Extracción y análisis de información de accidentes de tránsito desde redes sociales

Nestor Eduardo Suat-Rojas

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá D.C., Colombia
2021

Extracción y análisis de información de accidentes de tránsito desde redes sociales

Nestor Eduardo Suat-Rojas

Trabajo de profundización presentado como requisito parcial para optar al título de
Magister en Ingeniería de Sistemas y Computación

Director:
Cesar Augusto Pedraza Bonilla PhD.
Codirector:
Camilo Albeiro Gutierrez Osorio Msc.

Línea de Investigación:
Computación aplicada
Grupo de Investigación:
PLaS - Programming Languages and Systems

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá D.C, Colombia
2021

Dedicatoria

En el momento preciso, yo el Señor haré que las cosas ocurran. (Isaías 60:22)

Para mi mamá, Luz Marina. Para mi papá, Néstor. Y para mi querida familia.

Agradecimientos

Agradecimientos a mis directores, los señores Cesar Pedraza Bonilla y Camilo Gutierrez Osorio, quienes han brindado un apoyo de mentoría muy valiosa para la realización de este trabajo. Un agradecimiento y reconocimiento a los involucrados en llevar un pedazo de la maestría de la UNAL a las instalaciones de la Unillanos, los resultados de este convenio seguro beneficiará a la región de los Llanos en un avance por el reconocimiento como territorio intelectual y productivo. También quiero agradecer al grupo de investigación Programming Languages and Systems (PLaS) del departamento de sistemas e industrial, quienes desde sus seminarios aportamos ideas colaborativamente en cada uno de los proyectos. Finalmente quiero agradecer a quienes me apoyaron en el proceso de etiquetamiento de datos, Rafael, David, Andrés, los integrantes del PLaS y mis estudiantes de la Corporación Universitaria Autónoma de Nariño AUNAR, aprovecho para agradecer a la AUNAR Villavicencio por la confianza y disponibilidad que me brindaron para culminar mis estudios.

A todos gracias. A mi familia, mi Mamá, mi Papá, mis hermanos, mi adorada hermana Rosita, que de verdad son el tesoro que Dios me ha dado en esta vida. A Dios gracias siempre.

Resumen

Título en español: Extracción y análisis de información de accidentes de tránsito desde redes sociales

La detección de accidentes de tránsito es una estrategia importante para que los gobiernos implementen políticas que reduzcan este fenómeno. Usualmente usan técnicas como procesamiento de imágenes, dispositivos RFID, y otras. La detección en redes sociales ha surgido como una alternativa de bajo costo. Sin embargo las redes sociales presentan varios retos y desafíos como uso de lenguaje informal y falta de ortografía. Este trabajo propone un método para extraer y analizar los datos de accidentes de tránsito desde Twitter. Cuatro fases componen el método. La primera fase establece los mecanismos para obtener datos. El segundo consiste en representar vectorialmente los mensajes y clasificarlos como accidentes de tránsito o no. La tercera usa técnicas de reconocimiento de entidades nombradas para la detección de ubicaciones. En la cuarta estas ubicaciones pasan por un geocoder que devuelve sus coordenadas geográficas. Aplicamos este método para la ciudad de Bogotá y comparamos los datos de Twitter con la fuente oficial de tránsito, las comparaciones muestran una influencia en Twitter sobre la zona comercial e industrial de la ciudad. Los resultados revelan la efectividad de los accidentes reportados en Twitter como información adicional y su uso debe considerarse como fuentes complementarias a los métodos de detección existentes.

Palabras clave: Sistemas de transporte inteligente, redes sociales, accidente de tránsito, sensores sociales, procesamiento de lenguaje natural, aprendizaje automático, minería de texto, clasificación, reconocimiento de entidades nombradas, Twitter.

Abstract

Título en inglés: Extraction and analysis of traffic accident data from social networks

The detection of traffic accidents is an important strategy for governments to implement policies that reduce this phenomenon. They usually use techniques like image processing, RFID devices, and others. Social media detection has emerged as a low-cost alternative. However, social media presents several challenges such as use of non-formal language and misspelling. This work proposes a method to extract and analyze traffic accident data from Twitter. The method is composed of four phases. The first phase establishes the mechanisms for obtaining data. The second consists of representing the messages in vectors and classifying them as traffic accidents or not. The third uses named entity recognition techniques for

location detection. In the fourth, these locations go through a geocoder that returns their geographic coordinates. We apply this method for the city of Bogotá and compare the data on Twitter with the official transit source, the comparisons show an influence on Twitter on the commercial and industrial area of the city. The results reveal the effectiveness of the accidents reported on Twitter as additional information and their use should be considered as complementary sources to the existing detection methods.

Keywords: intelligent transportation system, social media, traffic accident, social sensors, natural language processing, machine learning, text mining, classification, named entity recognition, Twitter)

Este Trabajo Final de maestría fue calificado en diciembre de 2021 por el siguiente evaluador:

Ingeniero Julio Ernesto Suárez Páez PhD.
Colpatria Multibanca del Grupo Scotiabank

Contenido

Lista de Tablas

Lista de Figuras

1 Introducción

El crecimiento poblacional y económico de las ciudades son las primeras causas del aumento del volumen vehicular transitado en las calles [1], este aumento se ve reflejado en un mayor número de accidentes de tránsito, que para el 2016 en Colombia dejó cifras de 7 mil muertos y 45 mil lesionados, superando registros desde el año 2000, según registros del *Observatorio Nacional de Seguridad Vial* [2]. Identificar puntos de accidentalidad de tránsito brinda a los responsables de gobierno una herramienta para construir políticas y acciones que reduzcan los incidentes. La detección de incidentes viales permite descubrir la hora pico, los segmentos de carretera más problemáticos y otros factores que influyen en el flujo vehicular como eventos sociales [3], infraestructura vial [4], clima e iluminación, entre otros.

En los últimos años, en la detección de tráfico vehicular se han adelantado investigaciones que proponen el uso de dispositivos físicos para la detección y monitoreo del tráfico, como es la ubicación en puntos estratégicos de la ciudad de cámaras de video [5], Loop Inductors [6, 7] y lectores de RFID [8], inclusive aprovechando infraestructura existente como el monitoreo de los usuarios de la red celular GSM/3G/EDGE [9] y sistemas de navegación satelital instalados en vehículos de transporte público para modelar patrones de tráfico [10, 11]. Sin embargo, para no elevar los costos de los proyectos, los dispositivos son ubicados en segmentos de carretera principal, lo que reduce cobertura y dificulta el análisis en arterias o calles secundarias. Otros problemas comunes que reducen la previsión del tráfico son los costos de mantenimiento y errores de precisión en climas adversos.

Lo anterior, ha despertado un interés en los investigadores a estudiar la efectividad de las redes sociales como fuentes disponibles de información de incidentes de tránsito, logrando aumentar la cobertura en la detección y monitoreo del tráfico al incluir todos los actores viales [12, 13] y otros factores que inciden: como los eventos de masas y el estado de la carretera [14]. Varios autores han evaluado la credibilidad y efectividad de los datos de las redes sociales al compararlos con otras fuentes: Wang et al. [3] realiza la comparación con datos recolectados de GPS en buses de transporte público, Gu et al. [14] con fuentes oficiales y otras redes sociales y Zhang et al. [15] con fuentes oficiales y datos de loop inductors. En este sentido se han diseñado metodologías para la extracción de datos relevantes a incidentes de tráfico desde redes sociales, utilizando técnicas de aprendizaje automático y minería de texto [13, 14, 15, 16]. Estas técnicas permiten monitorear los incidentes de tráfico usando inteligentemente los recursos, algo de gran ayuda para que los gobiernos locales e investiga-

dores puedan ejercer control a estos tipos de siniestros en la ciudad.

El presente trabajo propone un método para la extracción de datos de accidentes utilizando las publicaciones de Twitter, un estudio realizado por los meses de octubre del 2018 a julio del 2019 en la ciudad de Bogotá (Colombia), considerando 4 fases: la recolección, clasificación de tweets relacionados a accidentes, reconocimiento de entidades para la extracción de ubicación y tiempo, y la geolocalización de los accidentes reportados. En la primera fase se contempla los mecanismos para la recolección de los datos en Twitter y la construcción de un dataset etiquetado usando *crowdsourcing data labeling*[1] con la colaboración de 30 personas y un mecanismo de selección de 3 votos por publicación; los tweets se recolectaron mediante la API de Twitter: Search API y Stream API; se diseñó un proceso de recolección teniendo en cuenta la búsqueda de tweets por palabras claves y recolección de tweets desde cuentas de usuario oficiales. La segunda fase implementa un método de clasificación automática con el algoritmo de aprendizaje supervisado Support Vector Machine entrenado con 3804 tweets, donde la mitad esta relacionado con accidentes de tránsito, este método permite filtrar los tweets relevantes a accidentes de tránsito con los no relevantes; en esta fase también se construye un modelo de *word embedding* de *doc2vec* con un millón de tweets. En la tercera fase, se usa una técnica de reconocimiento de entidades que permita la extracción de la ubicación y tiempo del accidente a partir del texto; para esta tarea se implementó la librería SpaCy aplicando re-entrenamiento del modelo es_core_news_lg[2] con el conjunto de datos propio de 1340 tweets usando el formato IOB. En la cuarta fase, después de extraída la entidad de ubicación se pasa por un *geocoder* que se encarga de devolver las coordenadas geográficas de la dirección mencionada.

1.1. Identificación del problema

Para cubrir un mayor rango en la detección de incidentes de tránsito a calles arteriales y secundarias, se debe utilizar fuentes de información disponibles que complementen a las técnicas de infraestructura física. Por lo tanto, existe la posibilidad de usar las redes sociales como fuentes de modelos de difusión de información [17]. Unas primeras metodologías definidas para extraer la información de incidentes de tránsito [16, 14, 18, 19, 20] demuestran que las publicaciones de los usuarios amplían el rango de cobertura en el monitoreo y análisis de accidentalidad [3, 15]. Estas técnicas permiten monitorear los incidentes de tráfico usando inteligentemente los recursos, algo de gran ayuda para que los gobiernos locales e investigadores puedan ejercer control a estos tipos de siniestros en la ciudad. Por esto, en favor de brindar soluciones que permitan explorar la problemática de accidentalidad vial en Bogotá, el actual trabajo propone un método para la extracción y análisis de datos referentes a la

[1] *Crowdsourcing data labeling* es el proceso de etiquetado de datos en la cual colaboran un grupo de personas en esta tarea.

[2] Autores en https://explosion.ai/

accidentalidad de tránsito desde las redes sociales en esta ciudad. Finalmente, para guiar el desarrollo de este trabajo se busca responder a la pregunta *"¿Cómo extraer información de las redes sociales relacionado con accidentes de tránsito en Bogotá?"*.

1.2. Caso de estudio: Reporte de accidentes de tránsito en Bogotá D.C. (Colombia)

En este trabajo se ha considerado la recolección y procesamiento de publicaciones en Twitter de accidentes de tránsito de la ciudad de Bogotá D.C. (Colombia), en un periodo de diez meses, de octubre a diciembre del 2018 y enero a julio del 2019. La ciudad de Bogotá tiene un área total de 1775 km^2 dividida en 20 divisiones administrativas o localidades, de los cuales 307 km^2 y 19 localidades corresponden al área urbana; la ciudad cuenta con una población aproximada de 7 millones de habitantes. Los accidentes reportados en la base de datos oficial de la secretaría de movilidad para el mismo periodo ascienden a 25299 incidentes, la mayoría de estos se reportan en la región urbana como se observa en la Figura **1-1**.

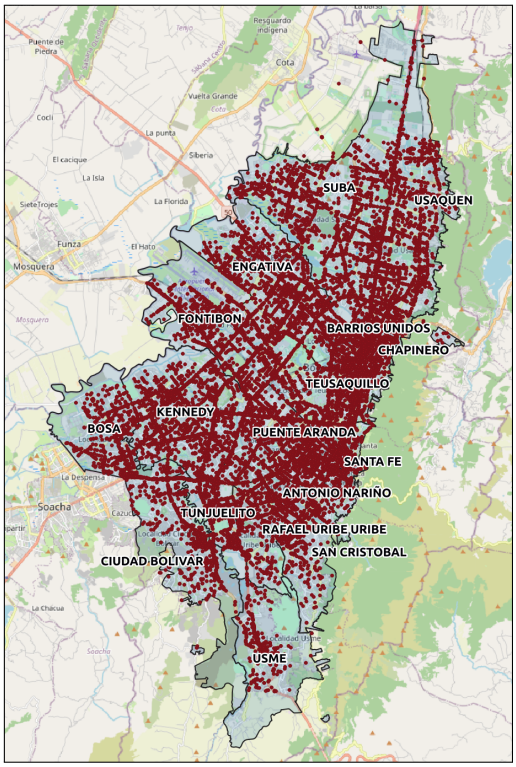

Figura 1-1: Ubicación de los accidentes de tránsito ocurridos en octubre del 2018 a julio del 2019.

1.3. Objetivo

El principal objetivo del trabajo de profundización es el de diseñar un método para la extracción y análisis estadístico-descriptivo de información sobre accidentalidad vial en la ciudad de Bogotá a partir de redes sociales. En orden para alcanzar este objetivo, los siguientes objetivos específicos son propuestos:

- Recolectar los datos que se encuentran en la red social seleccionada sobre información de accidentalidad para la ciudad de Bogotá.
- Aplicar una técnica con base en la literatura encontrada que contribuya a la extracción de información útil sobre accidentes de tránsito presente en las redes sociales.
- Analizar con estadística-descriptiva la información recolectada en redes sociales sobre accidentes de tránsito en Bogotá.
- Evaluar el desempeño del método propuesto con respecto a la extracción de información de accidentes de tránsito.

1.4. Contribuciones

La principal contribución de este trabajo es un método en la extracción de accidentes en Twitter para la ciudad de Bogotá, también aportamos un conjunto de datos de redes sociales y una publicación, como se menciona a continuación.

- Un conjunto de datos con 4,538,305 tweets relacionados con accidentes en Español recolectado en el periodo de octubre del 2018 a julio del 2019. Disponible en: *Gutierrez-Osorio, Camilo; Suat, Nestor; Pedraza, Cesar (2020), "Bogota city traffic accidents - Social media datasets", Mendeley Data, V1, doi: 10.17632/c2r6tk9hbg.1*
- Un conjunto de datos de 3804 tweets, donde 1902 están relacionados a reportes de accidentes de tránsito y 1902 no están relacionados. Disponible en: *Nestor Suat-Rojas, Camilo Gutierrez-Osorio, & Cesar Pedraza Bonilla. (2021). Dataset of accidents reported on Twitter Colombia (1.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5548475*
- Un conjunto de datos de 1340 tweets con etiquetamiento secuencial de entidades de ubicación y tiempo usando el estándar IOB y la herramienta Brat Annotation Tools. Disponible en: *Nestor Suat-Rojas, Camilo Gutierrez-Osorio, & Cesar Pedraza Bonilla. (2021). Dataset of accidents reported on Twitter Colombia (1.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5548475.*
- Un conjunto de datos con 26362 tweets de accidentes de tránsito con las coordenadas del incidente y la fecha de publicación. Disponible en: *Nestor Suat-Rojas, Camilo*

Gutierrez-Osorio, & Cesar Pedraza Bonilla. (2021). Dataset of accidents reported on Twitter Colombia (1.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5548475.

- Un conjunto de datos con 26362 tweets de accidentes de tránsito con las coordenadas del incidente y la fecha de publicación. Disponible en: *Nestor Suat-Rojas, Camilo Gutierrez-Osorio, & Cesar Pedraza Bonilla. (2021). Dataset of accidents reported on Twitter Colombia (1.0) [Data set]. Zenodo. https://doi.org/10.5281/zenodo.5548475.*
- Código fuente en Python de Notebook (Jupyter) de cada fase del proceso. Disponible en github: *https://github.com/solofruad/traffic-accidents/tree/master/model/notebook.*

Publicación

- Suat-Rojas, N.; Gutierrez-Osorio, C.; Pedraza, C. *Extraction and Analysis of Social Networks Data to Detect Traffic Accidents.* Information 2022, 13, 26. https://doi.org/10.3390/info13010026

1.5. Organización del trabajo

El resto del documento esta organizado de la siguiente forma:

- **Capítulo 2** se presenta una revisión de la literatura sobre la detección de accidentes usando Redes Sociales.
- **Capítulo 3** describe en términos generales el método propuesto para la extracción de accidentes en Twitter.
- **Capítulo 4** se describe el proceso de recolección de datos para la red social Twitter.
- **Capítulo 5** se discute y evalúa el modelo propuesto para la clasificación de publicaciones relacionadas a accidentes.
- **Capítulo 6** se discute y evalúa el modelo de extracción de ubicaciones mediante métodos de reconocimiento de entidades nombradas.
- **Capítulo 7** se realiza el procesamiento y análisis de tweets de accidentes de Bogotá, y se evalúa el método de recolección propuesto al comparar los datos extraídos con otras fuentes oficiales.
- **Capítulo 8** se amplía las conclusiones y recomendaciones de trabajos futuros.

2 Revisión de la literatura

En este capítulo se presenta una revisión de literatura relacionada al tema del trabajo de profundización. Primero se comienza con una breve descripción del estado de arte sobre el monitoreo de tránsito vehicular. Segundo se mencionan los trabajos de analítica de datos orientados a soluciones de detección de accidentes usando fuentes de dispositivos físicos y redes sociales. Finalmente se sintetiza los métodos existentes de extracción de incidentes desde redes sociales que sirvieron de base para el método propuesto de este trabajo.

2.1. Monitoreo de tránsito vehicular

Identificar puntos de accidentalidad de tránsito brinda a los responsables de gobierno una guía para construir políticas y acciones que reduzcan los incidentes. Por lo anterior, en los últimos años en la detección del tráfico vehicular se han adelantado investigaciones que proponen el uso de infraestructuras física, como la ubicación en puntos de la ciudad de cámaras de detección vehicular [7] y aprovechamiento de cámaras de seguridad [5], instalación de loop inductors en segmentos para el conteo vehicular y su uso para la estimación de velocidades promedio [6, 7], usando el GPS de los taxis o buses públicos para analizar el historial de tráfico [11], instalación de lectores RFID en automóviles y segmentos de calles para detectar el paso de vehículos e intercambiar información como velocidad y cantidad de ocupantes por vehículo [8, 21].

Sin embargo, para minimizar el costo de los proyectos, los dispositivos se ubican en segmentos de carretera principal, lo que reduce la cobertura y dificulta el análisis en arterias o calles secundarias. Ante esto, el uso de novedosas tecnologías como drones son utilizados para recorrer y cubrir mayores puntos de un segmento [22], pero aún presenta inconvenientes con la corta duración de la batería y funciona solo para segmentos de carretera recta. Por otro lado, hay trabajos que buscan aprovechar la infraestructura de tecnologías existentes llevándolas al monitoreo de tráfico, como es el ejemplo de utilizar la gran cantidad de usuarios conectados a la red celular GSM/3G/EDGE [9], pero todavía presentan retos para reducir el error de ubicación que llega hasta los 250 metros.

Como se mencionó anteriormente, entre las soluciones con infraestructura física se hallan algunos problemas en común que reducen la previsión del tráfico a gran escala para las ciudades. Como son los costos de mantenimiento y transmisión de datos, además la ubicación

fija de estos dispositivos limita la cobertura del monitoreo a carreteras principales. Estas desventajas están enumeradas en la tabla **2-1**, donde se observa que la mayoría de estos dispositivos físicos son aplicables o rastrean información solo de vehículos, dejando a un lado la información no recurrente como el estado del clima, los movimientos de masas y la infraestructura de la malla vial. Finalmente, los errores de precisión o inexactitud de los datos, como por ejemplo en climas adversos, determinan un ruido considerable a la hora de evaluar la fiabilidad de los datos obtenidos. Estos son retos que los autores coinciden identificar y resolver en trabajos posteriores para mejorar el monitoreo vehicular.

Tabla 2-1: Retos y dificultades en la detección de flujo de tráfico y accidentes

Autores y año	Variables		Dificultades				
	Flujo vehicular	Accidentes	Costos [1]	Cobertura limitada [2]	Monitorea solo vehículos [3]	Error en clima adverso [4]	Errores de precisión [5]
Cámaras detección							
Subaweh y Wibowo 2017 [5]	X		X	X	X	X	
Ke et al. 2017 [22]	X		X	X	X	X	
Zhang et al. 2017 [7]	X		X	X	X	X	
Loop inductors							
Li et al. 2018 [6]	X		X	X	X		
Kwak y Kho 2016 [23]		X	X	X	X		
Lectores RFID							
Krausz et al. 2017 [8]	X			X	X	X	X
Chao y Chen 2014 [21]	X			X	X	X	X
Sherif et al. 2014 [24]		X		X	X	X	X
Infraestructura GSM/3G							
Chaturvedi y Srivastava 2016 [9]	X						X
GPS							
Ya et al. 2017 [25]		X			X	X	X
Liu et al. 2016 [26]		X			X	X	X

[1] Costos en mantenimiento y transmisión de datos
[2] Ubicación fija para la recolección de datos y cobertura solo en carreteras principales
[3] El monitoreo solo es aplicable a vehículos
[4] Pierde exactitud en climas adversos
[5] El dispositivo de medición tiene error de precisión

2.2. Analítica de datos en accidentes de tránsito

Las anomalías de tránsito, como los accidentes o incluso vehículos varados, son difíciles de predecir en comparación con las congestiones producidas por las horas pico, las vacaciones, trabajos en la vía, eventos culturales o deportivos. Nikolaev et al. [27] divide las anomalías como eventos predecibles (planificables) e impredecibles (no planificables). Las autoridades necesitan detectar rápidamente las anomalías para medir los cambios producidos al flujo de

tránsito y en caso de accidente actuar prontamente con los servicios paramédicos. Liu et al. [26] identifica dos formas de detectar anomalías, una forma es un sistema diseñado para modelar patrones de tráfico normales; la anomalía se detecta cuando el tráfico observado se desvía de lo que el sistema predice; y la otra forma es comparando el estado actual con el historial de días adyacentes.

En la literatura se encuentran trabajos recientes para la detección de anomalías y accidentalidad de tránsito; usando como fuentes de datos el GPS [10, 26, 25], Loop inductors [23] y RFID [24]; tanto para el análisis como la detección en tiempo real. En la tabla **2-1** se comparan estos trabajos. Kwak y Kho [23] usaron loop inductors para identificar mediante análisis de regresión logística las variables de tráfico más significativas que ocasionan accidentes de tránsito, entre ellas las velocidades inseguras, falta de atención visual, distancias muy cercanas entre vehículos y condiciones climáticas. Los datos de GPS no son tan efectivos para capturar accidentes de tránsito, debido a la baja frecuencia de muestreo y error de precisión de los puntos, trabajos novedosos como Zuo et al. [10] usan el sistema de posicionamiento satelital impulsado por China, llamado *Beidou (BDS)*, donde la precisión es menor de un metro y se puede analizar la trazabilidad del vehículo momentos antes del accidente, como cambios de carril, sin embargo esta sigue siendo una solución local.

2.2.1. Análisis del patrón de accidentes

Los datos extraídos por diversas fuentes y la minería de datos permiten elaborar análisis estadísticos sobre patrones de accidentes en tiempo y espacio. Moncada [28] elabora un patrón espacial de accidentes estimando la intensidad o número de accidentes por metro cuadrado mediante el método no paramétrico basado en kernel o **Kernel Density Estimation *(KDE)***.

La estimación de densidad de kernel es un método no paramétrico para estimar la función de densidad de probabilidad de una variable aleatoria a partir de varias observaciones conocidas o muestras. En este caso se trata de predecir el valor desconocido de un punto de referencia, interpolando el valor de los puntos conocidos, por lo que este método también se conoce como Interpolación de densidad de kernel (o en inglés Kernel Density Interpolation). Dado un conjunto de datos $x = \{x_1, x_2, ..., x_n\}$, la función de distribución de densidad $f(x)$ puede aproximarse utilizando un KDE tal que:

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{2-1}$$

Donde, n, es el número de datos o puntos (observaciones). h, es el ancho de banda que controla cuánto se expande la influencia de cada observación, también se conoce como el parámetro de suavizado. K, es la función de Kernel que define la forma y distribución. Para

este análisis se selecciona el *kernel gaussiano* que se expresa en la fórmula (2-2) y asigna los pesos siguiendo la distribución normal con una desviación estándar equivalente al ancho de banda.

$$K(X;h)\ \alpha\ exp(-\frac{X^2}{2h^2}) \tag{2-2}$$

La selección del ancho de banda es importante para estimar la función de densidad, una mala elección puede ocasionar *overfitting* y *underfitting*, si selecciona un valor bajo o elevado, respectivamente. Por esta razón existen varias estrategias para seleccionar un valor adecuado, una de estas es la *scott's rule* (2-3).

$$h = 1{,}06\ \cdot\ \hat{\sigma}\ \cdot\ n\frac{-1}{d+4} \tag{2-3}$$

Donde d el número de dimensiones.

2.2.2. Información de accidentes de tránsito desde redes sociales

Extraer la información de tráfico desde una sola fuente de información es insuficiente para el análisis de accidentes, en la tabla **2-1** se muestra que la mayoría de dispositivos frecuentemente usados tienen limitaciones como cobertura, costos y precisión, además es difícil inferir el tipo de evento monitoreado. Esto ha motivado a los investigadores a interesarse por estudiar la efectividad del papel de las redes sociales como fuentes de datos para la detección de incidentes de tránsito, para ayudar a comprender las condiciones del tráfico, lograr una mayor cobertura e incluir todos los actores viales como peatones y pasajeros [12], y la posibilidad de analizar otros factores como los eventos masivos y el estado de la carretera [14].

Para detectar variables de tráfico, debido a que su API está ampliamente desarrollada y disponible, los trabajos recientes en extracción de información en redes sociales se concentran en Twitter como fuente de datos. Wang et al. [3] validó su efectividad al compararlo con datos recolectados de GPS en buses de transporte público, al igual que Gu et al. [14] también demuestra una primera aproximación al correlacionar el flujo de tráfico con el desarrollo de eventos sociales y la infraestructura vial. Zhang et al. [15] compara los reportes de accidentes extraídos en Twitter con datos obtenidos de loop inductors y bases de datos de la policía obteniendo resultados positivos. Diversos autores han diseñado métodos para la extracción de los datos relevantes a incidentes de tráfico desde Twitter [16, 14, 18, 19, 20], sintetizados en la tabla **2-2**, donde contemplan una fase para la adquisición de datos, clasificación, detección de ubicaciones y extracción de coordenadas de los incidentes publicados. Estos trabajos emplean algoritmos de aprendizaje automático y extraen categorías de incidentes como congestión, accidentes, eventos sociales, clima e intervenciones de obras viales.

Tabla 2-2: Cuadro comparativo de revisión de literatura en métodos y técnicas de extracción de información de transporte en redes sociales

Fases	**Gu et al. 2016 [14]**	**Zhang et al. 2018 [15]**	**Arias et al. 2019 [16]**	**Salas et al. 2017 [19]**	**Pereira et al. 2017 [29]**	**Wang et al. 2017 [12]**
Adquisición de datos						
Idioma	Inglés	Inglés	Español	Inglés	Portugues	Inglés
Región	EE.UU.	EE.UU.	Ecuador	Inglaterra	Brasil	EE.UU
Periodo	1 MES	1 AÑO	1 MES	3 MESES	1 MES	1 AÑO
Clasificación						
Clase	Incidentes	Accidentes	Incidentes	Incidentes	Viajes	Incidentes
Vector de características	BoW [1]	BoW	BoW	BoW	word2vec	Tokenizer
Clasificador	Naive Bayes	Deep Belief Network	SVM [2]	SVM	SVM	Similitud de palabras
Detección de ubicaciones						
Clases	Ubicación	Ubicación	Ubicación	Ubicación		Ubicación
Modelo NER	Basado en reglas	Geotagged [3]	Distancia Levenshtein	Stanford NER System		Geotagged
Geolocalización						
Método	Diccionario + ArcGIS	Geotagged				Geotagged
Validación						
Comparación de datos	GPS y Reportes	Loop inductors y Reportes				GPS

[1] En inglés Bag of Words o BoW. Es un método para representar vectorialmente las frases.
[2] En inglés Support Vector Machine o SVM. Modelo de aprendizaje automático supervisado.
[3] Información de las coordenadas de los usuarios de dispositivos móviles, sin embargo debe ser activado y en la práctica menos del 1 % de los tweets contienen esta información.

Los resultados obtenidos en Zhang et al. [15], Gu et al. [14] y Wang et al. [3] comprueban la efectividad de las redes sociales para la detección de accidentes, agregando que no todos los accidentes ocurridos son publicados en las redes sociales y su uso debe considerarse como complementarias, las cuales amplían la información obtenida por las técnicas con dispositivos actuales y aportan nuevos datos en segmentos de carreteras secundarias y arteriales. Por esta razón algunos trabajos han surgido para combinar las distintas fuentes disponibles y realizar una mejor previsión del tráfico, usando Big data [30], técnicas de Deep Learning [31, 32], factorización de matrices [3], entre otros. Finalmente, para mejorar los métodos de extracción en redes sociales varios autores determinan desafíos en común, como es la presencia de ruido en las publicaciones, errores de ortografía y evolución constante del lenguaje informal. En la tabla **2-3** se resumen los desafíos más comunes. Estos desafíos permanecen y evolucionan en

el tiempo por la naturaleza del lenguaje empleado en las redes sociales y la cultura local de una generación a otra de usuarios.

Tabla 2-3: Desafíos encontrados en los trabajos relacionados en detección de incidentes de tránsito con el uso de las redes sociales.

Autores y año	**Desafíos en las redes sociales**				
	Publicaciones altamente ruidosas	**Errores de ortografía**	**Los mensajes no tienen estructura fija**	**Evolución del lenguaje empleado en las redes**	**Ambigüedad en nombres de lugares**
Gu et al. 2016 [14]	X	X	X		X
Wang et al. 2015 [3]	X	X	X	X	X
Zhang et al. 2018 [15]	X		X	X	X
Kuflik et al. 2017 [13]	X	X	X	X	X
Caimmi & Vallejos 2016 [33]	X	X	X	X	
Kurniawan et al. 2016 [18]	X	X	X	X	X
Salas et al. 2017 [19, 20]	X	X	X	X	

2.3. Métodos de extracción de incidentes desde redes sociales

Twitter permite extraer información fácilmente dado que la mayoría de perfiles son públicos, sumado a esto son varias las propuestas de métodos de extracción de datos relevantes a incidentes de tráfico en esta red social, sintetizados en la tabla **2-2**, donde se contemplan varias fases para la adquisición, clasificación y reconocimiento de entidades como ubicación. En cada fase se emplean métodos de procesamiento de lenguaje natural para resolver varios de los retos encontrados de la tabla **2-3**, que tiene su origen debido a la naturaleza del uso del lenguaje en las redes sociales, como es el uso de abreviaciones y la aparición de nuevas expresiones informales del lenguaje.

2.3.1. Clasificación automática de tweets relacionados a accidentes

En la literatura existen métodos propuestos para clasificar tweets relacionados a eventos de tránsito [16, 29, 19] y accidentes de tránsito [34, 35]. Estos trabajos constituyen de un

clasificador automático usando algoritmos de aprendizaje automático [14] y técnicas de procesamiento de lenguaje natural [33, 36], estas fases incluyen el preprocesamiento de los tweets, extracción de características y modelos de clasificación.

- *Preprocesamiento.* En esta fase se recibe las publicaciones y se aplica un proceso de limpieza y normalización, dividiendo las frases en palabras o en segmentos llamados tokens (*tokenización*); se eliminan los enlaces de webs o urls, menciones de usuario (@), palabras vacías o *stopwords*; caracteres especiales y números; finalmente se reduce las palabras según su lema (*lematización*) o su raíz (*stemming* en inglés).

- *Extracción de características.* Las oraciones generadas en el paso anterior se representan en un espacio de características o vectores para mejorar el desempeño de los modelos matemáticos, en la literatura se ha implementado la técnica *TD-IDF* que representa la frecuencia de las palabras del tweet con respecto a la cantidad de tweets en el corpus [33, 37] y *word2vec* que es una red neuronal entrenada que genera arreglos semánticos más pequeños teniendo en cuenta el contexto de las palabras en la conformación de la frase [29, 38].

- *Clasificación automática.* Con los resultados de la fase anterior se construye un modelo de aprendizaje automático que clasifique si un tweet esta relacionado a un incidente o no, algunos algoritmos utilizados en la literatura para esta tarea son *Naive Bayes* [34, 14], *Support Vector Machine* [19, 36, 18] y *Convolutional Neural Networks* [38]. Cada modelo tiene parámetros diferentes que se deben configurar para obtener mejores resultados, la forma de buscar estos parámetros es mediante la búsqueda intensiva optimizada como es el método conocido en inglés como grid search.

2.3.2. Reconocimiento de Entidades Nombradas

El reconocimiento de entidades es una tarea de procesamiento de lenguaje natural que extrae información a partir del texto, como ubicación, organización, tiempo, persona, entre otros. En la literatura se a explorado diferentes técnicas para la extracción en Redes Sociales como *Rules-based* [33] y *Sequence Labeling* [39, 40], siendo esta última el estado del arte y sin requerir una construcción profunda de diccionarios o *gazetteers*; se basa en un conjunto de datos etiquetado manualmente que le permite entrenar, aprender y generalizar en nuevos datos. Para el caso de reportes de accidentes de transito, se busca extraer las entidades de Ubicación y Tiempo. En este caso, algunos autores [14, 16] siguen una metodología similar, empezando con un preprocesamiento de los tweets y seleccionando una técnica de reconocimiento de ubicaciones o también conocida en inglés como *geoparsing* [41].

- *Preprocesamiento.* Se limpia las urls, emoticones y algunos caracteres especiales. Los *@usernames* y *#hashtags* pueden contener información de ubicación y algunos autores no los eliminan, estas expresiones están conformadas generalmente por la unión

de dos o tres palabras sin espacios, por lo que se aplica un método para expandirlos [42, 43]. También, se corrigen los errores de ortografía y las abreviaciones empleadas comúnmente en las redes sociales [41]. Para el reconocimiento de entidades no se eliminan las *stopwords* porque estas palabras son claves para distinguir el contexto de la siguiente palabra y su ambigüedad.

- *Reconocimiento de entidades*. Hay diferentes técnicas que se pueden emplear para extraer entidades de tipo ubicación en el contenido del tweet, generalmente se agrupan en dos: *Based-Rules* y *Sequence Labeling*, algunos autores usan un híbrido de ambas técnicas [41, 44, 45, 14, 43]. La diferencia principal esta en el uso de recursos externos al contenido mismo del texto, como es el caso de las técnicas de *Based-Rules* que construyen diccionarios de términos, toponimias y metonimias locales, conocidos como *gazetteers*, además de conjunto de reglas o listado de expresiones regulares. Mientras que las técnicas de *Sequence Labeling* usan métodos de *machine learning* supervisado, haciendo uso únicamente de un conjunto de características y datos etiquetados a partir del texto; entrenando, aprendiendo y generalizando en datos no vistos. Los modelos con mejor desempeño para la tarea de *sequence labeling* en redes sociales son *CRF* [45], *BiLSTM* [39] y *BiLSTM+CRF* [40].

2.3.3. Geocodificación

Una vez extraídas las entidades de ubicación se pasan a un *geocoder* que devuelve una coordenada geográfica aproximada a la referencia del lugar, hay recursos externos que se pueden utilizar como *Google Maps*, *OpenStreetMap*, *Geonames*, *DBPedia*, entre otros. También hay autores que construyen *geocoders* propios que buscan coincidencias en un *Gazetteer* utilizando algoritmos de búsqueda como *String Match* o *Fuzzy String Match* [14].

3 Método propuesto

En este trabajo se propone un método que permita la extracción de reportes de accidentes en las publicaciones en español de Twitter en Bogotá, basado en los estudios previos de la extracción de información de tránsito como eventos [14, 16, 29] y accidentes [15, 34] en Twitter. El enfoque propuesto esta dividido en cuatro fases para: la recolección de datos, clasificación de tweets relacionados a accidentes, reconocimiento de entidades para la extracción de ubicación y tiempo, y la geolocalización de los accidentes reportados. En la figura **3-1** se muestra el diseño. En los siguientes capítulos se abordará en específico cada una de las fases del método propuesto.

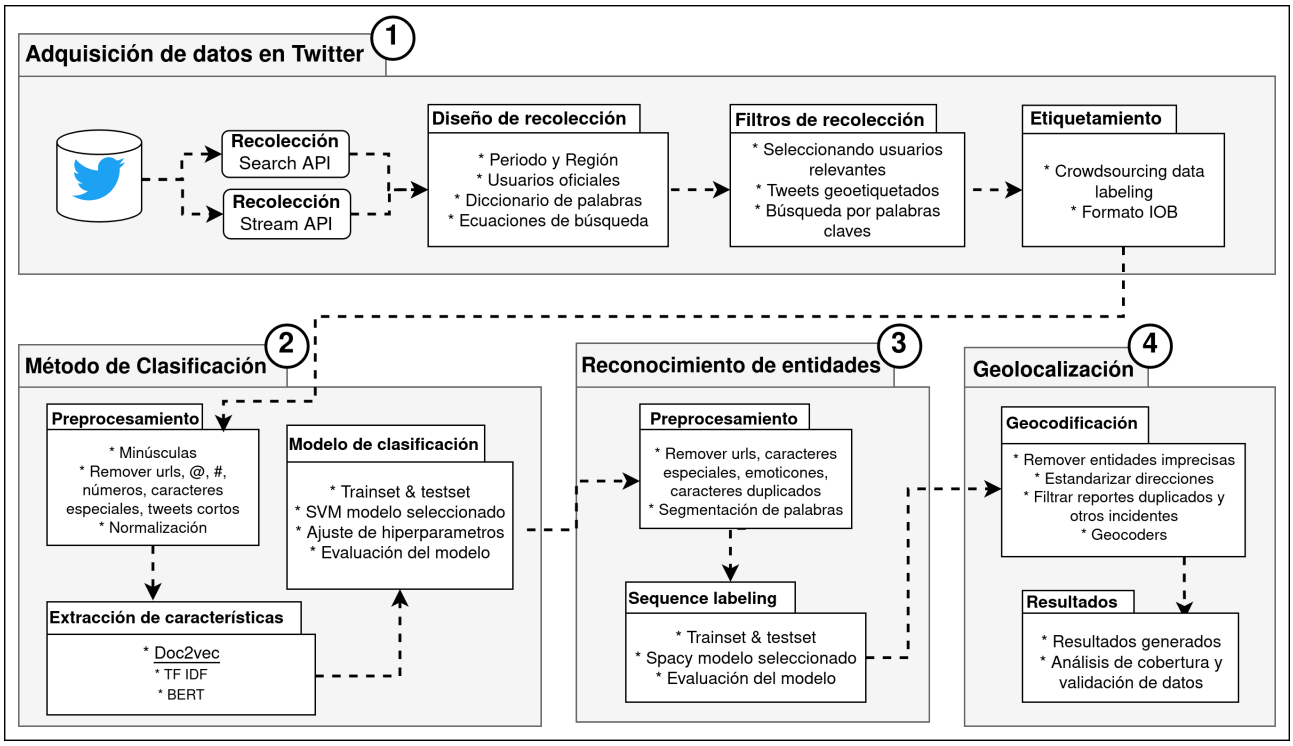

Figura 3-1: Método propuesto para la detección automática de accidentes de tránsito en Twitter (**1**) Adquisición de datos en Twitter; (**2**) Método de clasificación para filtrar automáticamente tweets de reportes de accidentes; (**3**) Extracción de entidades ubicación y tiempo del reporte del accidente de tránsito; (**4**) y geolocalización de los accidentes.

3.1. Fase 1. Adquisición de datos en Twitter

En la primera fase se contempla las herramientas para la recolección de los datos en Twitter y la construcción de un dataset etiquetado usando el mecanismo de *crowdsourcing data labeling*, en la cual colaboran un grupo de personas en esta tarea. Los tweets se recolectaron mediante la API de Twitter: *Search API* y *Stream API*. En el diseño del proceso de recolección se establece como objetivo la recolección de tweets para los usuarios de Bogotá, para recolectar la información de accidentes se define un listado de palabras claves relacionadas a tránsito y se combinan en ecuaciones de búsqueda. Se definen 4 esquemas de recolección donde se tiene en cuenta la búsqueda de tweets por palabras claves, tweets de cuentas de usuarios oficiales y tweets geoetiquetados dentro de la área urbana de la ciudad. Finalmente, para el proceso de etiquetado de datos para el modelo de clasificación se contó con la colaboración de 30 personas y un mecanismo de selección de 3 votos por tweet, para el modelo de *sequence labeling* se etiquetaron los datos usando la herramienta *Brat Annotation Tools* y el formato de etiquetamiento llamado *IOB*, por sus siglas en inglés *Inside-Outside-Beginning*.

3.2. Fase 2. Método de clasificación

La segunda fase implementa un método de clasificación automática con un algoritmo de aprendizaje supervisado, para la construcción de este modelo se requiere algunos pasos previos que preparan los datos para un mejor desempeño en el aprendizaje. Primero, los tweets pasan por un proceso de preprocesamiento que incluye limpieza y normalización de las publicaciones. Segundo, se requiere transformar las publicaciones en una representación numérica vectorial para el algoritmo de machine learning, para esto se evalúan diferentes métodos conocidos en la literatura como *TF IDF*, *doc2vec* y *BERT*. Finalmente, el algoritmo de clasificación seleccionado es *Support Vector Machine* construido con 3804 tweets, donde la mitad esta relacionado con accidentes de tránsito, este método permite filtrar los tweets relevantes a accidentes de tránsito con los no relevantes.

3.3. Fase 3. Reconocimiento de entidades

En la tercera fase se usa una técnica de *machine learning* para el reconocimiento de entidades que permita la extracción de ubicación y el tiempo del accidente a partir del texto. Para esta tarea también se requiere un proceso especial de limpieza y normalización de los datos, distinto a la fase anterior del método de clasificación. Se agrega en la normalización del tweet la segmentación de palabras para el tratamiento de los *#hashtags* y menciones de usuario (*@username*), por considerar que estas expresiones contienen información de ubicaciones [43]. Finalmente se construye un modelo de *sequence labeling* mediante el re-entrenamiento del modelo de la librería *Spacy* con el conjunto de datos propio de 1340 tweets usando el

estándar *IOB*.

3.4. Fase 4. Geolocalización

En la última fase, después de extraída la entidad de ubicación se pasa por un *geocoder* que se encarga de devolver las coordenadas geográficas de la dirección mencionada. Esta fase comienza con un preprocesamiento especial del texto sobre las entidades de ubicación extraídas de la fase anterior, primero se comienza con remover las entidades imprecisas, considerando entre ellas las direcciones demasiado cortas que no poseen claridad en la ubicación del accidente. Después, se pasa por un proceso de estandarización de direcciones, que se encarga de mejorar la legibilidad de las direcciones escritas por los usuarios, que en ocasiones carecen de un uso formal del lenguaje y es común el uso de abreviaciones y expresiones coloquiales. Con el propósito de eliminar el ruido en las publicaciones extraídas en Twitter y la redundancia para el análisis de accidentalidad, una vez normalizadas las direcciones detectadas se filtran reportes duplicados e incidentes de otras categorías distintas a accidentes. Una vez estandarizadas las direcciones un *geocoder* devuelve una coordenada geográfica. Finalmente con los datos generados en Twitter se realiza un análisis de los resultados y se valida la eficacia de los accidentes extraídos frente otras fuentes, como el reporte oficial de la secretaría de movilidad.

4 Adquisición de datos en Twitter

En este capítulo se presenta el proceso de recolección y construcción de los datos para este trabajo. Primero se debe diseñar la estrategia de recolección en Twitter, seleccionando el periodo de tiempo para la recolección, la ciudad destino, las cuentas de usuario y contenidos relacionados con los accidentes de tránsito. Segundo, aprovechando los diferentes mecanismo que Twitter brinda para acceder a sus datos se construye cuatro filtros de recolección, buscando lograr la mayor cantidad de tweets recolectados. Finalmente, una vez terminado el periodo de recolección se continúa con la construcción de los dos conjuntos de datos que se van a usar para los modelos, de clasificación y sequence labeling.

La recolección de tweets se lleva a cabo con dos recursos disponibles por la API de Twitter, el primer recurso es llamado ***Search API***, que permite buscar tweets por palabras claves y de cuentas de usuario seleccionadas, el segundo recurso es un servicio de ***Stream API*** que recibe en tiempo real los tweets publicados instantáneamente dentro de una región geográfica seleccionada, sin embargo la versión gratuita solo permite descargar el 1 % de los tweets. Debido a los límites impuestos por la API gratuita y con el propósito de recolectar la mayor cantidad de tweets, se construyen cuatro tipos de filtros de recolección que combinan ambos recursos mencionados anteriormente.

4.1. Diseño de la recolección

Antes de comenzar con el proceso de recolección de datos con la API de Twitter se debe definir previamente algunos criterios de búsqueda, como la construcción de un diccionario de palabras claves, establecer cuentas de usuario relevantes y definir la región geográfica y periodo de recolección.

- *Periodo y Región.* Los tweets en español recolectados corresponden a los meses de octubre a diciembre del 2018 y enero a julio del 2019, son diez meses de tweets de la ciudad de Bogotá, Colombia.

- *Usuarios oficiales.* Se seleccionaron algunas cuentas de usuarios oficiales por sus relevantes publicaciones constantes de tránsito sobre la ciudad de Bogotá. Las cuentas de usuario oficiales seleccionadas se muestran en la tabla **4-1**.

- *Diccionario de palabras*. La API de Twitter brinda la posibilidad de extraer tweets según una coincidencia de secuencia de palabras claves, para determinar un listado de estas palabras se extrajo manualmente palabras o secuencia de palabras llamadas *n-grams*, con mayor ocurrencia entre documentos, reportajes y tweets relacionados a accidentes de tránsito seleccionados previamente.

- *Ecuaciones de búsqueda*. A partir del diccionario de palabras se construyen algunas ecuaciones de búsqueda que optimicen la extracción, incluyendo también cuentas de usuario y hashtags en la búsqueda. Como se observa en la tabla **4-2**, estas ecuaciones también proporcionan la definición de exclusión de palabras irrelevantes para descartar algunos tweets, el uso del signo '-' indica exceptuar estas palabras.

Para las peticiones anteriores se realizó un programa en Python con la librería *Tweepy* que se conecta a la API de Twitter realizando peticiones diariamente durante todo el periodo de extracción, se recomienda realizar monitoreo constante de esta aplicación porque al tener acceso gratuito a la API es común que en ocasiones la conexión falle.

Tabla 4-1: Nombres de usuarios de cuentas oficiales de organizaciones y medios de comunicación.

@BogotaTransito	@Citytv	@RedapBogota	@WazeTrafficBog
@CIVICOSBOG	@rutassitp	@SectorMovilidad	@UMVbogota
@idubogota	@transmilenio	@IDIGER	

Tabla 4-2: Ecuaciones de búsqueda definidas para tweets en español y Bogotá.

Ecuaciones de búsqueda
("accidente" OR "choque" OR "incidente vial" OR "incidente" OR "choque entre") -RT -"plan de choque"
("atropello" OR "tráfico" OR "trafico" OR "tránsito" OR "transito" OR "#trafico" OR "#traficobogota" OR "sitp" OR "transmilenio") -RT

4.2. Filtros de recolección

Usando las dos herramientas que Twitter provee para acceder a los tweets, Stream API y Search API, se definieron cuatro tipos de filtros como diferentes métodos de recolección.

A. *Filtro Stream Bogotá*. Consulta para extraer los tweets publicados al instante en Bogotá con el recurso de Stream API, la versión gratuita solo extrae el 1 % de los tweets, además solo trae los tweets de los usuarios que tienen la geoetiqueta activada o celulares con el gps activado.

B. *Filtro Stream Follow/Timeline User*. Tweets recolectados usando Twitter Stream API. Se selecciona uno o varios usuarios; y se comienza una descarga automática en el momento que este usuario publique un tweet u otro usuario lo mencione o etiquete en alguna publicación.

C. *Filtro Search Token*. Tweets recolectados usando Twitter Search API. En este caso se utilizan las ecuaciones de búsqueda definidas en la tabla **4-2**; además se filtran los resultados para la ciudad de Bogotá y en español.

D. *Filtro Search Timeline User*. Tweets recolectados de cinco usuarios seleccionados de la tabla **4-1** usando Twitter Search API (*@BogotaTransito, @Citytv, @RedapBogota, @WazeTrafficBog, @CIVICOSBOG*). Se descargan los tweets publicados por estas cuentas o de sus timelines, la API permite extraer el historial de tweets del usuario.

Con este mecanismo, en total se recolectaron 4.973.900 tweets. En la tabla **4-3** se muestra la cantidad extraída según el filtro de recolección y en la tabla **4-4** se muestran algunos tweets extraídos.

Tabla 4-3: Cantidad de tweets recolectados según filtro.

Collection Filter	**# Collected Tweets**
Stream Bogotá	4 027 313
Stream Follow/Timeline User	574 816
Search Token	271 153
Search Timeline User	100 618

Tabla 4-4: Muestra de tweets recolectados en Bogotá.

Tweets	**Descrición**
Hueco causa accidentalidad Cra. 68c #10-16 sur, Bogotá	Accidente
Justo ahora 1:22pm en 21 angeles Av Suba gratamira (calle 145 Av Suba) complicaciones viales por accidente @BogotaTransito @ALCALDIASUBA11 @SectorMovilidad	Accidente
Semáforos de la Carrera 24 con Calle 9 en amarillo intermitente, Tanto por la calle como por la carrera con riesgo de incidente vehicular	No Accidente
Incidente vial entre bus y un motociclista en la calle 86a con carrera 111a. Unidad de @TransitoBta y asignadas.	@BogotaTransito
en la avenida Primero de Mayo con carrera 69 en sentido occidente - oriente chocan un taxi y una motocicletaen // la avenida de La Esperanza con carrera 68 A en sentido occidente - oriente chocan un vehículo particular y una camioneta	Dos reportes diferentes en el mismo tweet
#ArribaBogotá Por culpa de este hueco en la calle 27sur, una mujer sufrió un grave accidente de tránsito.	Ubicación imprecisa

4.3. Etiquetamiento de datos

Los primeros tweets recolectados se usan para construir un conjunto de datos inicial para el entrenamiento y evaluación de los modelos de *Machine Learning* de las siguientes fases. Para esta tarea el conjunto de datos se etiqueta para expresar la verdad fundamental (o en inglés *ground truth*) que se desea que los modelos aprendan. Dependiendo de la fase cambia el proceso y formato de etiquetado, como se muestra a continuación.

4.3.1. Etiquetamiento crowdsourcing de datos para la Clasificación

Para el entrenamiento del modelo de clasificación de tweets de accidentes se ha diseñado una estrategia de etiquetamiento colaborativo conocido como crowdsourcing, en la cual participaron 30 personas[1], a quienes se les dio instrucciones para realizar la tarea de etiquetado. Cada participante debía evaluar un tweet para clasificarlo manualmente en una de las tres categorías definidas como: relacionado a accidente de tránsito, no relacionado y no sabe/no responde. En la figura **4-1** se muestra un ejemplo de la votación. En este proceso se seleccionaron aleatoriamente 22582 tweets de los meses de octubre a diciembre del 2018, como resultado 3505 tweets se etiquetan como *"relacionado a accidentes de tránsito"*. Una vez un tweet es evaluado por 3 participantes, el proceso de selección de la etiqueta correcta se

[1]Entre los colaboradores se encuentran estudiantes del grupo de investigación PLaS de la Universidad Nacional de Colombia y estudiantes del programa de Ingeniería Informática de la Corporación Universitaria Autónoma de Nariño extensión Villavicencio.

Figura 4-1: Proceso de votación de tweets diseñado para la aplicación Tagenta.

realiza mediante una votación, donde la elección unánime de las 3 personas deben coincidir con la etiqueta seleccionada, de lo contrario se descarta el tweet para el conjunto de datos de entrenamiento y prueba. Este proceso demoró un mes, además requirió el desarrollo y despliegue de una aplicación web propia para este fin, que nombramos como Tagenta y se puede ver en la figura **4-1** y **4-2**.

4.3.2. Etiquetamiento para el modelo de Sequential Labeling

Para entrenar y evaluar el modelo de reconocimiento de entidades se seleccionó una muestra de 1340 tweets del conjunto de datos etiquetado anteriormente, estos tweets debían cumplir con el criterio de tener menciones de ubicaciones, lugares y direcciones. Posteriormente, este conjunto de datos fue etiquetado manualmente usando el formato *IOB (Inside-outside-beginning)*, para esta tarea se utilizó la herramienta de etiquetado llamada *Brat Annotation Tools*, como se ve en la imagen Figura **4-3**. Las etiquetas definidas son de ubicación (*Location*), que hace referencia al lugar del reporte; y tiempo (*Time*), que hace referencia a la hora o fecha ocurrida el incidente, generando así 5 etiquetas: *B-loc*, *I-loc*, *B-time*, *I-time* y *O* (ver Figura **5-1**), La etiqueta *'O'* hace referencia a *'Otros'*.

Figura 4-2: Aplicación Tagenta diseñada para etiquetar los tweets para el modelo de clasificación, las herramientas utilizadas fueron NodeJs, SailsJs y MongoDB.

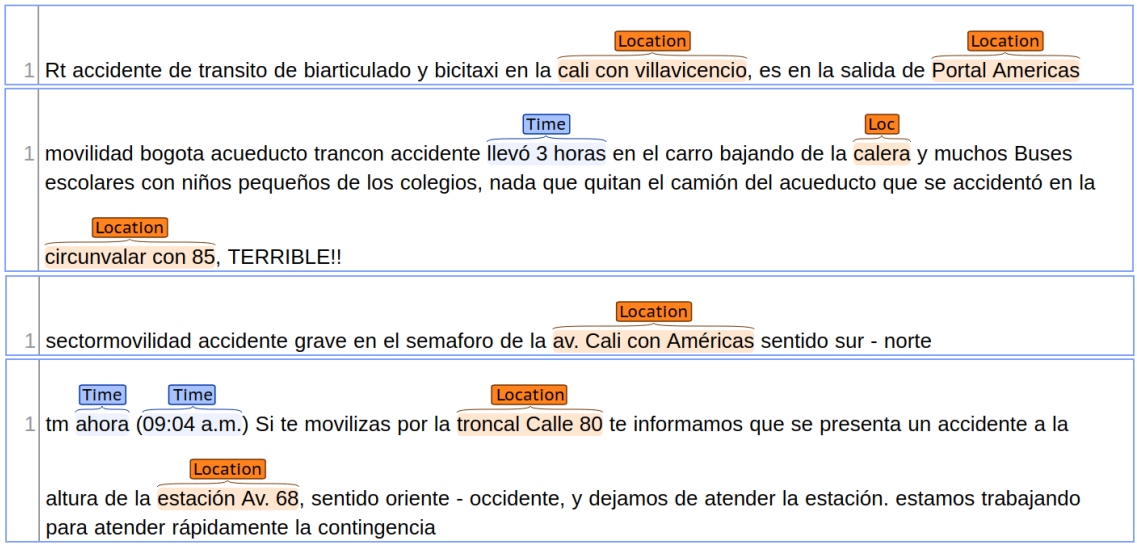

Figura 4-3: Proceso de votación de tweets diseñado para la aplicación Tagenta.

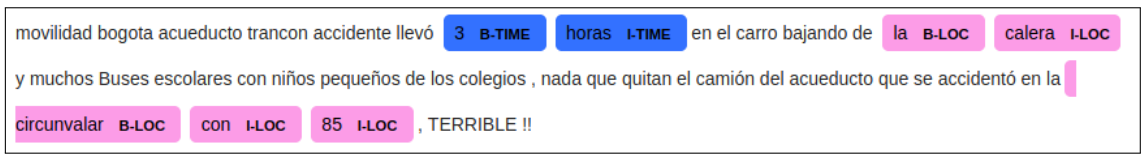

Figura 4-4: Aplicación Tagenta diseñada para etiquetar los tweets para el modelo de clasificación, las herramientas utilizadas fueron NodeJs, SailsJs y MongoDB.

5 Clasificación de tweets de accidentes de tránsito

Este capítulo se ha dividido en seis secciones que explican cada uno de los procesos involucrados para seleccionar el modelo deseado. Primero se explica brevemente el conjunto de datos seleccionado para el entrenamiento y evaluación del modelo. El proceso de normalización y limpieza por la que pasan los tweets son explicados en la segunda sección. Tercero, se explica la transformación empleada para representar el contenido del tweet en una representación vectorial numérica, conocido en inglés como *Word Embedding*. La cuarta sección se trata del modelo de clasificación automática que categoriza los tweets como relacionados a accidentes o no relacionados. Quinto y finalmente, se explica el experimento realizado donde se evaluó diferentes técnicas propuestas en la literatura para el procesamiento de publicaciones en redes sociales.

5.1. Conjunto de datos

Como se menciona en el capítulo **4**, una vez recolectado los primeros tweets se construye el conjunto de datos para entrenar y evaluar los modelos de *machine learning*. Después del proceso de etiquetamiento *crodwsourcing* (sección **4.3.1**) se obtuvieron 3505 tweets etiquetados como relacionados a accidentes de tránsito entre los meses de octubre a diciembre del 2018. Sin embargo, no se usaron todos esos tweets para el entrenamiento del modelo de clasificación. La mayoría de estas publicaciones, 2898 tweets, son de las cuentas oficiales de *@BogotaTransito* y *@rutasstip* y contienen tweets con el mismo formato de oración inicial: *"Incidente vial entre"*. Por esta razón se seleccionó una fracción de los tweets de estas cuentas oficiales, de manera que sus tweets solo representen un 30 % de la muestra seleccionada y los demás usuarios un 70 %, con el fin de evitar un sesgo en el conjunto de datos positivo o relacionados a accidentes de tránsito.

Teniendo en cuenta lo anterior, el conjunto de datos para el modelo de clasificación contiene **3804** tweets, donde la mitad del conjunto están relacionados a reportes de accidentes de tránsito (**Traffic accident TA, positive class**) y la otra mitad no están relacionados (**Non-traffic accident NTA, negative class**).

5.2. Preprocesamiento

Debido a su naturaleza, las publicaciones en las redes sociales suelen emplear lenguaje no formal, emoticones, menciones de usuario y etiquetas de temas como hashtags, entre otros. Para identificar un claro patrón en los reportes de accidentes de tránsito, estos mensajes deben pasar por un proceso que permita limpiar y normalizar su contenido. Los pasos aplicados a los contenidos del tweet son los siguientes.

A. **Conversión a minúsculas.** En el proceso de *word embedding* se construye un diccionario de palabras, y para que este no sea demasiado grande todas las palabras de los tweets se pasan a minúsculas, reduciendo el costo computacional.

B. **Limpieza de caracteres no alfabéticos:** Se elimina caracteres especiales, correo, urls, incluyendo los símbolos @ y # que suelen ser populares en Twitter.

C. **Normalización con Lematización:** En español las palabras tienen variaciones y están conjugadas en diferentes formas. Para construir un diccionario más generalizado, se aplica un proceso de normalización a los tweets que consiste en reducir la palabra a su lema original. Otra forma de normalización es *stemming* que en lugar de reducir a lema, reduce la palabra a su raíz, que consiste en quitar y reemplazar sufijos de las raíces de las palabras, en las siguientes secciones se evalúa el desempeño del método propuesto al aplicar *lematización* o *stemming*.

D. Se descartan tweets que después de los pasos anteriores su longitud de palabras o tokens sea menor a 3. Debido a que no cuentan con información suficiente del incidente.

Uso de las stopwords. Las *stopwords* son palabras que consisten en artículos, pronombres, preposiciones, entre otros. Estas palabras se consideran sin valor o vacías debido a que suelen estar presentes en publicaciones relacionadas y no relacionadas, por lo tanto no generan valor para discriminar un tipo de publicación con otra. Algunos modelos de *word embedding* de la sección **5.3** se pueden hacer cargo de estas stopwords sin tener que eliminarlas previamente, por ejemplo el algoritmo de *TFIDF* puede ignorar aquellas palabras que se repiten con mayor frecuencia en el corpus. Otros algoritmos basados en *Neural Networks*, como *doc2vec* y *BERT*, no eliminan las stopwords porque son claves para darle contexto a cada oración. En la tabla **5-1** se compara las diferencias de *TFIDF* si se elimina o no las stopwords, es posible que si se limpia estas palabras utilizando un diccionario de stopwords genérico, se puede perder palabras relevantes en el contexto de reportes de accidentes de tránsito, como las preposiciones *"entre"* y *"con"*. En la sección de evaluación (**5.5.3**) se compara los resultados obtenidos al aplicar o no este proceso.

Tabla 5-1: Resultado de limpieza y normalización del método *stem* utilizando *TFIDF* con o sin *stopwords*.

Texto Original [1]	**Limpiar Stopwords + Stemming**	**Stemming**
#Atención: se presenta siniestro vial entre un peatón y el tren a la altura de la Av. NQS con calle 67. Unidad de @TransitoBta asignada	altur asign atencion av call nqs peaton present siniestr tren unid vial	altur asign atencion av call con el entre nqs peaton present se siniestr tren un unid vial
Incidente vial entre taxi y motocicleta en la calle 75 con carrera 88 intersección.Unidad de TransitoBta asignada.	asign bta call carrer incident interseccion motociclet taxi transit unid vial	asign bta call carrer con entre incident interseccion motociclet taxi transit unid vial

[1] Para presentar esta tabla se eliminaron algunos caracteres especiales y emoticones.

5.3. Word Embedding

Después de aplicar el proceso de limpieza y normalización, el texto resultante se debe representar en un espacio vectorial numérico, este proceso es comúnmente conocido como *Word Embedding*. La técnica seleccionada en este trabajo es doc2vec [46] que ha demostrado buenos resultados con el tratamiento de publicaciones en Redes Sociales [38, 47, 40]. Un ejemplo es el trabajo de Pereira et al. [29] que implementaron doc2vec para la clasificación de tweets en portugués relacionados a tránsito de las ciudades de Sao Pablo y Río de Janeiro.

Doc2vec es un método de *embedding* propuesto por Le & Mikolov et al. [46] que extiende la implementación de word2vec [48] a secuencia de palabras, frases o párrafos. Consiste en una red neuronal de una sola capa oculta que tiene el objetivo de aprender sin supervisión el contexto de las oraciones y las palabras utilizadas, generando una representación vectorial que tiene en cuenta la semántica de las oraciones. Doc2vec tiene dos métodos diferentes que se pueden utilizar, en este caso seleccionamos ***DBOW*** o ***Distributed Bag of Words*** por su notable desempeño en Redes Sociales en trabajos recientes como el de Okur et al. [47] y Agilar et al. [40]. DBOW predice la representación de la palabra de contexto según una palabra destino.

Además para evaluar el desempeño de doc2vec en tweets en español se probaron otras técnicas de embedding usadas en la literatura como *TF IDF* y *BERT*.

- *TF IDF.* Este modelo calcula una representación teniendo en cuenta la frecuencia de ocurrencia de una palabra en el mismo tweet y en toda la colección del corpus.
- *BERT.* Se usa un método de Embedding pre-entrenado para extraer la representación vectorial de un texto, fué lanzado por Google a finales del 2018 [49] basándose en la idea de los *transformers* que son redes neuronales profundas que constan de varias capas de *encoders*. BERT dispone de dos versiones, una de 12 capas para el modelo

base y otra de 24 capas para el modelo extendido. Para este trabajo se utilizó el modelo entrenado sensible a mayúsculas de 12 capas de Cañete et al. [50].

5.4. Modelo de Clasificación

El último proceso de esta fase es la construcción de un modelo de *machine learning* para realizar una clasificación binaria, predecir si un tweet está relacionado o no a un accidente de tránsito. El modelo seleccionado es **Support Vector Machine** o **SVM** por su implementación en varios trabajos relacionados con tweets de tránsito que han demostrado su efectividad frente a otros modelos [29, 35, 37, 18, 36, 19]. De igual manera para comparar el desempeño de SVM con tweets en español, se evalúa otros modelos utilizados en la literatura como Naive Bayes (NB), Random Forest (RF) y Neural Network (NN), este último se selecciona como recomendación de los autores de BERT para su implementación en tareas de clasificación. Los resultados obtenidos de esta comparación se presentan en la siguiente sección de experimentos y resultados (sección **5.5.3**).

5.5. Experimentos y Evaluación

El objetivo es comparar la eficiencia de cada método de preprocesamiento en la sección **5.2**, cada método de embedding TFIDF, Doc2vec y BERT, y cada algoritmo de machine learning en la sección **5.4**. Las métricas utilizadas para esta comparación son **Accuracy, Recall, Precision y F1 score**, estos resultados permiten seleccionar la mejor técnica de clasificación de tweets relacionados con accidentes de tránsito. Para la evaluación del modelo este conjunto de datos se dividió usando dos estrategias distintas, **1)** *cross validation* con k igual a 10 para evaluar los modelos de clasificación *Support Vector Machine (SVM)*, *Random Forest (RF)* y *Naive Bayes (NB)*. Y **2)** en el caso de la *Neural Network (NNs)* se realizó la división del dataset en 70 % (2662 tweets) para entrenamiento y 30 % (1142 tweets) para prueba. Los parámetros utilizados para TFIDF, SVM, RF y NNs se encontraron mediante una búsqueda de hiperparámetros utilizando el método *grid search* de la librería *Sklearn* disponible en Python.

5.5.1. Métricas de evaluación

Para evaluar el desempeño de los modelos de *machine learning* propuestos, empleamos un grupo de métricas por clase llamadas en inglés como Accuracy, Recall, Precision y F1-score. Esta medidas son ampliamente usadas en trabajos relacionados para evaluar modelos de clasificación y *sequence labeling* [19, 38, 33, 35]. Como estas métricas son para una determinada clase con etiqueta l, se calcula los siguientes índices primero: *true positive (TP)* son la cantidad de instancias de la etiqueta l que se clasificaron correctamente con la etiqueta l,

true negative (TN) son la cantidad de instancias con cualquier otra etiqueta excepto l que fueron clasificadas correctamente. Por otro lado, los *false positive (FP)* son la cantidad de instancias con cualquier otra etiqueta excepto l que fueron erróneamente clasificadas con la etiqueta l, los *false negative (FN)* son la cantidad de instancias con la etiqueta l que fueron erróneamente clasificadas con cualquier otra etiqueta excepto l.

Accuracy (acc) indica la fracción de etiquetas clasificadas correctamente. Es la división de etiquetas clasificadas correctamente por el número total de etiquetas (5-1).

Precision (P) indica la fracción de instancias con la etiqueta l clasificados correctamente de todas las instancias clasificadas en esa clase. *Precision* mide la exactitud del clasificador(5-2).

Recall (R) indica la fracción de instancias con etiqueta l clasificados correctamente de toda la población de instancias que realmente pertenecen a esa clase. *Recall* mide la eficacia del clasificador (5-3).

F1-score (F1) es la media armónica entre *recision* y *recall* (5-4).

$$acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{5-1}$$

$$P = \frac{TP}{TP + FP} \tag{5-2}$$

$$R = \frac{TP}{TP + FN} \tag{5-3}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{5-4}$$

Para comparar los diferentes modelos de clasificación se definen las métricas anteriormente mencionadas. En este caso se evalúa como etiqueta l_1 para los reportes de accidentes de tránsito o en inglés *traffic accident* (TA) y la etiqueta l_2 para los no relacioandos a accidentes de tránsito o en inglés *non-traffic accident* (NTA).

5.5.2. Diseño del experimento

Comparación de preprocesamiento

Lematización vs. Stemming
Como se menciona en la sección **5.2** que describe el preprocesamiento aplicado para el modelo de clasificación de Tweets. Para evaluar el desempeño de aplicar normalización con Lematización se compara los resultados cuando se aplica Stemming.

Stopwords

De igual manera, en la sección **5.2** se menciona como influye en el desempeño del modelo al limpiar o no las stopwords del tweet. Para comparar los resultados se evalúa este aspecto con los métodos de *embedding* doc2vec, TFIDF y BERT.

Comparación de los métodos de clasificación

Se realizaron varios experimentos utilizando SVM, Naive Bayes, Random Forest y Neural Networks como modelo clasificador combinando la eficiencia de cada método de preprocesamiento y *embedding* presentado en la sección **5.4**.

SVM

Los parámetros utilizados para SVM, como **kernel, gamma y C,** se hallaron utilizando la técnica de *grid search* de la librería Sklearn.

Naive Bayes

Para este modelo no se realizó ninguna modificación a los parámetros, simplemente con el conjunto de datos de entrenamiento se calcula si un tweet está relacionado a un accidente. Se utilizó la clase *GaussianNB* de la librería *Sklearn*.

Random Forest

Para las pruebas con este modelo se hallaron los valores de hiperparametros de RF utilizando la técnica de *grid search* de la librería *Sklearn*, los parámetros seleccionados son el número de árboles, máxima profundidad de los árboles, el número mínimo de muestras necesarias para dividir un nodo interno y el número mínimo de muestras necesarias para estar en un nodo hoja.

Neural Network

Debido a que el algoritmo de NN es estocástico, para garantizar unos resultados reproducibles se realizaron 100 experimentos utilizando el conjunto de datos de 3804 tweets, particionando este conjunto en 70 % de tweets para entrenamiento y 30 % para prueba. Al final se calcula el promedio y la desviación estándar en los resultados obtenidos en cada experimento de *Accuracy, Recall, Precision y F1-score*.

La arquitectura seleccionada de la Neural Network es *Multilayer Perceptron* y se muestra en la figura **5-1**, tiene una capa oculta de *512 units* con una función de activación *ReLU* y una capa de salida con una sola neurona con una función de activación *sigmoide*, el entrenamiento se hizo con un *batch size* de 32, 25 *epochs*, *tasa de aprendizaje* de 1e-4 y el optimizador *Adam*.

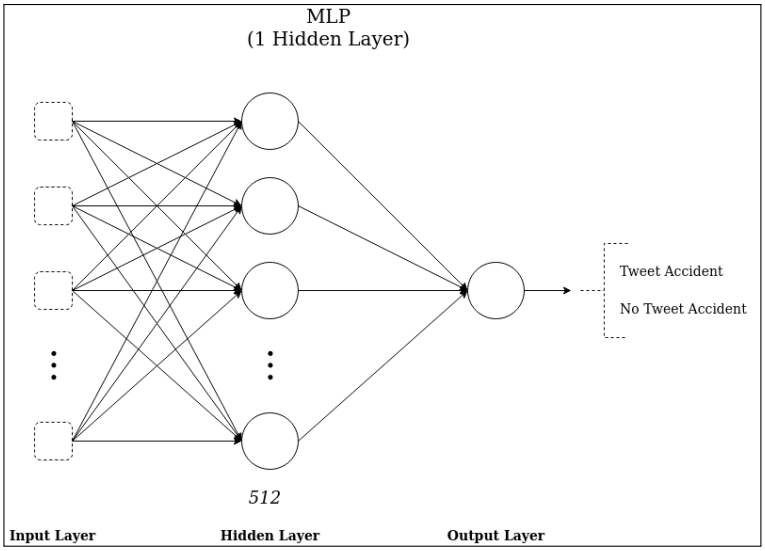

Figura 5-1: Multilayer Perceptron con 1 capa oculta. Dense(512).

5.5.3. Resultados

Como se especificó en el diseño del experimento, se compara el modelo propuesto de SVM con otros algoritmos de clasificación utilizados en la literatura en redes sociales. Además, se crean diferentes métodos combinando distintos procesos de normalización y *embedding*. Aunque se hizo el ejercicio de explorar diferentes procesos, en la tabla **5-2** se compara los resultados al aplicar normalización según lematización (lemma en inglés) o stemming (stem), la limpieza o no de stopwords y el modelo de clasificación SVM. Como se observa en la tabla **5-2**, en general se obtiene mejores resultados cuando no se elimina las stopwords, en este caso es recomendable que algoritmos como TFIDF se encargue del filtrado de las palabras con mayor ocurrencia, evitando eliminar preposiciones, como *"entre"* y *"con"*, que son utilizadas con regularidad para referir un reporte de accidente de tránsito. En caso de la normalización, tanto lematización y stemming obtienen resultados similares, por lo que se recomienda realizar una prueba con ambas y seleccionar la que mejor se adapte al problema de clasificación.

Los métodos de embedding del estado del arte son los algoritmos basados en Neural Networks, como por ejemplo los resultados de doc2vec en la tabla **5-2** supera en promedio con 0.5 % a TF IDF, en todas las métricas. En el caso de BERT entrenado con el dataset de Cañete et al. [50] sus resultados no superan a los de doc2vec y TFIDF, debido a que durante su entrenamiento el conjunto de Cañete et al. [50] no contemplaba tweets dentro del corpus, por lo que queda claro que para trabajar con tweets es mejor entrenar un método de *embedding* con publicaciones de redes sociales. En la tabla **5-2** se observa que el mejor resultado es nuestro enfoque propuesto que consiste en aplicar lematización y no eliminar las stopwords.

Por otro lado, al comparar los mejores resultados de desempeño de todos los modelos en la tabla **5-3**, los resultados entre SVM y NNs son similares pero con una diferencia en promedio de 0.57 % en todas las métricas, por lo que ambos algoritmos se pueden considerar al momento de trabajar con tweets, sin embargo nuestro enfoque propuesto con SVM obtuvo un desempeño superior con resultados de **96.8 % de accuracy y F1-score**. Los parámetros utilizados para SVM son *gamma de 0.2*, *C* igual a 7 y el *kernel RBF*. Finalmente, si bien por mucho tiempo el uso de Naive Bayes era común en la literatura para tareas de clasificación [34, 14], cuando se trabaja con textos informales en español como en Redes Sociales sus resultados estuvieron un 3 % por debajo de los otros, por lo que se puede considerar descartar para el tratamiento de reportes de accidentes. En la tabla **5-3** sólo se consideraron mostrar los mejores pasos de procesamiento de tweets por cada algoritmo de clasificación.

Tabla 5-2: Comparación de resultados de preprocesamiento, embedding y el modelo SVM.

Preprocesamiento	Embedding	Acc (%)	F1 (%)	R (%)	P (%)
Lemma	**Doc2vec***	**96.85**	**96.85**	**96.85**	**96.85**
	TF IDF	96.69	96.69	96.69	96.71
Lemma + Stopwords[a]	Doc2vec	96.63	96.63	96.63	96.65
	TF IDF	96.19	96.19	96.20	96.22
Stem	Doc2vec	96.14	96.14	96.14	96.16
	TF IDF	96.69	96.69	96.69	96.71
Stem + Stopwords[a]	Doc2vec	96.84	96.84	96.84	96.86
	TF IDF	96.03	96.03	96.04	96.05
Pasos de limpieza para BERT	BERT	95.37	95.37	95.37	95.40

[a] Indica que se han eliminado las stopwords

* Resultados del método seleccionado

Tabla 5-3: Comparación de los mejores resultados de los algoritmos de clasificación.

Método	Acc (%)	F1 (%)	R (%)	P (%)
doc2vec + lemma + SVM*	**96.85**	**96.85**	**96.85**	96.85
TF IDF + stem + NB	93.40	93.40	93.40	93.45
TF IDF + lemma + RF	96.08	96.08	96.08	96.11
TF IDF + stem + NN	96.22	96.12	95.17	**97.09**

* Resultados del método seleccionado

Comparación con la literatura relacionada

A continuación se selecciona y se compara nuestro enfoque con trabajos relacionados a detección de eventos de tránsito en redes sociales. Los trabajos se seleccionaron por tener

similitudes con la zona de estudio y tweets en español.

En comparación el modelo de clasificación propuesto para tweets de tránsito en la ciudad de Bogotá obtuvo resultados de **96.8 % en F1-score**. Lo cual es superior en trabajos similares como se muestra en la tabla **5-4**.

Tabla 5-4: Comparación entre trabajos relacionados y el **modelo de clasificación propuesta** en términos de idioma, región y la métrica F1.

Autor	**Idioma**	**Región**	**Clasificador**	**Clase**	**F1 (%)**
Método propuesto	Español	Bogotá, Colombia	Do2vec + SVM	Accidente	**96.85**
Arias et al. [16]	Español	Cuenca, Ecuador	BoW + SVM	Incidente	85.10
Caimmi et al. [33]	Español	Buenos Aires, Argentina	TF IDF + Ensemble SVM, SMO, NB	Incidente	91.44
Pereira et al. [29]	Portugués	Brasil	BoW + word2vec + SVM	Viajes de tránsito	85.48

6 Reconocimiento de entidades en Twitter

El siguiente capítulo explica los procesos involucrados para la selección del modelo de *sequence labeling* adecuado para los tweets en español. Primero se explica brevemente el conjunto de datos seleccionado para el entrenamiento y evaluación del modelo. El proceso de preprocesamiento para la normalización y limpieza de los tweets son explicados en la segunda sección, este proceso es distinto al modelo de clasificación. En la tercera sección se describe el modelo de *sequence labeling* seleccionado para la tarea de reconocimiento de entidades de ubicación y tiempo. Finalmente, se explica el experimento realizado donde se evaluó diferentes técnicas propuestas en la literatura para la tarea de reconocimiento de entidades a partir de texto.

6.1. Conjunto de datos

Para el entrenamiento del modelo de Reconocimiento de entidades se seleccionó una muestra de los tweets filtrados resultantes de la fase de clasificación anterior, se extrajeron **1340 tweets** dónde 800 son de usuarios "no oficiales", casi el 60 % de la muestra, estos tweets son reportes de usuarios sobre eventos de incidentes de tránsito ocurridos en Bogotá en el periodo de octubre del 2018 al julio del 2019, incluyendo otros tweets que contienen alguna referencias de ubicación como lo son reportes sobre el estado de infraestructura de la vía, igualmente se añadieron algunos tweets de los años 2016 y 2017. Aunque estas publicaciones no están relacionadas a accidentes propiamente se seleccionaron por contener información de ubicación, el propósito es entrenar un modelo que reconozca estas entidades, ya que previamente se construyó un clasificador de tweets relacionados a accidentes. Adicionalmente se realizó una partición del conjunto de datos, reservando 1072 tweets para entrenamiento del modelo y 268 para evaluación del mismo.

En los tweets extraídos hay publicaciones con muchas referencias vagas de direcciones como *"accidente en 10 con 15"* sin especificar calle o carrera o un punto de interés. También, hay tweets que hacen referencia a más de una ubicación en el mismo mensaje, mencionando dos eventos en diferente lugar. Algunos de estos casos se exponen en la tabla **4-4**.

Las etiquetas definidas son de ubicación (*Location*), que hace referencia al lugar del reporte; y tiempo (*Time*), que hace referencia a la hora o fecha ocurrida el incidente, generando así

5 etiquetas: *B-loc*, *I-loc*, *B-time*, *I-time* y *O*, la etiqueta 'O' hace referencia a 'Otros'. En la tabla **6-1** se describe la cantidad de entidades en ambos conjuntos de datos respectivamente.

Tabla 6-1: Cantidad de tokens por etiqueta y conjunto de datos.

	B-loc	**I-loc**	**B-time**	**I-time**	**O**
Trainset	1462	3768	131	112	20 242
Prueba	369	893	33	33	5038
Total	**1831**	**4661**	**164**	**145**	**25 280**

6.2. Preprocesamiento

Al igual que el modelo de clasificación, para un mejor desempeño del modelo de Reconocimiento de entidades (o NER por sus siglas en inglés *Named Entity Recognition*), debido a su naturaleza informal del lenguaje en redes sociales, cada tweet se aplica un proceso de limpieza y normalización. Los pasos de preprocesamiento para el modelo NER incluyen eliminar los caracteres especiales y urls. A diferencia del modelo de clasificación, los *@usernames* y *#hashtags* no se eliminan porque pueden contener información útil de ubicación y para esto se les aplica segmentación de palabras [51]. En concreto el preprocesamiento aplicado es el siguiente:

- Eliminar códigos ASCII, urls y saltos de línea innecesarios.
- Eliminar caracteres especiales y emoticones, exceptuando los signos de puntuación y tildes.
- **Eliminar letras o signos de puntuación repetidos consecutivamente.** Para el caso de las letras que se repitan más de 2 veces seguidas se eliminan las demás sobrantes, por ejemplo se reemplaza la expresión "gooooool" por "gol". Para el caso de los signos si estos se repite más de 3 veces seguidas.
- **Segmentación de palabras para #hashtags y @usernames.** Como sugiere Malmasi & Dras [43] estas expresiones pueden contener información de ubicación, sin embargo suelen ser una combinación de varias palabras. Para esta tarea se emplea el modelo de segmentación de palabras propuesto por Norvig [51].

6.2.1. Segmentación de palabras

En las redes sociales, los usuarios hacen uso de expresiones conocidas como *#hashtags*, estas sirven para remarcar un tema y pueda ser identificado o clasificado más adelante por otros usuarios, estas expresiones suelen ser una combinación y unión entre palabras y números.

Los *hashtags* (#) pueden contener información de ubicación [43], al igual que las menciones de usuario (@) que en ocasiones guarda el nombre de una localidad, barrio o punto de interés como edificios o parques. Por esta razón no se descarta este contenido y se emplea un modelo de segmentación de palabras propuesto por Norvig (2009) para separar las palabras.

- *Conjunto de datos.* Se construye un corpus con 400K unigramas usando el conjunto de datos en español de Cañete et al. [50] con cerca de tres mil millones de tokens y el conjunto de datos propio extraído desde Twitter con 76 millones de tokens.

- *Limpieza y normalización.* Se aplica el mismo preprocesamiento mencionado anteriormente, en este caso se elimina el acento de las palabras reemplazandolas sin tildes. Cada texto se divide en unigramas para construir un diccionario.

- *Modelo de lenguaje basado en Naive Bayes.* Con el conjunto de datos se construye un modelamiento de lenguaje que predice varios candidatos de segmentación de palabras y se selecciona la más probable.

Algunos resultados de segmentación de palabras se pueden ver en la tabla **6-2**. El segmentador reconoce nombres propios como de equipos de fútbol, puntos de interés y abreviaciones comunes mencionadas en redes sociales.

Tabla 6-2: Resultado del segmentador de palabras con tweets en español.

#hashtag o @username	Resultado
#puentearanda	puente aranda
#avenida68	avenida 68
@CorferiasBogota	corferias bogota

6.3. Modelo de etiquetamiento secuencial

Para la tarea de reconocimiento de entidades a partir de texto se emplea un modelo de *machine learning* basado en *sequence labeling*. El objetivo es entrenar un modelo que reconozca las entidades de ubicación y tiempo mencionadas en las publicaciones de los tweets. El modelo seleccionado en este caso esta disponible en **Spacy**[1] [52], una librería del estado del arte que se utiliza en Python para tareas de Procesamiento de Lenguaje Natural para

[1]Disponible en https://spacy.io/

varios idiomas. Spacy posee un modelo pre-entrenado de Deep Learning en español llamado es_core_news_lg[2], a este modelo aplicamos re-entrenamiento con nuestro conjunto de tweets etiquetados. Para el entrenamiento se usaron 500 iteraciones como sugiere la documentación de la librería.

Para evaluar el desempeño de Spacy con tweets en español se comparó con los siguientes modelos de *machine learning* usados en la literatura.

- ***Conditional Random Fields (CRF).*** Un modelo basado en características locales que son extraídas de las palabras y sus vecinas. Para esta implementación con tweets se tuvieron en cuenta algunas características basadas en los trabajos previos de Taufik et al. [53], Okur et al., [47] y García-Pablos et al [54]. Más adelante se menciona estas características.

- ***Bidirectional Long Short-Term Memory (BiLSTM).*** A diferencia de CRF no requiere de características locales. La arquitectura bidireccional consiste en la conexión de dos capas diferentes de redes neuronales recurrentes LSTM, una para analizar la secuencia de izquierda a derecha y la otra para analizar la secuencia de derecha a izquierda, de esta forma se tiene en cuenta el contexto de la palabra según las vecinas. Para la implementación de este modelo se usaron las recomendaciones de los trabajos de Peres et al. [39] y Aguilar et al. [40]).

- ***BiLSTM + CRF.*** En el trabajo de Aguilar et al. [40] con tweets sugiere agregar una capa de salida usando un modelo CRF, de esta forma se transfiere lo aprendido por el modelo BiLSTM hacia esta última capa.

6.4. Experimentos y Evaluación

Al igual que en el capítulo del método de clasificación (en el capítulo **5**), el objetivo es comparar la eficiencia de cada método de *sequence labeling*. Las métricas utilizadas para esta comparación son **Recall**, **Precision** y **F1 score**, como en muchos trabajos estos índices permiten seleccionar la mejor técnica. Las configuraciones de cada método a comparar se explican más adelante en la sección **6.4.2**.

6.4.1. Métricas de evaluación

Las métricas definidas para este experimento fueron *F1-score*, *Recall* y *Precision* para cada clase o etiqueta, como son varias clases del estándar *IOB*, al final se calcula una sola medida para el modelo usando la estrategia *weighted*, *F1-weighted*, *R-weighted* y *P-weighted*. La

[2]Autores en https://explosion.ai/

evaluación se hace sobre las etiquetas *B-loc*, *I-loc*, B-*time*, *I-time* y *O*. La formulación de estas métricas fueron definidas en el capítulo anterior (sección **5.5.1**).

6.4.2. Diseño del experimento

Para seleccionar un modelo automático para el reconocimiento de entidades en español sobre twitter se realiza una comparación del desempeño de cuatro métodos diferentes basados en el estado del arte: *CRF*, *BiLSTM*, *BiLSTM+CRF* y *Spacy*.

En el experimento se usaron los conjuntos de entrenamiento y prueba con 1072 y 268 tweets respectivamente. A excepción de la librería Spacy, para la configuración de los demás modelos mencionados se requiere que cada sentencia de tweet tenga la misma longitud de palabras o tokens. Para definir esta longitud se comparó todos los tweets del corpus seleccionado y se encontró una longitud máxima de 70 tokens como se muestra en la figura **6-1**, en este caso los tweets con longitud inferior se añade al final de la sentencia un token especial hasta alcanzar la longitud deseada.

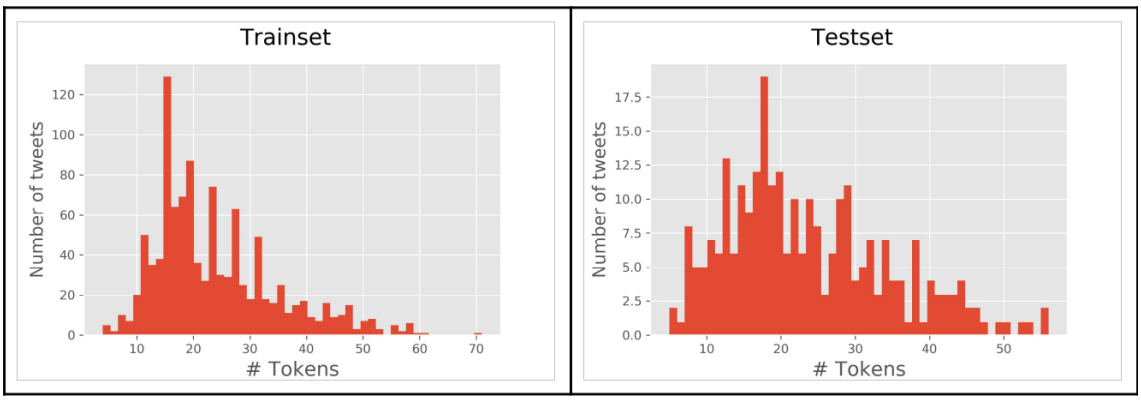

Figura 6-1: Distribución de número de tokens por número de tweets.

Los modelos de Neural Networks reciben como entrada de cada token o palabra una representación vectorial, en este experimento se implementó el modelo en español de *FastText* llamado *cc.es.300.bin*[3]. En el diseño del experimento se tiene en cuenta que las redes neuronales son algoritmos estocásticos, por lo tanto para los resultados presentados se realizó 25 experimentos y luego se calculó el promedio aritmético de cada métrica, para cada red se define un *batch size* de 16 y 40 épocas, este último usando la estrategia de parada temprana para quedarse con la mejor iteración. El número de experimentos se escogió en 25 por el tiempo de procesamiento computacional que puede gastar un número mayor.

[3]Disponible en https://fasttext.cc/docs/en/pretrained-vectors.html

CRF

Para la extracción de características con tweets se siguieron los pasos propuestos por Taufik et al. [53], Okur et al. [47] y García-Pablos et al. [54]. Se definen 13 funciones. En la figura **6-2** se muestra el diseño del CRF empleado.

- **Función Word.** Extracción de la palabra actual que se desea clasificar, como también sus palabras vecinas, dos palabras anteriores y dos siguientes.
- **Función Lower.** Transformación a minúsculas de la palabra actual, como de las palabras vecinas (antes y después).
- **Últimas tres letras.** Seleccionar el sufijo de las últimas 3 letras de la palabra actual.
- **Últimas dos letras.** Seleccionar el sufijo de las últimas 2 letras de la palabra actual.
- **Tamaño de caracteres** de la palabra actual.
- **Función de patrón.** Transformar la palabra a un respectivo patrón establecido como por ejemplo para las palabras Portal → Xxxxxx, Buses → Xxxxx, 85c → DDx.
- **Función isUpper.** Para determinar si todos las letras de la palabra actual están en mayúsculas, también para las dos palabras anteriores y las dos siguientes vecinas.
- **Función isTitle.** Para determinar si la primera letra de la palabra esta en mayúscula y las demás no, también se realiza para las dos palabras anteriores y las dos siguientes.
- **Función isDigit.** Determinar si todos los caracteres son dígitos o no, tanto de la palabra actual, como de las 2 siguientes y anteriores.
- **POS.** Función para determinar la etiqueta gramatical de la palabra actual (*Part-of-speech en inglés*), como también sus palabras vecinas, 2 palabras anteriores y 2 siguientes.
- **Extraer últimos dos caracteres del POS.** Función para extraer los últimos dos caracteres de la etiqueta gramatical de la palabra actual y las 2 anteriores y siguientes.
- **Etiquetas previas:** Predicción NER de 2 palabras previas usando CoreNLP Stanford.
- **Word Embeddings:** Representación vectorial de la palabra usando FastText, esta librería de Facebook permite hallar el *word embedding* de una palabra dada la conformación de las subpalabras que lo contiene, ideal para el tratamiento de palabras fuera del vocabulario (en inglés *OOV* o *Out-of-vocabulary*) comúnmente presentes en redes sociales.

Por último, el modelo CRF se entrena usando la librería de Python *sklearn_crfsuite* y ajustando al algoritmo **lbfgs**, y los parámetros **c1** y **c2** con 0.1, el parámetro de **all transitions = True** y **100 iteraciones**.

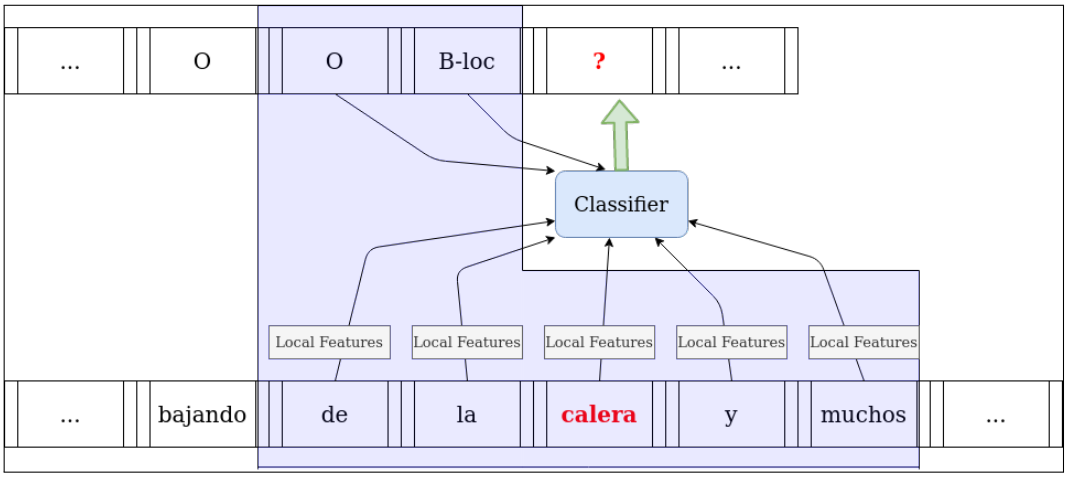

Figura 6-2: Etiquetamiento secuencial usando CRF y Local Features.

BiLSTM

Para los parámetros de la arquitectura BiLSTM consiste en dos redes *LSTM* cada una con *100 units* y *dropout 0.5*; la capa final es una *fully connected* con función de activación *softmax*, esta capa tiene como salida 5 unidades que representan cada etiqueta NER a predecir; finalmente, se utiliza el optimizador *ADAM* con una tasa de aprendizaje de 0.005, un *batch size* de 16 y 12 *epochs*. En la figura **6-3** se muestra un ejemplo de la arquitectura.

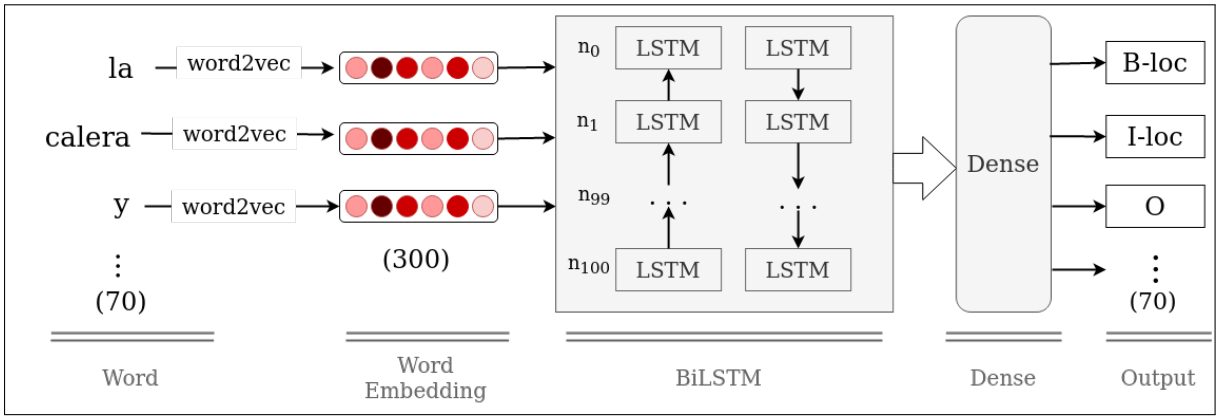

Figura 6-3: Etiquetamiento secuencial usando arquitectura BiLSTM.

BiLSTM + CRF

La implementación definida es similar al anterior modelo; se realiza *word embedding* con FastText; se aplica BiLSTM con 100 unidades y dropout de 0.5; la capa del *fully connected* en este caso usa la función de activación $ReLU$; la capa de salida es un capa CRF con 5 unidades que representa cada etiqueta NER resultante. Se utiliza el optimizador $ADAM$ con una tasa de aprendizaje inferior de 0.005, un *batch size* de 16 y 40 *epochs*. En la figura **6-4** se muestra la arquitectura empleada en este caso.

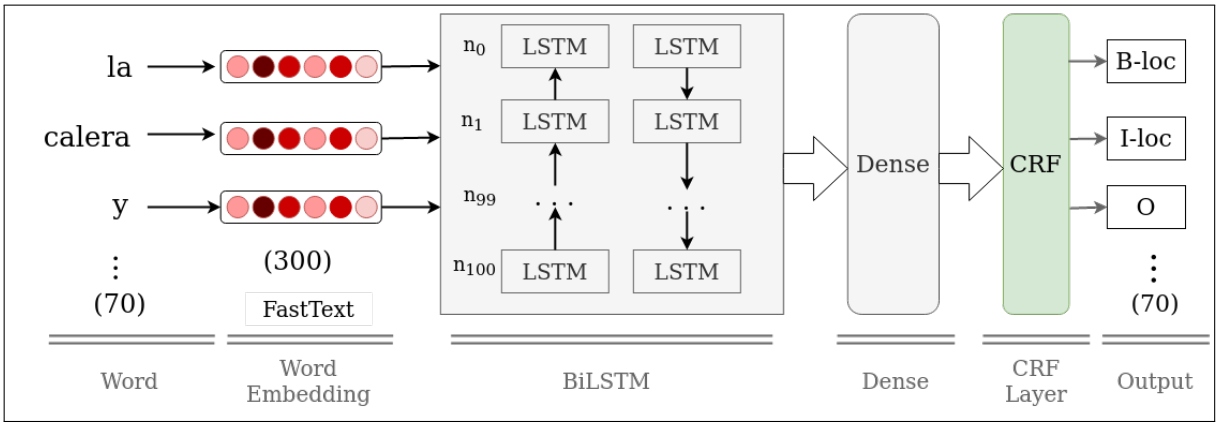

Figura 6-4: Etiquetamiento secuencial usando arquitectura BiLSTM + CRF.

Spacy

Esta librería ya tiene un modelo pre-entrenado disponible que reconoce varias entidades como PERSON, ORG, GPE, LOC, DATE, TIME, PERCENT[4], entre otros. Para nuestro enfoque aplicamos el re-entrenamiento del modelo en español es_core_news_lg usando nuestro propio dataset de tweets y únicamente para las etiquetas LOC (ubicación) y TIME (tiempo). En el re-entrenamiento se usaron 500 iteraciones como sugiere la documentación de la librería. Como entrada recibe cada secuencia de tweet con su respectiva etiqueta.

6.4.3. Resultados

Para evaluar el desempeño de los cuatro modelos presentados anteriormente se emplea el conjunto de datos de prueba con 268 tweets y distribución de etiquetas como se muestra en la tabla **6-1**. En el análisis de los resultados no se tendrá en cuenta la etiqueta Other, por lo que se quiere evaluar la detección de las etiquetas de ubicación y tiempo. En la tabla **6-3** se compara los resultados de las métricas para los modelos *CRF*, *BiLSTM*, *BiLSTM+CRF* y *SpaCy*, de esta manera el modelo que mejor desempeño obtiene es el modelo re-entrenado de **Spacy** con un puntaje **F1 de 91.97 %**, seguido de CRF con 91.66 %.

[4]Estas entidades son popularmente conocidas en inglés en la literatura de Procesamiento de Lenguaje Natural

Comenzar con un modelo entrenado previamente como el de SpaCy resulta atractivo para los desarrolladores e investigadores, debido a su fácil y rápida implementación. Además los resultados en la tabla **6-3** demuestran un mejor desempeño frente a los demás modelos. Finalmente, el detalle de los resultados del modelo re-entrenado de SpaCy se encuentra en la tabla **6-4**, este obtiene mejores resultados con las etiquetas que tienen mayor cantidad de tokens presentes en el corpus, como lo son las etiquetas de *location* o ubicación.

Tabla 6-3: Desempeño de los modelos propuestos (Mejor puntaje F1).

Modelo	**F1 (%)**	**Recall (%)**	**Precision (%)**
CRF	91.66	89.76	**93.80**
BiLSTM	90.88	89.77	92.25
BiLSTM + CRF	84.22	81.55	88.50
SpaCy	**91.97**	**91.97**	92.17

Tabla 6-4: Desempeño de Spacy re-entrenado

Mejor Modelo	**# Entidades**	**F1 (%)**	**Recall (%)**	**Precision (%)**
B-loc	369	88.70	87.82	89.60
I-loc	893	94.83	95.82	93.86
B-time	33	62.22	51.85	77.78
I-time	33	74.51	65.52	86.36
Overall	**1328**	**91.97**	**91.97**	**92.17**

Comparación con la literatura relacionada

El enfoque propuesto extrae las etiquetas de ubicaciones y tiempo obteniendo resultados de **91.97 % en F1**. Lo cual es superior en trabajos similares como se muestra en la tabla **6-5**.

Tabla 6-5: Comparación entre trabajos relacionados y el **modelo de Named Entity Recognition (NER) propuesto** en términos de idioma, región y la score F1.

Autor	**Idioma**	**Región**	**NER**	**Clases**	**F1 (%)**
Método propuesto	Español	Bogotá, Colombia	Spacy re-entrenado con tweets	Loc, Time	**91.97**
Arias et al. [16]	Español	Cuenca, Ecuador	Rule-Based	Loc	80.61
Gelernter & Zhang [44]	Español	Tweets en español	Rule-Based + NER Software + Translate	Topónimo	86.10
Sagcan & Karagoz [45]	Turco	Tweets en turco	Rule-Based + CRF	Loc	62.00

7 Geolocalización y Análisis de resultados

El siguiente capítulo valida y analiza los datos obtenidos en Twitter por el enfoque propuesto en este trabajo sobre accidentes de tránsito. Primero se describen brevemente los datos a trabajar. Segundo, se realiza un preprocesamiento de los datos recolectados en los tweets para extraer información de las coordenadas geográficas. En la tercera se sección realiza un análisis de los datos obtenidos de accidentes en Twitter para la ciudad de Bogotá. Finalmente, en la última sección, se comparan los datos en Twitter con los reportes oficiales de accidentes de la fuente de la secretaría de movilidad.

7.1. Datos

7.1.1. Reportes en Twitter

Los datos usados para este trabajo fueron recolectados usando el mecanismo definido en el capítulo **4** y los filtros de recolección de la sección 4.2 para el periodo de diez meses entre octubre del 2018 a julio del 2019. Se recolectaron en total 4.973.900 tweets. En la tabla **4-3** se muestra la cantidad extraída según el filtro de recolección. En la tabla **4-4** se muestran algunos tweets extraídos.

7.1.2. Conjunto de datos oficial

Se extrajeron de la página oficial de Datos Abiertos Bogotá[1] los accidentes reportados en la base de datos de la secretaría de movilidad para el mismo periodo. Se recolectaron en total 25.299 accidentes de tránsito reportados en la región urbana. De cada registro se trabaja con las coordenadas geográficas y la fecha/hora del accidente.

7.2. Preprocesamiento

Una vez extraídas las ubicaciones con el modelo de *sequence labeling* (capítulo **6**), estas pasan a un *geocoding* que devuelve las geocoordenadas. Con esta información se trabaja

[1]Página web de datos abiertos Bogotá, datos de accidentalidad disponible en https://datosabiertos.bogota.gov.co/dataset/historico-siniestros-bogota-d-c

posteriormente para validar la integridad de los datos en Twitter mediante un análisis de cobertura. Sin embargo, al tratarse de lenguaje informal se recomienda hacer unos pasos previos, como remover los tweets con información insuficiente e imprecisa de ubicación y aplicar un proceso de estandarización de direcciones. Este proceso se divide en cuatro tareas como se describe a continuación.

A *Remover tweets con entidades de ubicación imprecisas.* Algunos tweets hacen referencia a más de una ubicación, incluso referencias vagas a la ubicación del incidente sin especificar mayor detalle, lo anterior dificulta la tarea de geolocalizar las coordenadas del reporte. Por esta razón se decide aplicar las siguientes dos reglas para filtrar estos tweets.

- Descartar los tweets con menos de 4 palabras en la ubicación detectada. Por ejemplo, *"Accidente en Calle 22 trafico bogota"*, la entidad *"Calle 22"* no brinda información suficiente.
- Descartar los tweets con más de 4 entidades de ubicación reconocidas. Tweets que mencionan diferentes lugares o direcciones en el reporte. Por ejemplo *"- Calle 34 trancada al Oriente desde la calle 26 hasta la carrera 13 -choque en la Glorieta de la Avenida Primero de Mayo con carrera 68"*, es un tweet con dos reportes de eventos de tránsito en ubicaciones diferentes.

B *Estandarización de direcciones.* Las direcciones o ubicaciones detectadas en los tweets carecen de un uso formal del lenguaje, lo que limita la eficacia de los *geocoders*, quienes calculan las coordenadas del sitio en mención. Algunos inconvenientes son el uso de abreviaturas, diferentes toponimias para un mismo lugar y falta de precisión en la dirección o uso incompleto de los nombres de lugares. Estos problemas motivaron a emplear un método para enriquecer y estandarizar las ubicaciones detectadas en los mensajes de los tweets, para esto se usa la librería *Libpostal*[2]. Esta librería permite adaptar los diccionarios de palabras, toponimias, abreviaturas y entre otras modificaciones que realizamos para ajustarnos a las particularidades de la ciudad de Bogotá. Finalmente, *Libpostal* puede transformar una ubicación reconocida como *"cl 72 * cra 76"* → *"BOGOTA AVENIDA CALLE 72 CARRERA 76"*. Las modificaciones realizadas a los diccionarios de la librería *Libpostal* son los siguientes.

- Se agregaron a los diccionarios algunos topónimos particulares de la ciudad de Bogotá, como avenidas y localidades y otras reglas de abreviaciones usadas por los usuarios de Twitter.
- Eliminar palabras como *"con"*, *"por"* e *"y"*, preposiciones y conectores que dificultan la detección en los métodos de *geocoding* disponibles.

[2]Repositorio librería Libpostal en Github

- Para mejorar la granularidad de los *geocoders* se agrega al principio de toda frase la palabra "BOGOTA" (sin tilde) a todas las direcciones o ubicaciones.
- Se transforma toda la frase en mayúsculas y se eliminan las tildes para mayor precision con los *geocoding*.

C *Filtrar reportes duplicados y otros incidentes*. Para reducir el costo de usar *geocoders* de pago como *Google Maps API*, que cobran por consulta, se realiza un paso previo para eliminar los reportes duplicados y el ruido presente en Twitter. El objetivo es extraer información adicional en Twitter y no duplicar la existente. En este punto se eliminan tweets con ubicaciones similares, extraídas en el paso anterior, y que ocurren en la misma dirección en una ventana de tiempo de 1 hora (antes y después). Además para filtrar tweets de otros eventos o incidentes se realiza una última selección por coincidencia de palabras claves relacionadas a accidentes, de esta forma se eliminan reportes no relevantes.

D *Geocoders*. Una vez estandarizado las direcciones un *geocoder* devuelve una coordenada geográfica. Usamos una aplicación llamada *batch geocode*[3] que combina los recursos disponibles de *Google*, *OpenStreetMap* y *Geonames*. Se traen hasta 4 resultados por recurso y la herramienta de geocodificación asignará automáticamente una coordenada si todos los puntos caen dentro de una área de influencia.

7.3. Análisis del procesamiento de tweets de accidentes en Bogotá

Una vez entrenado los modelos de nuestro enfoque se aplican todos los pasos definidos en el proceso de la figura **3-1** del capítulo **3** del método propuesto. Como se mencionó anteriormente se recolectaron 4.973.900 tweets de octubre del 2018 a julio del 2019. En la tabla **7-1** se muestra la cantidad de tweets clasificados como TA (Accidente) y NTA (No accidente) según cada filtro definido en la sección **4.2**.

Como se muestra en la tabla **7-1** la cantidad de tweets filtrados por el clasificador varía según el filtro de recolección empleado, por ejemplo el 0.15 % (5832) de los tweets recolectados por el filtro de *Stream Bogotá* (geotagged o tweets geoetiquetados) son seleccionados como accidentes de tránsito, de estos tweets la mayoría son publicaciones del usuario *@BogotaTransito* y cerca de 2000 publicaciones son realizadas por otros usuarios, que en algunos casos son usuarios de periodistas o noticieros. Por otro lado, usando la herramienta *Twitter Search API* se puede extraer mayor cantidad de tweets relacionados a accidentes, con el filtro de

[3]Disponible en GISforHealth's Github repository

Search Token se clasificó 22.5 % de las publicaciones y del filtro *Search Timeline User* se clasificó 50 % de los tweets, este último conjunto es extraído directamente de usuarios oficiales seleccionados previamente. Hay que considerar que estos filtros pueden extraer publicaciones en común, proceso que se filtra en las siguientes etapas de detección de ubicaciones.

El método de clasificación entrenado tiene una precisión de 96.8 %, observando de cerca los tweets filtrados en el proceso de clasificación automática se logra ver que el modelo separa eficazmente los tweets que son de otros tipos de incidentes, como vehículos varados, protestas, tráfico lento, entre otros. El éxito de esto se debe a que estas características fueron incluidas al principio, durante las fases de recolección y entrenamiento. En la tabla **7-2** se muestra algunos ejemplos donde estas publicaciones comparten un contenido similar y no están relacionados con accidentes. Por otro lado, el 3.2 % de error de clasificación se mitiga en la siguiente fase de Reconocimiento de entidades, donde se descarta los tweets que no poseen referencia de ubicaciones.

La fase de Reconocimiento de entidades consiste en extraer las ubicaciones que se mencionan en los tweets de reportes de accidentes. Se tomaron los tweets recolectados por los filtros de recolección y se extrajeron las entidades de ubicaciones de los tweets, la cantidad de tweets filtrados que poseen ubicación se muestra en la tabla **7-1**. El siguiente paso es juntar las publicaciones extraídas por los cuatro filtros y eliminar los tweets repetidos entre sí, el resultado es un conjunto de datos con 84 262 tweets de accidentes con entidades reconocidas. Como se mencionó en el preprocesamiento para la geolocalización (sección **7.2**), algunos de estos tweets hacen referencia al mismo accidente pero reportados con un desfase de tiempo provocando ruido para los análisis. Por ejemplo los usuarios oficiales de la Secretaría de Movilidad (*@BogotaTransito*) y la empresa de transporte Transmilenio (*@rutasitp*) reportan el mismo incidente con más de 2 horas de diferencia, por esta razón se decide descartar los tweets de *@rutasitp*. Igualmente se descartan los tweets de accidentes duplicados que hacen referencia a la misma ubicación en una ventana de tiempo menor a 1 hora, y también se eliminan los tweets de otros tipos de incidentes o eventos de tránsito que el clasificador no logra filtrar, estos se eliminan mediante coincidencia de palabras claves como *"incidente"*, *"accidente"*, *"choque"* y otras relevantes a accidentes. De esta forma se logró eliminar spam tweets de bots como del usuario *@nikolai68843464* y algunos retweets innecesarios. En este proceso se logró extraer un conjunto final de 43 235 tweets únicos que posteriormente pasan por la herramienta de *geocoder* llamada *batch geocode*, combinando los resultados de *Google maps*, *Openstreetmap* y *Geonames*. Finalmente el resultado es la generación de las coordenadas de **26 362 tweets**, este último conjunto generado se considera el definitivo y es utilizado para el análisis de cobertura más adelante. En la tabla **7-3** se muestra la cantidad de tweets de accidentes con ubicaciones por mes antes y después del *geocoder*.

Tabla 7-1: Resultado cantidad de tweets clasificados como TA/NTA según el filtro de recolección; y cantidad de tweets extraídos con información de ubicación.

Filtro de recolección	**No accidentes**		**Accidentes**		**Tweets con ubicación**
	#*	%**	#	%	
Stream Bogotá	4 021 481	99.85	5 832	0.15	4 463
Stream Follow/Timeline User	487 545	84.8	87 271	15.2	80 277
Search Token	210 183	77.5	60 970	22.5	54 765
Search Timeline User	50 507	50.2	50 111	49.8	47 398

* Indica la cantidad de tweets clasificados del filtro de recolección.

** Indica el porcentaje de tweets clasificados del filtro de recolección.

Tabla 7-2: Análisis de Precisión correcta del método de clasificación propuesto.

Tweet	**Predicción**
Semáforos de la Carrera 24 con Calle 9 en amarillo intermitente, Tanto por la calle como por la carrera con riesgo de incidente vehicular	No accidente
Inicia marcha SENA Kra 30A esta hora inicia desplazamiento de estudiantes del SENA sede carrera 30 con calle 14 por toda la Av. NQS hacia el norte, utilizando calzada mixta con afectación de calzada de TransMilenio.	No accidente
Hueco causa accidentalidad Cra. 68c #10-16 sur, Bogotá A esta hora nuestras unidades brindan apoyo en la Av primero de mayo por 24, donde se presenta un choque entre un vehículo particular y una motocicleta.	Accidente Accidente

Tabla 7-3: Cantidad de Tweets de accidentes de tránsito con ubicación y coordenadas por mes.

Mes	# Tweets TA	# Tweets geocoordenados
Octubre	3682	2194
Noviembre	4072	2358
Diciembre	3634	2114
Enero	4316	2692
Febrero	4127	2534
Marzo	4029	2500
Abril	3697	2241
Mayo	4123	2545
Junio	5281	3287
Julio	6274	3897
Total	**43 235**	**26 362**

7.4. Análisis de cobertura de accidentes de tránsito entre Twitter y la fuente oficial

Se utilizan las coordenadas obtenidas por el método de *geoparsing* para realizar una comparación de patrones de accidentes entre Twitter y la fuente oficial. Los dos conjuntos están en el rango de octubre 2018 a julio 2019, la información oficial de accidentes contiene **25 299 registros** y el conjunto de tweets **26 362 registros**, ambos contienen las coordenadas latitud/longitud y fecha/hora del incidente.

7.4.1. Emparejamiento con el registro oficial de accidentes

Los accidentes de tránsito reportados en Twitter pueden contener información adicional, para responder esto se cuantifica el porcentaje de accidentes de los registros oficiales que son cubiertos por Twitter. Debido a que los accidentes no son reportados inmediatamente, se debe establecer un radio de tiempo y distancia cercano para hacer coincidir los reportes entre Twitter y la fuente oficial. Zhang et al. [15] y Gu et al. [14] proponen considerar aspectos de la zona de estudio como el tamaño, la congestión vehicular y tiempo en atención a emergencias. Teniendo en cuenta lo anterior para emparejar los registros entre Twitter y la fuente oficial se define **1 kilómetro como radio de distancia y 2 horas de diferencia**. La razón es que Bogotá fue la ciudad con mayor congestión vehicular en el mundo para el año 2019[4], lo que dificulta la atención de accidentes e impide conocer su hora exacta, con 1 o 2 horas de diferencia. Además, las ubicaciones descritas por los usuarios en Twitter en ocasiones son

[4]Según cifras de congestión vehicular para el año 2019 del Tráfico Global de INRIX

imprecisas y tienen un margen de error, también influye la precisión del geocoder que en las pruebas realizadas con el 1 % de los tweets obtuvo un 78.7 % de predicción de coordenadas correctas, en este caso aquellas con menos de 1 km de diferencia entre predicción y valor real.

En la tabla **7-4** se presentan los resultados del emparejamiento usando 1 km de diferencia y 2 horas de discrepancia. De los 26 362 tweets se emparejaron 8619, es decir un 32.7 % de estos con la fuente oficial, con una distancia promedio de 436 metros y 47 minutos de diferencia, además 2896 (34 %) de los tweets son publicados antes que el registro oficial y 5723 (66 %) publicados después, algunos ejemplos de estos tweets se muestran en la tabla **7-5**. Para seguir comparando los resultados, también se creó una muestra extra con 1431 tweets que fueron publicados por usuarios individuales (excluyendo BogotaTransito), siguiendo las recomendaciones de Gu et al. [14] para medir la proporción de información adicional, para este caso se emparejaron 455 publicaciones o 31.8 % con un promedio de diferencia de 460 metros y 51 minutos, en este caso 164 (36 %) se publicaron antes que el registro oficial y 291 (64 %) después.

Los tweets no emparejados se pueden considerar como reportes adicionales y corresponden a 17 743 tweets, de los cuales 976 son de usuarios individuales, como redes de apoyo (@RedapBogota) y reporteros. Examinando manualmente algunos registros y con apoyo de la tabla **7-6** se discute lo siguiente:

- En algunos casos existe 1 o 2 horas de diferencia entre la fuente oficial y Twitter, la demora de la atención de los accidentes y la congestión vial puede provocar desinformación de la hora exacta del siniestro.

- Encontramos diferencias de más de 1 km entre coordenada real y predecida, una causa es la imprecisión de la dirección del accidente descrito en Twitter y la disposición del *geocoding* utilizado en Bogotá.

- No todos los tweets de incidentes de *BogotaTransito* están publicados en la fuente oficial. Estos tweets se pueden considerar como información adicional confiable al ser publicaciones de un usuario oficial de tránsito, y se puede sumar junto con los usuarios individuales.

Tabla 7-4: Cantidad de accidentes reportados según la fuente de datos.

Fuente	**Número de reportes**	
Datos oficiales	25 299	
	Todos los reportes	**Sin BogotaTransito**
Datos de Twitter	26 362	1431
Twitter (emparejados con los datos oficiales con 1 kilometro y 2 horas)	8619	455
# Reportes "adicionales" en Twitter	17 743	976

Tabla 7-5: Tweets emparejados con la fuente oficial de accidentes.

Tweet*	**Tiempo de diferencia**
Calle 55 sur carrera 19B Choque con herido ya hay Ambulancia se necesita@TransitoPolicia @BogotaTransito @SectorMovilidad @TransitoBta	23 seg. después
Aparatoso accidente en la Av. Córdoba con calle 127 sentido sur - norte. Trancón, se recomienda usar vías alternas. @gusgomez1701 @CaracolRadio @SectorMovilidad	58 min. después
Incidente vial entre dos particulares en la Calle 19 con Carrera 34, sentido occidente - oriente. Unidad de @TransitoBta asignada.	1 h 37 min. después de BogotaTransito
@BogotaTransito buenos días, necesitamos su ayuda en la carrera 7 con calle 163, choque de sitp con taxi.	2 min. antes
@TransMilenio @PoliciaBogota Peatón atropellado en troncal calle 80, estación Av 68, Se requiere ambulancia urgente!!!!!	15 min. antes

* Se eliminaron algunos caracteres especiales y emoticones.

Tabla 7-6: Tweets no emparejados con la fuente oficial de accidentes.

Tweet*	**Dificultades**
@TransitoBta accidente en sentido S - en la 27 sur con 10 monumental trancon @Citytv @NoticiasCaracol @PoliciaColombia	Vaga descripción de ubicación
Incidente vial entre particular y un ciclista en la Calle 65A con Carrera 112, sentido occidente - oriente. Unidad de @TransitoBta y asignadas.	Predicción de coordenadas $>$ 1 km de diferencia; tweet publicado por *BogotaTransito*
Accidente de 2 vehículos calle 161 con carrera 7, sin heridos solo latas, generan afectación del tráfico.	No emparejado
Incidente vial entre particular y ciclista, en la Autonorte con calle 170, sentido sur-norte.Unidad de @TransitoBta y asignada.	No emparejado; tweet publicado por *BogotaTransito*

* Se eliminaron algunos caracteres especiales y emoticones.

7.4.2. Análisis del patrón de accidentes en tiempo y espacio

Ahora se examinan en el tiempo los accidentes entre Twitter y registros oficiales. En la figura **7-1** se muestra el porcentaje de reportes según la hora del día para los datos oficiales, tweets de *BogotaTransito* y usuarios individuales. Los reportes de accidentes se ajustan con el horario de mayor actividad de tránsito, entre las 5 a.m. y las 8 p.m. También las tres fuentes de datos coinciden con un pico a las 7 a.m. Los tweets de los usuarios individuales se ve condicionado por su jornada laboral, esto explica que solo hay dos picos para este tipo de tweets que coincide con las horas en que los usuarios la pasan en el tráfico, alrededor de las 7 a.m. y las 6 p.m., para las demás horas del día baja la actividad en redes sociales. Sorprende que la actividad de reportes entre los tweets de *BogotaTransito* y los datos oficiales no coincidan en las horas de la noche, a pesar que son datos reportados por la misma entidad oficial. En este caso, coincidimos con Gu et al. [14] al afirmar que una ventaja de la detección de accidentes en Twitter es tener mayor cobertura durante el día, sin embargo no es tan efectiva en la noche y la madrugada como sucede con los datos oficiales. En cuanto a la cantidad de reportes de accidentes según el día de la semana en la figura **7-2**, los datos también se ajustan a la jornada laboral habitual con mayor actividad en los días hábiles y disminución de tweets los fines de semana.

Por último, se realiza una comparación del patrón espacial de accidentes entre ambas fuentes con un estilo de mapa de calor, pero en su lugar usando el método de *Kernel Density Estimation* (KDE) para la intensidad de accidentes por metro cuadrado. En nuestro análisis para calcular el ancho de banda del método KDE se usa el factor de *Scott* y el *kernel gaussiano*.

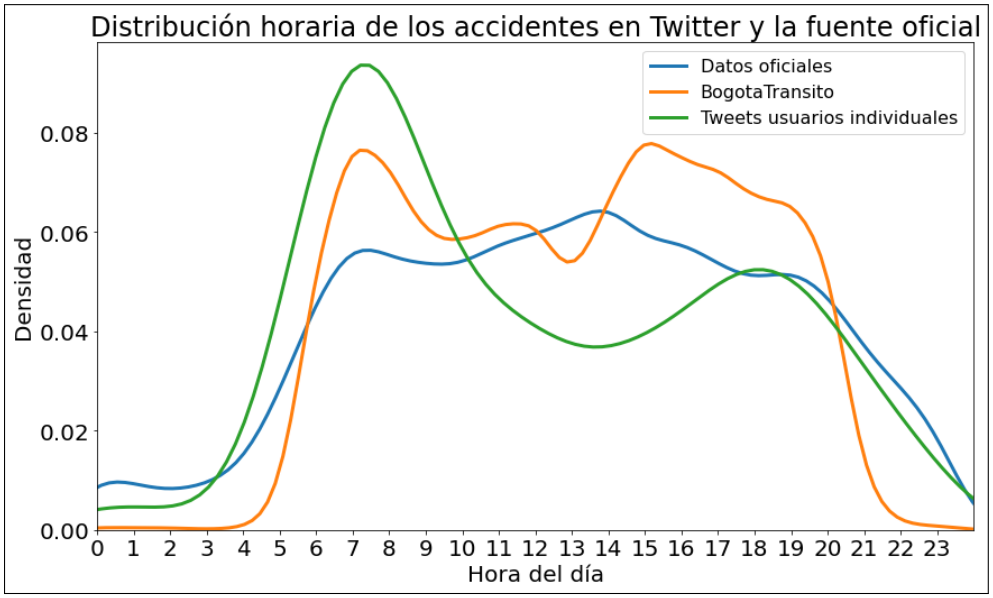

Figura 7-1: Distribución de accidentes reportados según la hora del día por Twitter y el Registro Oficial.

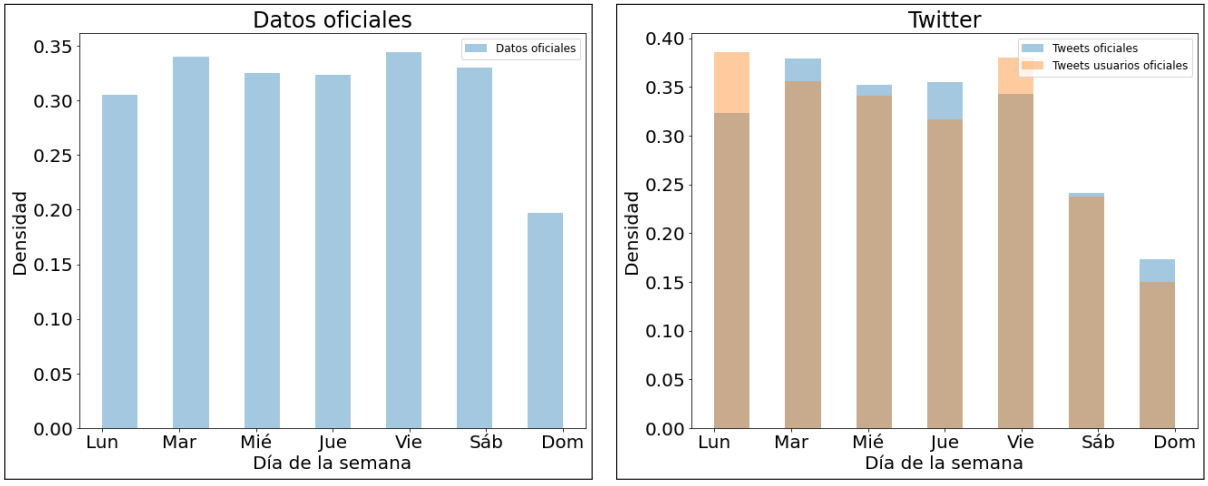

Figura 7-2: Distribución de accidentes reportados según el día de la semana por Twitter y Registro Oficial.

En la figura **7-3** se muestra la distribución de los accidentes durante los meses de octubre del 2018 a julio del 2019, según los datos de Twitter y oficiales. Los accidentes en Twitter se distribuyen espacialmente similar a los registros en la fuente oficial de datos, manteniendo mayor tasa de accidentalidad sobre la zona centro de Bogotá y disminuyendo cuando se aleja de este. Entre ambas fuentes algunas regiones coinciden con la concentración de accidentalidad, como son las localidades de Chapinero y los límites entre los Mártires y Puente Aranda, estas zonas se caracterizan por contener el centro principal del comercio y la zona industrial de Bogotá con mayor actividad de tráfico vehicular. En cuanto a la región occidente ambas fuentes de datos coinciden cercanamente con las zonas de accidentalidad en las localidades de Engativá, Fontibón y Kenedy, aunque la actividad de las publicaciones en Twitter baja un poco en estas zonas. Espacialmente los datos de Twitter y la fuente oficial coinciden en su mayoría con la densidad de los accidentes, manteniendo mayor ventaja para Twitter sobre la región comercial e industrial de Bogotá.

Finalmente, los resultados de las comparaciones anteriores, en tiempo y espacio, brinda mayor confianza y credibilidad sobre la efectividad de los accidentes reportados en Twitter como información adicional, en este sentido coincidimos con Zhang et al. [15] sobre el uso de estos datos como fuentes complementarias no sustituibles a los métodos de detección existentes, un caso validado para el contexto de la ciudad de Bogotá.

(a) Accidentes en Twitter (b) Fuente de datos oficial

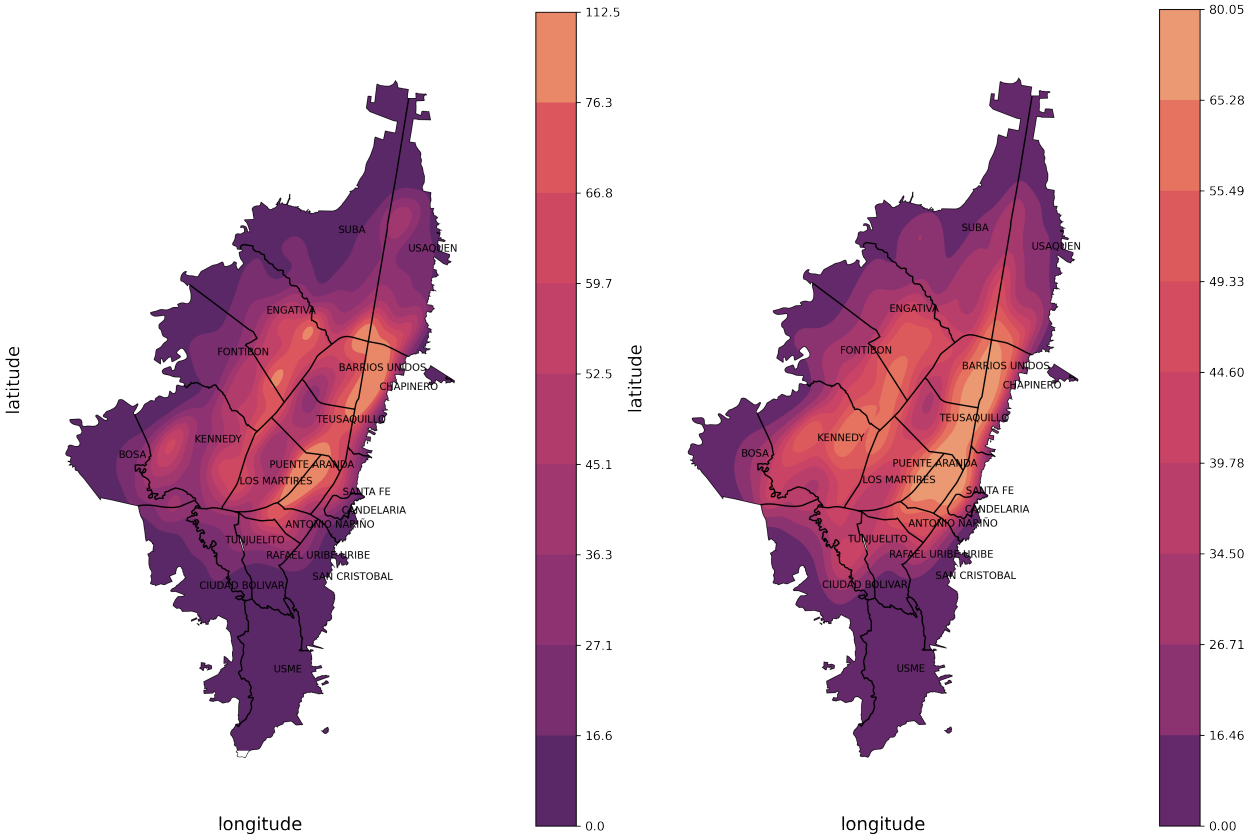

Figura 7-3: Incidencia geográfica de los accidentes reportados durante en el periodo de Octubre del 2018 a Julio del 2019 según los datos en Twitter (a) y la fuente oficial (b).

8 Conclusión y Trabajos Futuros

En este trabajo se presenta un método para la extracción de accidentes de tránsito en redes sociales, seleccionamos Twitter pero puede ser implementado en otros. La metodología se divide en cuatro fases, primero la recolección de tweets, segundo la clasificación de accidentes, tercero la detección de ubicaciones a partir del contenido del tweet y finalmente la generación de coordenadas del reporte. También realizamos una validación de los tweets de accidentes comparándolos con datos oficiales de la secretaría de movilidad de la ciudad de Bogotá.

En la clasificación de tweets se combina diferentes métodos de *word embedding* y clasificación para evaluar el desempeño de modelo propuesto en la separación de accidentes de tránsito sobre otros tipos de incidentes, para lograr este objetivo se incluyó este tipo de publicaciones desde la construcción colaborativa del conjunto de datos. La implementación propuesta de *doc2vec* y *SVM* en el clasificador logra una precisión de 96.8 % comparable con el estado del arte. Se recomienda no remover las stopwords en el preprocesamiento ya que los métodos de *word embedding* trabajan mejor con la mayoría del contenido del tweet.

En la extracción de ubicaciones a partir del texto se comparó algunas arquitecturas para el reconocimiento de entidades nombradas, sin embargo el modelo seleccionado es un modelo re-entrenado de *SpaCy* con tweets en español, el cual alcanzó un desempeño superior con 91.9 % en la métrica de *F1-score*. Una contribución clave en este trabajo es la fase de geolocalización, al tratarse de publicaciones de redes sociales se debe superar el uso del lenguaje informal y abreviaciones, para lograrlo estandarizamos las direcciones como paso previo para mejorar el desempeño de los *geocoders*. Las expresiones como *hashtags* también pueden incluir información de ubicación [43] y para esto realizamos una normalización de segmentación de palabras. También eliminamos reportes repetidos y algunos retweets, pues muchas veces son publicados con 5 horas de diferencia generando falsas alarmas.

Los accidentes de tránsito reportados en Twitter contienen información adicional, para responder esto se emparejó los datos de Twitter y la fuente oficial de tránsito Bogotá. Se encontró que cerca del 33 % de los tweets están reportados por la fuente oficial. Los tweets que no se emparejaron son reportes adicionales y la mayoría corresponden a usuarios oficiales como tránsito, redes de apoyo a emergencias y reporteros, estos tweets se consideran confiables al ser publicados por cuentas influyentes. Examinando manualmente los accidentes se encontró que en algunos casos existe 1 o 2 horas de diferencia entre el reporte de la fuente

oficial y Twitter, la demora en la atención y la congestión vial puede provocar desinformación de la hora exacta.

También realizamos una comparación en tiempo y espacio de los patrones de accidentes entre los tweets y los registros oficiales. En este caso, coincidimos con Gu et al. [14] al afirmar que una ventaja de la detección de accidentes en Twitter es tener mayor cobertura durante el día, sin embargo no es tan efectiva en la noche y la madrugada como sucede con los datos oficiales. También los datos de Twitter y la fuente oficial coinciden en su mayoría con la distribución espacial de los accidentes, manteniendo mayor ventaja para Twitter sobre la región comercial e industrial de Bogotá. Finalmente, los resultados brindan mayor confianza y credibilidad sobre la efectividad de los accidentes reportados en Twitter como información adicional, en este sentido coincidimos con Zhang et al. [15] sobre el uso de estos datos como fuentes complementarias no sustituibles a los métodos de detección existente, un caso validado para el contexto de la ciudad de Bogotá.

Encontramos varias oportunidades en la geolocalización que abren la posibilidad de una línea de trabajos futuros, como es resolver la imprecisión o referencia vaga a las ubicaciones del accidente en Twitter y los errores de ortografía comunes. La disponibilidad de las herramientas de *geocoding* como *Google Maps* y *OpenStreetMap* no es suficiente para ciudades no estadounidenses, actualmente estas herramientas pueden provocar diferencias de más de 1 km entre coordenada real y predecidas, un trabajo futuro es analizar a profundidad este tipo de error y diseñar un *geocoding* en específico para la ciudad en estudio con un mejor desempeño. Otra causa que distorsiona la predicción de los *geocoders* son los tweets que en la misma publicación reportan más de dos accidentes con diferentes direcciones, para este caso se debe crear un tratamiento especial para este tipo de publicaciones. Otros avances podría ser la construcción de sistemas de monitoreo en tiempo real de accidentes para la ciudad de Bogotá que incluya datos de redes sociales. Otras investigaciones futuras deberían orientarse en la integración o fusión de los tweets con otras fuentes para la predicción o monitoreo de accidentes de tránsito en la ciudad.

Bibliografía

[1] G. Cookson and B. Pishue, *INRIX Global Traffic Scorecard*. No. February, 2018.

[2] "Víctimas fallecidas y lesionadas valoradas por inmlcf. nacionales. agencia nacional de seguridad vial," 2017.

[3] S. Wang, L. He, L. Stenneth, P. S. Yu, and Z. Li, "Citywide traffic congestion estimation with social media," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '15*, pp. 1–10, 2015.

[4] X. Fan, B. He, C. Wang, J. Li, M. Cheng, H. Huang, and X. Liu, "Big Data Analytics and Visualization with Spatio-Temporal Correlations for Traffic Accidents," in *15th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2015)*, vol. 9531, (Zhangjiajie, China), pp. 255–268, 2015.

[5] M. B. Subaweh and E. P. Wibowo, "Implementation of Pixel Based Adaptive Segmenter method for tracking and counting vehicles in visual surveillance," in *2016 International Conference on Informatics and Computing, ICIC 2016*, no. Icic, pp. 1–5, 2016.

[6] L. Li, J. Zhang, Y. Zheng, and B. Ran, "Real-Time Traffic Incident Detection with Classification Methods," in *Green Intelligent Transportation Systems, Lecture Notes in Electrical Engineering*, vol. 419, pp. 777–788, 2018.

[7] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 3687–3696, 2017.

[8] N. Krausz, T. Lovas, and Á. Barsi, "Radio frequency identification in supporting traffic safety," *Periodica Polytechnica Civil Engineering*, vol. 61, no. 4, pp. 727–731, 2017.

[9] M. Chaturvedi and S. Srivastava, "Edge-level real-time traffic estimation with limited infrastructure," *2015 IEEE 82nd Vehicular Technology Conference, VTC Fall 2015 - Proceedings*, 2016.

[10] W. Zuo, C. Guo, J. Liu, X. Peng, and M. Yang, "A police and insurance joint management system based on high precision BDS/GPS positioning," *Sensors (Switzerland)*, vol. 18, no. 1, 2018.

[11] J. Aslam, S. Lim, X. Pan, and D. Rus, "City-scale traffic estimation from a roving sensor network," in *SenSys 2012 - Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems*, pp. 141–154, Association for Computing Machinery, 2012.

[12] S. Wang, X. Zhang, J. Cao, L. He, L. Stenneth, P. S. Yu, Z. Li, and Z. Huang, "Computing Urban Traffic Congestions by Incorporating Sparse GPS Probe Data and Social Media Data," *ACM Transactions on Information Systems*, vol. 35, no. 4, pp. 1–30, 2017.

[13] T. Kuflik, E. Minkov, S. Nocera, S. Grant-Muller, A. Gal-Tzur, and I. Shoor, "Automating a framework to extract and analyse transport related social media content: The potential and the challenges," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 275–291, 2017.

[14] Y. Gu, Z. S. Qian, and F. Chen, "From Twitter to detector: Real-time traffic incident detection using social media data," *Transportation Research Part C: Emerging Technologies*, vol. 67, pp. 321–342, 2016.

[15] Z. Zhang, Q. He, J. Gao, and M. Ni, "A deep learning approach for detecting traffic accidents from social media data," *Transportation Research Part C: Emerging Technologies*, vol. 86, no. November 2017, pp. 580–596, 2018.

[16] B. Arias, G. Orellana, M. Orellana, and M.-I. Acosta, *A Text Mining Approach to Discover Real-Time Transit Events from Twitter*, vol. 884. Springer International Publishing, 2019.

[17] K. Ikeda, T. Sakaki, F. Toriumi, and S. Kurihara, "An examination of a novel information diffusion model: Considering of twitter user and Twitter system features," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10002 LNAI, pp. 180–191, 2016.

[18] D. A. Kurniawan, S. Wibirama, and N. A. Setiawan, "Real-time traffic classification with Twitter data mining," in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 1–5, 2016.

[19] A. Salas, P. Georgakis, C. Nwagboso, A. Ammari, and I. Petalas, "Traffic Event Detection Framework Using Social Media," in *IEEE International Conference on Smart Grid and Smart Cities*, no. July, p. 5, 2017.

[20] A. Salas, P. Georgakis, and Y. Petalas, "Incident Detection Using Data from Social Media," *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 751–755, 2017.

[21] K.-H. Chao and P.-Y. Chen, "An Intelligent Traffic Flow Control System Based on Radio Frequency Identification and Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, p. 694545, 2014.

[22] R. Ke, Z. Li, S. Kim, J. Ash, Z. Cui, and Y. Wang, "Real-Time Bidirectional Traffic Flow Parameter Estimation from Aerial Videos," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 890–901, 2017.

[23] H.-c. Kwak and S. Kho, "Predicting crash risk and identifying crash precursors on Korean expressways using loop detector data," *Accident Analysis and Prevention*, vol. 88, pp. 9–19, 2016.

[24] H. M. Sherif, M. Shedid, and S. A. Senbel, "Real time traffic accident detection system using wireless sensor network," in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 59–64, 2014.

[25] N. Ya, A. E. Azhar, A. L. Yusof, S. S. Sarnin, and D. M. Ali, "Real Time Wireless Accident Tracker using Mobile Phone," no. October, pp. 2–3, 2017.

[26] Z. Li, S. K. Cha, C. Wan, B. Cui, N. Zhang, and J. Xu, "Detecting Anomaly in Traffic Flow from Road Similarity Analysis," in *17th International Conference, WAIM 2016, Proceedings, Part II* (B. Cui, N. Zhang, J. Xu, X. Lian, and D. Liu, eds.), vol. 9659 of *Lecture Notes in Computer Science*, (Cham), pp. V–VI, Springer International Publishing, 2016.

[27] A. B. Nikolaev, Y. S. Sapego, A. M. Ivakhnenko, E. Mel, and V. Y. Stroganov, "Analysis of the Incident Detection Technologies and Algorithms in Intelligent Transport Systems," vol. 12, no. 15, pp. 4765–4774, 2017.

[28] I. Moncada, "Análisis espacio-temporal de los accidentes de tránsito en Bogotá utilizando patrones puntuales," tech. rep., Universidad Nacional de Colombia, Bogotá, 2018.

[29] J. Pereira, A. Pasquali, P. Saleiro, and R. Rossetti, "Transportation in Social Media: An Automatic Classifier for Travel-Related Tweets," in *16th Portuguese Conference on Artificial Intelligence, EPIA 2013*, vol. 8154, pp. 355–366, 2017.

[30] Y. G. Petalas, A. Ammari, P. Georgakis, and C. Nwagboso, "A Big Data Architecture for Traffic Forecasting Using Multi-Source Information," in *ALGOCLOUD 2016, Springer International Publishing AG*, vol. 10230, pp. 65–83, 2017.

[31] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[32] Q. Chen, X. Song, H. Yamada, and R. Shibasaki, "Learning Deep Representation from Big and Heterogeneous Data for Traffic Accident Inference," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, pp. 338–344, 2016.

[33] B. Caimmi, S. Vallejos, L. Berdun, Ï. Soria, A. Amandi, and M. Campo, "Detección de incidentes de tránsito en Twitter," in *2016 IEEE Biennial Congress of Argentina, ARGENCON 2016*, pp. 1–6, 2016.

[34] H. Nguyen, W. Liu, P. Rivera, and F. Chen, "TrafficWatch: Real-Time Traffic Incident Detection and Monitoring Using Social Media Hoang," in *PAKDD: Pacific-Asia Conference on Knowledge Discovery and Data Mining* (J. Bailey, L. Khan, T. Washio, G. Dobbie, J. Z. Huang, and R. Wang, eds.), vol. 9651 of *Lecture Notes in Computer Science*, (Cham), pp. 540–551, Springer International Publishing, 2016.

[35] A. Schulz, P. Ristoski, and H. Paulheim, "I see a car crash: Real-time detection of small scale incidents in microblogs," in *The Semantic Web: ESWC 2013 Satellite Events. ESWC 2013. Lecture Notes in Computer Science*, vol. 7955 LNCS, pp. 22–33, 2013.

[36] C. Gutiérrez, P. Figueiras, P. Oliveira, R. Costa, and R. Jardim-goncalves, "An Approach for Detecting Traffic Events Using Social Media," in *Emerging Trends and Advanced Technologies for Computational Intelligence*, vol. 647, ch. An Approac, 2016.

[37] P. Anantharam, P. Barnaghi, K. Thirunarayan, and A. Sheth, "Extracting City Traffic Events from Social Streams," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 4, pp. 1–27, 2015.

[38] Y. Chen, Y. Lv, X. Wang, and F. Y. Wang, "A convolutional neural network for traffic information sensing from social media text," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2018-March, pp. 1–6, 2017.

[39] R. Peres, D. Esteves, and G. Maheshwari, "Bidirectional LSTM with a Context Input Window for Named Entity Recognition in Tweets," in *In Proceedings ofK-CAP 2017: Knowledge Capture Conference (K- CAP 2017).*, pp. 1–4, Association for Computing Machinery, 2017.

[40] G. Aguilar, A. P. López Monroy, F. González, and T. Solorio, "Modeling Noisiness to Recognize Named Entities using Multitask Neural Networks on Social Media," in *Proceedings of NAACL-HLT 2018 Association for Computational Linguistics*, vol. 1, pp. 1401–1412, Association for Computational Linguistics, 2018.

[41] J. Gelernter and S. Balaji, "An algorithm for local geoparsing of microtext," *GeoInformatica*, vol. 17, no. 4, pp. 635–667, 2013.

[42] A. Ritter, S. Clark, Mausam, and O. Etzioni, "Named entity recognition in tweets: an experimental study," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1524–1534, Association for Computational Linguistics, 2011.

[43] S. Malmasi and M. Dras, "Location Mention Detection in Tweets and Microblogs," in *PACLING 2015, CCIS*, pp. 123–134, Oxford University Press, may 2016.

[44] J. Gelernter and W. Zhang, "Cross-lingual geo-parsing for non-structured data," in *Proceedings of the 7th Workshop on Geographic Information Retrieval*, pp. 64–71, 2013.

[45] M. Sagcan and P. Karagoz, "Toponym Recognition in Social Media for Estimating the Location of Events," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 33–39, 2015.

[46] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, 2014.

[47] E. Okur, H. Demir, and A. Özgür, "Named entity recognition on twitter for Turkish using semi-supervised learning with word embeddings," *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, pp. 549–555, 2016.

[48] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, pp. 1–12, 2013.

[49] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, no. Mlm, 2019.

[50] J. Cañete, G. Chaperon, R. Fuentes, and J. Pérez, "Spanish Pre-Trained BERT Model and Evaluation Data," *PML4DC at ICLR 2020*, pp. 1–10, 2020.

[51] P. Norvig, "Natural Language Corpus Data," in *Beautiful Data: The Stories Behind Elegant Data Solutions*, pp. 219–242, 0'Reilly, 2009.

[52] M. Honnibal and M. Johnson, "An improved non-monotonic transition system for dependency parsing," *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, no. September, pp. 1373–1378, 2015.

[53] N. Taufik, A. F. Wicaksono, and M. Adriani, "Named entity recognition on Indonesian microblog messages," *Proceedings of the 2016 International Conference on Asian Language Processing, IALP 2016*, pp. 358–361, 2017.

[54] A. García-Pablos, N. Perez, and M. Cuadros, "Sensitive Data Detection and Classification in Spanish Clinical Text: Experiments with BERT," 2019.