

Algoritmo de detección de polaridad en análisis de sentimiento en Twitter basado en reglas de lógica difusa



Trabajo de Grado

Jason Paul Anturi Martínez
José Luis Paz Realpe

Director: MSc. Jimena Adriana Timaná Peña
Co-Director: PhD. Carlos Alberto Cobos Lozada

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo de I+D en Tecnologías de la Información
Línea de Investigación: Sistemas Inteligentes
Popayán, Diciembre de 2019

Agradecimientos

El presente trabajo investigativo lo dedicamos principalmente a Dios, por ser el inspirador y darnos fuerza para continuar en este proceso de obtener una de las tantas metas trazadas en nuestras vidas.

A nuestros padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes hemos logrado llegar hasta aquí y convertirnos en lo que somos.

Muchos agradecimientos a la Universidad del Cauca, a toda la Facultad de Ingeniería Electrónica y Telecomunicaciones, a mis profesores en especial a la MSc. Jimena Adriana Timaná Peña y el PhD. Carlos Alberto Cobos Lozada quienes con la enseñanza de sus valiosos conocimientos hicieron que pueda crecer día a día como profesional, gracias a cada uno de ustedes por su paciencia, dedicación, apoyo incondicional y amistad.

A todas las personas que nos han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que nos abrieron las puertas y compartieron sus conocimientos

Tabla de contenido

Lista de figuras	5
Lista de ecuaciones.....	6
Lista de tablas	7
Capítulo 1	9
Introducción.....	9
1.1 Planteamiento del problema	10
1.2 Objetivos	12
1.3 Aportes de investigación	12
1.4 Estructura del documento	13
Capítulo 2.....	15
Revisión sistemática.....	15
2.1 Revisión sistemática literatura científica	15
2.1.1 Preguntas de investigación.....	15
2.1.2 Proceso de búsqueda	16
2.1.3 Selección de artículos.....	18
2.1.4 Resultados del proceso de búsqueda.....	19
2.2 Proceso de desarrollo de la revisión sistemática (literatura gris)	31
2.2.1 Preguntas de investigación.....	32
2.2.2 Proceso de búsqueda	32
2.2.3 Selección de propuestas.....	34
2.2.4 Resultados del proceso de búsqueda de la literatura gris	35
2.3 Caracterización	42
2.3.1 Selección de atributos.....	44
2.3.2 Comparación de propuestas seleccionadas	45
Capítulo 3.....	49
Algoritmo	49
3.1 Exploración	50
3.1.1 Historias de usuario	50
3.1.2 Lógica Difusa	53
3.2 Planificación	56
3.3. Iteración	56
3.3.1 Pseudocódigo del algoritmo.....	56

3.3.2 Arquitectura	59
3.3.3 Implementación de historias de usuario.....	61
3.3.4 Pruebas de aceptación	65
3.3.5 Prototipo software	69
3.4 Producción	70
3.5 Mantenimiento.....	70
3.5.1 Optimización y refinamiento de valores numéricos de entrada.....	71
3.5.2 Pseudocódigo de optimización	71
3.6 Ciclo de muerte	73
3.6.1 Terminación del algoritmo.....	73
3.6.2 Terminación del prototipo software	73
3.6.3 Funcionamiento del prototipo software	73
Capítulo 4.....	78
Resultados	78
4.1 Evaluación de resultados	78
4.2 Comparación de Resultados	80
4.2.1 Test estadísticos no paramétricos	83
Capítulo 5.....	85
Conclusiones, experiencias y trabajo futuro.....	85
Referencias	88

Anexos

- A. Artículos publicados relacionados a la investigación
- B. Manual de usuario del prototipo software
- C. Código fuente del algoritmo y prototipo software

Lista de figuras

Gráfico 1. Conjunto de reglas de inferencia	54
Gráfico 2. Rangos de las reglas de lógica difusa	55
Gráfico 3. Ejemplo del flujo del algoritmo	59
Gráfico 4. Arquitectura del algoritmo	60
Gráfico 5. Cálculo de polaridades	60
Gráfico 6. Cálculo de polaridad final.....	61
Gráfico 7. Identificación de emoticonos.....	62
Gráfico 8. Identificación de hashtag	63
Gráfico 9. Identificación de contracciones	64
Gráfico 10. Menú principal	69
Gráfico 11. Menú configuraciones.....	69
Gráfico 12. Menú de medidas de precisión	69
Gráfico 13. Menú de selección de dataset	70
Gráfico 14. Menú de medidas de precisión	70
Gráfico 15. Paso 1. Ejecución opción 1.....	73
Gráfico 16. Paso 2. Ejecución opción 1.....	74
Gráfico 17. Paso 3. Ejecución opción 1.....	74
Gráfico 18. Paso 4. Ejecución opción 1.....	74
Gráfico 19. Paso 1. Ejecución opción 2.....	75
Gráfico 20. Paso 2. Ejecución opción 2.....	75
Gráfico 21. Paso 3. Ejecución opción 2.....	76
Gráfico 22. Paso 4. Ejecución opción 2.....	76
Gráfico 23. Paso 1. Ejecución opción 3.....	77

Lista de ecuaciones

Ecuación 1. Suma de criterios de calidad	18
Ecuación 2. Cálculo del puntaje de calidad 1	24
Ecuación 3. Cálculo del puntaje de calidad 2	24
Ecuación 4. Cálculo del puntaje de calidad para literatura gris	38
Ecuación 5. Cálculo de polaridad	64
Ecuación 6. Conversión de escalas	79
Ecuación 7. Resultado de precisión	79
Ecuación 8. Diagonal de matriz de confusión.....	80
Ecuación 9. Total de registros en la matriz de confusión	80
Ecuación 10. Exactitud.....	81
Ecuación 11. Medida F.....	81
Ecuación 12. Test de Friedman.....	84

Lista de tablas

Tabla 1. Definición de palabras claves	16
Tabla 2. Palabras categorizadas	17
Tabla 3. Criterios de selección	18
Tabla 4. Resultados cadena de búsqueda literatura científica	19
Tabla 5. Resultados artículos potenciales	19
Tabla 6. Propuestas primarias.....	22
Tabla 7. Evaluación criterios de calidad	24
Tabla 8. Clasificadores de polaridad por tipos.....	26
Tabla 9. Lista de propuestas de clasificadores de polaridad.....	29
Tabla 10. Clasificación por técnicas de extracción	31
Tabla 11. Palabras claves revisión sistemática literatura gris	33
Tabla 12. Categorización de palabras claves.....	33
Tabla 13. Criterios de selección de literatura gris.....	34
Tabla 14. Resultados cadena de búsqueda literatura gris	35
Tabla 15. Propuestas potenciales	37
Tabla 16. Aplicación de criterios de inclusión y exclusión	38
Tabla 17. Aplicación criterios de calidad	39
Tabla 18. Propuestas clasificadas.....	40
Tabla 19. Clasificación por tipos de datos	42
Tabla 20. Propuestas seleccionadas.....	44
Tabla 21. Lista de atributos	45
Tabla 22. Caracterización de las propuestas seleccionadas.....	45
Tabla 23. Tamaño de mensaje en caracteres	46
Tabla 24. Tipos de emoticonos	46
Tabla 25. Lista de precisiones.....	47
Tabla 26. Historias de usuarios	51
Tabla 27. Criterios de Aceptación de las HU.....	53
Tabla 28. Pseudocódigo del algoritmo	57
Tabla 29. Caso de prueba 1	66
Tabla 30. Caso de prueba 2.....	67
Tabla 31. Caso de prueba 3.....	68
Tabla 32. Caso de prueba 4.....	68
Tabla 33. Pseudocódigo de optimización	71
Tabla 34. Tipos de escalas por clasificador	79
Tabla 35. Matriz de confusión	80
Tabla 36. Precisión de polaridad de cada clasificador (Accuracy)	81
Tabla 37. Precisión de polaridad de cada clasificador (F-Score)	82
Tabla 38. Resultados precisión de polaridad Hill-Climbing	82

Tabla 39. Resultados competitivos dataset tweet_emevaltest	83
Tabla 40. Rangos del test de Friedman	83

Capítulo 1

Introducción

Realizar el proceso de análisis de sentimientos en mensajes publicados por los usuarios en sitios de Microblogging como Twitter, puede ofrecer grandes posibilidades [1], como comprender la tendencia del sentimiento social hacia una marca, persona o empresa en particular, también para filtrar o eliminar mensajes ofensivos publicados por los usuarios.

Sin embargo, la tarea de identificar y clasificar correctamente la polaridad¹ de un mensaje como positivo, negativo o neutro, es una tarea compleja que aborda entre otros aspectos el manejo de la sintaxis y la semántica. En la sintaxis² se debe tener en cuenta la forma cómo está escrito el mensaje, es decir, la estructura gramatical usada y si ésta incluye emoticonos, contracciones y elongaciones en las palabras. Por otra parte, la semántica³ se relaciona a la interpretación del mensaje, muchas veces ésta es subjetiva, debido a la cosmovisión de cada individuo que a la vez es afectada por factores culturales y modismos característicos de una región.

Acerca del contenido y la estructura de los mensajes de Twitter conocidos como tuits⁴, se suelen encontrar distintas formas de expresión usadas por los usuarios en sus mensajes, por lo que es frecuente encontrar palabras que cambian de significado cuando son combinadas con emoticones o acrónimos, expresiones escritas en forma de pregunta, con sarcasmo o ironía y hasta expresiones que cambian de significado según la localización geográfica. Por lo tanto, interpretar y analizar este tipo de mensajes resulta una tarea desafiante [2].

Investigaciones recientes que incluyen algoritmos que usan técnicas de inteligencia artificial y aprendizaje automático para el análisis de sentimientos en texto han sido usados para la clasificación de la polaridad de un mensaje [3].

En el caso particular de Twitter, la compañía ha implementado una herramienta que permite restringir publicaciones con contenido que sugiere algún comportamiento o sentimiento que pueda incumplir con sus reglas de publicación, limitando así

¹ Polaridad: Concepto semántico que describe la orientación sentimental (positiva, negativa y neutra) extraída del contexto de una expresión hablada o escrita.

² Sintaxis: Gram. Parte de la gramática que estudia el modo en que se combinan las palabras y los grupos que estas forman para expresar significados, así como las relaciones que se establecen entre todas esas unidades.

³ Semántica: Ling. Disciplina que estudia el significado de las unidades lingüísticas y de sus combinaciones.

⁴ Tuit: m. Mensaje digital que se envía a través de la red social Twitter® y que no puede rebasar un número limitado de caracteres.

de forma temporal ciertas funciones de la cuenta o bloquearla de forma indefinida [4].

En este trabajo investigativo se identifican y se describen los mejores clasificadores de polaridad en Twitter y en análisis de sentimientos con respecto a la medida de precisión (F1) y una serie de criterios de calidad que fueron definidos siguiendo los lineamientos de Kitchenham [5] a partir de una exhaustiva revisión sistemática. Además, se presenta una propuesta de un clasificador de polaridad en análisis de sentimiento en Twitter basado en reglas de lógica difusa el cual ha sido comparado con los mejores clasificadores mencionados en la revisión.

1.1 Planteamiento del problema

Diariamente se publican más de 700 millones de mensajes en el sitio de Microblogging Twitter [6], a estos mensajes digitales se les conoce como tuits. Los cuales, están limitados por un número fijo de caracteres en cada publicación. Al analizar los mensajes publicados es posible identificar un sentimiento que es asociado a una polaridad, la cual brinda información sobre el estado sentimental del usuario cuando publica un mensaje. La polaridad permite clasificar el mensaje como positivo, negativo o neutro [7], en donde se considera positivo al mensaje que contiene palabras que afectan de forma benévola la autoestima del lector, neutro al mensaje que no expresa emoción alguna o se encuentra en equilibrio entre sentimientos negativos y positivos y se considera mensaje negativo el que contiene palabras que atentan generalmente contra un sujeto sea una persona, marca, animal o cosa.

A causa de los mensajes negativos que atentan contra la dignidad e integridad de los usuarios, solo algunas redes sociales y sitios de Microblogging han desarrollado distintas herramientas que permiten calcular y definir la polaridad en análisis de sentimientos para publicaciones que involucran tanto documentos de texto como archivos multimedia que contienen audio, imagen o vídeo. Sin embargo, estas herramientas desarrolladas no implementan técnicas avanzadas para la correcta identificación y eliminación automática de sentimientos negativos, que llevan consigo tendencias terroristas, suicidas, sexistas, crueles e intimidantes, las cuales recaen con gran impacto sobre niños y adolescentes [8]. Es por ello, que la carencia de herramientas que permitan la eliminación automática de publicaciones con polaridad negativa en sitios de Microblogging como Twitter, conlleva a que sólo sea posible eliminarlas manualmente por los usuarios o por organizaciones autorizadas, a las cuales se les notifica mediante un mecanismo manual de denuncia de contenido ejecutado por los usuarios y les permite a las organizaciones analizar el contenido denunciado para determinar si es pertinente mantenerlo o eliminarlo

definitivamente [9]. Si el contenido inapropiado no es denunciado, entonces no será tenido en cuenta para ser revisado, ya que al no existir al menos una demanda para este tipo de contenido no se tomará en consideración para ser eliminado.

En el caso particular de Twitter, la compañía ha implementado una herramienta que permite restringir publicaciones con contenido inapropiado, pero sin que el propietario de la cuenta se entere que su tuit fue censurado. Esto ha desatado controversias debido a la falta de certeza para identificar contenidos inapropiados, lo que ha conllevado a clausurar o penalizar cuentas por un tiempo indefinido [10]. Además, las medidas implementadas por Twitter podrían no ser las más adecuadas para la eliminación de contenido detectado como inapropiado en una sociedad que es totalmente legítimo, ya que basta tan solo con cambiar la ciudad de origen de la cuenta para que cualquier persona incluyendo niños y adolescentes les sea permitido leerlo [9].

Acerca del contenido y la estructura de los mensajes de Twitter, en cada tuit se suelen encontrar distintas formas de cómo los usuarios se expresan en los mensajes, por lo que es frecuente encontrar palabras que cambian de significado cuando son combinadas con emoticonos o acrónimos, expresiones escritas en forma de pregunta, con sarcasmo o ironía y hasta expresiones que cambian de significado según la localización geográfica [11]. Por lo tanto, interpretar y analizar este tipo de mensajes para una posterior eliminación automática resulta una tarea compleja, aún más cuando el porcentaje de la medida F (F1) de clasificación de polaridad en análisis de sentimiento, utilizada como medida de precisión, sólo alcanza como máximo el 81,02% para mensajes de Twitter [12, 13].

Considerando que la medida F de clasificación de polaridad en análisis de sentimiento en general no es alta, en muchas ocasiones esta situación afecta negativamente el funcionamiento de las diferentes herramientas de restricción y eliminación automática de contenido, restringiendo contenido válido y aceptado en la comunidad. Además, debido a la mala clasificación de la polaridad se han eliminado publicaciones que no deberían eliminarse y en casos más críticos, se han aplicado sanciones e incluso suspensiones a cuentas con publicaciones consideradas inapropiadas [12]. La falta de precisión de la medida F en la clasificación de sentimientos con respecto a la polaridad ha llevado a la siguiente pregunta de investigación, ¿Cuál es la variación de la calidad de la clasificación de polaridad en análisis de sentimientos de un tuit si se implementa un algoritmo que integra reglas de lógica difusa y utiliza las polaridades entregadas por los mejores clasificadores de sentimiento del estado del arte?

Teniendo en cuenta lo anterior, en esta investigación se presenta una revisión sistemática, una caracterización y un algoritmo con el fin de conocer los mejores

clasificadores de polaridad en análisis de sentimiento con el fin de mejorar los resultados de detección de la medida de precisión F-Score.

1.2 Objetivos

1.2.1 Objetivo general

- Proponer un algoritmo que determine la polaridad de un tuit mediante la aplicación de reglas de lógica difusa a partir de los resultados entregados por los mejores clasificadores de análisis de sentimientos buscando mejorar la medida F (F1) de la clasificación de polaridad.

1.2.2 Objetivos específicos (OE)

- OE.1 Definir los mejores clasificadores de análisis de sentimientos basados en la medida F (F1) a través de una revisión sistemática para realizar la caracterización de herramientas, métodos, servicios y librerías usadas en la clasificación de polaridad en Twitter.
- OE.2 Modelar e implementar un algoritmo que a partir de la polaridad (positiva, negativa o neutra) entregada por los mejores clasificadores seleccionados del estado del arte, determine la polaridad resultante deducida mediante un conjunto de reglas de lógica difusa para intentar mejorar la medida F en la clasificación de polaridad.
- OE.3 Evaluar y comparar la variación de la polaridad obtenida del algoritmo propuesto frente a las polaridades entregadas por los mejores clasificadores del estado del arte, para determinar si hay mejora en la medida F de clasificación de polaridad usando los datasets de SentiStrength_twitter, vader_twitter y los test estadísticos no paramétricos de Friedman y Wilcoxon.

1.3 Aportes de investigación

Desde una perspectiva investigativa, la contribución de este trabajo de grado está dirigida a la propuesta de un prototipo software de código abierto que integra un algoritmo mediante reglas de lógica difusa, además de la caracterización de herramientas, métodos, servicios y librerías para definir los mejores clasificadores de polaridad en análisis de sentimientos en Twitter según la medida F. Teniendo en

cuenta que en la búsqueda realizada a la fecha, en varias bases de datos indexadas no se encontraron artículos publicados con una caracterización de herramientas métodos, servicios y librerías usadas para la clasificación de polaridad, se espera que ésta pueda ser utilizada en futuras investigaciones relacionadas con clasificación de sentimiento en Twitter y sirva como referente para agilizar y simplificar el trabajo investigativo tanto en tiempo como en esfuerzo.

Desde la perspectiva de innovación se desarrolló un algoritmo (integrado en un prototipo software) que mejoró la medida de precisión F (*F-Score*) de la clasificación de polaridad mediante la integración de reglas de lógica difusa a partir de las polaridades obtenidas por los mejores clasificadores de sentimiento, debido que hasta la fecha es necesario modificar el código fuente del algoritmo original o implementar nuevas técnicas de preprocesamiento de datos en el proceso de clasificación de polaridad en análisis de sentimiento si se quiere mejorar la medida de precisión de la clasificación de polaridad. Tanto el diseño del algoritmo como el prototipo software están disponibles para la comunidad académica y científica, de modo que pueden ser utilizados en futuras investigaciones relacionadas con análisis de sentimientos en sitios de Microblogging y lógica difusa. El algoritmo propuesto mejoró la precisión de polaridad usando como medidas *F-Score* y *Accuracy*, lo que hace viable que sea pueda ser integrado en una herramienta, método, algoritmo o librería que realice eliminación automática de texto con contenido inapropiado y con un margen de error reducido con respecto a los resultados que muestran los algoritmos existentes en el estado del arte con respecto a la precisión de polaridad.

1.4 Estructura del documento

A continuación, se presenta la estructura del documento de la monografía:

Capítulo 1 Introducción. En este capítulo se presenta la problemática que motivó a la realización de este proyecto de investigación. Adicionalmente, se presentan el planteamiento del problema, los objetivos y los aportes del proyecto.

Capítulo 2 Revisión sistemática. En este capítulo se presenta una revisión sistemática para clasificadores de polaridad en análisis de sentimientos en la literatura científica y en la literatura gris guiada por los lineamientos de Kitchenham, además se presenta la caracterización de los clasificadores de polaridad que resultaron a partir de la revisión sistemática mencionada.

Capítulo 3 Algoritmo. En este capítulo se presenta el algoritmo propuesto, las etapas y los procesos utilizados para la implementación del algoritmo que estuvo guiado por la estructura de la metodología XP [13].

Capítulo 4 Resultados. En este capítulo se presenta la comparación de resultados entre los clasificadores de sentimientos seleccionados y el algoritmo propuesto siguiendo los test estadísticos no paramétricos de Friedman y Wilcoxon para comprobar si realmente existen diferencias entre las medidas de precisión resultantes de los clasificadores de sentimientos seleccionados en la caracterización y el algoritmo propuesto.

Capítulo 5 Conclusiones y Trabajo Futuro. En este capítulo se presentan las conclusiones del trabajo realizado y trabajo futuro.

Anexos. Los anexos a este documento vienen divididos en tres partes, A, B y C. El Anexo A contiene la descripción y el nombre de los artículos publicados en revistas internacionales, el Anexo B contiene el manual de usuario del prototipo software y el Anexo C contiene la descripción del sitio web en donde se encuentra alojado el repositorio del código fuente.

Capítulo 2

Revisión sistemática

En esta sección se presenta el proceso que se llevó a cabo para la realización de la revisión sistemática de la literatura científica y de algunos documentos de literatura gris.

2.1 Revisión sistemática literatura científica

2.1.1 Preguntas de investigación

Se quiere compilar las propuestas científicas que abarquen la problemática del trabajo investigativo, para obtener una idea general del conocimiento requerido se ha planteado la siguiente pregunta de investigación:

“¿Qué estudios existen en la literatura sobre clasificación de polaridad en análisis de sentimiento?”

Además, para lograr que se cubra todo el estudio investigativo se han planteado una serie de preguntas de investigación, las cuales son:

- PI1. ¿Cuál es la clasificación de las propuestas existentes del estado del arte en clasificación de sentimientos?
- PI2. ¿Cuáles algoritmos o herramientas son utilizados en el proceso de clasificación de polaridad de las propuestas existentes?
- PI3. ¿Cuáles propuestas requieren, o no requieren, el preprocesamiento de datos en la clasificación de polaridad?
- PI4. ¿Cuáles propuestas utilizan técnicas de lógica difusa?

Con respecto a PI1, se quiere tener un panorama de las categorías que abarcan todas las propuestas existentes sobre clasificación de sentimientos, el tipo de categoría depende directamente del tipo de clasificador de sentimiento que se tenga con respecto a la naturaleza de la propuesta, es decir, si este es un algoritmo, método, servicio (API) o herramienta. Con respecto a PI2, se quiere obtener una lista de los algoritmos o herramientas que se han utilizado en el desarrollo del proceso de clasificación de las propuestas estudiadas. Con respecto a PI3, se desea identificar cuáles propuestas seleccionadas en el estado del arte utilizan técnicas de preprocesamiento de datos. Con respecto a PI4, se quiere analizar e identificar

cuáles propuestas seleccionadas del estado del arte utilizan técnicas de lógica difusa en su proceso de análisis de polaridad.

2.1.2 Proceso de búsqueda

En este proceso se define un protocolo de revisión de palabras claves, que incluye la definición de palabras clave, cadenas de búsqueda y criterios de selección.

2.1.2.1 Palabras clave

Las palabras claves vienen dadas en relación al tema de investigación que se desea abordar. En la Tabla 1 se observan las palabras claves que son de utilidad para construir la cadena de búsqueda.

Contexto	Palabras claves
Área	“sentiment analysis”, “opinion mining”
Criterio de clasificación	“polarity”
Clasificadores de sentimiento	“sentiment classifier”, “method”, “service”, “library”, “algorithm”, “framework”, “tool”
Plataforma social de estudio	“microblogging”, “twitter”

Tabla 1. Definición de palabras claves

En el proceso de construcción de la cadena de búsqueda se tuvieron en cuenta ciertos aspectos a saber:

- Las palabras clave categorizadas en el contexto “Área” fueron el foco principal para la búsqueda, es decir, los estudios que en términos generales abarcan todo el trabajo investigativo.
- Las palabras clave que están dentro de la categoría “Área” son utilizadas como filtros o términos alternos, lo que significa obtener resultados más exactos.
- Los sinónimos o palabras clave de la misma categoría se separan con OR y las palabras claves de distinta categoría se separan con AND.

En la Tabla 2, se definen los términos principales y alternativos para definir la cadena de búsqueda.

Términos principales	Términos alternativos
Análisis de sentimiento	“sentiment analysis”, “opinion mining”
Clasificadores de sentimientos	“sentiment classifier”, “method”, “service”, “library”, “algorithm”, “framework”, “tool”
Criterios de clasificación	“polarity”
Plataforma social de estudio	“twitter” OR “tweet”

Tabla 2. Palabras categorizadas

2.1.2.2 Cadena de búsqueda

La cadena de búsqueda resultante de acuerdo con los términos de la *Tabla 2* fue:

("sentiment analysis" OR "opinion mining" AND (("method" OR "model" OR "tool" OR "service" OR "library" OR "algorithm") AND "polarity" AND "twitter")).

2.1.2.3 Criterios de selección

Los criterios de selección (inclusión, exclusión y calidad) identificados y que permitieron filtrar y escoger los artículos científicos que resultaron de la aplicación de la cadena de búsqueda se presentan en la *Tabla 3*.

Inclusión
Publicados desde 2012 hasta la fecha (2019).
Los artículos o tesis que estén en idioma inglés.
Artículos completos (full papers), tesis, conference papers, books, book chapter.
Investigaciones que involucren datasets en idioma inglés.
Propuestas centradas en clasificación de polaridad.
Exclusión
Artículos cortos, revisiones (reviews), conference review, opiniones.
Investigación duplicada del mismo autor.
Contenido repetido en artículos similares.
Propuestas que contengan procesos de implementación y diseños de estudio sin estar claramente fijados o definidos.
Calidad
QA1: ¿La propuesta se puede implementar sólo a través del diseño presentado en su investigación? Las respuestas posibles son: (Si +1.5 No +0).

QA2: ¿La propuesta cuenta con un método de validación realizado a través de un estudio de caso o experimento? (Si +1 No +0).
QA3: ¿La propuesta presenta resultados detallados y obtenidos durante el método de validación? (Si +1 No +0).
QA4: ¿La propuesta presenta resultados de precisión detallados como la medida F1, recall y accuracy obtenidas durante el método de validación? (Presenta todas las medidas +1 Presenta al menos dos medidas 0.5 No presenta más de una medida +0).
QA5: ¿La propuesta está asociada al cálculo de polaridad en análisis de sentimiento con datasets que contienen mensajes de texto plano? (Si +1 No +0).
QA6: ¿La propuesta presenta información de tuits con medidas de precisión y polaridad? (Si +2 No +0).
QA7: ¿La propuesta ha sido publicada en una conferencia o revista relevante para la comunidad científica considerando los índices JCR para revistas y CORE para conferencias? (Q1/A* +2 Q2/A +1.5 Q3/B +1 Q4/C 0.5 Sin categoría +0)

Tabla 3. Criterios de selección

De los criterios de calidad de la Tabla 3, el puntaje total para evaluar una propuesta viene dado por la Ecuación 1.

$$P = QA1 + QA2 + QA3 + QA4 + QA5 + QA6 + QA7$$

Ecuación 1. Suma de criterios de calidad

donde P es el puntaje total y QAn es el puntaje obtenido en cada criterio.

2.1.3 Selección de artículos

Los artículos resultantes en la revisión sistemática fueron seleccionados mediante cuatro fases: Listado de artículos provenientes de una cadena de búsqueda, filtrado de artículos potenciales, aplicación de criterios de inclusión/exclusión y selección de artículos finales de acuerdo a los criterios de calidad establecidos. En la primera fase, se usó una cadena de texto para realizar la búsqueda en el indexador de artículos Scopus, de la cual se obtuvo como resultado una lista de 311 artículos. En la segunda fase, se seleccionó los artículos potenciales que fueron obtenidos después de filtrar las propuestas investigativas que contenían al menos dos palabras claves en su título y tres en el resumen (*abstract*), en el proceso se obtuvo 159 artículos. En la tercera fase, se aplicó los criterios de inclusión y exclusión definidos en la Tabla 3 a la lista de artículos potenciales, lo que permitió filtrar las propuestas que abarcaran la temática del trabajo investigativo y fueran convenientes para solucionar la pregunta de investigación. En la fase final, se aplicó los criterios de calidad a los artículos resultantes, obteniendo como resultado una

lista de propuestas que fueron puntuadas y ordenadas de mayor a menor. La puntuación se obtuvo mediante la suma total de los puntos de medida establecidos en cada criterio de calidad para cada propuesta.

2.1.4 Resultados del proceso de búsqueda

En esta sección se presentan los resultados obtenidos en la investigación de la revisión sistemática.

2.1.4.1 Cadena de búsqueda

En la Tabla 4, se muestran los resultados obtenidos al validar el protocolo de revisión en la base de datos Scopus, la cual se hizo mediante una cadena de texto conformada por varias palabras claves establecidas en el proceso de búsqueda establecido en la sección 2.1.2.2 Cadena de búsqueda. Se obtuvo una lista de estudios y propuestas que tienen como foco Análisis de Sentimiento (Sentiment Analysis) y clasificadores de polaridad en Twitter.

Base de datos	Cadena	Total
Scopus	("sentiment analysis" OR "opinion mining" AND ("method" OR "model" OR "tool" OR "service" OR "library" OR "algorithm") AND "polarity" AND "twitter")	311

Tabla 4. Resultados cadena de búsqueda literatura científica

2.1.4.2 Selección de artículos potenciales

Para el proceso de selección de artículos potenciales se estableció una cantidad mínima de palabras clave para el título y resumen (*abstract*). Las propuestas deben contener al menos dos palabras claves en el título y mínimo tres en el resumen. El número total de propuestas que cumplieron estas condiciones se presentan en la Tabla 5.

Palabras clave	Cantidad de artículos encontrados	Total de artículos potenciales
"sentiment analysis" "opinion mining" "polarity" "sentiment classifier" "method" "service" "library" "algorithm" "framework" "tool" "fuzzy logic" "microblogging" "twitter"	311	159

Tabla 5. Resultados artículos potenciales

2.1.4.3 Aplicación de criterios de inclusión y exclusión

Los criterios de inclusión y exclusión definidos en la sección 2.1.2.3 Criterios de selección, permitieron incluir las propuestas investigativas que responden a la pregunta de investigación y excluir las que no. Se pudo observar una disminución considerable en el número de propuestas de investigación con respecto al número de propuestas seleccionadas como artículos potenciales. Los estudios resultantes son considerados como propuestas primarias y se encuentran resumidos en la Tabla 6.

Nombre del artículo	Autor	Tipo de artículo	Año
A New Method for Sentiment Analysis Using Contextual Auto-Encoders [14].	Ameur, H.-Jamoussi, S.-Hamadou, A.B.	Article	2018
Enhanced Framework for Sentiment Analysis in Text using Distance based Classification Scheme [15].	Angadi, S-Venkata Siva Reddy, R.	Conference Paper	2018
Sentiment Analysis of Real Time Twitter Data Using Big Data Approach [16].	Rodrigues-Rao, A.-Chiplunkar-N.N.	Conference Paper	2017
Deep Learning Techniques for Polarity Classification in Multimodal Sentiment Analysis [17].	Mahendhiran, P.D.-Kannimuthu, S	Article	2018
Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings [18].	Xiong, S.-Lv, H.-Zhao, W.-Ji, D	Article	2018
An Ensemble Classification System for Twitter Sentiment Analysis [7].	Ankit, Saleena N.	Conference Paper	2018
Sentiment analysis on twitter using streaming API [19].	M.Trupthi-Suresh Pabboju-G.Narasimha	Conference Paper	2017
A Real-Time Twitter Sentiment Analysis using an unsupervised Method [20].	Azzouza, N.-Akli-Astouati, K.-Oussalah, A.-Ait Bachir, S.	Conference Paper	2017

Efficient Twitter sentiment classification using subjective distant supervision [21].	Sahni, T.-Chandak, C.-Chedeti, N.R.-Singh, M.	Conference Paper	2017
Sentiment analysis of Twitter data: Case study on digital India [22].	Prerna Mishra-Dr. Ranjana Rajnish-Dr.Pankaj Kumar	Conference Paper	2016
Efficient and parallel framework for analyzing the sentiment [23].	Sharma, A.-Nayak, G.K.	Conference Paper	2016
Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information [24].	Ahmed Sulaiman M Alharbi-Elise de Donckera	Article	2019
Tweet sentiment analyzer: Sentiment score estimation method for assessing the value of opinions in tweets [25].	Arun Manicka Raja M. Swamynathan S.	Conference Paper	2016
Incorporation of target specific knowledge for sentiment analysis on microblogging [26].	Kaewpitakkun, K. Shirai Kiyooki	Article	2016
Tweet analyzer: Identifying interesting tweets based on the polarity of tweets [27].	M. Arun Manicka Raja S. Swamynathan	Conference Paper	2015
Sentiment polarity detection from Amazon reviews: An experimental study [28].	Efstratios Sygkounas ¹ , Giuseppe Rizzo ² , Raphaël Troncy ¹	Conference Paper	2016
Twitter Sentiment Analysis for Large-Scale Data: An Unsupervised Approach [29].	Pandarachalil, R. Sendhilkumar, S Mahalakshmi, G.	Article	2015
Modelling public sentiment in twitter: Using linguistic patterns to enhance supervised learning [30].	Chikersal, P. Poria, S. Cambria, E. Gelbukh, A. Siong, C.	Conference Paper	2015
TOM: Twitter opinion mining framework using hybrid classification scheme [31].	Farhan Hassan Khan, Saba Bashir Usman Qamar	Article	2014

Sentiment Analysis in Twitter with Lightweight Discourse Analysis [32].	Subhabrata Mukherjee Pushpak Bhattacharyya	Conference Paper	2012
---	---	------------------	------

Tabla 6. Propuestas primarias

2.1.4.4 Aplicación de criterios de calidad

Se aplicó los criterios de calidad definidos en la Tabla 3 a todas las propuestas seleccionadas en la Tabla 6. La puntuación final obtenida para cada una de las propuestas resultantes después de aplicar la Ecuación 1 se presenta en la Tabla 7.

Además, se presenta el indicador de producción científica de cada revista o conference paper en donde fue publicada cada propuesta, en este caso el indicador de producción científica para revistas viene dado en cuartiles Q1, Q2, Q3 y Q4, en donde Q1 es el más alto y Q4 el más bajo. Para conference paper viene dado el factor impacto en A, B y C, en donde A es el más alto y C el más bajo. Entre más alto sea el indicador de producción científica más prestigiosa será la propuesta publicada.

Nombre del artículo	Revista de publicación	Puntaje
Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings.	Neurocomputing Q1.	$P = 0 + 1 + 1 + 1 + 1 + 0 + 2 = 6.0$
TOM: Twitter opinion mining framework using hybrid classification scheme.	Decision Support Systems Q1.	$P = 0 + 1 + 1 + 1 + 1 + 0 + 2 = 6.0$
A New Method for Sentiment Analysis Using Contextual Auto-Encoders.	Journal of Computer Science and Technology Q3.	$P = 0 + 1 + 1 + 1 + 1 + 0 + 1 = 5.0$
Tweet sentiment analyzer: Sentiment score estimation method for assessing the value of opinions in tweets.	ACM International Conference Proceeding Series Sin rango.	$P = 0 + 1 + 1 + 0 + 1 + 2 + 0 = 5.0$
Incorporation of target specific knowledge for sentiment analysis on microblogging.	IEICE Transactions on Information and Systems Q3.	$P = 0 + 1 + 1 + 1 + 1 + 0 + 1 = 5.0$

A Real-Time Twitter Sentiment Analysis using an unsupervised Method.	ACM International Conference Proceeding Series Sin rango.	$P = 0 + 1 + 1 + 0 + 1 + 2 + 0 + 0 = 5.0$
Deep Learning Techniques for Polarity Classification in Multimodal Sentiment Analysis.	International Journal of Information Technology and Decision Making Q1.	$P = 0 + 1 + 1 + 1 + 0 + 0 + 2 = 5.0$
Twitter Sentiment Analysis for Large-Scale Data: An Unsupervised Approach.	Cognitive Computation Q1.	$P = 0 + 1 + 1 + 1 + 0 + 0 + 2 = 5.0$
An Ensemble Classification System for Twitter Sentiment Analysis.	Procedia Computer Science Sin rango.	$P = 0 + 1 + 1 + 1 + 1 + 0 + 0 = 4.0$
Sentiment Analysis of Real Time Twitter Data Using Big Data Approach.	2nd International Conference on Computational Systems and Information Technology for Sustainable Solutions Sin rango.	$P = 0 + 1 + 0 + 0 + 1 + 2 + 0 + 0 = 4.0$
Sentiment polarity detection from Amazon reviews: An experimental study.	Communications in Computer and Information Science Sin rango.	$P = 0 + 1 + 1 + 1 + 1 + 0 + 0 = 4.0$
Modelling public sentiment in twitter: Using linguistic patterns to enhance supervised learning.	Annual Conference on Intelligent Text Processing and Computational Linguistics Sin rango.	$P = 0 + 1 + 1 + 1 + 1 + 0 + 0 = 4.0$
Efficient and parallel framework for analyzing the sentiment.	Advances in Intelligent Systems and Computing C.	$P = 0 + 1 + 1 + 0 + 1 + 0 + 0.5 = 3.5$
Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information.	Cognitive Systems Research Q4.	$P = 0 + 1 + 1 + 1 + 0 + 0 + 0.5 = 3.5$
Sentiment Analysis in Twitter with Lightweight Discourse Analysis.	International Conference on Computational Linguistics A.	$P = 0 + 1 + 1 + 0 + 0 + 0 + 1.5 = 3.5$
Tweet analyzer: Identifying interesting tweets based on the polarity of tweets.	International Conference on Computational Intelligence in Data Mining Sin rango.	$P = 0 + 1 + 1 + 0 + 1 + 0 + 0 = 3.0$

Efficient Twitter sentiment classification using subjective distant supervision.	International Conference on Communication Systems and Networks Sin rango.	$P = 0 + 1 + 1 + 0 + 1 + 0 + 0 = 3.0$
Sentiment analysis of Twitter data: Case study on digital India.	Institute of Electrical and Electronics Engineers C.	$P = 0 + 1 + 0 + 0 + 1 + 0 + 0.5 = 2.5$
Enhanced Framework for Sentiment Analysis in Text. using Distance based Classification Scheme.	2nd International Conference on Advances in Electronics, Computers and Communications, ICAECC 2018 Sin rango.	$P = 0 + 1 + 0 + 0.5 + 1 + 0 + 0 = 2.5$
Sentiment analysis on twitter using streaming API.	IEEE International Advanced Computing Conference Sin rango.	$P = 0 + 1 + 0 + 0 + 1 + 0 + 0 = 2.0$

Tabla 7. Evaluación criterios de calidad

La selección final de los estudios primarios está definida por un valor mínimo de calidad establecido en 6.0, el cual que deberá ser igualado o superado por cada una de las propuestas evaluadas. El método de cálculo del puntaje de calidad mínimo establecido se explica a continuación:

La suma de QA2, QA3, QA4, QA5 y QA6 debe ser 4, QA7 debe ser 2. Para QA1 Y QA7 se tiene dos escenarios a y b. El primer escenario en donde QA1 es 0 y QA7 es 2.0 y viene dado por la Ecuación 2; el segundo escenario en donde QA7 es como mínimo 0.5 y QA1 es 1.5 y viene dado por la Ecuación 3. Los dos escenarios vienen dados por las siguientes expresiones:

$$QA1(+0) + [QA2 + QA3 + QA4 + QA5] (+4) + QA6(+0) + QA7(+2) = 6.0$$

Ecuación 2. Cálculo del puntaje de calidad 1

$$QA1(+1.5) + [QA2 + QA3 + QA4 + QA5] (+4) + QA6(+0) + QA7(+0.5) = 6.0$$

Ecuación 3. Cálculo del puntaje de calidad 2

De esta manera se ha dado más peso e importancia a los criterios QA1 Y QA7, debido a que la complejidad en el proceso de implementación y el nivel de confianza al momento de emplear tiempo en desarrollo dependen directamente de estos. QA6 es tenido en cuenta sólo como bonificación para las propuestas que tengan

resultados de precisión y polaridad para cada uno de sus tuits analizados, lo que permitiría ahorrar el proceso de implementación.

Las propuestas seleccionadas como estudios primarios de calidad se encuentran resaltadas en la Tabla 7. A partir de ahora las tablas relacionadas a la revisión sistemática de la literatura científica vendrán resaltadas con las propuestas seleccionadas en la Tabla 7.

2.1.4.5 Solución a las preguntas de investigación

En esta sección se reportan los resultados de la investigación a través de cuatro fases, cada fase responde una de las preguntas de investigación mencionadas en la sección de 2.1.1 Preguntas de investigación. Se describen todos los estudios publicados hasta la fecha que han superado completamente el proceso de selección de propuestas.

Fase 1. ¿Cuál es la clasificación de las propuestas existentes del estado del arte en clasificación de sentimientos?

Aunque la lista de propuestas potenciales seleccionadas contiene en el contexto de cada propuesta científica el número mínimo de palabras clave establecidas en el proceso de búsqueda mencionado en la sección 2.1.2.1 Palabras clave, se decidió trabajar con las propuestas seleccionadas de la Tabla 6, debido a que algunos estudios potenciales no incluyen palabras clave en su proceso de clasificación de polaridad. En Tabla 8 se presenta la lista de propuestas clasificadas de acuerdo a la naturaleza de implementación de su clasificador de sentimiento y el año de publicación.

Tipo de Clasificador	Nombre del artículo	Año
	A New Method for Sentiment Analysis Using Contextual Auto-Encoders	2018
	Sentiment Analysis of Real Time Twitter Data Using Big Data Approach	2017
	Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings	2018
	An Ensemble Classification System for Twitter Sentiment Analysis	2018

Métodos	Tweet sentiment analyzer: Sentiment score estimation method for assessing the value of opinions in tweets	2016
	Incorporation of target specific knowledge for sentiment analysis on microblogging	2016
	Twitter Sentiment Analysis for Large-Scale Data: An Unsupervised Approach	2015
	Modelling public sentiment in twitter: Using linguistic patterns to enhance supervised learning	2015
	TOM: Twitter opinion mining framework using hybrid classification scheme	2014
	Sentiment analysis on twitter using streaming API	2017
	A Real-Time Twitter Sentiment Analysis using an unsupervised Method	2017
	Sentiment analysis of Twitter data: Case study on digital India	2016
Técnica	Efficient Twitter sentiment classification using subjective distant supervision	2017
Framework	Enhanced Framework for Sentiment Analysis in Text using Distance based Classification Scheme	2018
	Deep Learning Techniques for Polarity Classification in Multimodal Sentiment Analysis	2018
	Efficient and parallel framework for analyzing the sentiment	2016
Modelos	Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information	2019
	Tweet analyzer: Identifying interesting tweets based on the polarity of tweets	2015
	Sentiment Analysis in Twitter with Lightweight Discourse Analysis	2012
Algoritmo	Sentiment polarity detection from Amazon reviews: An experimental study	2016

Tabla 8. Clasificadores de polaridad por tipos

Las veinte propuestas presentadas en la Tabla 8 fueron identificadas como doce métodos, tres Framework, tres Modelos, una técnica y un algoritmo. No se encontraron herramientas, APIs, librerías o prototipos software que cumplieran con los criterios de selección. La mayoría de propuestas del estado del arte son métodos que implementan modelos e integran técnicas de inteligencia artificial en el proceso de clasificación de polaridad.

Fase 2. ¿Cuáles algoritmos o herramientas son utilizados en el proceso de clasificación de polaridad de las propuestas existentes?

Los algoritmos identificados que se utilizaron en el desarrollo de las propuestas seleccionadas en la Tabla 7 brindan un panorama de opciones y posibilidades al momento de diseñar un clasificador de sentimientos, influyendo directamente en la medida de precisión que se calcula en el proceso de clasificación de polaridad. En la Tabla 9 se lista las propuestas clasificadas por los algoritmos o herramientas utilizadas en el proceso de clasificación de polaridad, la rama de inteligencia artificial a la que pertenecen y el año de publicación.

Algoritmo / Herramienta	Rama de Inteligencia artificial	Nombre del artículo	Año
Support Vector Machine (SVM)	Supervised learning Machine learning	A New Method for Sentiment Analysis Using Contextual Auto-Encoders	2018
SentiWordNet (SWN)	Supervised learning Machine learning	Enhanced Framework for Sentiment Analysis in Text using Distance based Classification Scheme	2018
Apache Hadoop	Machine learning	Sentiment Analysis of Real Time Twitter Data Using Big Data Approach	2017
Support Vector Machine (SVM)	Deep learning Machine learning	Deep Learning Techniques for Polarity Classification in Multimodal Sentiment Analysis	2018
Support Vector Machine (SVM)	Deep learning Machine learning	Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings	2018

Naive Bayes - Random Forest - Support Vector Machine - Logistic Regression	Machine learning	An Ensemble Classification System for Twitter Sentiment Analysis	2018
Naive Bayes	Machine learning	Sentiment analysis on twitter using streaming API	2017
Método propio basado en SemEval	Unsupervised machine learning Machine learning	A Real-Time Twitter Sentiment Analysis using an unsupervised Method	2017
Naive Bayes - Support Vector Machine - Logistic Regression.	Machine learning	Efficient Twitter sentiment classification using subjective distant supervision	2017
NLTK Python	Machine learning	Sentiment analysis of Twitter data: Case study on digital India	2016
SentiWordNet (SWN)	Supervised learning Machine learning	Efficient and parallel framework for analyzing the sentiment	2016
Convolutional Neuronal Network (CNN)	Deep learning Machine learning	Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information	2019
Método propio apoyado en SentiWordNet	Machine learning	Tweet sentiment analyzer: Sentiment score estimation method for assessing the value of opinions in tweets	2016
Support Vector Machine (SVM)	Machine learning	Incorporation of target specific knowledge for sentiment analysis on microblogging	2016
Support Vector Machine - Naive Bayes - J48 - Conjunctive Rule - AdaBoostM1 - Sequential minimal - IBK	Machine learning	Tweet analyzer: Identifying interesting tweets based on the polarity of tweets	2015

SentiME	Machine learning	Sentiment polarity detection from Amazon reviews: An experimental study	2016
Método propio basado en SenticNet - SentiWordNet - SentislangNet	Machine learning	Twitter Sentiment Analysis for Large-Scale Data: An Unsupervised Approach	2015
Support Vector Machine (SVM) con n-gramas	Machine learning	Modelling public sentiment in twitter: Using linguistic patterns to enhance supervised learning	2015
SentiWordNet Classifier	Unsupervised machine learning Machine learning	TOM: Twitter opinion mining framework using hybrid classification scheme	2014
Support Vector Machine (SVM)	Machine learning	Sentiment Analysis in Twitter with Lightweight Discourse Analysis	2012

Tabla 9. Lista de propuestas de clasificadores de polaridad

De la Tabla 9 se determina que la mayoría de clasificadores de sentimientos del estado del arte usan como algoritmo de clasificación a Support Vector Machine (SVM). Tan sólo tres propuestas usan un método propio apoyándose de herramientas como SemEval y SentiWordNet Dictionary que permiten obtener la polaridad de una palabra específica.

Fase 3. ¿Cuáles propuestas requieren o no requieren, el preprocesamiento de datos en la clasificación de polaridad?

Los estudios científicos que utilizan técnicas de preprocesamiento de datos intentan incrementar la medida de precisión de la polaridad tratando directamente con los datos de entrada del mensaje. Debido a que el algoritmo propuesto no tiene relación directa con el preprocesamiento de datos, se quiere tener una idea cercana de los clasificadores de polaridad en análisis de sentimiento que no usen esta técnica, en la Tabla 10 se describe la lista de propuestas caracterizadas según su técnica de extracción de conocimiento.

Extracción de conocimiento	Nombre del artículo	Año
Preprocesamiento de datos	A New Method for Sentiment Analysis Using Contextual Auto-Encoders	2018
	Enhanced Framework for Sentiment Analysis in Text using Distance based Classification Scheme	2018
	Sentiment Analysis of Real Time Twitter Data Using Big Data Approach	2017
	Deep Learning Techniques for Polarity Classification in Multimodal Sentiment Analysis	2018
	Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings	2018
	An Ensemble Classification System for Twitter Sentiment Analysis	2018
	Sentiment analysis on twitter using streaming API	2017
	A Real-Time Twitter Sentiment Analysis using an unsupervised Method	2017
	Efficient Twitter sentiment classification using subjective distant supervision	2017
	Sentiment analysis of Twitter data: Case study on digital India	2016
	Efficient and parallel framework for analyzing the sentiment	2016
	Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information	2019
	Tweet sentiment analyzer: Sentiment score estimation method for assessing the value of opinions in tweets	2016
	Incorporation of target specific knowledge for sentiment analysis on microblogging	2016
Tweet analyzer: Identifying interesting tweets based on the polarity of tweets	2015	

	Twitter Sentiment Analysis for Large-Scale Data: An Unsupervised Approach	2015
	Modelling public sentiment in twitter: Using linguistic patterns to enhance supervised learning	2015
	TOM: Twitter opinion mining framework using hybrid classification scheme	2014
	Sentiment Analysis in Twitter with Lightweight Discourse Analysis	2012
Sin preprocesamiento de datos	Sentiment polarity detection from Amazon reviews: An experimental study	2016

Tabla 10. Clasificación por técnicas de extracción

De los veinte estudios del estado del arte clasificados por tipo de técnicas de extracción de conocimiento se puede conocer que sólo uno ejecuta un proceso de clasificación de polaridad con el dataset original, las diecinueve propuestas restantes manipulan y adaptan el dataset de prueba para obtener buenos resultados.

Fase 4. ¿Cuáles propuestas utilizan técnicas de lógica difusa?

Para la investigación propuesta es importante tener un panorama de estudios que implementen reglas de lógica difusa en el proceso de clasificación de polaridad. Encontrar estudios que usen estas técnicas permite trabajar sobre una base existente de procesos de clasificación de polaridad que se puede aprovechar para tener un punto de partida avanzado para crear nuevas versiones que intenten mejorar la medida de precisión en clasificación de polaridad. En la Tabla 9 se observa que no hay ninguna propuesta científica que implemente técnicas de lógica difusa en el proceso del cálculo de la polaridad, por lo que se llega a la conclusión que el estudio propuesto es novedoso al implementar un clasificador de sentimiento que calcula la polaridad mediante técnicas de lógica difusa.

2.2 Proceso de desarrollo de la revisión sistemática (literatura gris)

La revisión sistemática de la investigación propuesta en la literatura gris está basada en dos estrategias de búsqueda, la estrategia basada en los lineamientos

Kitchenham y la estrategia de bola de nieve⁵ utilizando el motor de búsqueda de Google.

2.2.1 Preguntas de investigación

Se quiere compilar las propuestas científicas que abarquen la problemática del trabajo investigativo y que aparecen en la literatura gris, para ello se ha planteado la siguiente pregunta de investigación:

“¿Qué herramientas, servicios y librerías existen sobre clasificación de polaridad en análisis de sentimiento?”

Además, para lograr que se cubra todo el estudio investigativo se han planteado una serie de preguntas de investigación, las cuales son:

- P11. ¿Cuáles algoritmos, herramientas o servicios son utilizados en el proceso de clasificación de polaridad de las propuestas existentes?
- P12. ¿Qué tipo de datos y cuales datos son arrojados en los resultados de clasificación de sentimiento?
- P13. ¿Cuáles clasificadores utilizan técnicas de lógica difusa?

Con respecto a P11, se quiere compilar una lista de los algoritmos, herramientas o servicios que han sido utilizados en el desarrollo del proceso de clasificación de las propuestas estudiadas. Con respecto a P12, se desea identificar qué tipo de datos y cuáles son arrojados como resultado de análisis de sentimiento por las propuestas del estado del arte. Los metadatos entregados como respuesta de los clasificadores influyen directamente con la implementación del algoritmo propuesto, debido a que esta toma los resultados arrojados por los clasificadores de sentimiento. Con respecto a P13, se quiere analizar e identificar cuáles propuestas seleccionadas del estado del arte utilizan técnicas de lógica difusa en el proceso de análisis de polaridad.

2.2.2 Proceso de búsqueda

En este proceso se define un protocolo de revisión para páginas web, que incluye la definición de: palabras clave, cadena de búsqueda, criterios de inclusión y exclusión, y criterios de calidad.

⁵ Bola de nieve: Estrategia que permite identificar estudios potenciales que son difíciles de encontrar a priori mediante cadenas de búsqueda, su método consiste en ir de un estudio a otro a través de las referencias citadas entre estos.

2.2.2.1 Palabras clave

Las palabras claves vienen dadas en relación al tema de investigación, en la *Tabla 11* se encuentran las palabras claves que son de utilidad para construir la cadena de búsqueda:

Contexto	Palabra clave
Área	“sentiment analysis”, “opinion mining”, “text analytics”
Criterio de clasificación	“polarity”
Clasificadores de sentimiento	“tool”, “service”, “library”, “API”
Técnicas	“fuzzy logic”
Plataforma social de estudio	“microblogging”, “twitter”

Tabla 11. Palabras claves revisión sistemática literatura gris

2.2.2.2 Cadena de búsqueda

En el proceso de construcción de la cadena de búsqueda se tuvieron en cuenta ciertos aspectos:

Las palabras clave categorizadas en el contexto “Área” fueron el foco principal para la búsqueda de propuestas. Los sinónimos o palabras clave de la misma categoría se separan con OR, las palabras dentro de comillas (“”) se usan para realizar una búsqueda específica de un término compuesto, debido a que, separar las palabras con un espacio se interpreta como palabras claves de distinta categoría [33].

Términos principales	Términos alternativos
Análisis de sentimiento	“sentiment analysis” OR “text analytics”
Clasificadores de sentimientos	“API” “classifier”

Tabla 12. Categorización de palabras claves

La cadena de búsqueda resultante usando los términos principales y alternativos resumidos en la Tabla 12 es la siguiente:

“sentiment analysis” OR “text analytics” “classifier” “API”

2.2.2.3 Criterios de selección de literatura gris

Los criterios de selección permitieron filtrar y escoger los clasificadores de polaridad en análisis de sentimientos que resultaron de la cadena de búsqueda planteada en el paso anterior. En la Tabla 13 se definen los criterios de selección para la literatura gris.

Inclusión
Permita al usuario calcular la polaridad de una cadena de texto ingresada.
La propuesta debe contar con un prototipo software que puede ser utilizado como aplicación web o descargado como aplicación de escritorio.
La documentación sobre el proceso de desarrollo de la propuesta debe estar publicada en su página.
Exclusión
Versiones de pago.
Que no sea posible usar el prototipo software con datasets o cadenas en inglés.
Calidad
QA1: ¿La propuesta calcula la polaridad de más de una palabra? Las respuestas posibles son: (Permite cargar un dataset completo +2 Calcula la polaridad de un mensaje +1 No +0).
QA2: ¿La propuesta presenta resultados detallados después de calcular la polaridad? Las respuestas posibles son: (Si +1 No +0).
QA3: ¿El ingreso de emoticonos dentro de la cadena de texto a clasificar influye directamente en el cálculo de la polaridad? Las respuestas posibles son: (Si +0 No +1).
QA4: ¿La propuesta presenta en los resultados de precisión alguna medida como F1 score, recall y accuracy obtenidas al clasificar la polaridad? (Presenta al menos una +1 Presenta una medida diferente +0.5 No presenta ninguna medida de precisión +0).
QA5: ¿La propuesta brinda soporte para ser implementada o consumida desde una aplicación externa? (Ofrece soporte +2 No se puede utilizar con otras aplicaciones +0).

Tabla 13. Criterios de selección de literatura gris

2.2.3 Selección de propuestas

Las propuestas resultantes en la revisión de literatura gris fueron seleccionadas mediante cuatro fases: Listado de sitios web provenientes de una cadena de búsqueda, filtrado de estudios potenciales, aplicación de criterios de inclusión/exclusión y selección de artículos finales de acuerdo a los criterios de calidad establecidos. En la primera fase, se usó una cadena de texto en el motor de búsqueda de Google, de la cual se obtuvo 169.000 resultados. Además, en base a

la estrategia bola de nieve se seleccionaron las propuestas difíciles de encontrar mediante la cadena de búsqueda y que posiblemente serían candidatas como estudios potenciales. En la segunda fase, se aplicaron una serie de condiciones dentro del proceso de selección preliminar, las que permitieron seleccionar las propuestas que tuvieran una palabra clave en el título y tres en la descripción. De este proceso se obtuvo 24 clasificadores. En la tercera fase, se aplicó una serie de criterios a la lista de clasificadores potenciales que permitió incluir o excluir las propuestas convenientes para responder la pregunta de investigación. En la fase final, se aplicó los criterios de calidad a los clasificadores resultantes, obteniendo como resultado una lista de clasificadores valorados y ordenados de mayor a menor por el puntaje de calidad calculado. La valoración de calidad de los clasificadores se obtuvo mediante la suma total de los puntos de medida establecidos en cada criterio de calidad.

2.2.4 Resultados del proceso de búsqueda de la literatura gris

En esta sección se presentan los resultados obtenidos en el proceso de desarrollo de la revisión sistemática de literatura gris.

2.2.4.1 Cadena de búsqueda

En la Tabla 14 se muestran los resultados obtenidos al validar el protocolo de revisión en el motor de búsqueda de Google, el cual utilizó una cadena de texto conformada por un conjunto de palabras claves establecidas. Se obtuvo una lista de clasificadores de polaridad en análisis de sentimiento que tiene como foco Análisis de Sentimiento (Sentiment Analysis) y clasificadores de polaridad en Twitter.

Motor de búsqueda	Cadena de búsqueda	Resultados
Google	"sentiment analysis" OR "text analytics" "classifier" "API"	169.000

Tabla 14. Resultados cadena de búsqueda literatura gris

2.2.4.2 Selección de propuestas potenciales

Para el proceso de selección de propuestas potenciales se estableció una cantidad mínima de palabras clave para el título y resumen (abstract). Las propuestas deben contener al menos una palabra clave en el título y mínimo tres en el resumen. El total de propuestas que cumplieron estas condiciones se presentan en la Tabla 15.

Nombre	Enlace de sitio web	Formato de despliegue
Text analytics [34]	https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/	Web
Python NLTK Text Classification [35]	https://text-processing.com/demo/sentiment/	Web
Sentiment Analysis [36]	https://app.monkeylearn.com/main/classifiers/cl_pi3C7JiL/	Web
Sentiment140 [37]	http://www.sentiment140.com/	Web
Tweetmood [38]	http://www.tweet-mood-check.appspot.com/	Web
Senti-py [39]	https://github.com/ayllote/senti-py	Repositorio
Analizador de sentimiento de las redes sociales [40]	https://www.mediatoolkit.com/	Web
API de análisis de texto [41]	https://ai-applied.nl/support/text-analysis-api-documentation	Consumo de API
Text analytics Demo [42]	https://www.paralldots.com/text-analysis-apis	Web
Tweepy [43]	https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/	Escritorio
Sentiment [44]	https://www.uclassify.com/browse/uclassify/sentiment	Web
Natural Language Understanding [45]	https://www.ibm.com/watson/services/natural-language-understanding/	Web
Applytics [46]	https://github.com/amitnesh/applytics	Repositorio
Sentiment Analysis LSTM [47]	https://github.com/artem-oppermann/Sentiment-Analysis-of-Netflix-Reviews	Repositorio

Text Classification Library [48]	https://github.com/jfilter/text-classification-keras	Repositorio
sentistrength_id [49]	https://github.com/masdevid/sentistrength_id	Repositorio
Sentiment viz [50]	https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/	Web
Mr Tuit [51]	http://www.mrtuit.com/	Web
25Trends [52]	https://twitter.25trends.me/services/twitter/public/	Web
Opinion Crawl [53]	http://opinioncrawl.com/	Web
Sentiment Analysis [54]	https://dandelion.eu/semantic-text/sentiment-analysis-demo/	Web
Senti4SD [55]	https://github.com/collab-uniba/Senti4SD	Repositorio
Text Analysis API Demo [56]	https://www.summarizebot.com/text_api_demo.html	Web
Linguakit [57]	https://linguakit.com/es/analisis-completo	Web

Tabla 15. Propuestas potenciales

2.2.4.3 Aplicación de criterios de inclusión y exclusión

Las aplicaciones de los criterios de inclusión y exclusión facilitan el filtrado de las propuestas que permiten responder la pregunta de investigación planteada como objetivo de la revisión sistemática. Los resultados de los estudios son considerados como propuestas primarias y se encuentran en la Tabla 16.

Nombre	Enlace de sitio web	Formato de despliegue	Desarrollador
Text analytics	https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/	Web	Microsoft Azure
Natural Language Understanding	https://www.ibm.com/watson/services/natural-language-understanding/	Web	Watson IBM

Sentiment Analysis	https://app.monkeylearn.com/main/classifiers/cl_pi3C7JiL/tab/demo/	Web	Monkeylearn
Python NLTK Text Classification	https://text-processing.com/demo/sentiment/	Web	NLTK
Text analytics Demo	https://www.paralldots.com/text-analysis-apis	Web	ParallelDots
Sentiment	https://www.uclassify.com/browse/uclassify/sentiment	Web	uClasificar
Text Analysis API Demo	https://www.summarizebot.com/text_api_demo.html	Web	SummarizeBot
Linguakit	https://linguakit.com/es/analisis-completo	Web	Linguakit
Tweepy	https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/	Escritorio	TextBlob

Tabla 16. Aplicación de criterios de inclusión y exclusión

2.2.4.4 Aplicación Criterios de calidad

Se aplicaron los criterios de calidad de todas las propuestas que resultaron del proceso de selección anterior. Los valores de puntuación de calidad para cada criterio se encuentran definidos en la sección criterios de calidad. La lista de propuestas al evaluar los criterios de calidad se presenta en la Tabla 17.

El proceso de evaluación de las propuestas de calidad seleccionadas son las que superan o igualan el valor de calidad mínimo calculado en 4.5. El método de cálculo del puntaje de calidad mínimo establecido en 4.5 como se muestra en la Ecuación 4.

$$[QA1 + QA2 + QA3](+2) + QA4 (+0.5) + QA5(+2) = 4.5$$

Ecuación 4. Cálculo del puntaje de calidad para literatura gris

La suma de QA1, QA2 y QA3 debe ser 2, QA4 y QA5 debe ser como mínimo 0.5 y 2 respectivamente.

De esta manera se ha dado más peso e importancia al criterio QA5, debido a que es que permite conocer la posibilidad de utilizar los datos arrojados como resultado desde una aplicación externa para el desarrollo del trabajo investigativo propuesto.

Nombre	Desarrollador	Enlace de sitio web	Puntaje
Text analytics	Microsoft Azure	https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/	$P = 1 + 1 + 0 + 0.5 + 2 = 4.5$
Natural Language Understanding	Watson IBM	https://www.ibm.com/watson/services/natural-language-understanding/	$P = 1 + 1 + 0 + 0.5 + 2 = 4.5$
Tweepy	TextBlob	https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/	$P = 1 + 0 + 1 + 0.5 + 2 = 4.5$
Sentiment Analysis	Monkeylearn	https://app.monkeylearn.com/main/classifiers/cl_pi3C7JiL/tabb/demo/	$P = 1 + 0 + 1 + 0 + 2 = 4.0$
Python NLTK Text Classification	NLTK	https://text-processing.com/demo/sentiment/	$P = 1 + 0 + 1 + 0 + 2 = 4.0$
Sentiment	uClasificar	https://www.uclassify.com/browse/uclassify/sentiment	$P = 1 + 1 + 0 + 0 + 2 = 4.0$
Text Analysis API Demo	SummarizeBot	https://www.summarizebot.com/text_api_demo.html	$P = 1 + 0 + 1 + 0 + 2 = 4.0$
Linguakit	Linguakit	https://linguakit.com/es/analisis-completo	$P = 1 + 1 + 1 + 0.5 + 0 = 3.5$
Text analytics Demo	ParallelDots	https://www.paralleldots.com/text-analysis-apis	$P = 1 + 1 + 0 + 0.5 + 0 = 2.5$

Tabla 17. Aplicación criterios de calidad

A partir de ahora, en todas las tablas relacionadas a la revisión sistemática de literatura gris estarán resaltadas las propuestas seleccionadas como estudios primarios de calidad, como se encuentran resaltadas en la Tabla 17.

2.2.4.5 Solución a las preguntas de investigación

En esta sección se reportan los resultados de la investigación basada en la literatura gris a través de tres fases, cada fase responde una de las preguntas de investigación mencionadas en la sección de 2.2.1 Preguntas de investigación.

Fase 1. ¿Cuáles algoritmos, herramientas o servicios son utilizados en el proceso de clasificación de polaridad de las propuestas existentes?

Se toma como punto de partida las propuestas primarias obtenidas en el proceso de revisión sistemática de literatura gris, debido a que se relacionan ampliamente con el estudio investigativo propuesto. Los algoritmos/herramientas utilizados en el desarrollo de las propuestas seleccionadas a partir de la Tabla 16 brindan un panorama de opciones y posibilidades al momento de diseñar un clasificador de sentimientos, influyendo directamente en la medida de precisión que se calcula en el proceso de clasificación de polaridad. En la Tabla 18 queda evidenciado la lista de propuestas clasificadas por algoritmos utilizados en el desarrollo del proceso de clasificación de polaridad y el formato de despliegue.

Algoritmo/Herramienta	Nombre de propuesta	Formato de despliegue
Azure Cognitive Services	Text analytics	Web
IBM Cloud API	Natural Language Understanding	Web
Naive Bayes	Tweepy	Escritorio
MonkeyLearn API	Sentiment Analysis	Web
Sentiment Analysis API	Python NLTK Text Classification	Web
ParallelDots API	Text analytics Demo	Web
UClassify API	Sentiment	Web
Text Analysis API	Text Analysis API Demo	Web
ClieisAPI	Linguakit	Web

Tabla 18. Propuestas clasificadas

El conjunto de propuestas clasificadas de la Tabla 18 sólo presentan ocho servicios o API's en el proceso de clasificación de polaridad y un algoritmo de clasificación conocido como Naive Bayes. El uso de API's en el proceso de clasificación de un algoritmo permite que el tiempo de desarrollo de un clasificador de sentimientos sea más corto con respecto a la implementación de un nuevo clasificador basado en Naive Bayes. Sin embargo, este tipo de servicios web también tienen una limitación en el área de desarrollo, ya que no es posible acceder al código fuente del algoritmo clasificador original. Lo que significa que para desarrollar versiones mejoradas que incrementen la medida de precisión se deben implementar algoritmos o métodos por fuera del proceso de clasificación de polaridad.

PI2. ¿Qué tipo de datos y cuáles datos son arrojados en los resultados de clasificación de sentimiento?

El tipo de dato arrojado en los resultados por un clasificador de sentimiento depende directamente de la implementación y diseño del algoritmo propuesto, debido a que se utilizan los resultados de los clasificadores de sentimiento para calcular una nueva polaridad. Si no se tiene una normalización en el formato y tipo de datos entregados por el clasificador entonces la medida de precisión se verá afectada directamente. La lista de propuestas clasificada por el tipo de datos que arrojan los resultados de sus clasificadores se encuentra descrita en la Tabla 19.

Formato datos / Estructura de datos	Nombre	Formato de despliegue
JSON Polaridad-Medida de precisión-Idiomas-Palabras clave-Entidades	Text analytics	Web
JSON Polaridad-Medida de precisión-Emoción-Palabras clave-Entidades-Categorías-Conceptos-Sintaxis-Roles semánticos	Natural Language Understanding	Web
Cadena de texto Polaridad-Medida de precisión	Tweepy	Escritorio
JSON Polaridad-Medida de precisión	Sentiment Analysis	Web
JSON Polaridad-Medida de precisión	Python NLTK Text Classification	Web

JSON Polaridad-Medida de precisión- Palabras Clave-Emoción-Entidades- Taxonomía	Text analytics Demo	Web
JSON Polaridad-Medida de precisión	Sentiment	Web
JSON Polaridad-Medida de precisión	Text Analysis API Demo	Web
Matriz Polaridad-Medida de precisión- Palabras clave-Multipalabras-Entidades- Frases Frecuentes	Linguakit	Web

Tabla 19. Clasificación por tipos de datos

De las propuestas evaluadas, una de las posibles selecciones es Linguakit, pero no tiene soporte para ser implementada o consumida por una aplicación externa, lo que hace imposible usar los datos de respuesta como datos de entrada para el algoritmo propuesto. Las otras cinco propuestas que no fueron seleccionadas no brindan resultados completos que generen confianza al momento de sustentar la medida de precisión calculada. Se considera resultados completos a las estadísticas mostradas después de calcular la polaridad, en las cuales se debe incluir como mínimo la polaridad, medida de precisión y palabras clave.

PI3. ¿Cuáles clasificadores utilizan técnicas de lógica difusa?

Para la investigación propuesta es importante tener un panorama de estudios que implementen reglas de lógica difusa en el proceso de clasificación de polaridad. Encontrar estudios que usen estas técnicas permite trabajar sobre una base existente de procesos de clasificación de polaridad que se puede aprovechar para tener un punto de partida avanzado para crear nuevas versiones que intenten mejorar la medida de precisión en clasificación de polaridad. En la Tabla 19 se observa que hasta la fecha, febrero de 2019 como resultado de búsqueda de la cadena planteada no se encuentran propuestas que implementen técnicas como las reglas de inferencia en lógica difusa en el proceso del cálculo de la polaridad, por lo que se llega a la conclusión que el estudio propuesto es novedoso al implementar un clasificador de sentimiento que calcula la polaridad mediante técnicas de lógica difusa.

2.3 Caracterización

La detección de polaridad en análisis de sentimiento en Twitter, es una tarea que permite identificar y extraer información de los datos contenidos en sus mensajes,

conocidos como tuit. Un tuit contiene opiniones y pensamientos expresados de forma escrita por los usuarios permitiendo calcular la polaridad, interpretada como un sentimiento positivo, negativo o neutro.

La polaridad es el estado que toma un mensaje al ser interpretado, el cual se mide en distintas métricas numéricas como *Accuracy*⁶, *Recall*⁷ y *F-Score*⁸. Para normalizar la medida de precisión de la polaridad con respecto a las medidas de precisión por defecto que traen los clasificadores de sentimientos seleccionados se ha tomado como base la medida F (*F-Score*). La medida de precisión F es uno de los atributos que permite comparar dos propuestas diferentes con respecto a la precisión en el proceso del cálculo de polaridad. Se han seleccionado una serie de atributos que permite comparar estudios involucrados con análisis de sentimiento, dando como resultado una lista clasificada de las mejores propuestas conocidas como caracterización de clasificadores de sentimiento.

La caracterización se pretendía llevar a cabo a partir de una lista de propuestas seleccionadas provenientes de los resultados obtenidos en las revisiones sistemáticas de la literatura científica y literatura gris, pero de las propuestas seleccionadas en la revisión sistemática de literatura científica no ha sido posible obtener el código fuente original, ni tampoco se obtuvo respuesta de las solicitudes enviadas a sus autores. Debido a ello, se optó por elegir una serie de clasificadores de sentimientos utilizados por la comunidad científica, que cuentan con grandes ventajas en su proceso de clasificación, pero también con limitaciones particulares. En la selección de los nuevos clasificadores se tuvo en cuenta los mismos criterios de calidad que se utilizaron en la literatura científica para elegir sólo las propuestas que se encuentren en el mismo contexto de esta investigación. Los tres clasificadores son:

- SentiStrength.
- Vader Sentiment.
- Senticnet.

El proceso de selección de los mejores clasificadores de sentimiento en la caracterización se desarrolla a partir de una comparación entre las propuestas seleccionadas mediante los atributos de selección establecidos. Los atributos de selección se establecieron de acuerdo a los parámetros más influyentes en el proceso del cálculo de polaridad, lo que permitió identificar y determinar cuáles son

⁶ Accuracy: Medida de precisión que permite conocer el porcentaje de acierto entre los registros obtenidos como positivos de un clasificador y los que están clasificados como positivos dentro del dataset evaluado.

⁷ Recall: Medida de precisión que permite conocer el porcentaje de acierto entre los registros que están bien clasificados entre un clasificador y el dataset evaluado.

⁸ F-Score: Medida de precisión que permite conocer el porcentaje de acierto a nivel global del conjunto de datos tomados del dataset evaluado. Esta medida relaciona el acierto entre los registros que están bien clasificados y los que no, tomando la forma de la media armónica.

las características más relevantes para el proceso de clasificación de polaridad del algoritmo propuesto. Las propuestas seleccionadas se evaluaron con dos datasets formados por mensajes de texto plano vader_twitter, sentistrength_twitter, tweet_semevaltest y stanford_tweets lo que permitió determinar las fortalezas y debilidades en el proceso del cálculo de polaridad. Las propuestas seleccionadas se encuentran listadas en la Tabla 20.

Identificación	Nombre del clasificador
C1	Text Analytics API
C2	Natural Language Understanding IBM API
C3	Tweepy
C4	SentiStrength
C5	Vader Sentiment

Tabla 20. Propuestas seleccionadas

Finalmente, la caracterización se toma como punto de partida en trabajos futuros para continuar con estudios de mejora de precisión de polaridad en análisis de sentimiento, debido a los resultados detallados obtenidos después de comparar cada una de las propuestas más viables del estado del arte.

2.3.1 Selección de atributos

Tomando como referencia los resultados obtenidos en la revisión sistemática y las nuevas propuestas seleccionadas a partir del estudio realizado, se han seleccionado los atributos con mayor influencia en el cálculo de polaridad de un mensaje. En la Tabla 21 se presenta la lista de atributos usados para categorizar los estudios seleccionados.

Identificación	Nombre del atributo	Descripción del atributo
A1	Tamaño del mensaje	Define la capacidad de un clasificador para calcular la polaridad de un mensaje.
A2	Contiene emoticonos	La capacidad de un clasificador para interpretar un mensaje con distintos emoticonos ASCII.

A3	Contiene Hashtag	Capacidad de interpretar palabras precedidas con el carácter “#”.
A4	Elongaciones	Interpretación de mensajes que contienen palabras deformadas por exceso de caracteres.
A5	Contracciones	Capacidad del clasificador de interpretar mensajes que contienen contracciones.
RESULTADO	Precisión de polaridad (medida F1)	Medida de precisión otorgada por el clasificador.

Tabla 21. Lista de atributos

2.3.2 Comparación de propuestas seleccionadas

La comparación de las propuestas seleccionadas a partir de los resultados de las revisiones sistemáticas se lleva a cabo mediante la lista atributos comparativos establecidos en la Tabla 21. Las propuestas seleccionadas obtenidas de la revisión sistemática de literatura gris pasaron por un proceso de implementación y diseño basados en la documentación expuesta por cada propuesta, además de que han fueron probadas directamente desde la demo web y del consumo de servicios web que ofrecen sus API's.

En la Tabla 22, Tabla 23, y Tabla 24 se presentan la caracterización de las propuestas seleccionadas del estado del arte.

		ATRIBUTOS				
		A1	A2	A3	A4	A5
CLASIFICADORES	C1	Textos medianos y largos.	Pocos	Si	Si	Si
	C2	Textos medianos y largos	Pocos	Si	No	Si
	C3	Textos cortos, medianos y largos.	Muchos	Si	No	No
	C4	Textos cortos, medianos y largos.	Muchos	Si	Si	No
	C5	Textos cortos, medianos y largos.	Medianamente	No	No	Si

Tabla 22. Caracterización de las propuestas seleccionadas

Tamaño de los mensajes en caracteres:

Medida	Medida
Ninguno	0
Pocos	1 - 3
Medianamente	4 - 45
Muchos	46 - 116

Tabla 23. Tamaño de mensaje en caracteres

Tipos de emoticonos:

Tamaño	Rango
Corto	1 - 15
Mediano	15 - 145
Largo	145 - 180

Tabla 24. Tipos de emoticonos

Para C1, se concluye que es un clasificador que tiene un rendimiento bastante bueno en textos de tamaño mediano y largo. Para textos cortos no funciona tan bien como se esperaría. Es bastante limitado en la interpretación de emoticonos, sólo reconoce 3 tipos distintos. En cuanto a las demás características no tiene inconvenientes para clasificar, tiene un rendimiento sobresaliente con hashtag, elongaciones y contracciones. Su medida de precisión es bastante baja si analiza un mensaje corto con muchos emoticonos diferentes.

Para C2, este clasificador tiene un rendimiento bueno en textos de tamaño mediano y largo. En textos cortos no tiene soporte (no arroja ningún resultado), no analiza ningún texto corto que contenga al menos 15 caracteres. No arroja resultados con textos que contienen elongaciones en las palabras. En cuanto a la presencia contracciones y hashtag en las palabras, tiene buen rendimiento.

Para C3, se identificó que tiene un alto rendimiento en textos de cualquier tamaño con palabras precedidas de hashtag y muchos emoticonos diferentes. Sin embargo, la precisión de polaridad es deficiente cuando el mensaje contiene elongaciones y contracciones.

Para C4, el clasificador más completo de todos, tiene un rendimiento en el cálculo de la precisión de polaridad alto, excepto cuando hay contracciones en el mensaje. Aunque interpreta el mensaje cuando están presentes, no arroja resultados precisos en el cálculo de la precisión de polaridad.

Para C5, se identificó que es un clasificador que arroja buenos resultados con cualquier tamaño de texto, interpreta alrededor de 45 emoticonos distintos y las

contracciones no son problema. Su precisión se ve afectada con mensajes que contienen hashtag y elongaciones.

Se evaluaron cuatro datasets (vader_twitter, sentistrength_twitter, tweet_semevaltest, stanford_tweets) en cada uno de los clasificadores de sentimiento seleccionados para obtener las precisiones (listen cuales), las cuales se indican en la Tabla 25.

		Text Analytics	Natural Understanding Language	Tweepy	SentiStrength	VaderSetiment
Dataset 1	Accuracy	0.79547619	0.534285714	0.3266667	0.718095238	0.59547619
	F-Score	0.814981015	0.600816327	0.438443209	0.776940467	0.681298068
Dataset 2	Accuracy	0.552333805	0.583451202	0.5264026	0.59806695	0.584629892
	F-Score	0.759701493	0.455134135	0.305084746	0.520393812	0.454826733
Dataset 3	Accuracy	0.684737966	0.594217184	0.5728602	0.598488582	0.639559717
	F-Score	0.664867296	0.477801268	0.323269131	0.540428733	0.5384939
Dataset 4	Accuracy	0.768802228	0.707520891	0.2729805	0.657381616	0.451253482
	F-Score	0.813186813	0.695652174	0.35235732	0.668463612	0.501265823

Tabla 25. Lista de precisiones

- Dataset 1: vader_twitter.
- Dataset 2: sentistrength_twitter.
- Dataset 3: tweet_semevaltest.
- Dataset 4: stanford_tweets.

De las precisiones obtenidas al evaluar cada clasificador de sentimiento con cada uno de los datasets seleccionados, se observa que TextAnalytics es superior en tres de los cuatro datasets y arroja un resultado competitivo con el dataset 2. SentiStrength supera en precisión al resto de clasificadores en el segundo dataset y arroja resultados competitivos con el dataset 1 y dataset 4. Además, el clasificador de IBM Watson Natural Language Understanding arroja un resultado competitivo con el dataset 4.

Cada clasificador de sentimiento cuenta con características propias descritas en la Tabla 22, que le permiten clasificar un mensaje con una medida de precisión alta o baja según el escenario en donde se encuentre, características que son vistas como fortalezas o debilidades del clasificador. Algunos clasificadores arrojan una medida de precisión bastante alta cuando se evalúan mensajes con textos cortos, otros ni siquiera son compatibles con textos cortos algunos son capaces de identificar distintos tipos de emoticonos, otros sólo unos cuántos, algunos son capaces de interpretar palabras con símbolos o contracciones, otros ni siquiera arrojan

resultados cuando las contracciones están presentes en un mensaje. Cada clasificador arroja como resultado una precisión de polaridad diferente de acuerdo a las características del tuit analizado, entre más parecidas sean las características del tuit a las características del clasificador, más alta será la precisión de polaridad resultante.

Además de estos aspectos cuantitativos, también existen otras características no funcionales como la licencia de uso. Aunque la mayoría de clasificadores son de uso libre, Text Analytics y Natural Language Understanding son demos con tiempo de caducidad, restricción de uso comercial y soporte directo con la desarrolladora, características que pueden ser desbloqueadas pagando la licencia en su sitio web oficial.

Finalmente, al analizar los resultados de la tabla 25 se puede indicar que no hay un clasificador que sea mejor que otro, sólo hay uno que es más adecuado dependiendo del escenario en que se encuentre, es decir, cuando las características del tuit son similares a las características naturales del clasificador usado.

Capítulo 3

Algoritmo

La precisión de polaridad en el proceso de clasificación de sentimientos varía en relación a las características naturales que posee el clasificador y las características que posee el tuit que se analiza, por lo tanto, un clasificador de sentimiento logra su mayor desempeño de precisión cuando se encuentra en escenarios en los cuales sus características son similares a las características del tuit. Por el contrario, en los escenarios en donde las características de un tuit no son similares a las características del clasificador de sentimientos, no se obtienen resultados precisos, de este modo, por ejemplo, si un tuit con menos de diez caracteres de tamaño es evaluado en un clasificador que soporta como mínimo quince caracteres, los resultados arrojados no tendrán una medida de precisión alta. En efecto, es interesante diseñar un clasificador de sentimientos que intente abarcar el mayor número de escenarios posibles, en donde las características de un tuit no sean limitaciones para el proceso de clasificación y la precisión de polaridad pueda verse aumentada con respecto a los clasificadores seleccionados en la caracterización (ver Tabla 20).

Se buscó aumentar la precisión de la polaridad en análisis de sentimientos mediante un algoritmo que funciona a partir de los resultados entregados por cada uno de los clasificadores de sentimientos de la Tabla 25, dando como resultado una polaridad unificada. Debido a que algunos clasificadores de sentimientos de acuerdo al escenario en que se encuentren arrojan una precisión de polaridad más alta que otros, se diseñó un módulo de reconocimiento de características para el tuit analizado, que permitirá tener una idea sobre sus características presentes y poder designar el clasificador más adecuado para clasificarlo. Con las características del tuit, el algoritmo propuesto busca aprovechar las ventajas y moderar las desventajas de cada clasificador de sentimiento mediante un sistema de prioridades. El sistema de prioridades intenta dar más peso matemático al clasificador que mejor se adapte a las características del tuit y dar menos peso al clasificador que tenga características distintas al tuit, debido a que probablemente no arrojará resultados precisos. Mediante este sistema se busca obtener una polaridad coherente en relación a las características del tuit con las características de los clasificadores. El diseño de este sistema de prioridades estará basado en técnicas de lógica difusa que funcionan mediante reglas de inferencia creadas manualmente de acuerdo a las características de cada clasificador de sentimientos.

El sistema de prioridades infiere los pesos de cada clasificador, el cual será utilizado junto a la precisión de polaridad calculada por un sistema estadístico propio, diseñado para calcular la polaridad final resultante.

Para llevar a cabo el desarrollo del algoritmo propuesto se siguió la metodología XP [13], la cual se divide en Exploración, Planificación, Iteración, Producción, Mantenimiento y Ciclo de muerte. En Exploración se definieron las historias de usuario y las reglas de lógica difusa. En Planificación se priorizaron las historias de usuario y se estimó el esfuerzo de cada una de ellas. En Iteración se diseñó la arquitectura, el pseudocódigo, la implementación de las historias de usuario, la elaboración y aplicación de pruebas de aceptación y el diseño de la interfaz gráfica para el prototipo software. En Producción se integró el prototipo software con el algoritmo propuesto. En la fase de mantenimiento se optimizó el algoritmo propuesto y en la fase Ciclo de muerte se da por terminado el algoritmo con la documentación del algoritmo y el prototipo software.

3.1 Exploración

En la fase de exploración se definió las historias de usuario del algoritmo junto a las reglas de lógica difusa. La definición de las 3.1.1 Historias de usuario se definen a continuación y la definición de reglas de inferencia se muestra en el módulo de 3.1.2 Lógica Difusa.

3.1.1 Historias de usuario

El prototipo software debe contar con al menos dos funcionalidades básicas que están implementadas en el algoritmo propuesto, como clasificar un dataset y mostrar los resultados de clasificación. La clasificación y la presentación de resultados de un dataset o de un tuit es una funcionalidad que está presente en todas las aplicaciones web y/o herramientas de análisis de sentimientos seleccionadas en el ítem 2.3 Caracterización.

Un prototipo software en análisis de sentimiento debe contar como mínimo con los siguientes requerimientos:

- Clasificación de una cadena de texto.
- Obtención de resultados que contengan al menos la precisión de polaridad, polaridad y palabras clave.
- Un demo web o herramienta que permita probar al usuario su funcionamiento.

A partir de los requerimientos básicos establecidos que debe tener el prototipo software del algoritmo propuesto se han desarrollado historias de usuario basadas en la siguiente estructura:

- Nombre breve y descriptivo.
- Descripción de la funcionalidad en forma de diálogo o monólogo del usuario describiendo la funcionalidad que desea realizar.
- Criterio de validación y verificación que determinará para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita.

En la Tabla 26 y la Tabla 27, se describen las HU del prototipo software del algoritmo propuesto con sus criterios de aceptación.

Enunciado de la Historia			
Identificador (ID) de la Historia	Rol	Característica / Funcionalidad	Razón / Resultado
HU001	Como un usuario	Necesito consultar la medida de polaridad Accuracy o F-Score, para valores por: (defecto o nuevos valores) en los datasets vader_twitter, sentistrength_twitter, tweet_semevaltest, stanford_tweets.	Con la finalidad de conocer el cálculo de polaridad final de cada uno de los datasets de tuits.
HU002	Como un usuario	Necesito optimizar los valores de los conjuntos difusos, para valores por: (defecto o nuevos valores) en los datasets vader_twitter, sentistrength_twitter, tweet_semevaltest, stanford_tweets.	Con la finalidad de conocer el cálculo de polaridad final con los valores optimizados de los conjuntos difusos para cada uno de los datasets de tuits.
HU003	Como un usuario	Necesito habilitar o deshabilitar el registro de interacción de usuario.	Con la finalidad de tener un histórico de los cálculos de polaridad y los procesos que ejecutados por el algoritmo.
HU004	Como un usuario.	Necesito realizar un menú con las opciones de calcular la polaridad, optimizar los valores, habilitar/deshabilitar los registros y cerrar programa.	Con la finalidad de poder seleccionar una opción a ejecutar.

Tabla 26. Historias de usuarios

Criterios de Aceptación de las Historias de Usuario				
Número (#) de Escenario	Criterio de Aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado
HU001				
1	Consulta exitosa	En caso que el usuario haya seleccionado correctamente las opciones de los valores y la medida de polaridad	Se debe seleccionar la opción: 1. valores por defecto o 2. nuevos valores. Una vez seleccionado los valores debe seleccionar el tipo de medida a consultar Accuracy o F-Score. Se debe mandar un mensaje de espera mientras se realiza el cálculo de la polaridad.	el sistema presentará por consola el cálculo de polaridad total de los 4 datasets.
2	Cancelar consulta	En caso que el usuario no desee realizar la consulta de polaridad.	En caso que el usuario no requiera consultar la polaridad en el algoritmo	El sistema debe presentar un mensaje donde de informe que está cancelando la consulta de la polaridad.
3	Fallo de conexión	En caso que el usuario realizó una acción que impidió realizar el cálculo de polaridad	Cuando se dé clic en el botón de cerrar la ventana en el momento de ejecución del algoritmo	El sistema se cerrará.
HU002				
1	Consulta exitosa	En caso que el usuario haya seleccionado correctamente las opciones de los valores y la medida de polaridad	Se debe seleccionar la opción, donde se escogerá entre: valores por defecto o nuevos valores. Una vez escogido los valores debe seleccionar el tipo de medida a consultar Accuracy o F-Score, después de elegir la opción el sistema debemos ingresar el valor del núcleo y el radio. Se debe mandar un mensaje de espera mientras se realiza el cálculo de la polaridad.	el sistema presentará por consola el cálculo de polaridad optimizado.

2	Cancelar consulta	En caso que el usuario no desee realizar la consulta de polaridad.	En caso que el usuario no requiera consultar la polaridad en el algoritmo	El sistema debe presentar un mensaje donde de informe que está cancelando la consulta de la polaridad.
3	Fallo de conexión	En caso que el usuario realizó una acción que impidió realizar el cálculo de polaridad	Cuando le dé clic en el botón de cerrar la ventana en el momento de ejecución del algoritmo	El sistema se cerrará.
HU003				
1	Habilitar el registro	En caso que el usuario realice la acción de habilitar los registros	Cuando escoja la opción habilitar los registros	el sistema mostrará un mensaje donde se da aviso de la habilitación de los registros
2	Deshabilitar el registro	En caso que el usuario realice la acción de deshabilitar los registros	Cuando escoja la opción deshabilitar los registros	el sistema mostrará un mensaje donde se da aviso de la deshabilitación de los registros
HU004				
1	Seleccionar un item	En caso que una categoría se seleccione.	Cuando se despliegue el listado de categorías a seleccionar.	A continuación se ejecutará la categoría seleccionada.
2	Cerrar menú	En caso que no se quiera ejecutar el programa	N/A	A continuación se muestra un mensaje de confirmación de cierre del programa y se cerrará el programa.

Tabla 27. Criterios de Aceptación de las HU

3.1.2 Lógica Difusa

El módulo de lógica difusa implementado mediante XFuzzy permitió obtener un valor numérico a partir del proceso de defuzzificación⁹ que funciona a partir de un conjunto de reglas de inferencia definidas para cada uno de los clasificadores, es decir, de acuerdo a las características de cada clasificador, se generarán sus propios conjuntos difusos y reglas de inferencia. Los conjuntos difusos contienen funciones matemáticas que se evalúan a partir de límites numéricos, los cuales son establecidos de forma manual. Las reglas de inferencia son establecidas de acuerdo a las características que posee cada del clasificador.

⁹ Defuzzificación: Proceso que permite obtener un valor numérico o cuantificable a partir de un conjunto de reglas difusas.

Por lo tanto, cada clasificador tendrá un valor numérico acorde a la similitud de características que exista entre sus características y las características del tuit. El valor numérico obtenido será conocido como peso (*weight*), el cual es la prioridad con la que un clasificador prima con respecto a los demás.

3.1.2.1 Definición de reglas de inferencia

A partir de la caracterización construida anteriormente se definió un conjunto de reglas de inferencia específicas para cada clasificador, en el cual las reglas del clasificador *X* no son las mismas reglas que las del clasificador *Y*, debido a que cada conjunto de reglas está relacionado directamente con las fortalezas y debilidades de cada uno de ellos. El número total de reglas de inferencia por clasificador es de 95, dentro de las cuales se abarca todas las combinaciones posibles de las cinco características definidas que podría tener un tuit. A continuación, en el Gráfico 1, se muestra la estructura lógica que se debe seguir para definir las reglas de inferencia de cualquier clasificador:

```
if(size == Short & emo == several & has == no & elo == no & con == yes) -> weight = E;
if(size == Short & emo == several & has == no & elo == no & con == no) -> weight = E;
if(size == medium & emo == nothing & has == yes & elo == yes & con == yes) -> weight = A;
if(size == medium & emo == nothing & has == yes & elo == yes & con == no) -> weight = A;
if(size == medium & emo == nothing & has == yes & elo == no & con == yes) -> weight = A;
```

Gráfico 1. Conjunto de reglas de inferencia

En el Conjunto de reglas de inferencia, se puede observar la estructura de las reglas de inferencia, para este caso *size* es el tamaño del tuit, *emo* es el número de emoticonos, *has* si el tuit tiene hashtag, *elo* si el tuit tiene elongaciones y *con* si el tuit tiene contracciones. El resultado se conoce como *weight* y refleja la prioridad con que debe ser tratado cada clasificador, viene dado en la escala A, B, C, D y E, donde E es baja prioridad y A es la más alta.

Además de definir las reglas de inferencia también se definieron los rangos numéricos para cada variable difusa (*size*, *emo*, *has*, *elo*, *con*), los cuales pueden variar de acuerdo a las características de cada clasificador. Por ejemplo, para el clasificador 1 (*Natural Language Understanding*) no se le podría considerar un tuit de tamaño *short* con el mismo número de caracteres que para el clasificador 2 (*Text Analytics*), debido a que el clasificador 1 no soporta tuits de menos de 16 caracteres, por lo tanto, no es correcto definir una regla difusa con los mismos valores que para el clasificador 2. Los rangos se encuentran definidos en el Gráfico 2.

```

type Size [0.0,180.0;256] {
  Short xfl.bell(0.0,15.0);
  medium xfsg.trapezoid(15.0,40.0,120.0,144.0);
  large xfl.bell(180.0,54.0);
}

type Emoji [0.0,116.0;256] {
  nothing xfl.rectangle(0.0,0.99);
  few xfl.rectangle(1.0,3.0);
  medium xfl.trapezoid(2.0,8.0,40.0,45.0);
  several xfl.rectangle(45.0,116.0);
}

type Binary [0.0,1.0;256] {
  no xfl.singleton(0.0);
  yes xfl.singleton(1.0);
}

type Weight [0.0,1.0;256] {
  E xfl.bell(0.0,0.15);
  D xfl.bell(0.25,0.15);
  C xfl.bell(0.5,0.15);
  B xfl.bell(0.75,0.15);
  A xfl.bell(1.0,0.15);
}

```

Gráfico 2. Rangos de las reglas de lógica difusa

En el Gráfico 2. Rangos de las reglas de lógica difusa, se asignaron los rangos numéricos para cada variable difusa, cada valor difuso de cada variable se encuentra asociado a una función matemática. La función matemática asociada a cada valor se asignó de acuerdo a la naturaleza de las características que se presentan en los tuits, entre las funciones matemáticas asignadas se encuentran la función de campana, trapezoidal, rectangular y conjunto unitario.

El primer tipo de dato para la regla de inferencia denominada Size, está compuesto por un rango entre 0 y 256 caracteres. Este rango se divide en tres partes, conocidas como short, medium, large. Cada parte está forma un rango numérico y tiene asignada una función matemática, si observamos short, tiene asignada la función matemática bell, que oscila entre 0.0 y 15.0. Igualmente, para medium y large, que contienen las funciones matemáticas trapezoid y bell entre los rangos 15.0, 40.0, 120.0, 144.0 y 180.0, 54.0 respectivamente. Bajo esta misma lógica está diseñado Emoji, Binary y Weight, que se pueden observar en el Gráfico 2.

Finalmente, se creó un total de 475 reglas entre todos los clasificadores de sentimientos seleccionados, reglas en las cuales se basa el sistema difuso para inferir la prioridad con la que un clasificador deba ser tratado frente a otro dependiendo el escenario en donde se encuentre.

3.2 Planificación

En la fase de planificación se priorizaron las historias de usuario, dando más importancia al desarrollo de la obtención de la precisión de polaridad. La integración del módulo de obtención de la precisión de polaridad (conformado internamente por el proceso de clasificación que tiene cada uno de los clasificadores seleccionados), el módulo de lógica difusa, el módulo de identificación de características y el módulo estadístico permitieron la implementación de toda la historia de usuario HU001, que está relacionada a la obtención de precisión de polaridad.

La historia de usuario HU002 relacionada a la optimización, integra los módulos de la historia de usuario HU001 de obtención de precisión de polaridad más un algoritmo de optimización llamado Hill Climbing, que permite buscar los rangos de los conjuntos difusos que mejor se ajustan al momento de clasificar un tuit, ya que por defecto son ingresados manualmente, por lo tanto, a esta historia de usuario se le asignó una prioridad más baja, debido a que primero se debía implementar el algoritmo de clasificación para que esta fuese posible implementarla.

Con respecto a las historias de usuario HU003 historial de registro y HU004 salir, se dejaron como prioridad baja, debido a que no influían directamente en los resultados de precisión de polaridad sino en usabilidad y utilidades al momento de usar el prototipo software.

3.3. Iteración

La fase de iteración se dividió en tres partes. La primera parte se conformó por el diseño del algoritmo en pseudocódigo, la arquitectura del algoritmo, la arquitectura del prototipo software y el diseño de la interfaz del prototipo software. La segunda parte incluyó la elaboración y aplicación de las pruebas de aceptación. La tercera parte integró el prototipo software con el algoritmo propuesto.

3.3.1 Pseudocódigo del algoritmo

A continuación, se presenta el algoritmo propuesto en pseudocódigo y su respectiva descripción. Ver Tabla 28.

Algorithm SentiFuzzy

```

INPUT
1. Dataset D [ ] // Array of datasets
2. Classifiers C [ ] // Selected classifiers
ASSIGN
3. Characteristics CH [ ] // Characteristics of a tweet
4. Result R [ ] // Results of every classifier
5. Weight W [ ] // Weight for each classifier
OUTPUT
6. FinalAccuracy FA [ ] // List of final accuracy for each dataset
START
7. for each dataset in D do
8.   for each tweet in dataset do
9.     for each classifier in C do
10.      CH ← getTweetCharacteristics(tweet) // identify the characteristics of each
          tweet
11.      R[R.length] ← C.classify() // getting accuracy scores
12.     end for each
13.     W ← fuzzyLogicProcess(CH) // getting weight scores
14.    end for each
15.    FA[FA.length] ← statisticProcess( W, R) // getting the final accuracy
16.  end for each
17. return FA
END

```

Tabla 28. Pseudocódigo del algoritmo

De la Tabla 28, en el bloque INPUT conformado por la línea 1 y 2 se encuentran declaradas las variables que son leídas como entradas, en este caso D es un arreglo de datasets y C es un arreglo de clasificadores de sentimiento. Tanto el arreglo de datasets como el arreglo de clasificadores contiene la dirección absoluta de la ubicación en donde se encuentran guardados cada dataset y cada clasificador, para que posteriormente sean cargados como archivos.

El bloque de ASSIGN conformado desde la línea 3 hasta la 5, contiene las variables que se utilizan como asignación a lo largo del flujo del algoritmo. CH es un arreglo que contiene las características de cada tuit de un dataset leído, R es un arreglo que contiene la precisión de polaridad de cada tuit evaluado en cada clasificador y W es un arreglo que contiene los pesos de cada clasificador en cada tuit evaluado.

El bloque OUTPUT conformado por la línea 6 contiene la variable FA que es un arreglo en donde se guardan las polaridades finales de cada clasificador en cada dataset.

En la línea 7 el algoritmo recorre la lista de datasets por medio de un ciclo repetitivo, en el cual se anida en la línea 8 otro ciclo repetitivo que recorre la lista de clasificadores de sentimiento hasta la línea 15. Cada dataset leído desde la ubicación en donde se encuentra almacenado contiene una lista de tuits, la lista de tuits empieza a ser recorrida de la línea 9 hasta la línea 12.

En la línea 10 un tuit leído de la lista de tuits es tomado como parámetro de entrada en el proceso de clasificación llamado *getTweetCharacteristics*, en el cual está implementado el módulo de 3.3.3 que retorna un arreglo como respuesta y es asignado a la variable CH. La línea 11 contiene el método *classify* que está presente en cada uno de los clasificadores. Este método nativo es llamado directamente de la librería o API de cada clasificador de sentimientos seleccionado; el método recibe como parámetro de entrada una cadena de texto y retorna la precisión de polaridad que finalmente es asignada al arreglo de respuestas R.

En la línea 14 el método *fuzzyLogicProcess* es llamado y este ejecuta al módulo de Lógica Difusa, que permite obtener el peso de cada clasificador en cada tuit que es analizado y es asignado al arreglo W. El módulo de Lógica Difusa se explica en el ítem Lógica difusa (sección 3.1.2 Lógica Difusa).

El módulo Estadístico es llamado en la línea 15 mediante el método *statisticProcess*, el cual usa como parámetros de entrada el arreglo de pesos W y el arreglo de precisiones de polaridad R. El módulo Estadístico está explicado en el ítem 3.3.3.2 Cálculo de la polaridad final. Finalmente, el arreglo de precisión de polaridad final FA es retornado por el algoritmo.

En el Gráfico 3 se puede observar el flujo del algoritmo propuesto, desde que un tuit es seleccionado a partir de un dataset leído, pasa por el módulo Characteristics Identifier, en donde se obtienen las características que conforman al tuit leído. El flujo se dirige hacía el módulo Sentiment Classifiers, en donde se calcula y se retorna la polaridad del tuit leído por cada uno de los algoritmos seleccionados en la 2.3 Caracterización. Después de ejecutarse el módulo Sentiment Classifiers el flujo pasa por el módulo Fuzzy Logic, en el cual se calcula y se retorna los pesos de cada clasificador de sentimientos seleccionado, tomando como dato de entrada las características obtenidas en el módulo Characteristics Identifier. Una vez se ha ejecutado el módulo Fuzz Logic el flujo pasa por el módulo Statistics module, en donde se calcula la polaridad final teniendo en cuenta la polaridad calculada en Sentiment Classifiers y los pesos calculados en Fuzzy Logic. Finalmente, la polaridad resultante es retornada como dato de salida.

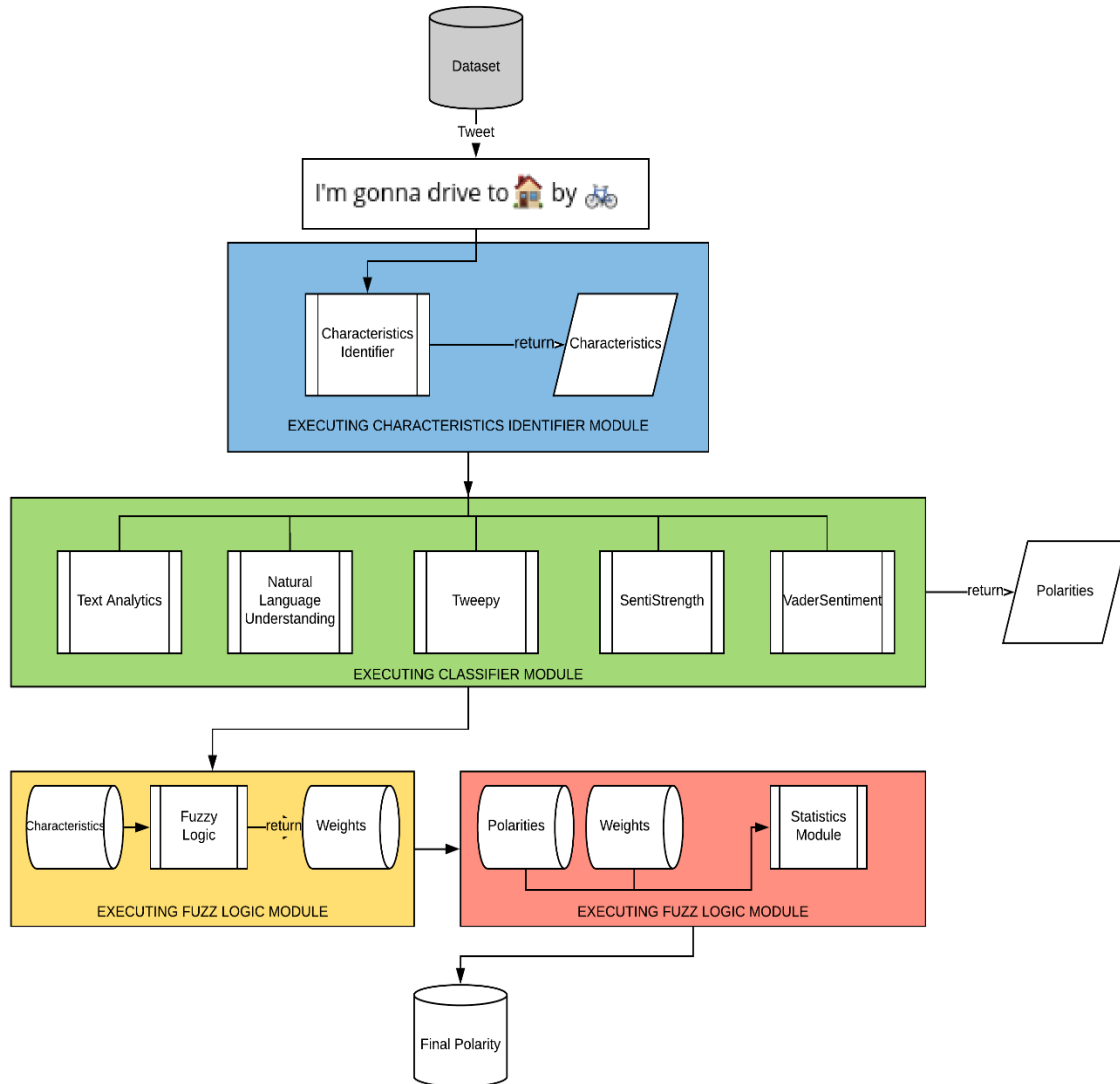


Gráfico 3. Ejemplo del flujo del algoritmo

3.3.2 Arquitectura

Se implementaron los clasificadores de sentimiento seleccionados en la Tabla 20, se desarrolló además un módulo de lógica difusa construido en XFuzzy [58] y algoritmos de búsqueda binaria para los procesos usados en módulo de reconocimiento de características del tuit.

En el Gráfico 4 se muestra la arquitectura del algoritmo propuesto. El algoritmo está conformado por cuatro módulos denominados Sentiment Classifiers, Characteristics Identifier, Fuzzy Logic y Statistic module. Además, cuenta con componente de persistencia, el cual permite cargar datos de entrada como dataset de tuits y guardar datos como la polaridad final. El algoritmo recibe como entrada un tuit que está representado como un dato persistente proveniente de un archivo de texto, el dato

de entrada se evalúa en el módulo Sentiment Classifiers, en el cual se clasifica al tuit y se obtiene la precisión de polaridad de cada clasificador. Seguidamente el tuit pasa por el módulo Identificador de características en donde se identifican los valores característicos del mensaje. El módulo de lógica difusa está contenido dentro del algoritmo y usa la respuesta del módulo de clasificadores de sentimientos e identificador de características. El módulo Statistic module es dependiente de los resultados de los resultados obtenidos en los tres módulos anteriores, es el que se encarga de calcular la polaridad final. Por lo tanto, Statistic module envía la polaridad final al componente de persistencia y este lo guarda como dato de salida en un archivo de texto.

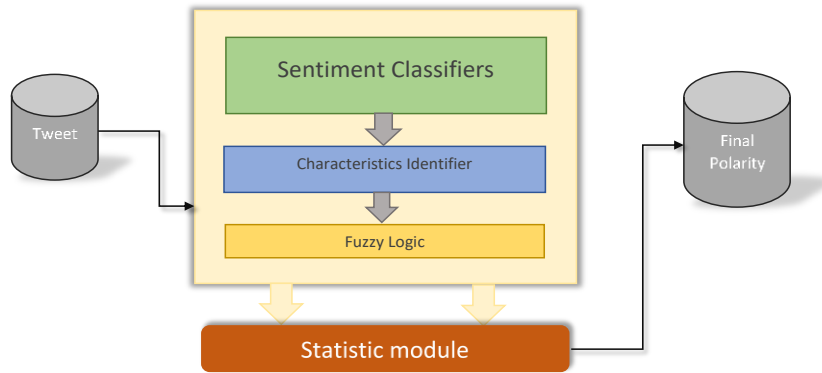


Gráfico 4. Arquitectura del algoritmo

En el Gráfico 5, se observa la estructura del módulo de cálculo de polaridades, en el cual se tomó un conjunto de datasets que pasa por el proceso de clasificación de cada uno de los clasificadores de sentimientos seleccionados. Del proceso de clasificación de cada clasificador se obtuvo las listas de medidas de polaridad que se usan como parámetros de entrada en el proceso estadístico de cálculo de polaridad final. El método que permite obtener la precisión de polaridad de cada clasificador se llama *classify()*.

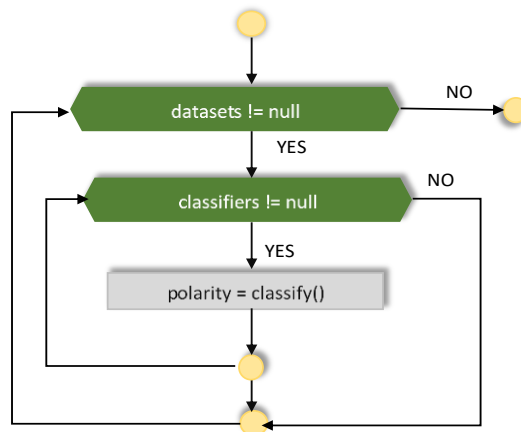


Gráfico 5. Cálculo de polaridades

En el Gráfico 6, se observa que en el proceso de cálculo de polaridad final se identifican las características de cada tuit contenido en cada uno de los datasets seleccionados, el método que permite identificar las características es conocido como *identifyCharacteristics()*. Estas características identificadas del tuit son el parámetro de entrada para el módulo de lógica difusa, del cual se obtiene una lista de pesos que indican al algoritmo propuesto con qué prioridad debe usarse cada algoritmo de clasificación, el método es llamado *fuzzyLogic(ch)*. El proceso estadístico llamado *statistic(weight, ch)* funciona a partir de la lista de pesos y la lista de valores de polaridad obtenidos en el método de clasificación de cada clasificador de sentimientos, el cual calcula la polaridad final.

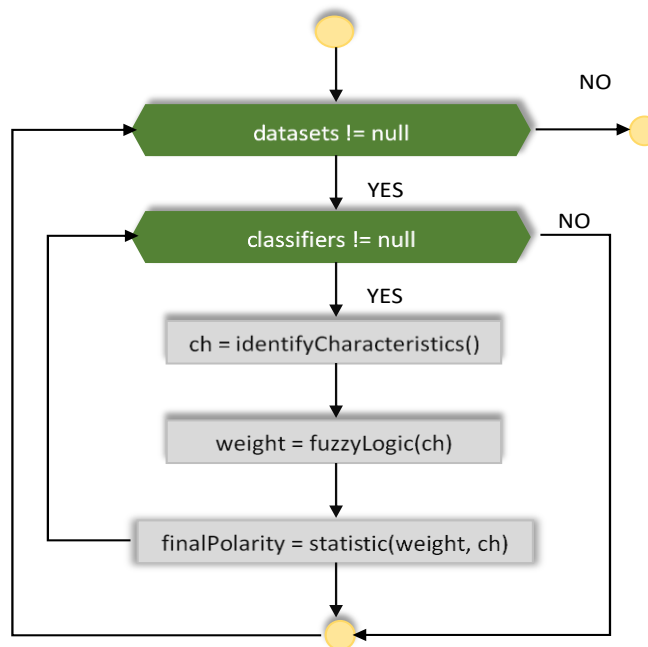


Gráfico 6. Cálculo de polaridad final

3.3.3 Implementación de historias de usuario

La implementación de las 3.1.1 Historias de usuario conllevó a implementar el módulo de detección de características, el módulo de 3.1.2 Lógica Difusa y el módulo estadístico.

3.3.3.1 Módulo de detección de características

El módulo de detección de características para un tuit está compuesto por cinco procesos: cálculo de tamaño, identificación de emoticonos, identificación de hashtag, identificación de elongación de palabras e identificación de contracciones. En el proceso de cálculo de tamaño se llama el método nativo en cada lenguaje que

retorna los caracteres que contiene una cadena, para Python se usó *len()* y para Java-SE se usó el atributo *length* del tipo de dato *String*.

La identificación de emoticonos divide el mensaje en una lista de palabras y determina si algún elemento de la lista de emoticonos coincide o está contenido, en una palabra. El método *Split(" ")* hace referencia al proceso de separar el tuit por espacios. El método *exist(word)* compara si la palabra guardada en la variable *word* se encuentra en el diccionario de emoticonos, el diccionario de emoticonos es una lista de caracteres, por lo tanto, el método *exist(word)* recorre una lista de tipo cadena comparando cada posición con la palabra *word*. El proceso anterior se representa en un diagrama de flujo que se muestra en el Gráfico 7.

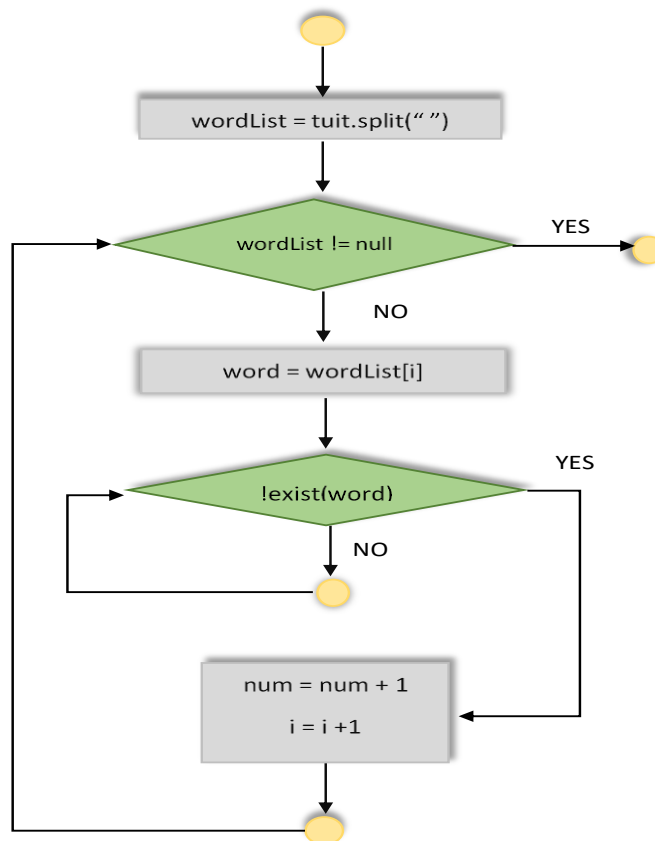


Gráfico 7. Identificación de emoticonos

En el proceso de identificación de hashtag se divide el mensaje en una lista de palabras y determina si cada palabra contiene el carácter "#". El proceso es similar al de identificación de emoticonos, sólo que en lugar de buscar en un diccionario de emoticonos se compara directamente con el carácter '#'. El proceso se muestra en el diagrama de flujo del Gráfico 8.

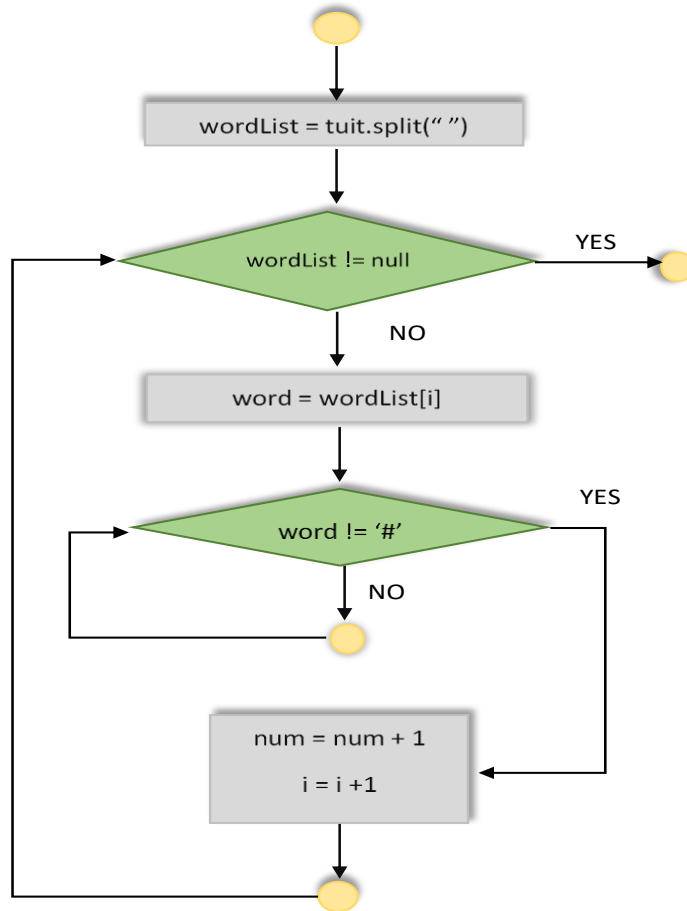


Gráfico 8. Identificación de hashtag

Para el proceso de identificación de contracciones se divide el mensaje en una lista de palabras y determina si una palabra coincide con algún elemento de la lista de contracciones. El proceso es similar al de identificación de emoticonos sólo que en lugar de utilizar una lista de emoticonos se usa una lista de contracciones. El método *contractions(word)* en este caso busca en la lista de contracciones si la palabra *word* está contenida en ella. El proceso se muestra en el Gráfico 9.

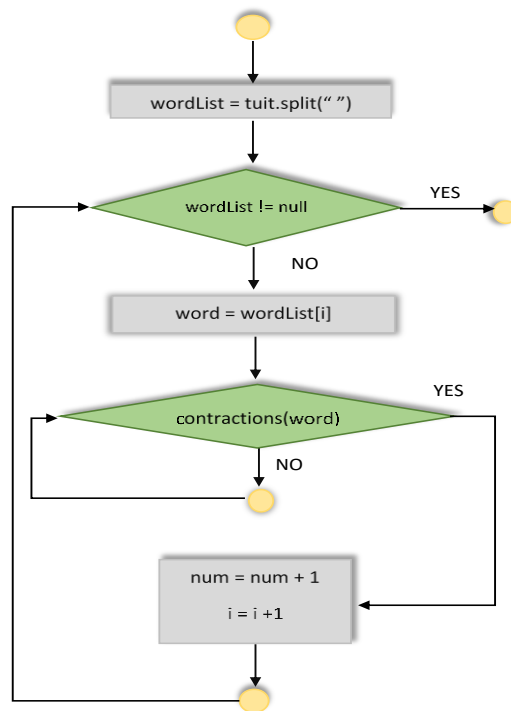


Gráfico 9. Identificación de contracciones

Finalmente, se retorna un arreglo multidimensional de cinco posiciones que contiene en cada posición el tamaño del mensaje, el número de emoticonos y verdadero o falso en las tres posiciones restantes.

3.3.3.2 Cálculo de la polaridad final

La polaridad final entregada es el resultado de un proceso matemático en el cual se involucran la precisión de polaridad y los pesos otorgados por el módulo de lógica difusa. El proceso de cálculo de polaridad se calcula usando la Ecuación 5:

$$\sum_{i=0}^n \frac{W_i}{\sum_{i=0}^n W_i} \times \sum_{i=0}^n P_i = FP$$

Ecuación 5. Cálculo de polaridad

En donde n es el número de clasificadores, W es el valor del peso de cada clasificador, P es el valor de la polaridad de cada clasificador y FP es el valor de la polaridad final.

3.3.4 Pruebas de aceptación

Las pruebas de aceptación representan las actividades que se llevan a cabo dentro del ciclo de vida de desarrollo del software en el que el equipo de desarrollo, los usuarios del producto software tienen que garantizar que el sistema desarrollado corresponda con los requerimientos definidos [59]. La propuesta del algoritmo está compuesta por una serie de iteraciones. Cada iteración se termina ejecutando un conjunto de pruebas de aceptación que permiten al cliente comprobar si está de acuerdo con el resultado de ejecución del algoritmo.

3.3.4.1 Elaboración

Para la elaboración del diseño de casos de prueba, el equipo de pruebas tuvo en cuenta los siguientes aspectos:

- Código: Cada caso de prueba debe tener un identificador (código) único y a que historia de usuario le pertenece el caso de prueba.
- Nombre: Título descriptivo del caso de prueba.
- Descripción: Descripción del caso de prueba, indicando sus elementos, funcionalidades y acciones a ser ejercidas en el caso de prueba.
- Condiciones de ejecución: Son las condiciones que se deben tener en cuenta para la ejecución del caso de prueba.
- Entrada/ejecución: Se especifican los pasos que se deben hacer para la ejecución del caso de prueba.
- Resultado de la prueba: Se especifica la salida que se espera de la ejecución de los casos de prueba con las entradas indicadas.
- Evaluación de la prueba: Se describe el resultado real obtenido de la ejecución del caso de prueba.

3.3.4.1 Aplicación

La aplicación de las pruebas de aceptación fue realizada en 2 computadoras portátiles con las siguientes características:

Ambiente de pruebas

Requerimientos Hardware:

- Asus con procesador core i7 de 6 generación, memoria Ram de 8 GB y sistema operativo Windows 10.
- Asus procesador Amd A10, memoria Ram de 8 GB y sistema operativo Linux.

Requerimientos Software:

Java Runtime Environment

A continuación en la Tabla 29, Tabla 30, Tabla 31 y Tabla 32, se presentan los casos de prueba realizados al algoritmo propuesto.

Casos de prueba
Código: 1 Historia de usuario: HU001
Nombre: Consultar polaridad
Descripción: Permitirá consultar la medida de polaridad
Condiciones de ejecución: Se llevará a cabo una vez el usuario seleccione la medida de polaridad a consultar.
Entrada/ejecución: Los parámetros de entrada: Valores: 1. Default 2. Best values Medida de polaridad: 1. Accuracy 2. F-Score a) En primer lugar, tenemos que ejecutar el algoritmo. b) Seleccionar el valor a consultar. c) Seleccionar la medida de polaridad a consultar.
Resultado esperado: Las medidas de polaridad de los datasets: vader_twitter, sentistrength_twitter, tweet_emevaltest y stanford_tweets.
Evaluación de la prueba: Después de hacer las pruebas, la ejecución del algoritmo arrojó las polaridades de los 4 datasets mencionados anteriormente. La HU quedó completa.

Tabla 29. Caso de prueba 1

Casos de prueba
Código: 2 Historia de usuario: HU002
Nombre: Optimizar valores
Descripción:

Permitirá consultar la medida de polaridad con valores optimizados.
Condiciones de ejecución: Se llevará a cabo una vez el usuario seleccione la medida de polaridad que se desea optimizar.
Entrada/ejecución: Los parámetros de entrada: Valores: 1. Default 2. Best values Medida de polaridad: 1. Accuracy 2. F-Score Optimización Hill-Climbing 1. Valor del núcleo. 2. Valor del Radio. a) En primer lugar, tenemos que ejecutar el algoritmo. b) Seleccionar el valor a consultar. c) Seleccionar la medida de polaridad a consultar. d) Ingresar el valor del núcleo y radio.
Resultado esperado: Las medidas de polaridad con los valores optimizados.
Evaluación de la prueba: Después de hacer las pruebas, la ejecución del algoritmo arrojo las polaridades optimizadas. La HU quedó completa.

Tabla 30. Caso de prueba 2

Casos de prueba
Código: 3 Historia de usuario: HU003
Nombre: Habilitar/deshabilitar registros
Descripción: Permitirá habilitar o deshabilitar el almacenamiento de los registros de las polaridades calculadas.
Condiciones de ejecución: Se llevará a cabo una vez el usuario seleccione la opción de habilitar o deshabilitar los registros.
Entrada/ejecución: Los parámetros de entrada:

<ol style="list-style-type: none"> 1. Habilitar registro. 2. Deshabilitar registro. <ol style="list-style-type: none"> a) En primer lugar, tenemos que ejecutar el algoritmo. b) Seleccionar la opción de habilitar o deshabilitar los registros.
<p>Resultado esperado:</p> <ol style="list-style-type: none"> 1. Que se habiliten el almacenamiento de los registros. 2. Que deshabiliten el almacenamiento de los registros.
<p>Evaluación de la prueba:</p> <p>Después de hacer las pruebas, el algoritmo permite realizar la habilitación o des habilitación de los registros. La HU quedo completa.</p>

Tabla 31. Caso de prueba 3

Casos de prueba
Código: 4 Historia de usuario: HU004
Nombre: Menú
<p>Descripción:</p> <p>El algoritmo de debe permitir seleccionar la opción a ejecutar por medio de un menú.</p>
<p>Condiciones de ejecución:</p> <p>Permitir seleccionar entre varias opciones la opción que desee realizar el usuario.</p>
<p>Entrada/ejecución:</p> <p>Los parámetros de entrada:</p> <p>N/A</p>
<p>Resultado esperado:</p> <p>Visualización del menú</p>
<p>Evaluación de la prueba:</p> <p>Después de hacer las pruebas, el algoritmo muestra el menú con las opciones:</p> <p>Menu</p> <ol style="list-style-type: none"> 1. Run Sentiment Algorithm 2. Opimize rule values 3. Enable/Disable Log 4. Exit <p>La HU quedó completa.</p>

Tabla 32. Caso de prueba 4

3.3.5 Prototipo software

El prototipo software es un entregable funcional del algoritmo propuesto, por tal motivo está conformado por un diseño gráfico que es usable por el usuario final. Se encuentra programado en Java SE, por lo que se puede ejecutar en cualquier Sistema Operativo que sea capaz de interpretar la máquina virtual de Java.

3.3.5.1 Diseño

El diseño de la interfaz del prototipo software está pensado para ser usado por el usuario desde la consola de comandos del sistema operativo en que es ejecutado. Sin embargo, se incluyó un sistema de elección de opciones conocido como *menú*, el cual consta de cuatro opciones *Run Sentiment Algorithm*, *Optimise rule values*, *Enable/Disable log*, *Exit*, que se puede observar en el Gráfico 10. El funcionamiento y la documentación detallada de cada una de las funcionalidades del prototipo software se encuentra en el ítem 3.6.3 Funcionamiento del prototipo software.

```
Menu
1. Run Sentiment Algorithm
2. Opimise rule values
3. Enable/Disable Log
4. Exit
Please select the option:
```

Gráfico 10. Menú principal

El submenú de selección de configuración está conformado por dos opciones, *Default* y *Best values*. A continuación, se muestra en el Gráfico 11.

```
What settings do you want to run?
1. Default
2. Best values
```

Gráfico 11. Menú configuraciones

El submenú de selección de medida de precisión está conformado por dos opciones, *Accuracy* y *F-Score*. En el Gráfico 12 se pueden observar.

```
What measure do you want??
1. Accuracy
2. F-Score
```

Gráfico 12. Menú de medidas de precisión

El submenú de selección de dataset para entrenar está conformado por cinco opciones, All, Dataset 0, Dataset 1, Dataset 2, Dataset 3. La interfaz se puede observar en el Gráfico 13.

```
What dataset do you want to optimise?  
1. All  
2. Dataset 0  
3. Dataset 1  
4. Dataset 2  
5. Dataset 3
```

Gráfico 13. Menú de selección de dataset

Finalmente, los resultados del algoritmo se ven listados de la siguiente forma en el Gráfico 14.

```
running, please wait a moment ...  
THE FINAL VALUES ARE:  
0.8866666666666667  
0.5994813767090995  
0.6430096927879086  
0.83008356545961
```

Gráfico 14. Menú de medidas de precisión

3.4 Producción

En la fase de 3.4 Producción se integró el algoritmo propuesto con el prototipo software diseñado. El prototipo software está conformado por la implementación de las 3.1.1 Historias de usuario definidas anteriormente. Cada historia de usuario está asociada a cada una de las funcionalidades del prototipo software. Al momento de integrar el algoritmo no se tuvo ningún inconveniente ya que todos los datos se cargan mediante un sistema de lectura y escritura de archivos de texto.

3.5 Mantenimiento

En la fase de mantenimiento se intentó optimizar los resultados obtenidos al ejecutar el algoritmo con valores ingresados manualmente para los rangos de los conjuntos difusos. Por este motivo se implementó un algoritmo de optimización que ayudó a encontrar los valores de los conjuntos difusos que mejor se ajustarán a la configuración del módulo de lógica difusa.

3.5.1 Optimización y refinamiento de valores numéricos de entrada

Para tratar de mejorar la precisión se buscó optimizar los valores numéricos de los límites de los conjuntos difusos y se implementó un algoritmo de optimización llamado Hill-Climbing [60]. El propósito de este algoritmo es obtener los rangos numéricos más apropiados en cada conjunto difuso sin establecerlos de forma manual.

Por lo tanto, para la optimización de rangos numéricos se creó un nuevo módulo en el cual se implementó Hill-Climbing. Este módulo de optimización funciona a partir de los valores numéricos ingresados manualmente en los conjuntos difusos para cada una de las funciones matemáticas que infieren a partir de las reglas difusas establecidas.

3.5.2 Pseudocódigo de optimización

En la Tabla 33, se presenta la optimización de el algoritmo propuesto en pseudocódigo y su respectiva descripción.

<pre> INPUT 1. Solution BEST // The init values established 2. BEST = new Solution (initValues) ASSIGN 3. Solution S 4. Function F // In this case the optimization function is the proposed algorithm 5. Accuracy A // The accuracy from best solution 6. Accuracy B // The accuracy from new solution 7. optimal = 1 // 100% START 8. A = BEST.fitness() 9. while times < n do 10. if (A = optimal) then 11. exit() 12. end if 13. S.newRandomSolution() 14. S.tweak(F) // it tweaks an ensemble values 15. B = S.fitness() 16. if (B > A) then 17. A = B 18. BEST = S 19. end if 20. times = times + 1 21. end while 22. return BEST END </pre>
--

Tabla 33. Pseudocódigo de optimización

De la Tabla 33, el bloque INPUT del algoritmo de optimización conformado por la línea 1 y línea 2, se recibe como parámetro un conjunto inicial de valores numéricos definidos manualmente en el archivo de configuración del módulo de 3.1.2 Lógica Difusa, estos valores representan los límites de cada función matemática definida en cada uno de los conjuntos difusos y se encuentran estructurados en un arreglo dentro de un tipo de dato llamado Solution. De la línea 3 a la línea 7 se declaran las variables que serán usadas a lo largo del flujo del algoritmo. S es la solución que se genera aleatoriamente, F es la función a evaluar, que en este caso es una instancia del algoritmo propuesto; A es el fitness o la mejor precisión de polaridad que el algoritmo ha encontrado en todas las repeticiones; B es el fitness o la precisión de polaridad de la nueva solución que se calcula aleatoriamente, la variable *optimal* va de 0 hasta 1, indicando el número óptimo al que se quiere llegar, en este caso 1 representa el 100% de precisión de polaridad.

Una vez los valores iniciales son tomados como punto de partida, en la línea 8 se calcula el fitness (precisión de polaridad) de la mejor solución hasta el momento mediante el método *fitness()* el cuál ejecuta la instancia de objeto del algoritmo propuesto, con los valores de rango difusos que tiene asignada la solución en ese momento. De la línea 9 hasta la línea 21 se ejecuta un ciclo *while* que itera *n* veces, en donde *n* es un número dado por el usuario. Desde la línea 10 hasta la 12 se declara un condicional en el cual se evalúa si A es mejor o igual que el valor óptimo establecido, es decir al 100%, de ser así el algoritmo termina mediante el método *exit()*, que finaliza la ejecución, sino el ciclo *while* seguirá iterando. En la línea 13 se encuentra una solución aleatoria mediante el método *newRandomSolution()*, en el cual cada posición del arreglo contenido en el tipo de dato Solution es llenado con valores aleatorios. La línea 14 mediante el método *tweak()* modifica esta solución aleatoria encontrada declarada como S, con un pequeño cambio numérico en cada valor del arreglo contenido como atributo.

En la línea 15 se calcula el fitness de la solución S y es asignado a la variable B. En la línea 16 hasta la línea 19 hay un condicional que evalúa si el fitness de la variable B es mejor que el fitness de la variable A. Si la condición se cumple entonces A pasa a tener el valor de B y la mejor solución que es BEST pasa a ser reemplazada por S. La línea 20 solamente se utiliza para incrementar un contador. Finalmente, en la línea 22 se retorna la solución BEST que es donde se guarda el mejor resultado.

De esta forma, mediante la implementación de Hill-Climbing se obtuvo un conjunto de valores numéricos que permitieron aumentar la precisión de polaridad cuando son usados como los rangos de las funciones matemáticas en los conjuntos difusos.

3.6 Ciclo de muerte

La fase final de la metodología *eXtreme Programming* conocida como ciclo de muerte está conformada por la terminación del algoritmo, prototipo software y la documentación para la configuración y uso del prototipo software.

3.6.1 Terminación del algoritmo

El algoritmo propuesto se da por terminado al obtener resultados satisfactorios, resultados que superaron en la mayoría de escenarios la medida de precisión más alta entre los algoritmos seleccionados en la revisión sistemática.

3.6.2 Terminación del prototipo software

La terminación del prototipo software se da por concluido cuando es integrado con el algoritmo propuesto y las pruebas de aceptación han sido superadas satisfactoriamente. El prototipo software cuenta con dos opciones que no se encuentran incluidas en ninguno de los prototipos software de clasificación de sentimientos del estado del arte, las cuales son optimizar la base de clasificación difusa y el historial de registros de cada acción ejecutada por el usuario.

3.6.3 Funcionamiento del prototipo software

El prototipo software cuenta con cuatro funcionalidades Run Sentiment Algorithm, Optimise rule values, Enable/Disable log y Exit.

La primera opción llamada *Run Sentiment Algorithm* permite ejecutar el algoritmo propuesto dentro del prototipo software mediante la consola de comandos. La funcionalidad retorna cuatro precisiones de polaridad, cada precisión corresponde a uno de los datasets seleccionados en la Tabla 25. Para ejecutar esta opción el usuario debe ingresar '1' en el menú principal, como se muestra en el Gráfico 15.

```
Menu
1. Run Sentiment Algorithm
2. Opimise rule values
3. Enable/Disable Log
4. Exit
Please select the option:
```

Gráfico 15. Paso 1. Ejecución opción 1

Una vez se ha digitado '1' se presenta el menú de selección de configuración, en donde se muestra *default* y *best*, *default* permite cargar al algoritmo los valores por defecto ingresados manualmente en los conjuntos difusos, *best* permite cargar los

valores más adecuados para los conjuntos difusos que el módulo de optimización ha encontrado hasta el momento. Se puede observar en el Gráfico 16.

```
What settings do you want to run?  
1. Default  
2. Best values  
2
```

Gráfico 16. Paso 2. Ejecución opción 1

Cualquiera de las dos opciones que se seleccione en el Gráfico 16, mostrará enseguida el menú de selección de medida de precisión Accuracy o F-Score. A continuación, se muestra en el Gráfico 17.

```
What measure do you want??  
1. Accuracy  
2. F-Score  
2  
Starting algorithm core for running...
```

Gráfico 17. Paso 3. Ejecución opción 1

Finalmente se muestran las precisiones para cada dataset seleccionado, como se muestra en el Gráfico 18.

```
THE FINAL VALUES ARE:  
0.9161085653859712  
0.5658062867365193  
0.6193977591036415  
0.8381962864721486
```

Gráfico 18. Paso 4. Ejecución opción 1

La segunda opción llamada *Optimise rule values* permite optimizar los valores numéricos de los conjuntos difusos. Entre los valores estén mejor adaptados a las funciones matemáticas asociadas a los conjuntos difusos la precisión será más alta. Para seleccionar esta opción se debe digitar '2', seguidamente se muestra el menú de configuraciones, como se muestra en el Gráfico 19.

```
Menu
1. Run Sentiment Algorithm
2. Opimise rule values
3. Enable/Disable Log
4. Exit
Please select the option:
2

Please wait a moment while it is running ...

What settings do you want to run?
1. Default
2. Best values
```

Gráfico 19. Paso 1. Ejecución opción 2

Una vez se ha seleccionado una de las dos configuraciones se despliega el submenú de que contiene para este caso cinco opciones All, Dataset 0, Dataset 1, Dataset 2, Dataset 3 como se observa en el Gráfico 20. La opción All es para optimizar todos los clasificadores en todos los datasets, Dataset 0 es para optimizar todos los clasificadores en el dataset 1, *Dataset 1* es para optimizar todos los clasificadores en el dataset 2, *Dataset 2* es para optimizar todos los clasificadores en el dataset 3, *Dataset 3* es para optimizar todos los clasificadores en el dataset 4.

```
What dataset do you want to optmise?
1. All
2. Dataset 0
3. Dataset 1
4. Dataset 2
5. Dataset 3
```

Gráfico 20. Paso 2. Ejecución opción 2

Una vez se ha seleccionado una de las cinco opciones se debe digitar el número de repeticiones del núcleo iterativo del algoritmo de optimización Hill Climbing, además de digitar el rango de búsqueda. En el Gráfico 21 se puede apreciar el proceso descrito.

```

Please type a value to loop core: 2
Please type a value for radio: 5
Optimising ...
C L A S S I F I E R  0
DATASET: 0
    
```

Gráfico 21. Paso 3. Ejecución opción 2

El algoritmo de optimización se ejecuta tomando como función de evaluación el algoritmo de clasificación de sentimientos propuesto, finalmente se muestra al usuario si desea guardar la nueva configuración de la optimización. El formato visual de los resultados se puede observar en el Gráfico 22.

```

Optimising ...
C L A S S I F I E R  0
DATASET: 0
DATASET: 1
Score 0.5639979602243753 has been improved to 0.5642857142857143
DATASET: 2
DATASET: 3
C L A S S I F I E R  1
DATASET: 0
DATASET: 1
DATASET: 2
DATASET: 3
C L A S S I F I E R  2
DATASET: 0
DATASET: 1
DATASET: 2
DATASET: 3
C L A S S I F I E R  3
DATASET: 0
DATASET: 1
DATASET: 2
DATASET: 3
C L A S S I F I E R  4
DATASET: 0
DATASET: 1
DATASET: 2
DATASET: 3
Do you want to save data?
1. yes
2. no
    
```

Gráfico 22. Paso 4. Ejecución opción 2

La opción número tres Enable/Disable Log se puede seleccionar digitando '3' en el menú principal, como se muestra en el Gráfico 23. Esta funcionalidad permite guardar las interacciones del usuario con el prototipo software, es decir, los resultados obtenidos, las opciones que ha seleccionado, los datasets que ha optimizado y los valores de configuración del módulo de lógica difusa. Mediante esta funcionalidad el usuario puede identificar cuál ha sido la mejor configuración para los conjuntos difusos hasta el momento e incluso exportarla para continuar la optimización en otro equipo de cómputo, el archivo `.log` se puede visualizar con cualquier lector de texto.

```
Menu
1. Run Sentiment Algorithm
2. Opimise rule values
3. Enable/Disable Log
4. Exit
Please select the option:
```

Gráfico 23. Paso 1. Ejecución opción 3

Capítulo 4

Resultados

Los resultados obtenidos en el trabajo propuesto indica una mejora en la precisión de polaridad en análisis de sentimientos en los datasets seleccionados de Twitter. Los resultados finales varían de acuerdo a la medida de precisión empleada a partir de una matriz de confusión. Para la investigación se tuvo en cuenta *Accuracy* y *F-Score*. Los resultados obtenidos con la medida *Accuracy* demuestran una mejora en tres datasets y resultados competitivos en uno, de los cuatro datasets totales evaluados, logrando que el algoritmo propuesto sea el más preciso en la mayoría de los escenarios que se presentaron. Los resultados obtenidos con la medida *F-Score* presentan resultados competitivos en dos datasets y mejoras en los otros dos, de los cuatro datasets totales evaluados. La mejora con medida *F-Score* alcanza hasta el 92% de precisión, lo que indica que el clasificador propuesto es uno de los más precisos del estado del arte en algunos escenarios específicos. Este capítulo se encuentra dividido en seis ítems donde se detallan los resultados obtenidos durante la investigación.

4.1 Evaluación de resultados

Los datasets evaluados en cada uno de los clasificadores de sentimientos no han sido modificados o alterados para el funcionamiento general del algoritmo propuesto, debido a que desde el principio de la investigación no se buscó la implementación de métodos y técnicas con preprocesamiento de datos. Los datasets se encuentran en su forma original desde el momento que son descargados de la web [referencia]. Los cuatro dataset de evaluación utilizados son: *vader_twitter*, *sentistrength_twitter*, *tweet semevaltest* y *stanford_tweets*. El número de tuits en los datasets son 1,2,3,4 respectivamente. Cada dataset está compuesto por dos archivos, uno en donde se encuentran guardados los tuits y otro en donde se encuentra la precisión de la polaridad de cada tuit. Los dos archivos son de texto plano y se encuentran en formato de texto (txt) lo que permite que sean de fácil legibilidad para las personas desde cualquier editor de texto.

En cuanto al estándar de la precisión de la polaridad, en todos los datasets se encuentran valores numéricos entre -1 y 1, en donde 1 es positivo, -1 negativo y 0 neutro. Para seguir el estándar de este formato primero se ha hecho una conversión

para cada uno de los resultados obtenidos por los clasificadores de sentimiento, ya que estos retornan las precisiones de polaridad en diferentes escalas numéricas.

En la Tabla 34, se describe el tipo de escalas manejadas por cada clasificador.

	Text Analytics	Natural Understanding Language	Tweepy	SentiStrength	VaderSetiment	Sentiment Algorithm (Proposed)
Range	[0, 1]	[-1, 1]	[-1, 1]	[-4, 4]	[0, 1]	[0, 1]

Tabla 34. Tipos de escalas por clasificador

Para hacer la conversión de escalas se utilizó la ecuación matemática Ecuación 6.

$$Z = \frac{x - \min}{\max - \min}$$

Ecuación 6. Conversión de escalas

en donde *min* es el límite mínimo, *max* el límite máximo y *x* es el valor a convertir.

En efecto, se tiene una precisión de polaridad que oscila entre -1 y 1 y se quiere estandarizar el valor $x = 0.5$ en la escala de 0 a 1, se obtuvo el siguiente resultado en la Ecuación 7:

$$Z = \frac{(0.5) - (-1)}{1 - (-1)} = \frac{1.5}{2} = 0.7$$

Ecuación 7. Resultado de precisión

De la Ecuación 7, se deduce que $x = 0.5$ entre los rangos -1 y 1 es equivalente a $x = 0.75$ entre los rangos 0 y 1.

Mediante este sistema de conversión de escalas se han estandarizado todos los resultados obtenidos por los clasificadores para que el algoritmo propuesto pueda procesar correctamente los valores numéricos. El resultado final debe evaluarse en un método propio de clasificación de polaridad en el que se clasifica la precisión de polaridad en un rango entre -1 y 1.

El método utilizado para clasificar la precisión de polaridad final estandarizada evalúa los valores entre 0 y 1, en donde la polaridad es negativa si la precisión de

polaridad final estandarizada se encuentra entre 0 y 0.45, neutro si está entre 0.45 y 0.55 y positiva si está entre 0.55 y 1. De esta forma el algoritmo propuesto entrega el resultado final de la precisión de polaridad en la escala de -1 a 1, donde -1 es negativo, 0 es neutro y 1 es positivo.

4.2 Comparación de Resultados

Los resultados de precisión de polaridad obtenidos por cada clasificador incluido el algoritmo propuesto, fueron comparados con los archivos de polaridad de cada dataset para obtener la matriz de confusión como se muestra en la Tabla 35.

		Predicted Polarity		
		Positives	Neutral	Negatives
Actual Polarity	Positives	True Positives (TP) <i>(a)</i>	False Neutral (FN) <i>(b)</i>	False Negatives (FNe) <i>(c)</i>
	Neutral	False Positives (FP) <i>(d)</i>	True Neutral (TN) <i>(e)</i>	False Negatives (FNe) <i>(f)</i>
	Negatives	False Positives (FP) <i>(g)</i>	False Neutral (FN) <i>(h)</i>	True Negative (TNe) <i>(i)</i>

Tabla 35. Matriz de confusión

La matriz de confusión es un método estadístico que permite calcular el desempeño de los clasificadores de sentimientos y el algoritmo propuesto. La matriz de confusión permite calcular la precisión de cada clasificador y el algoritmo propuesto mediante las siguientes ecuaciones:

La diagonal de la matriz de confusión permite obtener el porcentaje de acierto de las polaridades que han sido clasificadas correctamente, se describe en la Ecuación 8:

$$diagonal = a + e + i$$

Ecuación 8. Diagonal de matriz de confusión

El número total de registros de tuits es el valor numérico de todos los tuits, cada casilla viene dada por una letra, la suma de todas las casillas es el total de tuits del dataset evaluado en la matriz de confusión, viene dada en la Ecuación 9.

$$total\ tweet\ registers = a + b + c + d + e + f + g + h + i$$

Ecuación 9. Total de registros en la matriz de confusión

La medida de precisión *accuracy* permite conocer el porcentaje de acierto entre registros obtenidos como positivos de un clasificador y los que están clasificados como positivos dentro del dataset evaluado, el cálculo viene dado en la Ecuación 10:

$$accuracy = \frac{diagonal}{total\ tweet\ registers}$$

Ecuación 10. Exactitud

La medida de precisión F-Score permite conocer el porcentaje de acierto a nivel global del conjunto de datos tomados del dataset evaluado. Esta medida relaciona el acierto entre los registros que están bien clasificados y los que no, tomando la forma de la media armónica. El cálculo viene dado en la Ecuación 11:

$$F - score = 2 \frac{(precision \times recall)}{(precision + recall)}$$

Ecuación 11. Medida F

La precisión de polaridad medida en *Accuracy* de cada clasificador se puede apreciar en la Tabla 36.

	Text Analytics	Natural Understanding Language	Tweeepy	SentiStrength	VaderSetiment	Sentiment Algorithm (Proposed)
Dataset 1	0.79547619	0.534285714	0.326666667	0.718095238	0.59547619	0.874761904
Dataset 2	0.552333805	0.583451202	0.52640264	0.59806695	0.584629892	0.567421028
Dataset 3	0.684737966	0.594217184	0.572860194	0.598488582	0.639559717	0.615574174
Dataset 4	0.768802228	0.707520891	0.272980501	0.657381616	0.451253482	0.813370474

Tabla 36. Resultados Accuracy

De la Tabla 36, se puede concluir que el algoritmo propuesto supera en 2 de los 4 datasets evaluados al resto de clasificadores de sentimiento. Sin embargo, se optimizaron los rangos numéricos de las funciones matemáticas en los conjuntos difusos en el módulo de prioridades que usa lógica difusa.

	Text Analytics	Natural Understanding Language	Tweeepy	SentiStrengt h	VaderSetiment	Sentiment Algorithm (Proposed)
Dataset 1	0.814981015	0.600816327	0.438443209	0.776940467	0.681298068	0.909186906
Dataset 2	0.759701493	0.455134135	0.305084746	0.520393812	0.454826733	0.564760195
Dataset 3	0.664867296	0.477801268	0.323269131	0.540428733	0.5384939	0.616835487
Dataset 4	0.813186813	0.695652174	0.35235732	0.668463612	0.501265823	0.825396825

Tabla 37. Resultados F-score

De la Tabla 37, se puede concluir que el algoritmo propuesto supera en 2 de los 4 datasets evaluados al resto de clasificadores de sentimiento. Los resultados obtenidos indican que el algoritmo implementado alcanza una precisión de polaridad de 90.9%, porcentaje que demuestra que el algoritmo propuesto alcanza resultados con un alto nivel de precisión dependiendo el dataset o escenario en que se encuentre. En el estado del arte que se detalló en la revisión sistemática no se encontraron clasificadores de sentimiento que superaran el 90% de precisión en Accuracy como en F-Score.

En la Tabla 38 se puede apreciar la mejora en la precisión de polaridad del algoritmo propuesto con respecto a la Tabla 36.

	Text Analytics	Natural Understanding Language	Tweeepy	SentiStrength	VaderSetiment	Sentiment Algorithm (Proposed)
Dataset 1	0.79547619	0.534285714	0.326666667	0.718095238	0.59547619	0.866428571
Dataset 2	0.552333805	0.583451202	0.52640264	0.59806695	0.584629892	0.60348892
Dataset 3	0.684737966	0.594217184	0.572860194	0.598488582	0.639559717	0.612288484
Dataset 4	0.768802228	0.707520891	0.272980501	0.657381616	0.451253482	0.813370474

Tabla 38. Resultados medida Accuracy optimizados.

El algoritmo propuesto supera en precisión de polaridad con la medida *Accuracy* en tres de los cuatro datasets evaluados en los cinco clasificadores de sentimientos seleccionados en la revisión sistemática del estado del arte. En el dataset 3 tweet_semevaltest, el algoritmo propuesto arroja resultados competitivos con respecto a las otras cinco precisiones de polaridad. Ver Tabla 39.

	Text Analytics	Natural Understanding Language	Tweeepy	SentiStrength	VaderSetiment	Sentiment Algorithm (Proposed)
Dataset 1	0.814981015	0.600816327	0.438443209	0.776940467	0.681298068	0.916108565
Dataset 2	0.759701493	0.455134135	0.305084746	0.520393812	0.454826733	0.565806287
Dataset 3	0.664867296	0.477801268	0.323269131	0.540428733	0.5384939	0.619506216
Dataset 4	0.813186813	0.695652174	0.35235732	0.668463612	0.501265823	0.838196286

Tabla 39. Resultados medida F-score optimizados

El algoritmo propuesto supera en precisión de polaridad con la medida *F-Score* en dos de los cuatro datasets, con resultados competitivos en los dos restantes. El alto porcentaje de precisión de polaridad con 91,6% indica que el algoritmo propuesto puede ser el más preciso del estado del arte dependiendo los datasets evaluados o los escenarios en los que es ejecutado.

4.2.1 Test estadísticos no paramétricos

Los test estadísticos no parametrizados se usaron para comprobar si realmente existen diferencias entre las medidas de precisión que se obtuvieron de cada dataset para cada clasificador de sentimientos seleccionado incluyendo el clasificador diseñado con algoritmo propuesto. Se optó por realizar las pruebas no paramétricas debido a que no se tiene una distribución normal de los datos.

4.2.1.1 Test Friedman

El test no paramétrico de Friedman se usó a partir de la Tabla 34. El test consiste en reemplazar cada valor numérico por el equivalente a la posición entre los mismos valores de la fila, es decir, el valor numérico más bajo será el 1 y el más alto será el 6. En la Tabla 40 se observan los rangos del test de Friedman.

	Text Analytics	Natural Understanding Language	Tweeepy	SentiStrength	VaderSetiment	Sentiment Algorithm (Proposed)
Dataset 1	5	2	1	4	3	6
Dataset 2	6	3	1	4	2	5
Dataset 3	6	2	1	4	3	5
Dataset 4	5	4	1	3	2	6
Range	22	11	4	15	10	22

Tabla 40. Rangos del test de Friedman

Una vez los valores de las precisiones son reemplazados por el equivalente a su posición se debe usar la Ecuación 12.

$$F = \frac{12}{nk(k+1)} \sum_{j=1}^k R^2 - 3n(k+1)$$

Ecuación 12. Test de Friedman

De la Ecuación 12 se tiene que n es el número de filas, k el número de columnas o variables y R es el rango de cada columna. Al reemplazar los valores y calcular la ecuación obtenemos que el valor chi-cuadrado estadístico es 18.142857, la probabilidad 0.002773 con un grado de libertad de 5. El test dice que para chi-cuadrado con libertad 5 es mayor al grado a la probabilidad entonces se concluye que todos los clasificadores de sentimiento presentan diferencias en cada una de las medidas de precisión.

4.2.1.2 Test de Wilcoxon

El test no parametrizado de Wilcoxon se usa para verificar la conclusión obtenida a partir de los resultados del test de Friedman. El test se usó a partir de la Tabla 36. Resultados Accuracy. Debido a que el test de Wilcoxon se implementa en muestras de dos en dos, se debería comparar cada muestra de los clasificadores con la muestra del algoritmo propuesto. La muestra se conoce al conjunto de resultados obtenidos como precisión de polaridad en cada dataset. Sin embargo, para esta investigación no fue posible aplicar el test de Wilcoxon debido a que el tamaño de las muestras es demasiado pequeño con respecto al mínimo de pruebas aceptado. Son cuatro datasets de los que se obtienen valores y se necesita como mínimo 6 para obtener el valor crítico del test. En futuras investigaciones se puede tratar de utilizar 6 o más dataset para hacer posible la ejecución de esta prueba.

Capítulo 5

Conclusiones, experiencias y trabajo futuro

La revisión sistemática exhaustiva de clasificadores de sentimiento en análisis de sentimientos, se llevó a cabo, tanto en la literatura científica basados en los lineamientos de Kitchenham como también en la literatura gris en base a Kitchenham y el método de bola de nieve. Mediante la revisión sistemática fue posible conocer la estructura de los clasificadores de sentimiento y saber qué tipos de algoritmos utilizan en el proceso de clasificación, lo que permitió concluir que la mayoría usan técnicas de preprocesamiento de datos para normalizar los mensajes analizados antes del proceso de clasificación y entregar resultados de polaridad más precisos con respecto a otros clasificadores de sentimientos.

Los resultados obtenidos después de realizar el proceso de búsqueda y selección aplicando la cadena de búsqueda seleccionada y los diferentes criterios propuestos se observó que seis clasificadores superaron el cálculo del puntaje de calidad mínimo establecido y en ninguno se usa técnicas de lógica difusa en el proceso del cálculo de la precisión de polaridad en análisis de sentimiento. En lugar de ello se encontraron propuestas en análisis de sentimientos que usan preprocesamiento de datos y modelos de entrenamiento para predecir la precisión de polaridad. El preprocesamiento de datos es usado para normalizar los mensajes debido a la mala ortografía, elongaciones, jergas, hashtags, contexto, sarcasmo, abreviaciones, siglas, contracciones, entre muchas otras más, por lo que se concluye que esta técnica influye directamente en la precisión de la polaridad en un clasificador de sentimiento.

La revisión sistemática se llevó a cabo hasta la fecha de febrero de 2019, es conveniente realizar una actualización a la revisión sistemática posterior a la fecha, para así poder abarcar nuevas propuestas.

Se diseñó un algoritmo basado en reglas de lógica difusa que funciona a partir de los resultados de cinco clasificadores de sentimientos. Los clasificadores de sentimientos con los que funciona el algoritmo se seleccionaron mediante una caracterización en la cual se describen las características de cada clasificador y se evidencian virtudes y debilidades, la caracterización surgió como resultado de la revisión sistemática.

La caracterización de clasificadores ofreció un panorama de características de cada clasificador, con el cual fue posible crear una serie de reglas de inferencia y conjuntos difusos que conforman el módulo de lógica difusa, módulo con que se dieron pesos matemáticos a los clasificadores de acuerdo a su naturaleza de

clasificación. De esta manera, mediante este proceso se obtuvo resultados que permiten saber si un clasificador es el más adecuado para la naturaleza de un tuit que sea analizado. Los modelos difusos no necesitan dataset de entrenamiento, no necesitan aprender en cada momento que un dataset o escenario cambie para mejorar los resultados, para obtener buenos resultados es viable hacer ajustes en las reglas de inferencia o en los rangos numéricos de los conjuntos difusos, por esta razón se implementó Hill-Climbing, para optimizar los valores que se definieron de forma manual, obteniendo una mejor precisión de polaridad.

Para un algoritmo que no utiliza modelos de entrenamiento para el preprocesamiento de datos cada vez que es ejecutado, se han obtenido resultados satisfactorios, en los cuales se ve una mejora en la precisión de polaridad, tanto en medida *Accuracy* como *F-Score*.

El algoritmo propuesto alcanzó un porcentaje alto de precisión de polaridad en *Accuracy*, lo que demostró que superó en tres de los cuatro datasets a los mejores clasificadores de sentimientos del estado del arte seleccionados de acuerdo a unos criterios de calidad establecidos en la revisión sistemática. Además, la medida *F-Score* alcanza el 91,6% máximo de precisión, permitiendo tener un margen de error menor con respecto a los clasificadores de sentimientos del estado del arte.

La variación de la calidad de la clasificación de polaridad en análisis de sentimientos de un tuit basado en reglas de lógica difusa y con la utilización de las polaridades entregadas por los mejores clasificadores de sentimiento del estado del arte fue de 1.02% con respecto a los mejores clasificadores seleccionados en el estado del arte.

En cuanto a las experiencias en el desarrollo de la investigación, se tuvieron en cuenta distintas áreas de las ciencias de computación como estructuras de datos, estructura de lenguajes, inteligencia artificial, sistemas distribuidos y aplicaciones web. Todas estas áreas de conocimiento fueron aplicadas en el desarrollo y el diseño del algoritmo propuesto, debido a que se cuenta con módulos de identificación de características, en las cuales se implementaron algoritmos de búsqueda binaria y algoritmos recursivos. Las áreas de estructura de lenguajes y sistemas distribuidos se usaron al momento de buscar que la compatibilidad entre distintos lenguajes de programación fuera posible, en los cuales se utilizó Python y Java SE. El área web se usó para consumir servicios web y usar tecnología para consumo de API's como CURL. Todo este proceso permite conocer a fondo la implementación de un clasificador de sentimiento a alto nivel.

La implementación del algoritmo fue el proceso más complejo dentro de la etapa de desarrollo. El primer obstáculo a superar fue entender el funcionamiento del código después de mapear las reglas de lógica difusa a Java SE. Se necesitaba un algoritmo que permitiera ejecutar y optimizar reglas de inferencia en tiempo de ejecución, para ello se optó por parametrizar los constructores de cada clase que

implementan la clase abstracta que define cada regla difusa, logrando que los parámetros fueran leídos mediante archivos de texto. Superado este inconveniente, la integración del módulo de lógica difusa con el algoritmo propuesto fue todo un desafío de conceptos y abstracción, debido a que era frecuente perder el flujo de conocimiento cada vez que iban creciendo los procesos. En la medida que fue creciendo el número de procesos se fue aplicando métodos y funciones binarias en el algoritmo, que también presentaron un alto grado de complejidad, ya que controlar miles de datos sin que el rendimiento del algoritmo disminuyera al momento de calcular la polaridad, fue una tarea ardua y de varias semanas de trabajo. Además, se requería que fuese posible integrar varios clasificadores de sentimientos implementados en distintos lenguajes de programación, tecnología y sistemas operativos para que trabajaran como un solo clasificador, es por ello que se optó por diseñar rutinas de ejecución en archivos de lotes para la ejecución de cada uno de los clasificadores seleccionados; para evitar problemas de compatibilidad se decidió usar un lenguaje que funcionase en cualquier sistema operativo, en donde solo fuera necesario la instalación de una máquina virtual, como Java SE.

Para un trabajo futuro, es interesante que el algoritmo pueda interpretar y detectar el tiempo semántico en que está expresado un tuit a través de un módulo de lenguaje natural. No debería tener polaridad positiva un tuit como "I was good", debido a que es tiempo pasado, en donde el autor puede expresar felicidad pero que en el tiempo presente es posible que no ya no lo sea, se podría clasificar como un tuit con polaridad neutra, pero actualmente los clasificadores del estado del arte lo toman como positivo, ya que la palabra "good" está asociada a puntuación positiva.

Referencias

- [1] B. Liu, *Análisis de sentimientos: opiniones mineras, sentimientos y emociones*. Cambridge: Cambridge University Press, 2015.
- [2] D. Tang, "Sentiment-Specific Representation Learning for Document-Level Sentiment Analysis," *WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, pp. 447-451, 2015.
- [3] M. López, *et al.*, "E2SAM: Evolutionary ensemble of sentiment analysis methods for domain adaptation," *Information Science*, vol. 480, pp. 273-286, 2018.
- [4] Twitter. (Accessed: Jun 30, 2019). *Ayuda con las cuentas bloqueadas o limitadas*. Available: <https://help.twitter.com/es/managing-your-account/locked-and-limited-accounts>
- [5] B. Kitchenham, *et al.*, "Systematic literature reviews in software engineering – A tertiary study," *Information and Software Technology*, vol. 52, pp. 792-805, 2010.
- [6] I. I. stats. (Accessed: Jul 1, 2019). *1 second*. Available: <https://www.internetlivestats.com/one-second/>
- [7] Ankit and N. Saleena, "An Ensemble Classification System for Twitter Sentiment Analysis," *Procedia Computer Science*, vol. 132, pp. 937-946, 2018.
- [8] G. García, *et al.*, "Ciberbullying: forma virtual de intimidación escolar," *Revista Colombiana de Psiquiatría*, vol. 40, pp. 115-130, 2011.
- [9] Twitter. (Accessed: jun 27, 2019). *About country withheld content*. Available: <https://help.twitter.com/en/rules-and-policies/tweet-withheld-by-country>
- [10] R. Andrea. (10 Marzo 2017). *Twitter marcará las cuentas que publiquen "contenido sensible"*. Available: <https://www.baquia.com/redes-sociales/twitter-marcara-las-cuentas-usuarios-publicuen-contenido-sensible>
- [11] M. Piñeiro. (Accessed: May 15, 2019). *¿Porqué falla el análisis de sentimiento?* Available: <http://www.concepto05.com/2011/03/por-que-falla-el-analisis-de-sentimiento/>
- [12] A. Bastida. (18 Marzo 2015). *Las fotos amamantando ya no serán consideradas obscenas*. Available: <https://www.bebesymas.com/lactancia/facebook-insiste-las-fotos-amamantando-ya-no-seran-consideradas-obscenas-a-menos-que-se-vea-el-pezon>
- [13] D. Wells. (Accessed: Jun 27, 2019). *Extreme Programming: A gentle introduction*. Available: <http://www.extremeprogramming.org/>
- [14] H. Ameur, *et al.*, "A New Method for Sentiment Analysis Using Contextual Auto-Encoders," *Journal of Computer Science and Technology*, vol. 33, pp. 1307-1319, November 01 2018.
- [15] S. Angadi and V. Siva Reddy, "Enhanced Framework for Sentiment Analysis in Text using Distance based Classification Scheme," presented at the Second International Conference on Advances in Electronics, Computers and Communications (ICAIECC), Bangalore, 2018.

- [16] A. Rodrigues, *et al.*, "Sentiment Analysis of Real Time Twitter Data Using Big Data Approach," presented at the 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), Bangalore, 2017.
- [17] D. Mahendhiran and S. Kannimuthu, "Deep Learning Techniques for Polarity Classification in Multimodal Sentiment Analysis," *International Journal of Information Technology & Decision Making*, vol. 17, pp. 883-910, 2018.
- [18] S. Xiong, *et al.*, "Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings," *Neurocomputing*, vol. 275, pp. 2459-2466, 2018.
- [19] M. Trupthi, *et al.*, "Sentiment Analysis on Twitter Using Streaming API," presented at the IEEE 7th International Advance Computing Conference (IACC), Hyderabad, 2017.
- [20] N. Azzouza, *et al.*, "A real-time Twitter sentiment analysis using an unsupervised method," presented at the Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, Amantea, Italy, 2017.
- [21] T. Sahni, *et al.*, "Efficient Twitter sentiment classification using subjective distant supervision," presented at the 9th International Conference on Communication Systems and Networks (COMSNETS), 2017.
- [22] P. Mishra, *et al.*, "Sentiment analysis of Twitter data: Case study on digital India," presented at the International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds, Noida, 2016.
- [23] A. Sharma and G. K. Nayak, "Efficient and Parallel Framework for Analyzing the Sentiment," presented at the Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications. Advances in Intelligent Systems and Computing, Singapore, 2017.
- [24] A. Sulaiman, *et al.*, "Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information," *Cognitive Systems Research*, vol. 54, pp. 50-61, 2019.
- [25] A. Manicka Raja and S. Swamynathan, "Tweet Sentiment Analyzer: Sentiment Score Estimation Method for Assessing the Value of Opinions in Tweets," 2016, pp. 1-6.
- [26] Y. Kaewpitakkun and K. Shirai, "Incorporation of Target Specific Knowledge for Sentiment Analysis on Microblogging," *IEICE Transactions on Information and Systems*, vol. E99.D, pp. 959-968, 2016.
- [27] M. A. M. Raja and S. Swamynathan, "Tweet Analyzer: Identifying Interesting Tweets Based on the Polarity of Tweets," New Delhi, 2016, pp. 307-316.
- [28] E. Sygkounas, *et al.*, "Sentiment Polarity Detection from Amazon Reviews: An Experimental Study," Cham, 2016, pp. 108-120.
- [29] R. Pandarachalil, *et al.*, "Twitter Sentiment Analysis for Large-Scale Data: An Unsupervised Approach," *Cognitive Computation*, vol. 7, pp. 254-262, April 01 2015.
- [30] P. Chikersal, *et al.*, "Modelling Public Sentiment in Twitter: Using Linguistic Patterns to Enhance Supervised Learning," Cham, 2015, pp. 49-65.

- [31] F. H. Khan, *et al.*, "TOM: Twitter opinion mining framework using hybrid classification scheme," *Decision Support Systems*, vol. 57, pp. 245-257, 2014/01/01/ 2014.
- [32] S. Mukherjee and P. Bhattacharyya, *Sentiment Analysis in Twitter with Lightweight Discourse Analysis*, 2013.
- [33] Google. (2019). *Cómo definir mejor las búsquedas web*. Available: <https://support.google.com/websearch/answer/2466433>
- [34] M. Azure. (Accessed: Jun 27, 2019). *Text Analytics* Available: Available: <https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/>
- [35] NLTK. (Accessed: Jun 28, 2019). *Sentiment Analysis with Python NLTK Text Classification*. Available: <https://text-processing.com/demo/sentiment/>
- [36] monkeylearn. (Accessed: Jun 30, 2019). *Sentiment Analysis*. Available: https://app.monkeylearn.com/main/classifiers/cl_pi3C7JiL/
- [37] Sentiment140. (Accessed: Jun 30, 2019). *API*. Available: <http://www.sentiment140.com/>
- [38] I. Baqapuri. (Accessed: Jul 2, 2019). *Tweetmood*. Available: <http://www.tweet-mood-check.appspot.com/>
- [39] ayllote. (Accessed: Jun 27, 2019). *senti-py*. Available: <https://github.com/ayllote/senti-py>
- [40] mediatoolkit. (Accessed: Jun 27, 2019). *Monitor online mentions of your brand in real-time*. Available: <https://www.mediatoolkit.com/>
- [41] AI and t. technology. (Accessed: Jun 27, 2019). *Text Analysis API*. Available: <https://ai-applied.nl/support/text-analysis-api-documentation>
- [42] Paralleldots. (Accessed: Jun 25, 2019). *Text Analysis*. Available: <https://www.paralleldots.com/text-analysis-apis>
- [43] Geeksforgeeks. (Accessed: Jul 6, 2019). *Tweepy*. Available: <https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/>
- [44] Uclassify. (Accessed: Jul 6, 2019). *Sentiment*. Available: <https://www.uclassify.com/browse/uclassify/sentiment>
- [45] IBM. (Accessed: Jul 1, 2019). *Natural Language Understanding* Available: <https://www.ibm.com/watson/services/natural-language-understanding/>
- [46] amitness. (Accessed: Jul 5, 2019). *Applytics*. Available: <https://github.com/amitness/applytics>
- [47] artem-oppermann. (Accessed: Jul 5, 2016). *Sentiment Analysis LSTM*. Available: <https://github.com/artem-oppermann/Sentiment-Analysis-of-Netflix-Reviews>
- [48] R. Kotikalapudi, *et al.* (Accessed: Jul 9, 2019). *Text Classification Keras*. Available: <https://github.com/jfilter/text-classification-keras>
- [49] D. H. Wahid and S. N. Azhari. (2016). *sentistrength_id*. Available: https://github.com/masdevid/sentistrength_id
- [50] tweet_viz. (Accessed: Jun 26, 2019). *Sentiment viz* Available: https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/
- [51] M. d. ideas. (Accessed: Jun 27, 2019). *Mr.Tuit*. Available: <http://www.mrtuit.com/>
- [52] 25Trends. (Accessed: Jul 10, 2019). *Twitter analytics for the masses*. Available: <https://twitter.25trends.me/services/twitter/public/>
- [53] S. E. LLC, "OpinionCrawl " Accessed: Jul 10, 2019.

- [54] SpazioDati. (Accessed: Jul 10, 2019). *Sentiment Analysis API*. Available: <https://dandelion.eu/semantic-text/sentiment-analysis-demo/>
- [55] N. Novielli and F. Calefato, "Senti4SD," Accessed: Jul 10, 2019.
- [56] SummarizeBot. (Accessed: Jun 20, 2019). *Text Analysis API Demo*. Available: https://www.summarizebot.com/text_api_demo.html
- [57] Linguakit. (Accessed: Jun 27, 2019). *Análisis lingüísticos*. Available: <https://linguakit.com/es/analisis-completo>
- [58] Xfuzzy. (2019). *Fuzzy Logic*. Available: http://www2.imse-cnm.csic.es/Xfuzzy/Xfuzzy_3.0/download_sp.html
- [59] J. P. Gonzalez, *et al.*, "Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión documental," *Ibersid*, vol. 8, pp. 73-80, 2014.
- [60] S. Edelkamp and S. Schrödl, "Hill Climbing," in *Heuristic Search*, S. Edelkamp and S. Schrödl, Eds., ed San Francisco: Morgan Kaufmann, 2012.