a un típico Random Forest, pero difiere en la manera de cómo los umbrales se han fijado para cada subconjunto al azar, los umbrales son fijados aleatoriamente para cada característica candidata y los mejores se seleccionan en lugar de buscar los más discriminantes. Esto tiende a reducir aún más la varianza del modelo interno.

También en 2016 [39], se detectó la polaridad en tweets usando los modelos de regresión de contracción (*shrinkage*) de Lasso y Ridge (que realiza la selección de características y la clasificación al mismo tiempo. Además, es capaz de estimar las puntuaciones del sentimiento de los términos en los tweets en comparación con otros algoritmos de selección o clasificación de características) proporcionando puntuaciones de sentimiento para los términos que aparecen en los tweets. A continuación, se identifica los temas principales a través de un modelo de análisis latente de Dirichlet (LDA) y estima el grado de polaridad en los temas usando puntuaciones de sentimientos de los términos. Los resultados indican que la

[1]Las formas superficiales de las palabras son las que se encuentran en cualquier texto. La forma léxica correspondiente de una forma superficial es el lema seguido por la información gramatical (por ejemplo, la parte del discurso, el género y el número). En inglés ‘give’, ‘gives’, ‘giving’, ‘gave’ y ‘given’ son formas superficiales del verbo ‘give’. La forma léxica sería el verbo " give".

exactitud del modelo Ridge es aproximadamente un 7% más alto que el de los modelos SVM.

En 2017 [40], se analizó el impacto de las emociones en el análisis de la polaridad en textos con lenguaje figurado en Twitter mediante un estudio exhaustivo sobre la capacidad de distintos recursos léxicos de emociones, detallando el impacto de cada uno de los recursos sobre distintas formas de lenguaje figurado como pueden ser la ironía y el sarcasmo. Los resultados obtenidos muestran indicios que apuntan a que la inclusión de información relativa a las emociones ayuda a clasificar correctamente la polaridad tanto a nivel global como a nivel del lenguaje figurado o literal.

También en 2017 [41], se desarrolló un enfoque híbrido basado en un léxico de sentimientos, reglas semánticas, manejo de negaciones, gestión de ambigüedades y variables lingüísticas. El método se aplica a dos conjuntos de datos y los resultados alcanzados obtienen puntuaciones de precisión y exactitud más altas que demuestran superioridad.

En 2018 [42], se propuso un enfoque novedoso para predecir los sentimientos expresados por los emojis en los tweets. Para este propósito, se construyó automáticamente un nuevo léxico de sentimiento de emoji usando un sistema de análisis de sentimiento no supervisado basado en las definiciones dadas por los creadores de emoji en Emojipedia[2]. Se evaluó y comparó al incluirlo en el análisis del sentimiento y los resultados confirman la competitividad del enfoque.

Es preciso comentar que SemEval (Semantic Evaluation, disponible en http://alt.qcri.org/semeval2016) es una serie continua de evaluaciones de sistemas computacionales de análisis semántico para investigar interrelaciones entre elementos de una oración (por ejemplo, el etiquetado del rol semántico), las relaciones entre oraciones (por ejemplo, la co-referencia) y la naturaleza de lo que se dice (relaciones semánticas y análisis de sentimientos). En análisis de sentimientos en Twitter se ha llevado a cabo en los últimos años una subtarea que consiste en clasificación de polaridad de mensajes [43, 44] (Dado un Tweet, predecir si el tweet es de sentimiento positivo, negativo o neutral). De las ultimas 3 evaluaciones (2015, 2016 y 2017) existe una gran cantidad de trabajos participantes [45-90], de los cuales se seleccionan los siguientes como los más relevantes:

En 2015 [54], se reprodujeron cuatro enfoques de clasificación de sentimientos de Twitter que participaron en ediciones anteriores de SemEval con diversos conjuntos de características. Los enfoques reproducidos se combinan en un conjunto, promediando las puntuaciones de confianza de los clasificadores individuales para las tres clases (positivo, neutral, negativo) y la polaridad del sentimiento de decisión basada en estos promedios. El conjunto de los enfoques reproducidos ocupa el primer lugar en el conjunto de pruebas de 2015.

[2] https://emojipedia.org/

En 2016 [61], se describió un sistema basado en *AlchemyAPI* y *SentiWordNet* para crear 43 características en las cuales se seleccionó un subconjunto de características como representación final. Los autores usan *Active Learning* para filtrar los tweets ruidosos del conjunto de entrenamiento proporcionado, dejando un conjunto más pequeño de sólo 900 tweets que usan para entrenar un clasificador *Multinomial Naive Bayes* y predecir las etiquetas del conjunto de pruebas.

También en 2016 [91], se propuso un clasificador basado en el enfoque de inclusión de oraciones convolucionales. Se aprovechó la disponibilidad de grandes cantidades de datos previamente clasificados para entrenar un conjunto de redes neuronales convolucionales de 2 capas cuyas predicciones se combinan usando un clasificador Random Forest. Este enfoque fue evaluado en los conjuntos de datos de la competencia SemEval-2016 superando a todos los otros enfoques para la tarea de clasificación de polaridad de mensaje.

En 2017 [92], se presentó un sistema de aprendizaje profundo que compitió en SemEval-2017 usando redes del tipo LSTM (Long Short-Term Memory - memoria a corto y largo plazo) aumentadas con dos tipos de mecanismos de atención, además de incrustaciones de palabras (word embeddings) pre entrenadas en una gran colección de mensajes de Twitter. Adicionalmente, se presentó una herramienta de procesamiento de texto adecuada para mensajes de redes sociales, que realiza tokenización, normalización de palabras, segmentación y corrección ortográfica. Además, no utiliza características hechas a mano o léxicos de sentimiento, este sistema ocupo el primer lugar de la competencia y logró un puntaje f1 de 0.677.

También en 2017 [93], se propuso un clasificador de sentimiento de Twitter de última generación que utiliza redes neuronales convolucionales (CNN) y memorias de corto a largo plazo (LSTM). Aprovecha una gran cantidad de datos no etiquetados para pre-entrenar incrustaciones de palabras. A continuación, utiliza un subconjunto de los datos sin etiquetar para ajustar las incrustaciones utilizando supervisión a distancia. Los CNN y LSTM finales reciben entrenamiento sobre el conjunto de datos de Twitter SemEval-2017, donde las incrustaciones se vuelven a ajustar. Para aumentar en rendimiento, se combinan varias CNN y LSTM, el sistema ocupo el segundo lugar y un puntaje f1 de 0.685.

Finalmente en 2017 [94], se presentó un sistema con un conjunto de modelos de Red Neuronal Profunda (DNN): Red Neural Convolucional (CNN) y Red Neuronal Recurrente de Memoria de Corto a Largo Plazo (RNN-LSTM). Se inicializa la representación de entrada de DNN con diferentes conjuntos de incrustaciones formadas en grandes conjuntos de datos. El conjunto de DNN se combinan utilizando un enfoque de fusión de nivel de puntaje, el sistema ocupa el tercer lugar en Semeval 2017 y un puntaje f1 de 0.674.

También se encontraron algunos trabajos donde se evalúan diferentes enfoques de selección de características para análisis de sentimientos, como en [95-97] (trabajos presentados en 2016), donde se comparan varios métodos buscando identificar el

mejor enfoque posible. O en [98, 99] (de 2015 y 2016), donde se discuten los efectos del preprocesamiento sobre el rendimiento de la clasificación de sentimientos y se evalúa los efectos de las URLs, palabras vacías, letras repetidas, la negación y los acrónimos.

Capítulo 4

4 Framework de prueba

En este capítulo, se inicia describiendo los datasets de prueba y los resultados obtenidos en 3 competencias de Semeval (2013, 2016 y 2017), así como los resultados de los sistemas para detección de polaridad en análisis de sentimientos que compitieron en dichos eventos. También se describe el dataset Sentiment140 [31] que se usó en la fase de experimentación de la presente tesis. Igualmente se presentan algunos algoritmos relevantes para la detección de polaridad en AST encontrados en la revisión del estado del arte y que fue posible encontrar su implementación y evaluar su rendimiento con los datasets obtenidos, lo cual permitió definir el framework de prueba y cumplir así con el primer objetivo específico.

4.1 Semeval

SemEval (Evaluación Semántica) es una serie continua de evaluaciones de sistemas de análisis semántico computacional. Este trabajo comenzó con intentos aparentemente simples de identificar los sentidos de las palabras y ha evolucionado para investigar las interrelaciones entre los elementos de una oración (por ejemplo, etiquetado de roles semánticos), las relaciones entre oraciones (por ejemplo, correferencia) y la naturaleza de lo que se está diciendo (relaciones semánticas y análisis de sentimientos).

En el presente trabajo, se ha tenido en cuenta los resultados de las evaluaciones correspondientes a los años 2013, 2016 y 2017 de los cuales se logró obtener la mayoría de los tweets de sus datasets (debido a que los datasets constan de los identificadores de los tweets y estos se ven afectados por las restricciones de privacidad de Twitter que cambian con el tiempo, hecho que no permitió obtener todos los tweets). A continuación, se muestra los resultados de las competencias de los años 2013, 2016 y 2017 respectivamente.

4.1.1 Semeval 2013 [100]

Para el año 2013 las estadísticas sobre el corpus creado para la subtarea de “Clasificación de polaridad del mensaje” (Dado un mensaje, decida si es positivo, negativo o neutral. Para los mensajes que transmiten tanto un sentimiento positivo como uno negativo, cualquiera que sea el más fuerte debe ser elegido), se presentan en la **Tabla 4**.

Los resultados para la subtarea se muestran en la **Tabla 5**. Los sistemas se clasifican por sus puntajes en F1 o medida F (resultado del promedio armónico de la precisión y el recuerdo).

Corpus	Positive	Negative	Objective/ Neutral
Twitter - Training	3,662	1,466	4,600
Twitter - Dev	575	340	739
Twitter - Test	1,573	601	1,640

Tabla 4. Estadísticas del dataset para “Clasificación de polaridad del mensaje” en 2013

Sistema	Puntaje F1
NRC-Canadá	69.02
GU-MLT-LT	65.27
Teragram	64.86
BOUNCE	63.53
KLUE	63.06
AMI&ERIC	62.55
FBM	61.17
AVAYA	60.84
SAIL	60.14
UT-DB	59.87
FBK-irst	59.76
nlp.cs.aueb.gr	58.91
UNITOR	58.27
LVIC-LIMSI	57.14
Umigon	56.96
NILC USP	56.31
DataMining	55.52
ECNUCS	55.05
nlp.cs.aueb.gr	54.73

Sistema	Puntaje F1
ASVUniOfLeipzig	54.56
SZTE-NLP	54.33
CodeX	53.89
Oasis	53.84
NTNU	53.23
UoM	51.81
SSA-UO	50.17
SenselyticTeam	50.10
UMCC DLSI	49.27
bwbaugh	48.83
senti.ue-en	47.24
SU-sentilab	45.75
OPTWIMA	45.40
REACTION	45.01
uottawa	42.51
IITB	39.80
IIRG	34.44
sinai	16.28

Tabla 5. Resultados para “Clasificación de polaridad del mensaje” en 2013

4.1.2 Semeval 2016 [43]

Para el año 2016 en la subtarea A “Dado un tweet, predecir si es de sentimiento positivo, negativo o neutral”, las estadísticas del dataset son las siguientes:

Corpus	Positive	Negative	Neutral	Total
TRAIN	3,094	2,043	863	6,000
DEV	844	391	765	2,000
DEVTEST	994	325	681	2,000
TEST	7,059	3,231	10,342	20,632

Tabla 6. Estadísticas del dataset para“Clasificación de polaridad” en 2016

Los resultados para la subtarea de muestran en la **Tabla 7**. Los sistemas están ordenados por su puntuación F1.

#	System	F1
1	SwissCheese	0.633
2	SENSEI-LIF	0.630
3	UNIMELB	0.617
4	INESC-ID	0.610
5	aueb.twitter.sentiment	0.605
6	SentiSys	0.598
7	I2RNTU	0.596
8	INSIGHT-1	0.593
9	TwiSE	0.586
10	ECNU	0.585
11	NTNUSentEval	0.583
12	MDSENT	0.580
12	CUFE	0.580
14	THUIR	0.576
14	PUT	0.576
16	LYS	0.575
17	IIP	0.574

#	System	F1
18	UniPI	0.571
19	DIEGOLab16	0.554
20	GTI	0.539
21	OPAL	0.505
22	DSIC-ELIRF	0.502
23	UofL	0.499
23	ELiRF	0.499
25	ISTI-CNR	0.494
26	SteM	0.478
27	Tweester	0.455
28	Minions	0.415
29	Aicyber	0.402
30	mib	0.401
31	VCU-TSA	0.372
32	SentimentalITists	0.339
33	WR	0.330
34	CICBUAPnlp	0.303

Tabla 7. Resultados para "Clasificación de polaridad del mensaje" en 2016

#	System	AvgRec	F1		#	System	AvgRec	F1
1	DataStories	0.681	0.677		20	NILC-USP	0.612	0.595
1	BB twtr	0.681	0.685		21	Ti-Senti	0.607	0.577
3	LIA	0.676	0.674		22	BUSEM	0.605	0.587
4	Senti17	0.674	0.665		23	EICA	0.595	0.555
5	NNEMBs	0.669	0.658		24	OMAM	0.590	0.542
6	Tweester	0.659	0.648		25	Adullam	0.589	0.552
7	INGEOTEC	0.649	0.645		26	NileTMRG	0.578	0.515
8	SiTAKA	0.645	0.628		27	Amobee-C-137	0.575	0.520
9	TSA-INF	0.643	0.620		28	ej-za-2017	0.571	0.539
10	UCSC-NLP	0.642	0.624		28	LSIS	0.571	0.561
11	HLP@UPENN	0.637	0.632		30	XJSA	0.556	0.519
12	YNU-HPCC	0.633	0.612		31	Neverland-THU	0.555	0.507
12	SentiME++	0.633	0.613		32	MI&T-Lab	0.551	0.522
14	ELiRF-UPV	0.632	0.619		33	diegoref	0.546	0.527
15	ECNU	0.628	0.613		34	xiwu	0.479	0.365
16	TakeLab	0.627	0.607		35	SSN MLRG1	0.431	0.344
17	DUTH	0.621	0.605		36	YNUDLG	0.340	0.201
18	CrystalNest	0.619	0.593		37	WarwickDCS	0.335	0.221
19	deepSA	0.618	0.587		37	Avid	0.335	0.163

Tabla 8. Resultados para " Clasificación de polaridad" en 2017 (inglés)

4.1.3 Semeval 2017 [101]

Para el año 2017 se incluyó un nuevo idioma para la competencia, el árabe, debido a esto se mantuvieron los datasets de las competencias anteriores y se agregó nuevos datasets en árabe, en este trabajo solo se tienen en cuenta los resultados de la competencia en inglés. Los resultados para la subtarea se muestran en la **Tabla 8**. Los sistemas se clasifican por sus puntajes en “average recall”.

4.1.4 Sentiment140 dataset [31]

Este conjunto de datos. Contiene 1,600,000 tweets extraídos usando la API de Twitter. Los tweets se han anotado (0 = negativo, 2 = neutro, 4 = positivo) y se pueden usar para detectar el sentimiento. Contiene los siguientes 6 campos:

- Objetivo: la polaridad del tweet (0 = negativo, 2 = neutral, 4 = positivo)
- Ids: la identificación del tweet
- Fecha: la fecha del tweet
- Flag: La consulta. Si no hay consulta, este valor es NO_QUERY.
- Usuario: el usuario que tuiteó.
- Texto: el texto del tweet.

De acuerdo con los creadores del conjunto de datos: "Nuestro enfoque fue único porque nuestros datos de entrenamiento se crearon automáticamente, en lugar de tener anotaciones manuales de seres humanos. En nuestro enfoque, suponemos que cualquier tweet con emoticones positivos, como ‘:)’, fueron positivos, y tweets con emoticones negativos, como ‘:(‘, fueron negativos. Utilizamos la API de búsqueda de Twitter para recopilar estos tweets mediante el uso de la búsqueda de palabras clave". Al contrario del dataset de entrenamiento, el dataset de prueba no se creó automáticamente, las anotaciones se realizaron manualmente por seres humanos y su tamaño es de 500 tweets.

4.2 Algoritmos implementados

En internet se encontraron varios algoritmos (código fuente) que participaron en las competencias de Semeval, pero no se encontró el código completo y replicar el algoritmo con lo presentado en los artículos y el código disponible no fue posible realizar por falta de información, información que los autores de los artículos no suministraron a pesar de las diversas solicitudes realizadas por el autor del presente trabajo. Los algoritmos que se implementaron gracias a la información y código de sus autores fueron los siguientes:

4.2.1 Stem

Stem [61] se basa en AlchemyAPI[3] y SentiWordNet para crear 43 características de las cuales se selecciona un subconjunto de características como representación

[3] https://www.ibm.com/watson/alchemy-api.html

final. Además, utiliza aprendizaje activo para filtrar los tweets ruidosos del conjunto de entrenamiento, dejando un conjunto más pequeño de solo 900 tweets que se utiliza para entrenar un clasificador Multinomial Naive Bayes y predecir las etiquetas del conjunto de pruebas con un puntaje F1 de 0,478 en Semeval del año 2016.

El experimento se replicó gracias al repositorio compartido y la asesoría de su autor[4], a diferencia de su experimento no se obtuvo un conjunto más pequeño de instancias mediante aprendizaje activo, sino que se entrenó con todo el conjunto de entrenamiento de 2013 y de 2016. La **Tabla 9** muestra los resultados de stem entrenado con el dataset SemEval 2013-Train+dev y evaluado con **SemEval 2013 Test,** los dos datasets están descritos más adelante en la **Tabla 18**.

	Negativo	**Neutral**	**Positivo**
Negativo	204	276	121
Neutral	111	1255	274
Positivo	94	488	990
F1: 63,47 %			

Tabla 9. Resultados de stem con datasets de Semeval 2013

La **Tabla 10** muestra los resultados de stem entrenado con el dataset SemEval 2016 Train+dev y evaluado con **SemEval 2016 Test,** los dos datasets descritos en la **Tabla 18**.

	Negativo	**Neutral**	**Positivo**
Negativo	74	100	114
Neutral	92	238	300
Positivo	52	188	656
F1: 51,9 %			

Tabla 10. Resultados de stem con datasets de Semeval 2016

Como se evidencia, en la **Tabla 9** y **Tabla 10**, los resultados difieren a los obtenidos en la competencia (puntaje F1 de 0,478), lo que se debe principalmente a:

- El uso de distintos grupos de entrenamiento y pruebas.
- La falta de la implementación de la etapa de aprendizaje activo.
- Las características aportadas por AlchemyAPI no se pudieron incluir, ya que esta API ya no es de uso libre y su alto costo no permitió su uso en la investigación.

Los resultados que obtuvo Stem se lograron con base en las siguientes 43 características:

[4] Stefan Räbiger <stefan@sabanciuniv.edu>

	Característica	Descripción
1	Pos_emo	Número de emoticones positivos
2	Neg_emo	Número de emoticones negativos
3	Elongated	Número de palabras alargadas
4	Upper	Número de palabras MAYÚSCULAS
5	has_ht	¿El tweet contiene al menos un hashtag?
6	neg_ht	hashtags con sentimiento negativo
7	neu_ht	hashtags con sentimiento neutral
8	pos_ht	hashtags con sentimiento positivo
9	neg_words_ht	Número de palabras negativas en hashtag
10	neu_words_ht	Número de palabras neutrales en hashtag
11	pos_words_ht	Número de palabras positivas en hashtag
12	pol_words_ht	Número de palabras de polaridad en hashtag
13	negat_words_ht	Número de palabras de negación en hashtag
14	neg_words_ht_sum	Suma del sentimiento negativo en hashtag
15	neu_words_ht_sum	Suma del sentimiento neutro en hashtag
16	pos_words_ht_sum	Suma de sentimiento positivo en hashtag
17	punct	Número de apariciones de '!', '??', '!?', '?!'
18	start_len	Número de palabras antes del preprocesamiento
19	end_len	Número de palabras después del preprocesamiento
20	avg_len	Número promedio de palabras por oración
21	adj_frac	Porcentaje de adjetivos
22	adv_frac	Porcentaje de adverbios
23	v_frac	Porcentaje de verbos
24	nn_frac	Porcentaje de sustantivos
25	al_t_pol	Polaridad del tweet usando AlchemyAPI
26	al_t_type	Tipo de polaridad del tweet usando AlchemyAPI
27	al_e_pol	Media de la polaridad de la entidad usando AlchemyAPI
28	al_e_type	Mediana de la polaridad de la entidad usando AlchemyAPI
29	al_neg_e	Número de entidades con nombre negativo usando AlchemyAPI
30	al_neu_e	Número de entidades con nombre neutral usando AlchemyAPI
31	al_pos_e	Número de entidades con nombre positivo usando AlchemyAPI
32	al_mixed	¿El tweet contiene sentimientos mixtos? según AlchemyAPI
33	neg	Sentimiento negativo usando SentiWordNet
34	neu	Sentimiento neutral usando SentiWordNet

35	Pos	Sentimiento positivo usando SentiWordNet
36	neg_words	Número de palabras negativas usando SentiWordNet
37	neu_words	Número de palabras neutras usando SentiWordNet
38	pos_words	Número de palabras positivas usando SentiWordNet
39	negat_words	Número de palabras de negación usando SentiWordNet
40	pol_words	Número de palabras de polaridad usando SentiWordNet
41	neg_words_sum	Suma de sentimientos negativos usando SentiWordNet
42	neu_words_sum	Suma de sentimiento neutral usando SentiWordNet
43	pos_words_sum	Suma de sentimientos positivos usando SentiWordNet

Tabla 11. Características utilizadas en stem (Adaptado de [61])

Los resultados mostrados en la **Tabla 9** y **Tabla 10** sirven para comparar el método propuesto de selección de características, evaluar la mejora en la clasificación al agregar más características que se consideran importantes y realizar con esto un proceso que permite identificar las características más relevantes y que permita mejorar la clasificación o hacer más simple el modelo manteniendo los niveles de clasificación.

A nivel de implementación, en la presente tesis se usaron la mayoría de las características (excepto las basadas en AlchemyAPI) presentadas por el autor de Stem.

4.2.2 Hlp@upenn

Hlp@upenn [102] es un sistema supervisado de clasificación de texto que combina representaciones vectoriales dispersas y densas de palabras, y las representaciones generalizadas de palabras a través de clústers (embeddings o incrustaciones). Los vectores dispersos se generan a partir de secuencias de n-gramas de palabras (1-3). Las representaciones de vectores densos de palabras (incrustaciones) se aprenden al entrenar una red neuronal para predecir las palabras vecinas en un conjunto de datos grande sin etiquetar. Para clasificar un segmento de texto, las diferentes representaciones de vectores se concatenan, y la clasificación se realiza utilizando máquinas de soporte vectorial (SVM). Dado un conjunto de entrenamiento, el sistema genera automáticamente los vectores de entrenamiento, optimiza los hiper-parámetros relevantes para el clasificador SVM y entrena el modelo de clasificación. El sistema se evaluó en la tarea de análisis de sentimiento en inglés de SemEval2017. En términos de puntaje promedio F1, el sistema obtuvo la 8ª posición de 39 con un puntaje promedio F1 de 0.632.

El experimento se replicó gracias al repositorio[5] y la documentación en línea del algoritmo compartida por su autor. El algoritmo se entrenó con todo el conjunto de entrenamiento de SemEval 2013 y de 2016, los resultados se presentan en la **Tabla**

[5] https://bitbucket.org/ pennhlp/hlp-upenn_semeval_2017_task4

12. Los resultados difieren de los publicados en el artículo debido a que el proceso se realizó con diferentes datasets. Los autores reportan el uso de un total de 49,484 tweets, de los cuales 19,597 (39.6%) fueron etiquetados como positivos, 7692 (15.5%) como negativos y 22,195 (44.9%) como neutros.

Dataset	F1 Score (%)
Semeval 2013	63,1
Semeval 2016	49,0

Tabla 12. Resultados de sistema Hlp@upenn

Un punto importante de este trabajo es que emplea distintas representaciones del tweet para realizar el proceso de entrenamiento y clasificación, mostrando que la representación multimodal permite cubrir aspectos que podrían faltar usando una sola representación.

4.3 Otros sistemas de interés

Algunos sistemas no se lograron implementar debido a que no se encontró el código fuente, se encontró solo parte de este, la explicación dada en el artículo no era lo suficientemente detallada o los datos de entrenamiento no estaban disponibles, entre otras razones, sin embargo, se consideran importantes algunos aspectos de estos sistemas que pueden ser usados en futuras investigaciones.

4.3.1 DataStories

DataStories [92] es un sistema que ocupo el primer lugar en la competencia Semeval 2017. En este trabajo se usó una red LSTM (memoria a corto y largo plazo) de 2 capas, equipada con un mecanismo de atención e incrustaciones de palabras pre-entrenadas en una gran colección de mensajes de Twitter.

En este trabajo se aprovechó una gran colección de mensajes de Twitter para generar incrustaciones de palabras, con un tamaño de vocabulario de 660K palabras, utilizando GloVe[6]. Las incrustaciones de palabras preentrenadas fueron utilizadas para inicializar la primera capa de la red neuronal (Capa de incrustaciones). De igual forma se desarrolló una herramienta para procesar los textos y utilizar la mayor parte de información del texto, esta herramienta contribuye a realizar tokenización sensible al sentimiento, corrección ortográfica, normalización de palabras, segmentación de palabras (para dividir hashtags) y anotación de palabras.

Las redes neuronales convolucionales (CNN) no tiene noción de orden, por lo tanto, al aplicarlas a las tareas de Procesamiento de Lenguaje Natural (PLN), se pierde la información crucial del orden de las palabras. Una opción más natural es utilizar redes neuronales recurrentes (RNN), pero estas redes son difíciles de entrenar, porque los gradientes pueden crecer o decaer exponencialmente en secuencias

[6] https://nlp.stanford.edu/projects/glove/

largas, una forma de superar estos problemas es mediante el uso de una de las variantes más sofisticadas de las RNN, la red de memoria a corto y a largo plazo (LSTM), o las recientemente propuestas unidades recurrentes cerradas (Gated Recurrent Units, GRU). Ambas variantes introducen un mecanismo de activación de puerta, lo que garantiza la propagación correcta del gradiente a través de la red. En esta propuesta se utilizó LSTM, porque se desempeñó ligeramente mejor que GRU.

Es importante mencionar que un RNN actualiza su estado oculto a medida que procesa una secuencia y, al final, el estado oculto contiene un resumen de toda la información procesada. Para ampliar la contribución de elementos importantes en la representación final, se utilizó un mecanismo de atención, que agrega todos los estados ocultos utilizando su importancia relativa. Se puede observar en la **Tabla 13** que el mecanismo de atención implementado aumenta los resultados e impacta positivamente en el sistema.

RNN	F1
Regular	0.673
Mecanismo de atención	**0.675**

Tabla 13. Resultados y el impacto del mecanismo de atención

4.3.2 BB twtr

BB twtr [93] ocupa el segundo lugar en Semeval 2017, utiliza Redes neuronales convolucionales (CNN) y redes de memoria a corto y largo plazo (LSTM), además, utiliza una gran cantidad de datos sin etiquetar para entrenar previamente las incrustaciones de palabras, luego se ajustan las incrustaciones utilizando supervisión a distancia con un subconjunto de los datos.

Para representar el tweet, a cada palabra se le asigna una representación de vector de palabra, es decir, una palabra incrustada, de manera que un tweet completo se puede mapear a una matriz de tamaño s × d, donde s es el número de palabras en el tweet y d es la dimensión del espacio de incrustación (ver **Figura 3**).

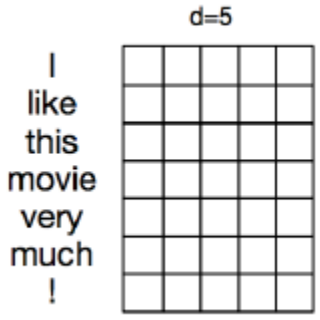


Figura 3. Representación del tweet utilizada en el sistema BB twtr (Tomado de [93])

El preprocesamiento llevado a cabo en cada uno de los tweets, realiza algunas tareas como reemplazar las URL por el token <url>, reemplazar los emoticones por los tokens: <smile>, <sadface>, <lolface> o <neutralface>. Además, cualquier letra repetida más de 2 veces, en una palabra, se sustituye por 2 repeticiones de esa letra (por ejemplo, "sooooo" se reemplaza por "soo") y finalmente todas las letras de los tweets se convierten en minúsculas.

Se encontraron algunos problemas como por ejemplo que la RNN simple sufre el problema de la explosión y desaparición del gradiente durante la etapa de propagación hacia atrás en el entrenamiento. Los LSTM resuelven este problema al tener una estructura interna más compleja que permite a los LSTM recordar información, ya sea a corto o largo plazo. De la misma manera, en CNN Para reducir el sobre ajuste, se agregó una capa dropout después de la capa max-pooling y después de la capa oculta, con una probabilidad de abandono del 50% durante el entrenamiento.

Las incrustaciones aprendidas en la fase no supervisada contienen muy poca información sobre la polaridad del sentimiento de las palabras, ya que el contexto para una palabra positiva (por ejemplo, "bueno") tiende a ser muy similar al contexto de una palabra negativa (por ejemplo, "malo"). Para agregar información de polaridad a las incrustaciones, se realizó un ajuste de las incrustaciones a través de una fase de entrenamiento distante. Para ello, se utilizó una CNN inicializando las incrustaciones con las aprendidas en la fase no supervisada. Luego se usó el conjunto de datos distantes para entrenar a la CNN para clasificar los tweets positivos frente a los tweets negativos. Después de esta etapa de entrenamiento, las palabras con una polaridad de sentimiento muy diferente (por ejemplo, "bueno" contra "malo") están muy separadas en el espacio de incrustación.

4.3.3 LIA

El sistema LIA [94] ocupó el tercer lugar en Semeval 2017. El sistema es un conjunto de modelos de red neuronal profunda (DNN): red neuronal convolucional (CNN) y memoria de corto plazo a largo plazo (RNN-LSTM) de red neuronal recurrente. Las incrustaciones de palabras preentrenadas se utilizan para inicializar las representaciones de palabras, que luego se toman como entrada. El enfoque propuesto consiste en entrenar clasificadores para cuatro tipos de incrustaciones, basadas en las arquitecturas CNN y RNN-LSTM. Cada conjunto de incrustaciones de palabras modela el tweet de acuerdo con un punto de vista diferente y se aplica un paso final de fusión.

Las incrustaciones de palabras son un enfoque para la semántica distributiva que representa palabras como vectores de números reales. Dicha representación tiene propiedades de agrupamiento útiles, ya que agrupa palabras que son semántica y sintácticamente similares. Por ejemplo, la palabra "café" y "té" estarán muy cerca del espacio creado. El objetivo es utilizar estas funciones como entrada para un clasificador de DNN. Sin embargo, con la tarea de análisis de sentimientos en mente, las incrustaciones típicas de palabras extraídas del contexto léxico pueden

no ser las más precisas porque los antónimos tienden a ubicarse en la misma ubicación en el espacio creado.

Por lo anteriormente mencionado, se exploraron cuatro diferentes enfoques para integrar la polaridad sentimental de las palabras, a saber:

- Incrustaciones léxicas: estas incrustaciones se obtienen con el modelo clásico de diagramas. La representación se crea utilizando la capa oculta de una red neuronal lineal para predecir una ventana de contexto desde una palabra central. Este método típicamente extrae una representación que cubre tanto la sintaxis como la semántica hasta cierto punto.

- Incrustaciones de sentimiento (aprendizaje multitarea): Uno de los problemas con el enfoque básico de diagramas (incrustaciones léxicas) es que el modelo ignora la polaridad de sentimiento de las palabras. Como resultado, las palabras con polaridad opuesta, como "bueno" y "malo", se asignan en vectores cercanos. En trabajos previos, se propone abordar este problema codificando la información del sentimiento en la representación continua de palabras, mediante una red neuronal que predice dos tareas: el contexto de la palabra y la etiqueta de sentimiento de la oración.

- Incrustaciones de sentimientos (supervisión a distancia): la supervisión a distancia es otra solución para integrar la polaridad de sentimientos en palabras. Un DNN (CNN o RNN-LSTM) se entrena en tweets supervisados masivos y distantes seleccionados por emoticones positivos y negativos. Los emoticones positivos y negativos se utilizan como etiquetas supervisadas. Durante el entrenamiento, la DNN refina automáticamente la palabra incrustada para capturar la polaridad del sentimiento. Las incrustaciones de palabras refinadas se pueden utilizar como una nueva representación.

- Incorporación de opiniones (muestreo negativo): el enfoque de muestreo negativo es una forma eficiente de calcular el softmax[7]. Para hacer frente a la dificultad de tener demasiados vectores de salida que necesitan actualizarse, la idea principal del muestreo negativo es actualizar no todas las palabras, sino solo algunas palabras como muestras negativas (por lo tanto, "muestreo negativo"). En lugar de seleccionar palabras al azar, como es habitual en esta técnica, se decidió seleccionar palabras con polaridades opuestas. Por ejemplo, para la palabra "bueno" se seleccionaron las palabras "malo", "terrible", etc. para el muestreo negativo.

[7] La función softmax, o función exponencial normalizada, es una generalización de la función logística. Se emplea para "comprimir" un vector K-dimensional, de valores reales arbitrarios en un vector K-dimensional, de valores reales en el rango [0, 1].

Se propuso agregar información a nivel de tweet e inyectar esta información en el modelo. Para incorporar esta fuente de información en el sistema, un conjunto de características de nivel de oración se concatena con la última capa oculta en el modelo.

Las siguientes características se extraen a nivel de oración:

- Léxico: frecuencia de los lemas que se combinan en MPQA[8], el Léxico de opinión de Bing Liu [103] y el léxico Emotion NRC [104].

- Emoticones: número de emoticones que se agrupan en categorías positivas, negativas y neutrales.

- Mayúsculas: número de palabras en mayúsculas.

- Unidades alargadas: número de palabras en las que los caracteres se repiten más de dos veces (por ejemplo: looooool).

- Puntuación: número de secuencias contiguas de varios períodos, signos de exclamación y signos de interrogación.

Se utilizó un enfoque de profesor estudiante (Mimic model), que consiste en capacitar a un modelo avanzado (modelo de maestro) y luego capacitar a un nuevo modelo (modelo de estudiante) para imitar el modelo de maestro. El modelo de mímica (modelo de estudiante) no está entrenado en las etiquetas originales, pero está entrenado para aprender los objetivos predichos por el modelo del maestro. Sorprendentemente, un modelo de imitación entrenado en objetivos predichos por el modelo del maestro puede ser más preciso que el modelo del maestro entrenado en las etiquetas originales.

Hay una variedad de razones por las que esto puede suceder:

- Si algunas etiquetas tienen errores, el modelo del maestro puede eliminar algunos de estos errores, lo que facilitará el aprendizaje para el alumno.

- El modelo mimético se puede ver como una forma de regularización que ayuda a evitar el ajuste excesivo del modelo.

Se observó que el mejor sistema es el modelo CNN Mimic que utiliza incrustación de sentimientos (muestreo negativo); la incorporación de sentimientos (muestreo negativo) obtiene para cada modelo de DNN los mejores resultados. Con respecto a los modelos DNN, el enfoque CNN proporciona mejores resultados que los modelos RNN-LSTM.

[8] https://mpqa.cs.pitt.edu/

Las salidas de todas las redes neuronales profundas se concatenan para formar un único vector de características. Este vector luego se alimenta a un 'Multi Layer Perceptron' (MLP) que se entrena para predecir la polaridad. El MLP contiene una capa oculta de 128 neuronas y la función de activación 'tanh'. Los resultados obtenidos por el sistema de fusión. Logró el tercer puesto en Semeval 2017.

4.3.4 Procesos importantes observados

Se observó que los trabajos anteriores cuyos resultados reportan los más altos puntajes en el último evento, tienen en común el uso de redes LSTM, esto puede ser un aspecto importante a futuro para lograr una mejor calidad en la clasificación.

Una representación bastante usada en estos trabajos son las incrustaciones (embedings) pre-entrenadas, este aspecto se incluyó como una característica en el presente trabajo, pero es preciso evaluar otras representaciones que tengan en cuenta la polaridad de las palabras en la construcción de las incrustaciones.

Asimismo, se puede notar que el preprocesamiento del tweet es un aspecto importante que se debe realizar muy cuidadosamente. Procesos como la tokenización especial para Twitter, la corrección de ortografía, entre otros, tienden a mejorar los subsecuentes procesos aplicado a los tweets.

Capítulo 5

5 El método propuesto

El método propuesto necesita primero contar con una serie de características que se usan para representar cada Tweet (vista minable), para ello se han definido 91 características de distintos aspectos o representaciones (representación multimodal). La **Tabla 14** presenta una descripción general de las 91 características inicialmente extraídas de los Tweets. Entre ellas se cuenta con características que, por ejemplo, exploran si los emoticones se correlacionan con el sentimiento, contando el número de emoticones y número de palabras de sentimiento positivo, neutro y negativo, para hacerlo, se utilizó un léxico que abarca 81 emoticones positivos y negativos comunes basados en Wikipedia [61].

Se examina también si los hashtags comparten una relación similar con el sentimiento general del Tweet, así como con los emoticones [105]. Los signos de exclamación también sugieren un sentimiento amplificado del tweet. La longitud de un tweet afecta su sentimiento: si los tweets son más largos, es más probable que contengan polaridad mixta y, por lo tanto, es más difícil etiquetarlos, entre otros.

Las 91 características se han agrupado en 8 grupos, cada característica pertenece a uno o varios grupos y estos grupos indican un tipo, que puede ser:

- Conteo de palabras: se cuenta el número de palabras o apariciones de la palabra que cumple con el criterio de la característica que se obtiene en el tweet.

- Hashtag: se asigna este grupo a las características que se evalúan sobre palabras que aparecen en los hashtags y que generalmente son usados por los usuarios de Twitter para agrupar los tweets sobre un tema en particular bajo una misma etiqueta.

- Emoticones: Se asigna este grupo a las características evaluadas sobre emoticones.

- Polaridad: Se asigna este grupo a las características que llevan un conteo de la polaridad de las palabras, indicando aquellas negativas, positivas y neutras entre otras.

- Etiquetado gramatical (POS tag): Se relaciona con las características de las cuales se lleva un conteo de adjetivos, adverbios, verbos, y sustantivos presentes en el tweet (también conocido como etiquetado de partes del discurso).

- Reglas semánticas: registran el número de veces que se cumplen las reglas mencionadas en [41] y resumidas en la **Tabla 14** como descripción de las características 49 a 63.

- Trigramas polaridad: registran el número de veces que se obtiene un trigrama de polaridad en el tweet (explicado más detalladamente en sección 5.3).

- Doc2Vec: representación del tweet como características de tamaño fijo en un espacio de representación generado con una red neuronal, también llamado incrustación de frases (*phrase embeddings*).

Id	Características	Descripción	Tipo
1	start_len	Número de palabras antes del preprocesamiento.	Conteo de palabras
2	Punct	Número de apariciones de los siguientes caracteres '!', '??', '!?', '?!'	Conteo de palabras
3	avg_len	Número de palabras promedio por oración.	Conteo de palabras
4	has_ht	Indica si el tweet contiene al menos un hashtag.	Hashtag
5	neutral_emojis	Cuenta los emoticones neutros, negativos y positivos, que permiten intensificar los tweets. El intensificador va aumentando o disminuyendo según el emoticon, los positivos se suman, negativos se restan, si al final el intensificador es cero se observa el ultimo emoticono y si este es positivo se asigna 1.01, si es negativo o neutro se asigna -1.01.	Emoticones
6	negative_emojis		Emoticones
7	positive_emojis		Emoticones
8	intensify_emoji		Emoticones
9	count_elongated	Número de palabras elongadas.	Conteo de palabras
10	count_uppercase	Número de palabras que comienzan en mayúscula.	Conteo de palabras
11	end_len	Cuenta el número de palabras al terminar el preprocesamiento.	Conteo de palabras
12	neg_words_ht	Número de hashtags negativos, neutros y positivos considerando factores como: vecinos, mayúsculas, negaciones, intensificadores, elongaciones, entre otros, de acuerdo con las listas de Bing Liu [103].	Hashtag
13	neu_words_ht		Hashtag
14	pos_words_ht		Hashtag
15	neg_words_ht_sum		Hashtag
16	neu_words_ht_sum		Hashtag
17	pos_words_ht_sum		Hashtag
18	neg_ht	Número de sentimientos de los hashtags basado en 'NRC Emotion and Sentiment Lexicons' [104].	Hashtag
19	pos_ht		Hashtag
20	neu_ht		Hashtag
21	pol_words_ht		Hashtag

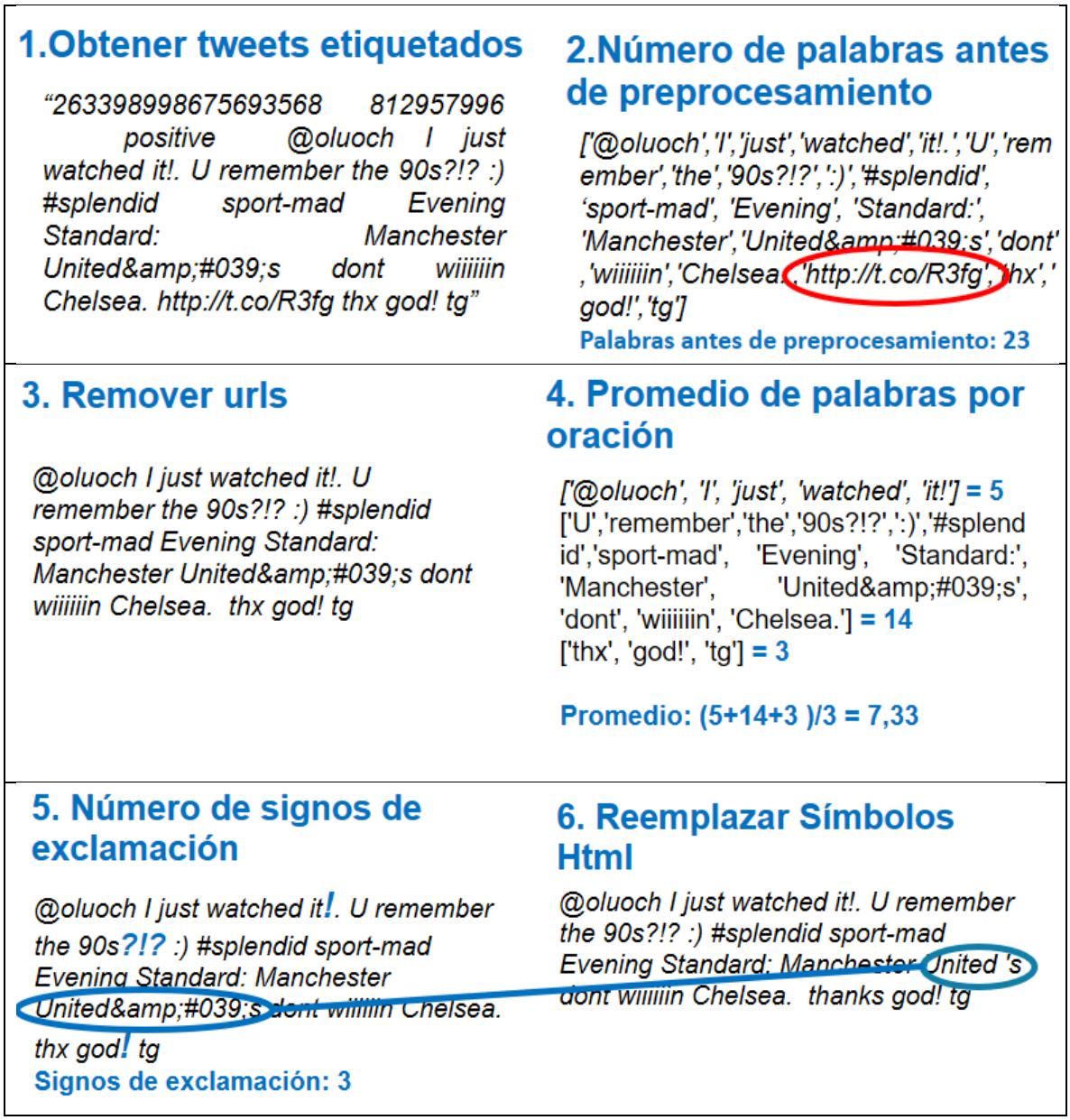
22	neg_words_tweet	Número de palabras negativas, neutras y positivas en el tweet de acuerdo con las listas de Bing Liu [103].	Polaridad
23	neu_words_tweet		Polaridad
24	pos_words_tweet		Polaridad
25	negat_words_ht	Número de palabras de negación en el tweet y en los hashtags, basado en lista de negaciones.	Hashtag, Polaridad
26	negat_words_tweet		Polaridad
27	adj_frac	Número de adjetivos, adverbios, verbos, y sustantivos basado en NLTK pos tagger English.	Etiquetado gramatical
28	adv_frac		Etiquetado gramatical
29	v_frac		Etiquetado gramatical
30	nn_frac		Etiquetado gramatical
31	Neg	Se importan todos los valores de las palabras que tienen polaridad usando SentiWordNet en los hashtags. neg: Acumulado de polaridad negativa en el tweet. neu: Acumulado de polaridad neutral en el tweet. pos: Acumulado de polaridad positiva en el tweet. neg_words: Número de palabras negativas. neu_words: Número de palabras objetivas o neutrales. pos_words: Número de palabras positivas. neg_words_sum: Suma del sentimiento negativo según SentiWordNet. neu_words_sum: Suma del sentimiento neutral según SentiWordNet. pos_words_sum: Suma del sentimiento positivo según SentiWordNet. Una palabra se considera objetivo, si su puntaje objetivo es > = 0.8. pol_words: Número de palabras de polaridad (no todas las palabras tienen un valor de polaridad en SentiWordNet). negat_words: Número de palabras de negación. Se considera palabra de negación una palabra cuyo puntaje negativo es > = 0.8.	Polaridad
32	Neu		Polaridad
33	Pos		Polaridad
34	neg_words		Polaridad
35	neu_words		Polaridad
36	pos_words		Polaridad
37	neg_words_sum		Polaridad
38	new_words_sum		Polaridad
39	pos_words_sum		Polaridad
40	pol_words		Polaridad
41	negat_words		Polaridad
42	neg_words_ht_lists		Hashtag

43	neu_words_ht_lists	Número de palabras negativas, neutras y positivas de los hashtags pertenecientes a un tweet según las listas de Bing Liu [103].	Hashtag
44	pos_words_ht_lists		Hashtag
45	neg_ht_NRC	Número de palabras de sentimiento de los hashtags basado en la lista de NRC Emotion and Sentiment Lexicons.	Hashtag
46	pos_ht_NRC		Hashtag
47	pol_words_ht_NRC		Hashtag
48	neu_ht_NRC		Hashtag
49	r1	Número de veces que se cumple la regla R1 en el tweet: R1- existe una negación en el tweet. E.g.: 'not bad'.	Reglas semánticas
50	r2	Número de veces que se cumple la regla R2 en el tweet: R2- existe "of" en medio de dos sustantivos o pronombres. E.g.: 'Lack of crime in rural areas'.	Reglas semánticas
51	r3	Número de veces que se cumple la regla R3 en el tweet: R3- existe un verbo luego de un sustantivo. E.g.: 'Crime has decreased'.	Reglas semánticas
52	r4	Número de veces que se cumple la regla R4 en el tweet: R4 - existe un sustantivo seguido de un verbo 'to be', seguido de un adjetivo. E.g.: 'Damage is minimal'.	Reglas semánticas
53	r5	Número de veces que se cumple la regla R5 en el tweet: R5 - existe un sustantivo, seguido de 'of', seguido de un verbo. E.g.: 'Lack of killing in rural areas'.	Reglas semánticas
54	r6	Número de veces que se cumple la regla R6 en el tweet: R6 - existe un adjetivo seguido de 'to', seguido de un verbo. E.g.: 'Unlikely to destroy the planet'.	Reglas semánticas
55	r7	Número de veces que se cumple la regla R7 en el tweet: R7 - existe un verbo seguido de un sustantivo. E.g.: 'Destroyed terrorism'.	Reglas semánticas
56	r8	Número de veces que se cumple la regla R8 en el tweet: R8 - existe un 'to' en medio de dos verbos. E.g.: 'Refused to deceive the man'.	Reglas semánticas

57	r9	Número de veces que se cumple la regla R9 en el tweet: R9 - existe 'as' seguido de un adjetivo, seguido de 'as' y luego una frase nominal o sustantivo. E.g.: 'As ugly as a rock'.	Reglas semánticas
58	r10	Número de veces que se cumple la regla R10 en el tweet: R10 - existe una negación, seguida de 'as', seguida de un adjetivo, seguida de 'as' y luego una frase nominal o sustantivo. E.g.: 'That wasn't as bad as the original'.	Reglas semánticas
59	r11	Número de veces que se cumple la regla R11 en el tweet: R11 - Contiene "but". E.g.: 'And I've never liked that director, but I loved this movie'.	Reglas semánticas
60	r12	Número de veces que se cumple la regla R12 en el tweet: R12 - Contiene "despite". E.g.: 'I love the movie, despite the fact that I hate that director'.	Reglas semánticas
61	r13	Número de veces que se cumple la regla R13 en el tweet: R13 - Contiene "unless". E.g.: 'Everyone likes the video unless he is a sociopath'.	Reglas semánticas
62	r14	Número de veces que se cumple la regla R14 en el tweet: R14 - Contiene "while". E.g.: 'While they did their best, the team played a horrible game'.	Reglas semánticas
63	r15	Número de veces que se cumple la regla R15 en el tweet: R15 - Contiene "however". E.g.: 'The film counted with good actors. However, the plot was very poor'.	Reglas semánticas
64 a 90	Trigramas formados por la polaridad de 3 palabras consecutivas	Se describe más adelante en la sección 5.3.	Trigramas polaridad
91	Doc2Vec	Se describe más adelante en la sección 5.5.	Doc2Vec

Tabla 14. Descripción de las características extraídas

Las 91 características mencionadas se obtienen mediante el proceso que se detalla a continuación en la **Tabla 15** numerado paso por paso y con ejemplos.



Dividir el tweet en tokens, eliminar menciones y reducir tamaño de las elongaciones

['I', 'just', 'watched', 'it', '!', '.', 'U', 'remember', 'the', '90s', '?', '!', '?', ':)', '#splendid', 'sport-mad', 'Evening', 'Standard', ':', 'Manchester', 'United', "'", 's', 'dont', 'wiiin', 'Chelsea', '.', 'thanks', 'god', '!', 'tg']

8.Remover signos de puntuación

['I', 'just', 'watched', 'it', 'U', 'remember', 'the', '90s', ':)', '#splendid', 'sport-mad', 'Evening', 'Standard', 'Manchester', 'United', 'dont', 'wiiin', 'Chelsea', 'thanks', 'god', 'tg']

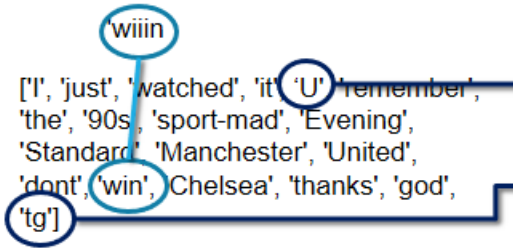
9 y 10. Obtener los hashtags y determinar si los hay

['I', 'just', 'watched', 'it', 'U', 'remember', 'the', '90s', ':)', 'sport-mad', 'Evening', 'Standard', 'Manchester', 'United', 'dont', 'wiiin', 'Chelsea', 'thanks', 'god', 'tg']
Hashtags= ['splendid']
Tiene hashtag: SI

11. Emoticones positivos, negativos, neutros e intensidad

['I', 'just', 'watched', 'it', 'U', 'remember', 'the', '90s', ':)', 'sport-mad', 'Evening', 'Standard', 'Manchester', 'United', 'dont', 'wiiin', 'Chelsea', 'thanks', 'god', 'tg']
Emoticones neutrales: 0
Emoticones negativos: 0
Emoticones positivo: 1
Intensidad: 1 (positivo)

12. Palabras elongadas



Numero de palabras elongadas: 1

13. Reemplazar jerga de Twitter

['I', 'just', 'watched', 'it', 'you', 'remember', 'the', '90s', 'sport-mad', 'Evening', 'Standard', 'Manchester', 'United', 'dont', 'win', 'Chelsea', 'thanks', 'god', 'that is great']

14. expandir contracciones en las negaciones y en los hashtags

['I', 'just', 'watched', 'it', 'you', 'remember', 'the', '90s', 'sport-mad', 'Evening', 'Standard', 'Manchester', 'United', 'do not', 'win', 'Chelsea', 'thanks', 'god', 'that is great']

Hashtags= ['splendid']

15. remover números positivos y negativos del tweet

['I', 'just', 'watched', 'it', 'you', 'remember', 'the', 's', 'sport-mad', 'Evening', 'Standard', 'Manchester', 'United', 'do not', 'win', 'Chelsea', 'thanks', 'god', 'that is great']

16. separar aquellas palabras que tienen guion

['I', 'just', 'watched', 'it', 'you', 'remember', 'the', 's', 'sport mad', 'Evening', 'Standard', 'Manchester', 'United', 'do not', 'win', 'Chelsea', 'thanks', 'god', 'that is great']

17. separar palabras que se reemplazaron por 2 o más

['I', 'just', 'watched', 'it', 'you', 'remember', 'the', 's', 'sport', 'mad', 'Evening', 'Standard', 'Manchester', 'United', 'do', 'not', 'win', 'Chelsea', 'thanks', 'god', 'that', 'is', 'great']

Hashtags= ['splendid']

18. Eliminar palabras no pertenecientes al lenguaje ingles

[''I', just', 'watched', 'it', 'you', 'remember', 'the', 'sport', 'mad', 'Evening', 'Standard', 'United', 'do', 'not', 'win', 'thanks', 'god', 'that', 'is', 'great']

Hashtags= ['splendid']

19 y 20. Contar número de palabras que comienzan en mayúscula y palabras después de preprocesamiento

['I', 'just', 'watched', 'it', 'you', 'remember', 'the', 'sport', 'mad', 'Evening', 'Standard', 'United', 'do', 'not', 'win', 'thanks', 'god', 'that', 'is', 'great']

Número de palabras Mayúscula: 3
Número de palabras: 20

21. Contar los hashtags negativos, positivos y neutros

Se compara palabra a palabra con los listados proporcionados por Bing Liu. "Sentiment Analysis and Subjectivity."
Hashtags neutrales: 0
Hashtags negativos: 0
Hashtags positivos: 1
Hashtags= ['splendid']

22. Contar los sentimientos de los hashtags

Este conteo se realiza comparando cada palabra con las palabras pertenecientes al NRC Emotion and Sentiment Lexicons.
Hashtags neutrales: 0
Hashtags negativos: 0
Hashtags positivos: 1
Hashtags con polaridad: 1
Hashtags= ['splendid']

23. Determinar la polaridad de los hashtag

Para hacerlo, se consideran factores especiales, como la polaridad de los hashtags vecinos, hashtags en mayúsculas, negaciones e intensificadores, entre otros.

Polaridad negativa: 0 Palabras de polaridad: 1
Polaridad neutral: 0 Negaciones: 0
Polaridad positiva: 0 Palabras negativas: 0
Palabras neutrales: 0 Palabras positivas: 1
Promedio palabras negativas 0.0
Promedio palabras neutrales: 0.0
Promedio palabras positivas: 0.625

24. Detectar la polaridad de las palabras del tweet

Palabras negativas: 1
Palabras neutrales 17
Palabras positivas 2

25. Contar el numero de palabras de negación en el tweet y en los hashtags

Palabras de negación Tweet: 1
Palabras de negación Hashtag: 0

26. Adjetivos, Adverbios, Verbos y Sustantivos

Adjetivos: 1
Adverbios: 2
Verbos: 6
Sustantivos: 6

27. Polaridad usando SentiWordNet en el tweet y en los hashtags

Polaridad negativa: 1.125
Polaridad neutral: 14.625
Polaridad positiva: 2.25
Palabras negativas: 1
Palabras neutrales: 13
Palabras positivas: 7
Promedio palabras negativas 0.625
Promedio palabras neutrales: 12.5
Promedio palabras positivas: 2.0
Palabras de polaridad: 18
Negaciones: 0

28. Ocurrencias de Reglas Semanticas

R1: 1
R2: 0
R3: 1
R4: 0
R5: 0
R6: 0
R7: 0
R8: 0
R9: 0
R10: 0
R11: 0
R12: 0
R13: 0
R14: 0
R15: 0

29. Ocurrencias de Trigramas de polaridad de 3 palabras consecutivas	30. Representación Doc2Vec
pos_pos_pos = 0.0 pos_pos_neg = 0.0 pos_pos_neu = 0.0 pos_neg_pos = 0.355672028058 ... neu_neg_neu = 0.183605404779 neu_neu_pos = 0.28502955861 neu_neu_neg = 0.178612052447 neu_neu_neu = 0.736618602824	[-0.04013254866, -0.087734490633, ..., -0.10277813673, -0.0965704619884, 0.0255022179335, 0.183166623116, -0.0420833006501, -0.030747005716, -0.0133021101356, -0.0272661130875, -0.0716953352094, 0.0833471789956, -0.0108029926196, -0.0104782106355, 0.0137445395812, 0.104885593057, 0.0708704516292, -0.0964932441711, 0.0185575317591, -0.010923105292, 0.00180098903365]

Tabla 15. Pasos para obtener las características que representan cada Tweet

Para eliminar URLs se utilizó expresiones regulares evaluando cada token y eliminando aquellos que detecta como urls y menciones, además, se reemplazó jerga y abreviaturas, similar a lo que se hizo en [61]: Primero se eliminan los identificadores de Twitter (@nombre de usuario) y las URLs. Se eliminaron las fechas y números con expresiones regulares y se canonicalizaron las abreviaturas comunes, jergas y negaciones usando un léxico reunido a partir de recursos en línea y una lista de negaciones que abarca: don't, mustn't, shouldn't, isn't, aren't, wasn't, weren't, not, couldn't, won't, can't, wouldn't. Los cuales se reemplazaron con las formas respectivas y lo mismo para sus formas libres de apóstrofes (por ejemplo, 'cant'), que es más probable que ocurran en hashtags.

Con la finalidad de promediar las palabras por oración, se realizó el conteo de las palabras que hay hasta cada punto (.), asumiendo que cada oración se finaliza con un punto, y se realiza el promedio, si no hay puntos, el promedio es el total de palabras en el tweet.

Cuando los hashtags son conformados por varias palabras, se obtiene cada una de las palabras que los conforman.

En la eliminación de palabras elongadas se reemplaza los caracteres alargados por una sola ocurrencia o dos si la palabra en su forma normal posee dos, por ejemplo: 'wooooooow' por 'Wow' o 'cooooooooool' por 'cool'.

Con el objetivo de determinar las etiquetas POS en el texto canonicalizado resultante, se determinan las etiquetas 'part of speech' (POS) utilizando el etiquetador POS de Stanford [106].

Finalmente, se elimina cualquier puntuación y token incoherente o que no pertenezca al idioma inglés.

Para realizar el conteo de palabras y emoticones especiales se utilizó los léxicos de sentimientos descritos más adelante en la **sección 5.2** (Léxicos de sentimiento), los cuales fueron utilizados para obtener las siguientes características:

5 - neutral_emojis
6 - negative_emojis
7 - positive_emojis
8 - intensify_emoji
12 - neg_words_ht
13 - neu_words_ht
14 - pos_words_ht
15 - neg_words_ht_sum
16 - neu_words_ht_sum
17 - pos_words_ht_sum
18 - neg_ht
19 - pos_ht
20 - neu_ht
21 - pol_words_ht
22 - neg_words_tweet
23 - neu_words_tweet
24 - pos_words_tweet
25 - negat_words_ht
26 - negat_words_tweet
31 – neg
32 – neu
33 – pos
34 - neg_words
35 - neu_words
36 - pos_words
37 - neg_words_sum
38 - new_words_sum
39 - pos_words_sum
40 - pol_words
41 - negat_words
42 - neg_words_ht_lists
43 - neu_words_ht_lists
44 - pos_words_ht_lists
45 - neg_ht_NRC
46 - pos_ht_NRC
47 - pol_words_ht_NRC
48 - neu_ht_NRC

Estos léxicos fueron utilizados para comparar cada palabra del tweet y llevar el registro de la cantidad de apariciones.

Para obtener las características 64 a 90 - Trigramas formados por la polaridad de 3 palabras consecutivas, se tiene en cuenta lo descrito más adelante en la **sección 5.3** (Trigramas formados por la polaridad de tres palabras consecutivas).

Las características 49 a 63 son una serie de reglas descritas en [41] y resumidas en la **Tabla 14** como descripción de cada características asociada a la regla.

La representación de Doc2Vec (incrustación de frase) se añade al vector de características obtenidas anteriormente y se cuenta como una sola característica atómica.

Al ejecutar los pasos que se muestran en la **Tabla 15**, se obtiene una vista minable similar a la **Figura 4**. A esta representación se le denomina matriz de características (MC) y tiene un tamaño n x k donde n es el número de tweets y k es el número de columnas o características (91 originalmente) más la columna de clase o polaridad. Cada fila es la representación del tweet y cada columna es una característica de las 91 mencionadas, las características que representan conteos de palabras y cumplimiento de reglas se normalizan en función de la longitud del tweet, las de los trigramas con TF-IDF y la característica de Doc2Vec con su representación vectorial, obteniéndose una representación multimodal del tweet.

	1	2	3		k = 91			
	start_len	**punct**	**avg_len**	**…**	**Doc2Vec**			**Class**
1	18	2	9	…	1.3	…	0.5	Positive
2	21	3	7	…	1.4	…	0.6	Negative
…	…	…	…	…	…			…
n	5	1	5		0.8	…	0.7	Neutral

Figura 4. Ejemplo de la representación de los Tweets, Matriz de Características (MC) original

Teniendo la representación presentada para cada uno de los tweets, se procede a realizar el muestreo de filas con remplazo y el de columnas con arreglos de cobertura. Al igual que en bagging, se crea un número M de muestras boostrap, es decir, M muestras de filas seleccionadas aleatoriamente del dataset de entrenamiento (matriz de características MC), donde M se define con base en el número de filas de un arreglo de cubrimiento (covering array, CA [15]) previamente seleccionado.

En este caso se usó un arreglo de cobertura binario denotado por CA (M = 206; k = 91, v=2, t = 5), que es una matriz de M filas y k columnas, donde M, el número de muestras boostrap a realizar y filas en el CA es decir 206, k es el número de factores o características de MC, 91 en este caso para la MC original, v es el número de símbolos del CA que en este caso es binario, donde 0 implica que la característica no será tenida en cuenta en la muestra y 1 que si será incluida en ésta, y t es el grado de interacción entre los parámetros, es decir la fuerza en el CA. En [16] y [17] se describen en detalle las propiedades de los CA y su uso en el diseño de experimentos y en las pruebas de caja negra de software y hardware.

A manera de ejemplo, en la **Tabla 16** se muestra el CA (6; 10, 2, 2). La fuerza de este arreglo de cobertura es 2 (*t=2*), con 10 factores (*k=10*) y un alfabeto binario (*v=2*) representado por los símbolos (0 y 1). En el método propuesto, este CA indica que se deben generar 6 muestras boostrap para una MC de 10 características y cada fila del CA define cuál de las diez características es incluida en cada muestra boostrap. Es así como en la primera muestra (fila del CA) se incluyen todas las características, en la segunda muestra se incluyen sólo las características 4, 5, 6 y 7, y así sucesivamente con las otras filas del CA.

1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	0	0	0
0	1	0	1	0	0	0	0	1	1
1	1	1	0	0	1	0	0	0	0
0	0	1	0	0	0	1	1	1	0
1	0	0	0	1	0	0	1	0	1

Tabla 16. Ejemplo de CA (6; 10, 2, 2)

La **Figura 5** muestra cuáles serán las columnas o características seleccionadas según la fila del arreglo de cobertura. En la parte superior en fondo gris se ve la fila del CA y donde hay un uno (1) se selecciona la fila, lo que se representa con un recuadro de color rojo. En este caso la matriz resultante MR_i es la presentada en la **Figura 6**.

0	1	0	0	1	1	1	0	1	0	0	0	1	0	

Pos emo	Neg emo	Elongated	Upper	has ht	neg ht	neu ht	pos ht	neg words ht	neu words ht	pos words ht	pol words ht	negat words ht	neu words ht sum	...	Class
3	4	3	0	1	1	1	0	1	0	0	0	1	0	...	Positive
2	3	0	0	0	0	0	0	0	2	0	0	0	1	...	Neutral
1	4	0	0	0	0	0	0	0	0	3	0	0	0	...	Neutral
4	1	1	0	0	0	0	0	0	0	0	0	1	2	...	Positive
5	1	0	0	1	2	0	0	2	3	1	1	0	0	...	Positive
2	3	3	1	1	1	1	0	1	0	2	3	0	0	...	Neutral
4	2	1	0	0	0	0	0	0	0	4	0	1	0	...	Negative
5	1	0	0	0	0	0	0	0	0	0	1	0	3	...	Negative
2	5	0	2	1	0	0	0	3	0	2	0	1	0	...	Negative
4	3	0	0	1	2	0	0	2	0	0	0	0	0	...	Positive
5	2	0	0	1	1	1	0	1	4	1	0	0	0	...	Positive
2	3	0	0	0	0	0	0	0	0	0	2	0	2	...	Positive
4	1	1	0	0	0	0	0	1	0	0	0	1	3	...	Negative
5	0	0	3	0	0	0	0	0	0	0	0	0	0	...	Negative
3	2	0	0	0	2	0	0	2	4	0	0	0	0	...	Negative
2	3	1	0	0	1	1	0	1	0	0	0	3	3	...	Positive
1	4	0	0	1	0	0	0	1	0	2	0	0	2	...	Neutral
4	1	0	0	1	0	0	0	0	0	0	1	0	4	...	Negative

Figura 5. Columnas seleccionadas basado en el arreglo de cobertura

Neg emo	has ht	neg ht	neu ht	neg words ht	negat words ht	Class
4	1	1	1	1	1	Positive
3	0	0	0	0	0	Neutral
4	0	0	0	0	0	Neutral
1	0	0	0	0	1	Positive
1	1	2	0	2	0	Positive
3	1	1	1	1	0	Neutral
2	0	0	0	0	1	Negative
1	0	0	0	0	0	Negative
5	1	0	0	3	1	Negative
3	1	2	0	2	0	Positive
2	1	1	1	1	0	Positive
3	0	0	0	0	0	Positive
1	0	0	0	1	1	Negative
0	0	0	0	0	0	Negative
2	0	2	0	2	0	Negative
3	0	1	1	1	3	Positive
4	1	0	0	1	0	Neutral
1	1	0	0	0	0	Negative

Figura 6. Matriz Resultante (MR_i)

El proceso de creación de muestras, inducción de reglas y selección de atributos se resume en la **Figura 7**. En este paso del método propuesto se toman las M muestras bootstrap (MB_1, MB_2, … MB_M) previamente construidas y haciendo uso de algoritmos de inducción de reglas (C4.5 [20, 21], CART [20, 22] y JRIP [20]) se crean 2 árboles (uno n-ario y otro binario) y una lista de reglas para cada muestra bootstrap. Paso seguido, se reúnen las distintas características incluidas en los M árboles o M reglas generadas (controlando un número mínimo de instancias por hoja en los árboles, para realizar una poda que permita encontrar la mínima cantidad de atributos necesarios para su clasificación, los que aportan más información) y

estos se toman como las características finales del proceso de selección de características. En la parte central e inferior de la figura se muestra la vista minable resultante MR, que corresponde a la matriz (o dataset) de representación de los tweets, pero solamente con las p (p<< k) características seleccionadas, el cual puede ser usado como el dataset de entrenamiento de cualquier clasificador, entre ellos, Naive Bayes, Regression Lineal, Random Forest, C4.5. Support Vector Machines y Multi-Layer Perceptron.

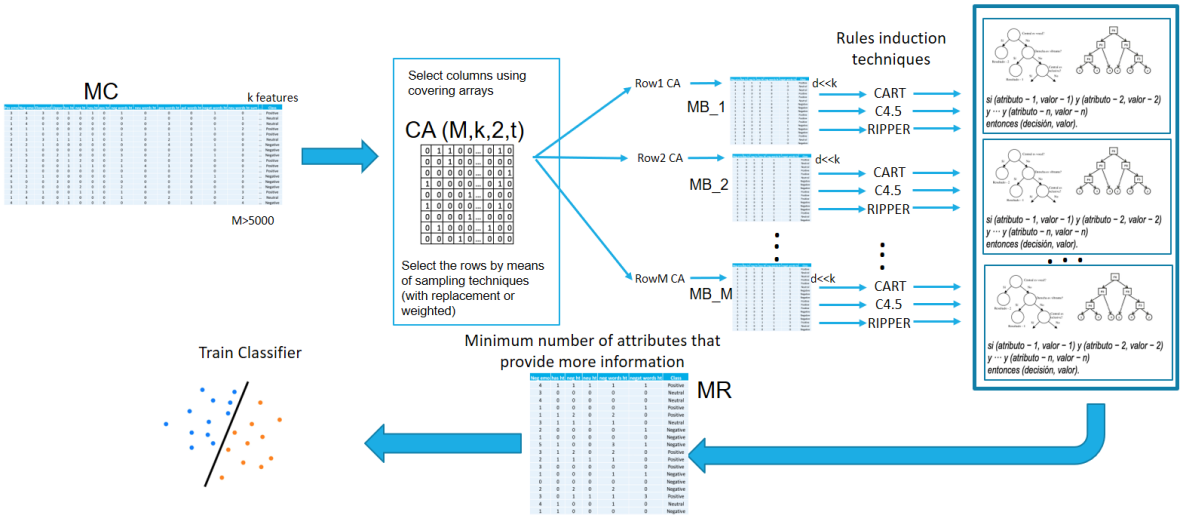


Figura 7. Propuesta de selección de características en Tweets

Determinar las características más apropiadas para la clasificación (detección de polaridad) de tweets permite disminuir el preprocesamiento de estos, disminuir el tiempo de construcción de los modelos de clasificación y obtener resultados que son más legibles. Además, según los resultados de la experimentación, dichos modelos entregan una calidad de clasificación que es similar o superior a la que se obtiene usando las 91 características originales.

Es preciso aclarar que la elección del método de muestreo, muestreo de filas aleatorio con remplazo (bagging) y no muestreo ponderado (manipulando el peso de los objetos mal clasificados como en boosting), obedece a pruebas realizadas en la experimentación, donde los resultados del primer método superaron a los del segundo, esto se detalla más adelante en la **sección 6.3**.

5.1 Obtención de tweets y herramientas

En necesario realizar el preprocesamiento de los tweets antes de utilizar el método de selección de características propuesto, esto con el fin de convertir un tweet obtenido de un dataset, tal como el siguiente:

"263398998675693568 812957996 positive @oluoch @victor_otti @kunjand I just watched it! Sridevi's comeback.... U remember her from the 90s?? Sun mornings on NTA ;)"

Que tiene la siguiente estructura:

<SID><tab><UID><tab><TOPIC><tab><positive|negative|neutral|objective><tab><TWITTER_MESSAGE>

A una representación vectorial más fácil de interpretar para el método de selección de características propuesto, donde cada celda del vector corresponde a una característica de las descritas en la **Tabla 14**.

Para lograr llegar a la representación vectorial, se debe iniciar cargando el dataset de entrenamiento, uno de los datasets utilizados en este trabajo es SemEval[9] (Semantic Evaluation) [43], que fue generado inicialmente para evaluar la "subtarea A" relacionada con la detección de polaridad (Dado un tweet, predecir si es de sentimiento positivo, negativo o neutral), el formato es el siguiente:

id<TAB>label<TAB>tweet
Donde "label" puede ser 'positive', 'neutral' o 'negative'.

A continuación, se describe el entorno de desarrollo y herramientas con las que se realizó el preprocesamiento:

El preprocesamiento fue realizado con el lenguaje Python 2[10], mediante la distribución Anaconda[11], la cual es utilizada en ciencia de datos y aprendizaje automático y está orientado a simplificar el despliegue y administración de las aplicaciones y paquetes necesarios para procesar grandes volúmenes de datos, análisis predictivo y cómputo científico.

Mediante Anaconda fue posible establecer un ambiente de desarrollo para Python llamado Spyder[12] y a la vez instalar la lista de paquetes que se muestran en la **Tabla 17**.

Nombre del paquete/modulo	Descripción
Os	Este módulo proporciona la forma de utilizar la funcionalidad del sistema operativo.

[9] SemEval-2016 Task 4 website: http://alt.qcri.org/semeval2016/task4/
[10] https://www.python.org/download/releases/2.7.2/
[11] https://www.anaconda.com/download/
[12] https://anaconda.org/anaconda/spyder

Random	Este módulo implementa generadores de números pseudo aleatorios para varias distribuciones.
Gensim	Gensim es una librería de Python para el modelado de temas, la indexación de documentos y la recuperación de similitudes con grandes corpus. Su objetivo es el procesamiento de lenguaje natural (NLP) y recuperación de información (IR).
Pandas	Pandas es una librería de Python destinada al análisis de datos, que proporciona unas estructuras de datos flexibles y que permiten trabajar con ellos de forma muy eficiente. Pandas ofrece las siguientes estructuras de datos: Series, DataFrame, Panel, Panel4D y PanelND.
Numpy	NumPy es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.
Nltk	NLTK, es un conjunto de bibliotecas y programas para el procesamiento del lenguaje natural (PLN) simbólico y estadísticos para el lenguaje de programación Python. NLTK incluye demostraciones gráficas y datos de muestra.
Sklearn	Scikit-learn es una biblioteca de aprendizaje de máquina, Cuenta con varios algoritmos de clasificación, regresión y agrupación, incluyendo máquinas de vectores de soporte, bosques aleatorios, aumento de gradiente, k-means y DBSCAN. Se utilizó en el preprocesamiento para obtener el tf-idf (la frecuencia de ocurrencia del término en la colección de documentos) de los trigramas de los sentimientos de las palabras consecutivas.
WordNet	WordNet es una gran base de datos léxica de inglés. Los sustantivos, verbos, adjetivos y adverbios se agrupan en conjuntos de sinónimos cognitivos (synsets), cada uno expresando un concepto distinto. Los synsets están interrelacionados por medio de relaciones semántico-conceptuales y léxicas.
SentiWordNet	SentiWordNet es un recurso léxico para la minería de opinión. SentiWordNet asigna a cada synset de WordNet tres puntajes de sentimiento: positividad, negatividad, objetividad.
Pattern	Tiene herramientas para la minería de datos, procesamiento de lenguaje natural (etiquetadores de voz parcial, búsqueda

	de n-gramas, análisis de sentimientos, WordNet), aprendizaje automático (vector modelo espacial, agrupamiento, SVM), análisis de red y visualización. Se utilizó mayormente para convertir palabras a singular y obtener la forma de los verbos sin conjugar.

Tabla 17. Paquetes utilizados en el preprocesamiento

Con estas herramientas se procedió a cargar el listado de Tweets y a cada uno de ellos se le realizó el proceso descrito en la **Tabla 15**.

5.2 Léxicos de sentimiento

Para realizar el conteo de determinadas palabras y emoticones que pertenecen a cada tweet se utilizaron los siguientes léxicos:

- **Palabras que transmiten opinión**, es decir, palabras con significado positivo o negativo tomadas del léxico utilizado por Bing Liu en [103].

- **SentiWordNet** [107] se utilizó para extraer la polaridad de las palabras que tienen sentido de opinión, así como sus etiquetas POS asociadas. También se registró el conteo del número de sustantivos, verbos, adjetivos y adverbios por tweet [108-111].

- **NRC Hashtag Lexicon** [104] se utilizó para tener en cuenta las 78 palabras semillas con hashtags positivos y negativos tales como #good, #excelent, #bad y #terrible, de tal forma que resulta útil como un indicador de la polaridad del tweet.

- **Léxico de emoticones de Wikipedia**, con 81 emoticones que expresan sentimientos positivos y negativos [61].

5.3 Trigramas formados por la polaridad de tres palabras consecutivas

Se agrega una representación del tweet que consiste en 27 valores, donde cada una representa la frecuencia de aparición de la polaridad de 3 palabras consecutivas, por ejemplo:

"*i love you lucy*"

'I' = Neutral
'Love' = Positiva
'You' = Neutral

'Lucy' = Neutral

Obtendrá 2 apariciones, una en neu_pos_neu (i, love, you) y otra en pos_neu_neu (love, you, lucy), los demás trigramas tendrían un 0 de frecuencia.

5.4 Reglas semánticas

En [41], se afirma que el resultado final de un algoritmo de clasificación podría verse afectado por la negación y el uso de partículas específicas de PoS. Por lo tanto, se proponen una serie de reglas que pueden desempeñar un papel clave en la clasificación semántica de las oraciones, en este trabajo se tiene en cuenta la frecuencia de aparición de estas reglas en el tweet, las 15 reglas mencionadas en [41], están descritas como las características 49 a 63 en la **Tabla 14**.

5.5 Doc2Vec

El objetivo de doc2vec es crear una representación numérica de un documento, independientemente de su longitud. Mientras que los vectores de palabras representan el concepto de una palabra, el vector del documento (doc2vec) intenta representar el concepto de un documento como un vector [112].

Para entender Doc2Vec es necesario entender primero que es Word2Vec. Word2Vec pretende representar las palabras como un vector en un espacio multidimensional de forma que las palabras similares o relacionadas se encuentren representadas por puntos cercanos. De esta forma se captura información semántica, por ejemplo, palabras como "rojo", "negro" y "blanco" se encontrarán en una misma zona de ese espacio multidimensional y lo mismo pasaría con palabras como "león", "tigre" y "leopardo". Dado un conjunto de frases (también llamado corpus) el modelo analiza las palabras de cada sentencia y usa cada palabra para predecir las palabras vecinas. Por ejemplo, a la palabra "Caperucita" le seguirá "Roja" con más probabilidad que cualquier otra palabra [113].

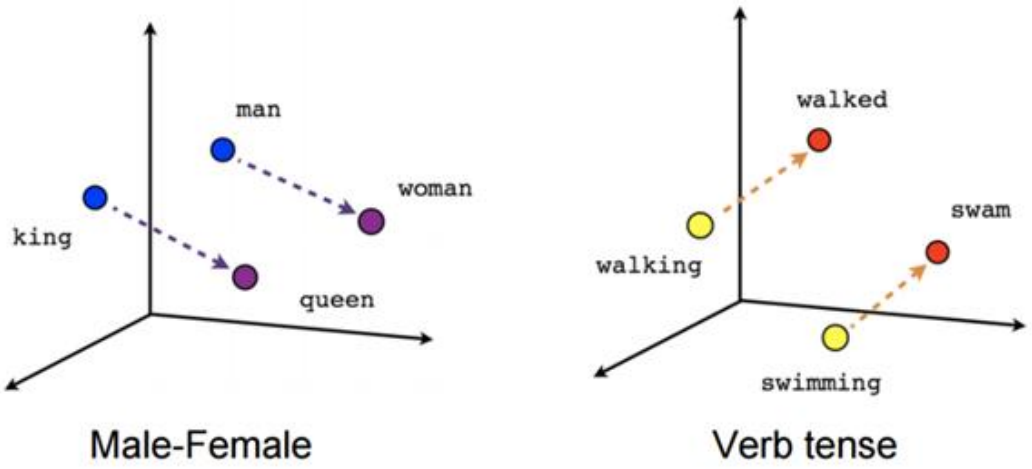


Figura 8. Relaciones entre palabras que se evidencian usando Word2Vec

Tales representaciones, encapsulan diferentes relaciones entre palabras, como sinónimos, antónimos o analogías, o expresan un concepto significativo como el género o el tiempo verbal, como se muestra en la **Figura 8** (dimensionalidad reducida) [114, 115].

Como se mencionó previamente, el objetivo de doc2vec es crear una representación numérica de un documento, independientemente de su longitud. Pero a diferencia de las palabras, los documentos no vienen en estructuras lógicas como las palabras, por lo que se utiliza el método de Mikilov y Le, quienes utilizaron el modelo de word2vec y agregaron otro vector (ID de párrafo a continuación), así [114, 115]:

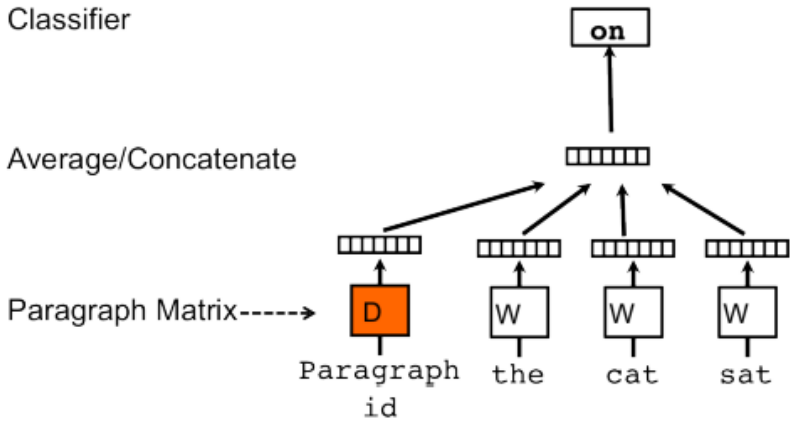


Figura 9. El modelo de versión de memoria distribuida de vector de párrafo (PV-DM)

Entonces, al entrenar los vectores de palabras W, el vector de documento D también se entrena, y al final del entrenamiento, obtiene una representación numérica del documento. El modelo actúa como una memoria que recuerda el contexto actual o el tema del párrafo. Mientras que los vectores de palabras representan el concepto de una palabra, el vector del documento intenta representar el concepto de un documento [114, 115].

En [116] se evidencio que usar modelos *(Word embedings)* word2vec pre entrenadas de Wikipedia[13,14] para obtener nuevas representaciones de los párrafos, aumenta el rendimiento de la clasificación. Por lo anterior, en este trabajo se empleó una representación de 300 dimensiones de cada tweet usando el modelo Doc2Vec pre entrenado en [112, 117] con documentos de Wikipedia [116] y disponible en https://github.com/jhlau/doc2vec#pre-trained-doc2vec-models.

[13] Los autores utilizaron el volcado de Wikipedia del 2015-12-01, limpiado usando WikiExtractor: https://github.com/attardi/wikiextractor.
[14] El tamaño del vector definido en los parámetros por los autores del modelo preentrenado fue de 300.

Capítulo 6

6 Resultados obtenidos

Los experimentos realizados buscaron en cada dataset de prueba, primero determinar la efectividad del proceso de selección de características cuando los datos de entrenamiento y prueba se obtienen del mismo dataset (66% y 34% respectivamente), es decir, cuando se usa la misma distribución de los datos. Luego, se realizó un experimento en donde el dataset de entrenamiento y de prueba son diferentes, con lo cual se busca evaluar la calidad de los datasets de entrenamiento y su efecto en el proceso de selección de atributos.

6.1 Descripción de los conjuntos de datos y las medidas de evaluación

La **Tabla 18** resume los datasets usados para la experimentación. En la columna "Total Original" se muestra el número de Tweets reportados originalmente en la referencia según la columna "Ref", luego se muestra el "Total" de tweets que se lograron descargar debido al cambio en las políticas de Twitter. Seguidamente el número de tweets positivos, negativos y neutros. Si el dataset se formuló originalmente para entrenamiento y desarrollo, se muestra el número de tweets que se descargaron en cada tarea. Los datasets en negrilla fueron los usados para la evaluación y comparación.

Dataset	Total Original	Ref	Total	Positives	Negatives	Neutral	Train	Dev
SemEval 2013-Train+dev	11382	[100]	11338	4215	1798	5325	9728	1654
SemEval 2013 Test	3814	[100]	3813	1572	601	1640		
SemEval 2016 Train+dev	8000	[43]	7350	3606	1148	2596	6000	2000
SemEval 2016 Test	2000	[43]	1814	896	288	630		
SemEval 2016 Eval	20632	[101]	16167	5620	2383	8164		
Sentiment140 Test	498	[31]	498	182	177	139		

Tabla 18. Resumen de los datasets usados en la experimentación

Como medidas de evaluación y comparación se usan la exactitud (o porcentaje de instancias correctamente clasificadas, ICC) y la medida F (F1).

6.2 Resultados experimentales y discusiones

La **Tabla 19** presenta los principales resultados de la experimentación en el dataset "SemEval 2013 Test". Esta tabla muestra en la primera línea el resultado del porcentaje de instancias correctamente clasificadas (ICC) y de medida F (F1) tomando el dataset con las 91 características originalmente definidas en la **Tabla 14** y los clasificadores Linear Regression (LR), Simple Linear Regression (Simple LR), Naive Bayes, una implementación de Support Vector Machines (SMO), Multi-Layer Perceptron (MLP), Random Forest, JRIP, J48, CART y SVM.

SemEval 2013 Test	LR		Simple LR		Naive Bayes		SMO		MLP		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Original (66% / 34%)	65,0	65,0	**66,8**	**66,7**	63,1	62,9	**66,1**	66,2	60,0	59,1	91
FS (66% / 34%)	64,4	64,5	**65,3**	**65,1**	61,7	61,9	**64,2**	64,3	61,7	61,6	21
Training: SemEval 2013-Train+dev	**66,2**	65,7	**66,8**	66,2	58,5	57,1	65,0	64,4	63,4	62,3	91
FS Base Line	**65,4**	64,8	**65,5**	64,8	60,0	60,0	65,1	64,6	59,1	58,8	22
SemEval 2013 Test	JRIP		CART		J48		RANDOM FOREST		SVM		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Original (66% / 34%)	59,2	57,4	61,4	60,6	54,0	53,9	64,9	62,7	53,4	48,9	91
FS (66% / 34%)	61,0	59.8	62,5	62,1	54,3	54,3	62,8	60,3	53,5	48,8	21
Training: SemEval 2013-Train+dev	62,2	61,5	61,8	60,0	63,2	62,2	63,0	61,1	55,2	48,8	91
FS Base Line	60,3	58,4	61,9	60,7	63,2	62,2	61,9	59,7	55,2	48,5	22
F1 reportado estado del arte	**69.02** [104] con un clasificador SVM entrenado con dos corpus, uno de tweets positivos y negativos que tenían como hashtags palabras de 'NRC Hashtag Sentiment Lexicon'[105] con 775,000 tweets, 54,129 unigramas y 316,531 bigramas. y otro de tweets con emoticones que contiene 1.6 millones de tweets [31], 62,468 unigramas y 677,698 bigramas.										

Tabla 19. Resultados de la experimentación en SemEval 2013 Test (mejores resultados en negrita)

Los resultados de la segunda línea muestran que en general el proceso de selección de características (Feature Selection, FS) obtiene resultados similares en calidad (medida en ICC y F1) pero reduciendo de 91 a 21 características (columna k). Las líneas 3 y 4 de la tabla muestran el mismo análisis cambiando el dataset usado para soportar el proceso de selección de características. En este caso, se observa la misma situación previa, la calidad disminuye muy poco, pero hay una notable reducción en el número de características (de 91 a 22 - reducción del 75% de características). Comparando los resultados de los dos experimentos, se observa que a pesar de que el dataset del segundo experimento es mucho más grande, la calidad de los resultados no mejora significativamente.

En la **Tabla 19** también se puede evidenciar que la mayor exactitud lograda con todas las características es de 66.8% y al realizar la selección su valor es de 65.3%, en este caso se está perdiendo 1.5% de exactitud, pero se logra una reducción del 77% de las características. Se puede afirmar que, para este dataset, la propuesta obtiene una representación más simple que mantiene un nivel de calidad similar al que se puede lograr con todas las características. Así mismo, se puede afirmar que el dataset utilizado en la segunda prueba (SemEval 2013-Train+dev) tiene una distribución muy similar porque su mayor valor de exactitud con todas las características es el mismo (66.8%) y su resultado al utilizar la selección, es comparable al obtenido al validarlo con el 34% siendo 65.5% su exactitud y logra una reducción similar de 76%.

SemEval 2016 Test	LR		Simple LR		Naive Bayes		SMO		MLP		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Original (66% / 34%)	50,2	50,7	**55,9**	**54,0**	46,2	47,6	54,9	54,4	53,2	52,5	91
FS (66% / 34%)	50,1	50,5	**58,5**	**57,1**	52,0	52,7	53,2	52,3	50,9	50,1	25
Base line: SemEval 2016 Train+dev	**57,2**	56,2	**56,6**	**55,4**	46,9	48,2	56,4	55,0	48,3	48,7	91
FS Base Line	**56,7**	55,5	**57,6**	**56,3**	51,0	51,1	55,8	54,1	52,5	52,7	15
SemEval 2016 Test	JRIP		CART		J48		RANDOM FOREST		SVM		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Original (66% / 34%)	48,9	42,8	49,7	48,1	46,6	46,9	53,9	47,0	48,7	32,0	91
FS (66% / 34%)	54,7	50,8	49,7	48,1	49,5	49,4	54,9	48,6	48,7	32,0	25
Base line: SemEval 2016 Train+dev	51,0	44,0	54,1	49,1	52,0	50,2	54,6	49,0	49,3	32,7	91
FS Base Line	50,3	38,7	54,1	51,8	53,1	51,3	54,7	49,0	49,4	32,8	15
F1 reportado estado del arte	**63.3** [91] usando un enfoque de incrustación de frases convolucionales (convolutional sentence embedding). Aprovechan grandes cantidades de datos para entrenar un conjunto de redes neuronales convolucionales de dos capas cuyas predicciones se combinan utilizando Random Forest.										

Tabla 20. Resultados de la experimentación en SemEval 2016 Test (mejores resultados en negrita)

Los resultados sobre el dataset “SemEval 2016 Test” se presentan en la **Tabla 20**. En el primer experimento, a diferencia del anterior dataset, en este se aprecia una leve mejora en la calidad de la clasificación cuando se toma el conjunto de 25 características seleccionada con el método propuesto. Luego, en el segundo experimento se observa que el método reduce levemente su calidad, pero es mucho más sencillo (sólo 15 características de las 91- reducción del 83%). Además, se aprecia que a pesar de que el dataset es mucho más grande, la calidad no mejora sustancialmente ni en la línea base ni con el proceso de selección, lo que muestra que un proceso activo de selección de instancias puede ser requerido.

De la misma forma se puede evidenciar que con el método propuesto se obtuvo una mejor calidad, aproximadamente 2.6% y un 72% de reducción de características, teniendo una relación directa con el dataset de entrenamiento, con el cual se puede obtener una mejora de 1.0% y sobre todo una reducción del 83% de características siendo muy comparables los resultados.

SemEval 2016 Eval	LR		Simple LR		Naive Bayes		SMO		MLP		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Original (66% / 34%)	62,4	61,7	**63,2**	**62,3**	58,0	58,0	62,3	61,2	58,8	58,4	91
FS (66% / 34%)	61,9	61,1	**62,6**	**61,5**	56,7	57,2	61,9	60,7	55,9	55,6	19
Base Line: SemEval 2013-Train+dev + SemEval 2013 Test + SemEval 2016 Train+dev +SemEval 2016 Test	60,2	60,2	**60,7**	**60,7**	53,5	53,1	60,3	60,3	58,0	58,2	91
FS Base Line	59,6	59,6	**59,7**	**59,8**	53,3	54,3	59,9	59,9	52,2	52,5	28
SemEval 2016 Eval	JRIP		CART		J48		RANDOM FOREST		SVM		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Original (66% / 34%)	57,9	55,7	58,5	56,0	51,0	51,2	60,2	56,0	54,0	41,8	91
FS (66% / 34%)	59,1	57,5	58,3	56,0	62,6	61,5	59,3	54,2	50,8	34,3	19
Base Line: SemEval 2013-Train+dev + SemEval 2013 Test + SemEval 2016 Train+dev +SemEval 2016 Test	50,2	48,4	56,3	55,9	56,6	56,4	57,9	56,9	56,1	51,8	91
FS Base Line	51,5	50,8	56,3	55,9	56,9	56,5	57,7	56,8	57,4	52,6	28
F1 reportado estado del arte	**63.3** [91] usando dos redes convolucionales combinadas con Random Forest										

Tabla 21. Resultados de la experimentación en SemEval 2016 Eval (mejores resultados en negrita)

Los resultados sobre el dataset “SemEval 2016 Eval” se presentan en la **Tabla 21**. En el primer experimento, similar al primer dataset, se obtiene una leve perdida en la calidad de la clasificación cuando se toma el conjunto de 19 características seleccionadas con el método propuesto. Luego, en el segundo experimento se observa que el método también reduce levemente su calidad, pero es más sencillo (28 características de las 91 - reducción del 69%). Además, se aprecia que a pesar

de que el dataset es mucho más grande, la calidad no mejora ni en la línea base ni en el proceso de selección, mostrando que la unión de datasets (SemEval 2013-Train+dev + SemEval 2013 Test + SemEval 2016 Train+dev +SemEval 2016 Test) no cuenta con una distribución similar al dataset de prueba, por eso pierde entre un 2% y 3% de calidad. Lo anterior, motiva a pensar que se requiere un proceso activo de selección de instancias para mejorar la calidad de la clasificación.

Sentiment140 Test	LR		Simple LR		Naive Bayes		SMO		MLP		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Original (66% / 34%)	44,4	43,5	66,2	66,2	**68,0**	**67,3**	60,4	60,1	63,3	63,1	91
FS (66% / 34%)	42,6	42,7	65,6	65,5	**70,4**	**69,9**	61,5	61,3	64,5	64,3	36
Base Line: SemEval 2013-Train+dev + SemEval 2013 Test + SemEval 2016 Train+dev + SemEval 2016 Test	**73,1**	**72,7**	71,1	70,4	63,2	63,4	**73,1**	**73,0**	67,5	67,7	91
FS Base Line	**71,5**	**71,1**	69,1	68,2	67,6	67,3	**71,9**	**71,8**	62,7	62,5	28
Sentiment140 Test	JRIP		CART		J48		RANDOM FOREST		SVM		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Original (66% / 34%)	56,8	56,5	56,8	56,5	56,8	56,5	64,4	63,8	37,8	37,4	91
FS (66% / 34%)	66,2	65,7	66,2	65,7	57,3	57,2	64,4	64,2	38,4	37,8	36
Base Line: SemEval 2013-Train+dev + SemEval 2013 Test + SemEval 2016 Train+dev + SemEval 2016 Test	55,8	49,2	59,2	56,1	62,2	59,6	57,4	53,2	48,5	38,3	91
FS Base Line	60,8	57,3	62,6	61,3	62,4	59,7	61,2	58,0	50,2	39,5	28
F1 reportado estado del arte	**80.0** [105] con Naive Bayes, MaxEnt y SVM, usando un dataset de entrenamiento de 1.6 millones de tweets con emoticones para aprendizaje supervisado a distancia (distant supervised learning)										

Tabla 22. Resultados de la experimentación en Sentiment140 Test (mejores resultados en negrita)

Los resultados sobre el dataset “Sentiment140 Test” se presentan en la **Tabla 22**. En el primer experimento, se aprecia una leve mejoría en la calidad de la clasificación cuando se toma el conjunto de 36 características seleccionadas con el método propuesto. Luego, en el segundo experimento se observa que el método reduce levemente su calidad, pero es mucho más sencillo (28 características de las

91, 69% de reducción). Además, se aprecia que, con el dataset de mayor tamaño, la calidad mejora tanto en la línea base como con el proceso de selección indicando que la distribución de éste aporta más información para la clasificación.

Después de la experimentación con los 4 datasets, las características que tuvieron más relevancia son 22, ya que fueron seleccionadas en la mayoría de los experimentos. Se puede inferir que la relevancia de estas características se da por la estructura y composición de los tweets del dataset. Debido a que los datasets de prueba contienen pocos emoticones y pocos hashtags se observa que la relevancia de estas características es muy baja. También se observa que:

- Del grupo de conteo de palabras se tienen 2 características: 'punct' y 'count_uppercase', que se relacionan con la aparición de algunos signos de puntuación y el uso de palabras en mayúscula. Lo que indica su importancia en la determinación de la polaridad de un tweet.

- Del grupo 'POS tag' se tienen 2 características: 'adv_frac' y 'nn_frac', relacionados con el peso de los adverbios y los sustantivos en el tweet. Esto implica que al procesar las partes del discurso se deben contabilizar principalmente estas dos características.

- Del grupo de polaridad las más recurrentes fueron 13 características: 'neg', 'neg_words_tweet', 'neu_words_tweet', 'pos_words_tweet', 'negat_words_tweet', 'neu', 'pos', 'neg_words', 'neu_words', 'pos_words', 'neg_words_sum', 'pos_words_sum' y 'negat_words'. Este grupo es el que más aporta características para determinar la polaridad de un tweet. La lista de seleccionadas deja por fuera a 'negat_words_ht', 'neu_words_sum', 'pol_words' (número de palabras de negación en el hashtag, suma del sentimiento negativo según SentiWordNet y número de palabras de polaridad según SentiWordNet) que inicialmente se consideraban importantes.

- De las reglas semánticas solo 1 característica fue seleccionada, 'r1', que indica si hay una negación en el tweet, la cual puede apoyar la detección de tweets que tienen formas no directas de comunicar su polaridad como el sarcasmo.

- En los trigramas de polaridad, las de mayor importancia fueron tan solo 3 de las 27 características, 'neu_neu_neg', 'pos_neu_neu', 'neu_neu_neu'. Esta selección puede deberse a que las palabras neutras son las más frecuentes dentro de los tweets y la elección de estas 3 características complementan información con los conteos de polaridad.

- La representación Doc2Vec fue fundamental en todos los experimentos, ya que se experimentó con el método sin incluir las incrustaciones de Doc2Vec y los resultados presentaron una reducción clara en la calidad frente a los experimentos que la incluían.

- Del grupo hashtag y emoticones no se seleccionó ninguna característica.

Como se evidencia en los resultados, de 91 características originalmente definidas, se logró reducir sustancialmente el número de características sin reducir o en muchos casos mejorando la calidad de la clasificación, con 'SemEval 2013 Test' se redujo de 91 a 22, de 'SemEval 2016 Test' se redujo de 91 a 15, de 'SemEval 2016 Eval' se redujo de 91 a 28 y con 'Sentiment140 Test' se redujo de 91 a 28, en el mejor de los casos se utilizó un 16.5% de las características originales.

A pesar de que no se logró implementar los sistemas que ocupan los primeros lugares en las competencias de Semeval, se logró realizar una comparación con 2 sistemas que tienen características particulares, el primero en el que también se realiza un proceso de selección de características (Stem) y el segundo (Hlp@upenn) que cuenta con una representación multimodal que incluye incrustaciones, estos sistemas ocupan el puesto 26 en el año 2016 y número 11 en 2017 respectivamente.

Como se evidencia en la **Tabla 23**, el método propuesto (columna FS) supero los dos sistemas implementados del estado del arte, al obtener mejor precisión en la clasificación, pero el avance más significativo es la notable reducción de características para obtener estos resultados, y que en rendimiento puede significar menores tiempos de preprocesamiento y recursos ya que, Stem redujo de 43 a 28, mucho menos del 50% de las características originales e Hlp@upenn incluía un numero bastante grande de características que consistía en la unión de incrustaciones, representación de frecuencias de trigramas y representación de bolsa de palabras.

	Stem	**Hlp@upenn**	**FS**
Semeval 2013 (F1 %)	63,4	63,1	**64,8**
Semeval 2016 (F1 %)	51,9	49,0	**56,3**

Tabla 23. Comparación de los resultados del método propuesto con los implementados del estado del arte

Lo que también se debe resaltar de la propuesta de esta tesis es que el proceso de clasificación se realizó con algoritmos tradicionales de aprendizaje automático, y los mejores resultados que determinan el estado del arte utilizan redes profundas, las cuales utilizan métodos más complejos y que necesitan hardware y herramientas especializadas, convirtiéndola en una opción para ambientes no tan especializados (sin hardware costoso).

Del grupo mencionado de 22 características más relevantes, solo 7 fueron seleccionadas en todos los experimentos, estas características fueron: 'neg_words_tweet', 'neu_neu_neg', 'pos', 'pos_words', 'pos_words_tweet', 'punct', 'Doc2vecFeature', lo que indica que estas no dependen de un conjunto de datos en particular, se experimentó con solo estas 7 características y en la **Tabla 24** y **Tabla 25** se presentan los resultados:

SemEval 2013 Test	LR		Simple LR		Naive Bayes		SMO		MLP		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Training: SemEval 2013-Train+dev	66,2	65,7	66,8	66,2	58,5	57,1	65,0	64,4	63,4	62,3	91
FS Base Line	**65,4**	**64,8**	**65,5**	**64,8**	**60,0**	**60,0**	**65,1**	**64,6**	**59,1**	**58,8**	22
7 Características	63,6	62,6	63,2	62,0	55,8	55,8	62,6	62,7	55,1	54,7	7
SemEval 2013 Test	JRIP		CART		J48		RANDOM FOREST		SVM		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Training: SemEval 2013-Train+dev	62,2	61,5	61,8	60,0	63,2	62,2	63,0	61,1	55,2	48,8	91
FS Base Line	**60,3**	**58,4**	**61,9**	**60,7**	**63,2**	**62,2**	**61,9**	**59,7**	**55,2**	**48,5**	22
7 Características	59,5	57,2	60,7	60,0	50,0	50,1	58,2	54,1	53,2	45,8	7

Tabla 24. Comparación de las 7 caracteristicas mas relevantes en Semeval 2013

SemEval 2016 Test	LR		Simple LR		Naive Bayes		SMO		MLP		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Base line: SemEval 2016 Train+dev	57,2	56,2	56,6	55,4	46,9	48,2	56,4	55,0	48,3	48,7	91
FS Base Line	**56,7**	**55,5**	**57,6**	**56,3**	**51,0**	**51,1**	55,8	54,1	**52,5**	**52,7**	15
7 Características	56,4	55,1	56,9	55,3	49,8	49,9	**56,1**	**54,3**	51,1	50.1	7
SemEval 2016 Test	JRIP		CART		J48		RANDOM FOREST		SVM		k
	ICC	F1	ICC	F1	ICC	F1	ICC	F1	ICC	F1	
Base line: SemEval 2016 Train+dev	51,0	44,0	54,1	49,1	52,0	50,2	54,6	49,0	49,3	32,7	91
FS Base Line	50,3	38,7	**54,1**	**51,8**	**53,1**	**51,3**	54,7	49,0	**49,4**	**32,8**	15
7 características	**51,8**	**44,5**	52.8	46,7	46,1	46,4	**56,8**	**50.7**	49,4	32,7	7

Tabla 25. Comparación de las 7 caracteristicas mas relevantes en Semeval 2016

6.3 Muestreo aleatorio con reemplazo frente al ponderado

Para evaluar si en el método propuesto era mejor utilizar muestreo ponderado o aleatorio, se realizó un experimento con los dos tipos de muestreo. Los resultados se muestran en la **Tabla 26** y la **Tabla 27**. Las diferencias en muchos casos se establecen en el segundo decimal.

	Logistic Regression		Simple Logistic Regression		Naive Bayes		JRIP		CART	
	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg
Muestreo Ponderado	64,8	64,2	65,4	64,7	**60,0**	**0,6**	60,0	57,9	61,9	60,0
Muestreo Aleatorio	**65,4**	**64,8**	**65,5**	**64,8**	**60,0**	**0,6**	**60,4**	**58,4**	**62,0**	**60,7**
	J48		RANDOM FOREST		SVM		SMO		MULTILAYER PERCEPTRON	
	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg
Muestreo Ponderado	62,8	62,1	60,9	58,2	**56,0**	**49,4**	64,6	64,1	58,6	58,3
Muestreo Aleatorio	**63,2**	**62,2**	**61,9**	**59,7**	55,2	**48,5**	**65,1**	**64,6**	**59,1**	**58,8**
	CARACTERISTICAS SELECCIONADAS						Numero de características seleccionadas		Numero de características originales.	
Muestreo Ponderado	punct, negat_words_tweet, neg, neu, pos, pos_words, neu_neg_neu, negative_emojis, positive_emojis, pol_words_ht, pos_neu_neu, neg_neg_neu, neg_neu_neu, neu_neg_neg, neu_words_ht_sum, neg_words_tweet, neu_words_tweet, neg_words, negat_words, r1, pos_words_tweet, adv_frac, v_frac, neg_words_sum, pos_words_sum, 300Doc2vecFeatures						26		91	
Muestreo Aleatorio	punct, neg_words_tweet, pos_words_tweet, nn_frac, neg, pos, neg_words, r1, negative_emojis, pos_words, neg_words_sum, pos_words_sum, neu_neu_neg, adv_frac, v_frac, neg_neu_neu, positive_emojis, neu_neg_neu, count_uppercase, pos_neu_neu, neu_neu_neu, 300Doc2vecFeatures						22		91	

Tabla 26. Comparación utilizando el dataset Semeval 2013

	Logistic Regression		Simple Logistic Regression		NAIVE BAYES		JRIP		CART	
	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg
Muestreo Ponderado	**56,7**	55,4	56,8	54,4	49,2	49,3	**52,1**	**4,42**	**54,4**	48,5
Muestreo Aleatorio	**56,7**	**55,5**	**57,7**	**56,3**	**51,0**	**51,1**	50,3	38,7	54,1	**51,8**

	J48		RANDOM FOREST		SVM		SMO		MULTILAYER PERCEPTRON	
	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg	CCI	F1 Avg
Muestreo Ponderado	**53,6**	49,2	54,5	47,6	49,4	32,7	**56,9**	**55,0**	49,6	49,4
Muestreo Aleatorio	53,1	**51,3**	**54,8**	**49,3**	**49,4**	**32,8**	55,8	54,1	**52,5**	**52,7**
	CARACTERISTICAS SELECCIONADAS						Numero de características seleccionadas		Numero de características originales.	
Muestreo Ponderado	punct, pos, neg_words_tweet, neu, 300Doc2vecFeatures						5		91	
Muestreo Aleatorio	punct, pos_words_tweet, negat_words_tweet, neg, pos_words, negat_words, neu_neu_neg, neg_words_tweet, pos, neg_words, r1, neu, neu_neg_neu, neu_neu_pos, 300Doc2vecFeatures						15		91	

Tabla 27. Comparación utilizando el dataset Semeval 2016

Se puede observar que en las pruebas con los datasets de 2013 y 2016 en la mayoría de los casos el muestreo aleatorio obtiene mejores resultados en calidad de clasificación y F1, de igual manera, se observa que con el dataset de 2013 el muestreo aleatorio proporciona el menor número de características seleccionadas y en el dataset de 2016 es el muestreo ponderado el que reduce más características, lo que no permite tener en cuenta el número de características seleccionadas como un factor para decidir qué tipo de muestreo escoger. Por lo anterior, se decide agregar el muestreo aleatorio como el tipo de muestreo seleccionado para el método propuesto.

Capítulo 7

7 Conclusiones, recomendaciones y trabajo futuro

7.1 Conclusiones

A pesar de que no fue posible replicar los experimentos con los sistemas que logran los primeros lugares en el estado del arte, debido a que los repositorios no se encontraron completos, no se contaba con todos los recursos que habían utilizado o usaban plataformas que ya no estaban disponibles, se logró realizar un framework de prueba replicando 2 sistemas pertenecientes al estado del arte, estos 2 sistemas tenían métodos distintos y usaban representaciones de las características que son similares parcialmente con características que se utilizaron en el método propuesto. El trabajo para establecer esta línea base, además del framework, permitió complementar información de la descripción y particularidades de los sistemas que ocupan los 3 primeros puestos en Semeval 2017 y se encontró que los mejores resultados en detección de polaridad en Twitter se están logrando con redes neuronales profundas usando LSTM.

Con el objetivo de proponer un nuevo método de selección de características, que proporcione la mayor calidad posible en la detección de polaridad en el análisis de sentimientos en Twitter reduciendo las características de los tweets y el tiempo de preprocesamiento, se planteó un método basado en: 1) el uso de arreglos de cobertura para definir el número de muestras bootstrap y las columnas, atributos o características que se usan en cada muestra, 2) el muestreo aleatorio ponderado de filas (usado en técnicas de bagging) sobre el dataset de entrenamiento para generar cada una de las muestras con el mismo número de filas que el dataset de entrenamiento, 3) tres técnicas de inducción de reglas (C4.5, CART y RIPPER) que se aplican sobre las muestras y de las cuales se obtienen las características que aportan más información en el Análisis de Sentimientos en Twitter.

Es preciso destacar el uso de los arreglos de cobertura en el método propuesto, ya que con ellos se logra cubrir la mayor cantidad posible de interacciones (en nuestras pruebas se empleó de interacción 5) entre las características con el menor esfuerzo posible, en este caso el menor número de casos de prueba o experimentos en comparación con otros enfoques, por ejemplo, con algoritmos metaheurísticos. Con este trabajo se evidencio que los arreglos de cobertura se pueden aplicar al área de selección de características y aprovechar su potencial en el diseño óptimo de experimentos, convirtiéndose esta tesis sino la primera, una de las primeras en usarlos para soportar esta tarea (un área no tradicional de la aplicación de los arreglos de cobertura).

Con el método propuesto se logró reducir entre un 69% y un 83% las características totales sin disminuir la calidad de la clasificación, medida por la exactitud (accuracy) o medida F (F1). Además, en un experimento se logró mejorar la calidad usando dichas métricas, lo que muestra también que la calidad del dataset de entrenamiento tiene relación directa con los resultados del proceso de selección de características propuesto. Al realizar la comparación del método propuesto con los dos algoritmos del estado del arte que se incluyeron en el framework de prueba, se logró observar que el método propuesto obtiene una mayor precisión, en el caso del dataset Semeval 2016 de un 4.4% mayor a la reportada por Stem y de 7.3% por encima de Hlp@upenn y con el dataset Semeval 2013, supero a Stem por 1,4% y a Hlp@upenn por 1.7%. Esta comparación indica que el método de selección de características propuesto mejoró la precisión y supero el framework de prueba.

Como experiencia en el desarrollo de la presente tesis y según lo indagado en el estado del arte, el preprocesamiento es fundamental para crear una adecuada representación de los tweets y con ello obtener una buena calidad en la detección de polaridad y otras tareas relacionadas. Es importante contar con diferentes recursos, entre ellos, los recursos léxicos, los algoritmos para identificar las partes del discurso, las redes neuronales que permitan definir una representación semántica del texto en los tweets, corrector ortográfico para identificar palabras mal escritas y expresiones regulares para encontrar patrones y detectar palabras o reglas semánticas.

Según se investigó, en el estado del arte y se evidenció en este proyecto, las incrustaciones (word or phrase embeddings) son una muy importante representación del tweet que a diferencia de una matriz de términos por frecuencia (modelo espacio vectorial o bolsa de palabras) y n-gramas, mejora la calidad de la clasificación al lograr captar el contexto de los componentes de cada tweet. Los experimentos fueron claros en mostrar que la calidad de la clasificación es mejor cuando se usan incrustaciones Doc2Vec que cuando no se usan, ver **Anexo 3**.

Se evidenció en los resultados, un subconjunto de 7 características las cuales se seleccionaron en todos los experimentos, por lo que encontramos que estas 7 características no dependen del conjunto de datos, y por sí solas brindan buenos resultados en la clasificación e indica que son un recurso muy útil que se debe tener en cuenta en los procesos de análisis de sentimientos. Los recursos léxicos con que se obtienen estas características fueron las listas de palabras positivas y negativas de Bing Liu, los trigramas formados por la polaridad de 3 palabras consecutivas, SentiWordNet, la puntuación y la representación vectorial del tweet.

7.2 Recomendaciones y trabajo futuro

Se espera que los resultados de esta investigación permitan mejorar los resultados de otros sistemas de detección de polaridad del estado del arte, ya que la mayoría de los algoritmos de clasificación son sensibles a la representación de los tweets y

por consiguiente a las características que lo componen. Si las características seleccionadas ya no son irrelevantes, redundantes o ruidosas se verían beneficiados al entrenarse con las características que realmente aportan información valiosa para la clasificación.

En este proyecto se desarrolla un método de selección de características que busca mejorar la precisión en el análisis de sentimientos en Twitter, se logró probar que superaba los algoritmos implementados del framework de prueba, sin embargo, no se lograron reproducir aquellos sistemas que tenían más interés y son aquellos que ocupaban los primeros lugares, esto se dio por que los repositorios compartidos no contenían todos los algoritmos y recursos necesarios para replicar los experimentos. Se recomienda que, si se publica un nuevo método o sistema, la información para replicar los experimentos, sea más exacta, mejor explicada, y así mismo las versiones de los paquetes que se deben usar. Además, que los autores estén más disponibles para responder los correos que se envían solicitando información detallada de lo que se presenta en los artículos o que liberen en forma completa y correcta todo su código fuente.

Como trabajo futuro se espera: 1) evaluar el método de selección de características propuesto en otros contextos diferentes al análisis de sentimientos en Twitter, 2) implementar una red neuronal profunda para el AST usando las características seleccionadas por el método propuesto, 3) usar un método diferente al muestreo por reemplazo de filas o muestreo ponderado que haga posible la selección de las instancias más útiles para el entrenamiento, 4) evaluar covering arrays de mayor nivel de interacción (parámetro de fuerza), 5) seleccionar datos de entrenamiento de mayor calidad y evaluar la calidad de los resultados con el método propuesto, 6) usar un modelo Doc2Vec preentrenado solamente con tweets y 7) utilizar incrustaciones de word2vec generadas con un modelo entrenado solo con tweets e incluir en su generación, la polaridad de la palabra, de tal forma que palabras opuestas se alejen en el espacio vectorial tal como en [94].

Capítulo 8

8 Referencias

[1] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Systems,* vol. 89, pp. 14-46, 2015.

[2] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*, 2015.

[3] Y.-M. Li and T.-Y. Li, "Deriving market intelligence from microblogs," *Decision Support Systems,* vol. 55, pp. 206-217, 2013.

[4] D. Kang and Y. Park, "Review-based measurement of customer satisfaction in mobile service: Sentiment analysis and VIKOR approach," *Expert Systems with Applications,* vol. 41, pp. 1041-1050, 2014.

[5] H. Rui, Y. Liu, and A. Whinston, "Whose and what chatter matters? The effect of tweets on movie sales," *Decision Support Systems,* vol. 55, pp. 863-870, 2013.

[6] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal,* pp. 1093-1113, 2014.

[7] A. Giachanou and F. Crestani, "Like it or not: A survey of Twitter sentiment analysis methods," *ACM Computing Surveys,* vol. 49, p. 28, 2016.

[8] N. F. F. Da Silva, L. F. S. Coletta, and E. R. Hruschka, "A survey and comparative study of tweet sentiment analysis via semi-supervised learning," *ACM Computing Surveys,* vol. 49, p. 15, 2016.

[9] D. Barber, *Bayesian reasoning and machine learning*, 2012.

[10] A. Amolik, N. Jivane, M. Bhandari, and M. Venkatesan, "Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques.," p. 15, 2016.

[11] N. Omar, M. Albared, T. Al-Moslmi, and A. Al-Shabi, "A comparative study of feature selection and machine learning algorithms for arabic sentiment classification," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 8870, pp. 1-7, 2014.

[12] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-basedmethods for sentiment analysis," *Computational Linguistics,* vol. 37, pp. 267-307, 2011.

[13] O. Appel, F. Chiclana, J. Carter, and H. Fujita, "A hybrid approach to the sentiment analysis problem at the sentence level," *Knowledge-Based Systems,* vol. 108, pp. 110-124, 2016.

[14] A. Pak and P. Paroubek, "Twitter based system: Using Twitter for disambiguating sentiment ambiguous adjectives," in *Proceedings of the 5th International Workshop on Semantic Evaluation*, 2010, pp. 436-439.

[15] M. B. Cohen, C. J. Colbourn, and A. C. H. Ling, "Constructing strength three covering arrays with augmented annealing," *Discrete Mathematics,* vol. 308, pp. 2709-2722, 2008.

[16] H. A. George, J. T. Jiménez, and V. H. García, *Verificación de Covering Arrays*: Lambert Academic Publishing, 2010.

[17] Y. Jun, "Backtracking Algorithms and Search Heuristics to Generate Test Suites for Combinatorial Testing," 2006, pp. 385-394.

[18] H. Avila George, "Constructing Covering Arrays using Parallel Computing and Grid Computing," Ph.D., Departamento de Sistemas Informáticos y Computación, Universitad Politécnica de Valencia, Valencia, Spain, 2012.

[19] J. W. Grzymala-Busse, "Rule induction," in *Data mining and knowledge discovery handbook*, ed: Springer, 2009, pp. 249-265.

[20] C. C. Aggarwal, *Data classification: algorithms and applications*: CRC Press, 2014.

[21] J. R. Quinlan, *C4. 5: programs for machine learning*, 2014.

[22] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*, 1984.

[23] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*: Morgan Kaufmann, 2017.

[24] F. Sancho Caparrini. (2018). *Métodos combinados de aprendizaje*. Available: http://www.cs.us.es/~fsancho/?e=106

[25] A. Esuli, F. Sebastiani, and A. M. Fernández, "Distributional correspondence indexing for cross-lingual and cross-domain sentiment classification," *Journal of Artificial Intelligence Research,* vol. 55, pp. 131-163, 2016.

[26] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on human language technology and empirical methods in natural language processing*, 2005, pp. 347-354.

[27] H. Saif, Y. He, and H. Alani, "Semantic Sentiment Analysis of Twitter," pp. 508-524, 2012.

[28] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," pp. 1320-1326, 2010

[29] Y.-J. Tai and H.-Y. Kao, "Automatic Domain-Specific Sentiment Lexicon Generation with Label Propagation," p. 53, 2013,.

[30] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research,* vol. 50, pp. 723-762, 2014.

[31] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," 2009.

[32] E. Kouloumpis, T. Wilson, and J. D. Moore, "Twitter sentiment analysis: The good the bad and the omg!," *Icwsm,* vol. 11, pp. 538-541, 2011.

[33] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 2010, pp. 36-44.

[34] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the workshop on languages in social media*, 2011, pp. 30-38.

[35] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu, "Combining lexicon- based and learning-based methods for twitter sentiment analysis," *HP Laboratories, Technical Report HPL-2011,* vol. 89, 2011.

[36] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 2011, pp. 151-160.

[37] R. Mansour, M. F. A. Hady, E. Hosam, H. Amr, and A. Ashour, "Feature Selection for Twitter Sentiment Analysis: An Experimental Study," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2015, pp. 92-103.

[38] J. M. Cotelo, F. L. Cruz, F. Enríquez, and J. A. Troyano, "Tweet categorization by combining content and structural knowledge," *Information Fusion,* vol. 31, pp. 54-64, 2016.

[39] H. G. Yoon, H. Kim, C. O. Kim, and M. Song, "Opinion polarity detection in Twitter data combining shrinkage regression and topic modeling," *Journal of Informetrics,* vol. 10, pp. 634-644, 2016.

[40] M. A. E. Pérez, M. G. Fayos, and P. Rosso, "El impacto de las emociones en el análisis de la polaridad en textos con lenguaje figurado en Twitter," *Procesamiento del Lenguaje Natural,* vol. 58, pp. 85-92, 2017 2017.

[41] O. Appel, F. Chiclana, J. Carter, and H. Fujita, "Successes and challenges in developing a hybrid approach to sentiment analysis," *Springer International Publishing,* pp. 1176-1188., 2017.

[42] M. Fernández-Gavilanes, J. Juncal-Martínez, S. García-Méndez, E. Costa-Montenegro, and F. J. González-Castaño, "Creating emoji lexica from unsupervised sentiment analysis of their descriptions," *Expert Systems with Applications,* vol. 103, pp. 74-91, 2018.

[43] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "SemEval-2016 task 4: Sentiment analysis in Twitter," in *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016), San Diego, US (forthcoming)*, 2016, pp. 502-518.

[44] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter, and V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in twitter," *Proceedings of SemEval-2015,* pp. 451-463, 2015.

[45] M. Lango, D. Brzezinski, and J. Stefanowski, "PUT at SemEval-2016 Task 4: The ABC of Twitter Sentiment Analysis," *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016), San Diego, US,* pp. 126-132, 2016.

[46] A. Severyn and A. Moschitti, "UNITN: Training deep convolutional neural network for Twitter sentiment classification," 2015, pp. 464-469.

[47] R. Collins, D. May, N. Weinthal, and R. Wicentowski, "SWAT-CMW: Classification of Twitter Emotional Polarity using a Multiple-Classifier Decision Schema and Enhanced Emotion Tagging," *SemEval-2015,* pp. 669-672, 2015.

[48] W. Boag, P. Potash, and A. Rumshisky, "TwitterHawk: A Feature Bucket Approach to Sentiment Analysis," *SemEval-2015,* p. 640, 2015.

[49] Y. Alhessi and R. Wicentowski, "SWATAC: A Sentiment Analyzer using One-Vs-Rest Logistic Regression," *SemEval-2015,* p. 636, 2015.

[50] F. Uzdilli, M. Jaggi, D. Egger, P. Julmy, L. Derczynski, and M. Cieliebak, "Swiss-Chocolate: Combining Flipout Regularization and Random Forests with Artificially Built Subsystems to Boost Text-Classification for Sentiment," *SemEval-2015,* pp. 608-612, 2015.

[51] A. Kumar, V. K. Akella, and A. Ekbal, "IITPSemEval: Sentiment Discovery from 140 Characters," *SemEval-2015,* pp. 601-607, 2015.

[52] P. Basile and N. Novielli, "UNIBA: Sentiment analysis of English tweets combining micro-blogging, lexicon and semantic features," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 595-600.

[53] S. Guha, A. Joshi, and V. Varma, "Sentibase: Sentiment analysis in twitter on a budget," *SemEval-2015,* pp. 590-594, 2015.

[54] M. Hagen, M. Potthast, M. Büchner, and B. Stein, "Webis: An ensemble for twitter sentiment detection," pp. 582-589, 2015.

[55] F. Pla, M. Giménez, and L.-F. Hurtado, "ELiRF: A Support Vector Machine Approach for Sentiment Analysis Tasks in Twitter at SemEval-2015," pp. 1-18, 2015.

[56] H. Hamdan, P. Bellot, and F. Bechet, "lsislif: Feature extraction and label weighting for sentiment analysis in twitter," in *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015, pp. 568-573.

[57] Z. Zhang, G. Wu, and M. Lan, "Ecnu: Multi-level sentiment analysis on twitter using traditional linguistic features and word embedding features," *Proceedings of SemEval-2015,* pp. 561-567, 2015.

[58] E. Castillo, O. Cervantes, D. Vilarino, D. Báez, and A. Sánchez, "Udlap: sentiment analysis using a graph based representation," *SemEval-2015,* p. 556, 2015.

[59] M. Nabil, A. Atyia, and M. Aly, "CUFE at SemEval-2016 Task 4: A gated recurrent model for sentiment classification," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016, pp. 52-57.

[60] G. Da San Martino, W. Gao, and F. Sebastiani, "QCRI at SemEval-2016 Task 4: Probabilistic methods for binary and ordinal quantification," *Proceedings of SemEval-2015,* pp. 58-63, 2016.

[61] S. Räbigera, M. Kazmia, Y. Saygına, P. Schüllerb, and M. Spiliopoulouc, "SteM at SemEval-2016 Task 4: Applying active learning to improve sentiment classification," *Proceedings of SemEval,* pp. 64-70, 2016.

[62] Z. Zhang, C. Zhang, F. Wu, D.-Y. Huang, W. Lin, and M. Dong, "I2RNTU at SemEval-2016 Task 4: Classifier fusion for polarity classification in Twitter," *Proceedings of SemEval,* pp. 71-78, 2016.

[63] D. Vilaresa, Y. Dovala, M. A. Alonsoa, and C. Gómez-Rodrıgueza, "LYS at SemEval-2016 Task 4: Exploiting neural activation values for Twitter sentiment classification and quantification," *Proceedings of SemEval,* pp. 79-84, 2016.

[64] G. Balikas and M.-R. Amini, "TwiSE at SemEval-2016 Task 4: Twitter Sentiment Classification," *arXiv preprint arXiv:1606.04351,* 2016.

[65] A. Esuli, "ISTI-CNR at SemEval-2016 Task 4: Quantification on an ordinal scale," *Proceedings of SemEval,* pp. 92-95, 2016.

[66] S. Giorgis, A. Rousas, J. Pavlopoulos, P. Malakasiotis, and I. Androutsopoulos, "aueb. twitter. sentiment at SemEval-2016 Task 4: A weighted ensemble of SVMs for Twitter sentiment analysis," *Proceedings of SemEval,* pp. 96-99, 2016.

[67] V. Yadav, "thecerealkiller at SemEval-2016 Task 4: Deep learning based system for classifying sentiment of tweets on two point scale," *Proceedings of SemEval,* pp. 100-102, 2016.

[68] B. E. Jahren, V. Fredriksen, B. Gambäck, and L. Bungum, "NTNUSentEval at SemEval-2016 Task 4: Combining general classifiers for fast Twitter sentiment analysis," *Proceedings of SemEval,* pp. 103-108, 2016.

[69] J. Juncal Martınez, T. Alvarez López, M. Fernández Gavilanes, E. Costa Montenegro, and F. J. González Castano, "GTI at SemEval-2016 Task 4: Training a naive Bayes classifier using features of an unsupervised system," *Proceedings of SemEval,* pp. 115-119, 2016.

[70] S. Du, Z. Xi, and T. China, "Aicyber at SemEval-2016 Task 4: i-vector based sentence representation," pp. 120-125, 2016.

[71] M. Lango, D. Brzezinski, and J. Stefanowski, "PUT at SemEval-2016 Task 4: The ABC of Twitter sentiment analysis," *Proceedings of SemEval,* pp. 126-132, 2016.

[72] V. Cozza and M. Petrocchi, "MIB at SemEval-2016 Task 4a: Exploiting lexicon-based features for sentiment analysis in Twitter," *Proceedings of SemEval,* pp. 133-138, 2016.

[73] H. Gao and T. Oates, "MDSENT at SemEval-2016 Task 4: A Supervised System for Message Polarity Classification," *Proceedings of SemEval,* pp. 139-144, 2016.

[74] H. Gómez-Adorno, G. Sidorov, A. J. de Dios Bátiz, D. Vilarino, and D. Pinto, "CICBUAPnlp at SemEval-2016 Task 4-A: Discovering Twitter Polarity using Enhanced Embeddings," *Proceedings of SemEval,* pp. 145-148, 2016.

[75] E. Palogiannidi, A. Kolovou, F. Christopoulou, F. Kokkinos, E. Iosif, N. Malandrakis*, et al.*, "Tweester at SemEval-2016 Task 4: Sentiment analysis in Twitter using semantic-affective model adaptation," *Proceedings of SemEval,* pp. 155-163, 2016.

[76] O. Abdelwahab and A. Elmaghraby, "UofL at SemEval-2016 Task 4: Multi domain word2vec for Twitter sentiment classification," *Proceedings of SemEval,* pp. 164-170, 2016.

[77] S. Ruder, P. Ghaffari, and J. G. Breslin, "INSIGHT-1 at SemEval-2016 Task 4: Convolutional Neural Networks for Sentiment Classification and Quantification," *arXiv preprint arXiv:1609.02746,* 2016.

[78] X. Xu, H. Liang, and T. Baldwin, "UNIMELB at SemEval-2016 Tasks 4A and 4B: An Ensemble of Neural Networks and a Word2Vec Based Model for Sentiment Classification," *Proceedings of SemEval,* pp. 183-189, 2016.

[79] A. Balahur, "OPAL at SemEval-2016 Task 4: the Challenge of Porting a Sentiment Analysis System to the" Real" World," *Proceedings of SemEval,* pp. 262-265, 2016.

[80] H. Hamdan, "SentiSys at SemEval-2016 Task 4: Feature-based system for sentiment analysis in Twitter," *Proceedings of SemEval,* pp. 190-197, 2016.

[81] V. Martínez, F. Pla, and L.-F. Hurtado, "DSIC ELIRF at SemEval 2016 Task 4 Message Polarity Classification inTwitter using a Support Vector Machine Approach," *Proceedings of SemEval-2016,* pp. pages 198–201, 2016.

[82] M. Rouvier and B. Favre, "SENSEI-LIF at SemEval-2016 Task 4: Polarity embedding fusion for robust sentiment analysis," *Proceedings of SemEval,* pp. 202-208, 2016.

[83] A. Sarker and G. Gonzalez, "DIEGOLab16 at SemEval-2016 Task 4: Sentiment analysis in Twitter using centroids, clusters, and sentiment lexicons," *Proceedings of SemEval,* pp. 209-214, 2016.

[84] G. Briones, K. Amarasinghe, and B. T. McInnes, "VCU-TSA at Semeval-2016 Task 4: Sentiment Analysis in Twitter," *Proceedings of SemEval,* pp. 215-219, 2016.

[85] G. Attardi and D. Sartiano, "UniPI at SemEval-2016 Task 4: Convolutional neural networks for sentiment classification," *Proceedings of SemEval,* pp. 220-224, 2016.

[86] J. Friedrichs, "IIP at SemEval-2016 Task 4: Prioritizing classes in ensemble classification for sentiment analysis of tweets," *Proceedings of SemEval,* pp. 225-229, 2016.

[87] S. Amir, R. F. Astudillo, W. Ling, M. J. Silva, I. Trancoso, and R. A. Redol, "INESC-ID at SemEval-2016 Task 4-A: Reducing the Problem of Out-of-Embedding Words," *Proceedings of SemEval,* pp. 238-242, 2016.

[88] C. Florean, O. Bejenaru, E. Apostol, O. Ciobanu, A. Iftene, and D. Trandabăţ, "SentimentalITsts at SemEval-2016 Task 4: building a Twitter sentiment analyzer in your backyard," *Proceedings of SemEval,* pp. 243-246, 2016.

[89] C.-C. Ciubotariu, M.-V. Hrişca, M. Gliga, D. Darabană, D. Trandabăţ, and A. Iftene, "Minions at SemEval-2016 Task 4: or how to build a sentiment analyzer using off-the-shelf resources?," *Proceedings of SemEval,* pp. 247-250, 2016.

[90] Y. Zhou, Z. Zhang, and M. Lan, "ECNU at SemEval-2016 Task 4: An empirical investigation of traditional NLP features and word embedding features for sentence-level and topic-level sentiment analysis in Twitter," *Proceedings of SemEval,* pp. 256-261, 2016.

[91] J. Deriu, M. Gonzenbach, F. Uzdilli, A. Lucchi, V. De Luca, and M. Jaggi, "SwissCheese at SemEval-2016 Task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision," *Proceedings of SemEval,* pp. 1124-1128, 2016.

[92] C. Baziotis, N. Pelekis, and C. Doulkeridis, "Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 747-754.

[93] M. Cliche, "BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs," *arXiv preprint arXiv:1704.06125,* pp. 1480-1489, 2017.

[94] M. Rouvier, "LIA at SemEval-2017 Task 4: An Ensemble of Neural Networks for Sentiment Classification," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 760-765.

[95] M. Trupthi, S. Pabboju, and G. Narasimha, "Improved feature extraction and classification—Sentiment analysis," in *Advances in Human Machine Interaction (HMI), 2016 International Conference on*, 2016, pp. 1-6.

[96] B. Agarwal and N. Mittal, "Prominent feature extraction for review analysis: an empirical study," *Journal of Experimental & Theoretical Artificial Intelligence,* vol. 28, pp. 485-498, 2016.

[97] F. Luo, C. Li, and Z. Cao, "Affective-feature-based sentiment analysis using SVM classifier," in *Computer Supported Cooperative Work in Design (CSCWD), 2016 IEEE 20th International Conference on*, 2016, pp. 276-281.

[98] Z. Jianqiang, "Pre-processing Boosting Twitter Sentiment Analysis?," in *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*, 2015, pp. 748-753.

[99] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," in *Information, Intelligence, Systems & Applications (IISA), 2016 7th International Conference on*, 2016, pp. 1-5.

[100] P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson, "Semeval-2013 task 2: Sentiment analysis in Twitter," *Atlanta, Georgia, USA,* vol. 312, 2013.

[101] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 502-518.

[102] A. Sarker and G. Gonzalez, "HLP $@ $ UPenn at SemEval-2017 Task 4A: A simple, self-optimizing text classification system combining dense and sparse vectors," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 640-643.

[103] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 168-177.

[104] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets," *arXiv preprint arXiv:1308.6242,* 2013.

[105] S. M. Mohammad, "# Emotional tweets," in *Association for Computational Linguistics*, 2012, pp. 246-255.

[106] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit.," in *ACL (System Demonstrations)*, 2014, pp. 55-60.

[107] S. Baccianella, A. Esuli, and F. Sebastiani, "SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining," in *LREC*, 2010, pp. 2200-2204.

[108] V. Hatzivassiloglou and K. R. McKeown, "Towards the automatic identification of adjectival scales: Clustering adjectives according to meaning," in *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, 1993, pp. 172-182.

[109] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Association for Computational Linguistics*, 1997, pp. 174-181.

[110] J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke, "Using WordNet to Measure Semantic Orientations of Adjectives.," in *LREC*, 2004, pp. 1115-1118.

[111] J. Wiebe, "Learning subjective adjectives from corpora," vol. 20, p. 0, 2000.

[112] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International Conference on Machine Learning*, 2014, pp. 1188-1196.

[113] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111-3119.

[114] S. Huang. (2018). *Word2Vec and FastText Word Embedding with Gensim*. Available: https://towardsdatascience.com/word-embedding-with-word2vec-and-fasttext-a209c1d3e12c

[115] A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vectors," *arXiv preprint arXiv:1507.07998,* pp. 1-8, 2015.

[116] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," *arXiv preprint arXiv:1607.05368,* 2016.

[117] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010.