Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

HERRAMIENTAS NER FRENTE A ALGORITMOS ADAPTADOS A LA EXTRACCIÓN DE ENTIDADES PARA IDENTIFICAR ETIQUETAS DE LOCALIZACIÓN EN TEXTOS CORTOS EN ESPAÑOL

Trabajo de graduación previo a la obtención del título de Ingeniero de Sistemas y Telemática

Autor:

Ruth Catalina Fárez Crespo

Director:

Ing. Marcos Patricio Orellana Cordero

Cuenca – Ecuador

2020

DEDICATORIA

A Dios, por permitirme alcanzar esta meta, brindándome salud y sabiduría.

A mis padres, por brindarme su apoyo en todo momento, por la paciencia y el gran amor que me han brindado.

A mis hermanos, por su apoyo y cariño incondicional durante esta etapa, por estar siempre conmigo dándome consejos.

AGRADECIMIENTO

A mi familia, por darme la oportunidad de haberme formado y apoyado en este proceso.

A mi director de trabajo de titulación, por haberme guiado y apoyado en el desarrollo de este trabajo.

A la Universidad del Azuay, por haberme dado la oportunidad de ser parte de ella y enriquecerme en conocimiento.

ÍNDICE

Índice de contenido

Índice de tablas e ilustraciones

Tablas

Ilustraciones

RESUMEN

El reconocimiento de entidades nombradas (NER) es un área importante en el campo de la extracción de información de textos. En los últimos años han surgido varias herramientas NER, las mismas que emplean diferentes metodologías, identifican diferentes entidades y trabajan en varios idiomas. Estas circunstancias hacen difícil al usuario determinar la herramienta adecuada. Ante esta necesidad, se propuso evaluar herramientas NER y un algoritmo adaptado a la extracción de entidades (Algoritmo de Levenshtein). Esta evaluación se enfocó en la identificación de etiquetas de localizaciones en textos en idioma español. A partir de los resultados obtenidos, se obtuvo el método adecuado y las librerías o algoritmos vinculados.

Palabras claves: reconocimiento de entidades nombradas, extracción de información, Twitter, Levenshtein, herramientas NER.

ABSTRACT

Named-entity recognition (NER) is an important area in the field of text information extraction. In recent years, several NER tools have emerged with different methodologies, identifying different entities and working in several languages. These circumstances make it difficult for the user to determine the right tool. Given this need, it was proposed to evaluate NER tools and an algorithm adapted to entity extraction (Levenshtein algorithm). This evaluation focused on the identification of location tags in Spanish-language texts. From the results obtained, the appropriate method and the linked libraries or algorithms were obtained.

Keywords: named-entity recognition, information extraction, Twitter, Levenshtein, NER tools.

UNIVERSIDAD DEL AZUAY
Dpto. Idiomas

Translated by
Ing. Paúl Arpi

1. INTRODUCCIÓN

En la actualidad existe cuantiosa información escrita en lenguaje natural proveniente de sitios web y redes sociales (Getoor, 2003). La información que se obtiene de estos medios es muy valiosa y representa un gran potencial para el conocimiento (Derczynski et al., 2015). No obstante, esta información es transmitida en textos no estructurados y resulta difícil analizarla (Eken & Tantug, 2010). Este inconveniente dio lugar a la necesidad de contar con herramientas que permitan analizar información escrita en lenguaje natural, lo que condujo a la aparición de técnicas de extracción de información (IE) (Piskorski & Yangarber, 2013).

El proceso de IE consiste en estructurar datos escritos en lenguaje natural a través de la extracción automática de frases nominales (personas, organizaciones y referencias geográficas), atributos y relaciones (Maedche, Neumann, & Staab, 2012; Sarawagi, 2007). Su propósito es analizar textos escritos en lenguaje natural y extraer información de interés para el usuario, los datos resultantes pueden ser analizados con mayor facilidad, inclusive pueden ser almacenados en bases de datos (Cuningham, 2006).

La extracción de información es utilizada en varios campos, principalmente en el ámbito empresarial, científico, web y personal (Sarawagi, 2007). Sobre la base de estos campos de estudio, se han encontrado trabajos que han realizado seguimiento de noticias para el rastreo automático de eventos específicos (Sarawagi, 2007), recopilación de información para localizar brotes de enfermedades (Grishman, Huttunen, & Yangarber, 2002), análisis del estado de ánimo de los clientes en conversaciones telefónicas de soporte con la empresa (Jansche & Abney, 2001), extracción de etiquetas de direcciones para identificar direcciones de clientes que corresponden a un mismo lugar (Borkar, Deshmukh, & Sarawagi, 2001), entre otros.

Si considera el contexto de accidentes de tránsito, por ejemplo, para la extracción de información sobre eventos de accidentes en redes sociales, se busca identificar a los principales actores involucrados, la ubicación del accidente y los afectados. La recolección de información de localizaciones es importante, debido a que ayuda a comprender el impacto de un siniestro. Por esta razón, investigar las posibilidades de aplicar técnicas de extracción de entidades de localización en *microblogs* es

significativo para las entidades de emergencia, ya que permite ubicar zonas en donde se requiere asistencia.

La extracción de información está dividida en cinco tareas, cada una de ellas se enfoca en un tema específico, tal es el caso del sistema de reconocimiento de entidades con nombre (NER) (Cuningham, 2006; Piskorski & Yangarber, 2013). Este sistema cuenta con herramientas que clasifican diferentes tipos de entidades nombradas (Persona, Ubicación, Organización, entre otros.) aplicando distintas metodologías en distintos dominios e idiomas (Jiang, Banchs, & Li, 2016; Srihari & Li, 2000). Entre las herramientas más aplicadas de NER están SpaCy, NLTK, Stanford CoreNLP, también existen otras herramientas como Freelyng, Gensim, OpenNLP, AllenNLP, entre otras (Domino Data Lab, 2019).

Otra técnica que también se vincula a la extracción de información son los algoritmos adaptados a la extracción de entidades, es el caso del algoritmo de distancia de *Levenshtein*. Este algoritmo mide la distancia de las palabras y calcula los costos requeridos para cambiar una palabra mediante inserción o sustitución (Beijering, Gooskens, & Heeringa, 2008).

Varias investigaciones se han desarrollado con respecto a la técnica de reconocimiento de entidades nombradas (NER). En ciertos casos se han evaluado herramientas de NER para determinar la herramienta con mejor desempeño en la identificación de etiquetas de tipo persona, ubicación y organización para textos en idioma ingles (Jiang et al., 2016); también se han evaluado herramientas de NER para determinar si retienen conceptos teóricos e identifican de manera automática las reseñas de hoteles verdaderas y falsas (Kleinberg, Mozes, Arntz, & Verschuere, 2018). Por otro lado, se han evaluado herramientas de NER con el fin de conocer el rendimiento para identificar etiquetas en un conjunto de datos en idioma portugués (Pires, Devezas, & Nunes, 2017).

También se han desarrollado investigaciones que aplican algoritmos adaptados a la extracción de entidades. Existe un estudio para descubrir la ubicación de eventos de tránsito en tiempo real, mediante la aplicación del algoritmo de distancia de *levenshtein* (Zhañay et al., 2019). También se propone nuevos métodos para el reconocimiento de entidades y aplican el algoritmo de distancia de *levenshtein* para corregir errores de escritura (Cheng, Zheng, & Li, 2013; Eken & Tantug, 2010). En Su

et al. (2008) se investiga la aplicación del algoritmo de distancia de *levenshtein* para la detección de plagio en textos académicos

Contar con estos métodos hace posible extraer información útil de un dominio de interés, para establecer la información de manera estructurada y realizar un mejor análisis de la información (Cardie, 1997). Sin embargo, al existir varios métodos que trabajan con la extracción de información resulta difícil determinar la técnica con mejores características y que se adapte a las necesidades particulares como es el caso de la extracción de etiquetas de localización en idioma español.

De acuerdo a los trabajos antes mencionados, se puede evidenciar que existen varias investigaciones que aplican la técnica NER y pocas investigaciones que apliquen el algoritmo de distancia de *levenshtein* como un método aceptado para realizar extracción de identidades. Por lo general se utiliza el algoritmo de distancia de *levenshtein* para corregir errores de escritura. Por esta razón, es importante evaluar las técnicas disponibles para la extracción de información debido a que permitirá identificar el método apropiado para realizar la extracción de etiquetas. Otro aspecto importante que se puede destacar es la falta de investigaciones que apliquen herramientas de NER, y el algoritmo de distancia de *levenshtein* a textos en idioma español, muchos de los trabajos revisados se enfocan en idioma inglés, portugués, entre otros.

Ninguna de las investigaciones previas propuso realizar una evaluación de las distintas técnicas disponibles para la extracción de entidades, como es la técnica NER y los algoritmos adaptados a la extracción de entidades (algoritmo de *levenshtein*). No obstante, tampoco se han realizado evaluaciones a textos en idioma español, aún menos se propone la extracción de entidades nombradas centradas en una sola categoría como pueden ser etiquetas de personas, localizaciones, organizaciones.

Frente a este vacío de conocimiento, se ve la necesidad de evaluar las técnicas de extracción de entidades y centrarse en una sola categoría debido a que se puede determinar la mejor técnica en una categoría específica, de igual manera es importante aplicar en textos de idioma español por la escasa cantidad de investigaciones existentes. Por lo tanto, el objetivo de este trabajo será realizar una evaluación experimental en la extracción de etiquetas de localización en textos cortos en idioma español utilizando herramientas NER y algoritmo de *levenshtein*. Con la finalidad de

identificar la técnica ideal para procesos de extracción de direcciones con base en parámetros de precisión, *recall* y *f-measure*.

2. FUNDAMENTOS TEÓRICOS

2.1 Extracción de información

La extracción de información analiza textos escritos en lenguaje natural con la finalidad de extraer entidades, atributos y relaciones que son de interés para el usuario, logrando que se ignore información no relevante para el dominio. La información resultante puede ser almacenada y analizada con mayor facilidad (Cuningham, 2006; Grishman et al., 2002; Sarawagi, 2007; Srihari & Li, 2000).

IE, se encuentra dividida en cinco tareas:

1. **Reconocimiento de entidades con nombre (NER). -** identifica etiquetas de personas, lugares, organizaciones, etc.
2. **Resolución de conferencia (CO). -** identifica relaciones existentes entre las entidades de los textos.
3. **Construcción de elementos de plantilla (TE). -** permite asociar información descriptiva entre las entidades existentes.
4. **Construcción de relación de plantilla (TR). -** obtiene relaciones útiles de entre las entidades encontradas.
5. **Producción de plantilla de escenarios (ST). -** permite vincular las entidades de (TE) con las relaciones de (TR).

Para el propósito de esta investigación, se enfocará en la tarea de reconocimiento de entidades nombradas (NER), debido a que es un sistema simple, confiable y puede alcanzar la medida de precisión de hasta 0.95 (Cuningham, 2006).

2.2 Métodos para extracción de información.

2.2.1 Reconocimiento de entidades nombradas (NER)

Es una tarea clave en el campo de extracción de información, este sistema consiste en extraer información de importancia de un texto escrito en lenguaje natural, para posteriormente clasificar en categorías, tales como etiquetas de personas, localizaciones, organizaciones, entre otras. Esta herramienta posee un algoritmo de

aprendizaje automático lo que hace que sea efectivo en términos de calidad al momento de reconocer entidades. (Derczynski et al., 2015; Inkpen, Liu, Farzindar, Kazemi, & Ghazi, 2015; Rodriquez, Bryant, Blanke, & Luszczynska, 2012; Santos & Guimarães, 2015; Tkachenko & Simanovsky, 2012).

Ejemplo.

Entrada.

> anoche, un choque entre un vehículo y una moto dejó una persona herida en la Turuhuayco y avenida de las Américas

Salida.

> anoche un choque entre un vehículo y una moto dejó una persona herida en la **[turuhuayco]** (Localización) y **[avenida de las américas]** (Localización)

2.2.2 Algoritmo de Levenshtein

El algoritmo de *levenshtein* o distancia de *levenshtein*, es una métrica que mide la diferencia entre dos palabras o identifica el grado de similitud basándose en una comparación. Esta métrica se define por la cantidad de ediciones requeridas para trasformar una palabra en otra. La métrica utilizada se denomina *threshold*, esta medida varía en una escala de valores de 0 a 1.

Este algoritmo considera tres operaciones fundamentales inserción, eliminación y sustitución (Beijering, Gooskens, & Heeringa, 2008; Hicham, Abdallah, & Mostapha, 2012; Putera Utama Siahaan et al., 2018; Sri Nandhini & Sheeba, 2015).

Ejemplo.

Sustitución:

saka → sala Sustitución de ‘k’ por ‘l’.

Inserción:

comedors → comedores Inserción de ‘e’ entre ‘o’ y ‘s’.

Eliminación:

corrredor → corredor Eliminación de ‘r’ entre ‘r’ y ‘e’.

En la Tabla 1, se puede observar cómo influye el valor de *threshold*, por ejemplo, si se establece un valor de 0.6 las palabras confrontadas tienen un nivel de similitud bajo, con un *threshold* de 0.7, se sigue manteniendo este nivel. Sin embargo, con un *threshold* de 0.8 y 0.9, el nivel de similitud va aumentando, ya que las palabras se diferencian entre sí por un solo carácter.

Tabla 1: Ejemplo escala de valores para threshold.

Palabra 1	**Palabra 2**	**Resultado comparación**	**Threshold**
gerona	personas	0.62	0.6
francia	tranvia	0.71	0.7
espana	españa	0.83	0.8
estevez de toral	esteves de toral	0.93	0.9

2.3 Herramientas de reconocimiento de entidades nombradas

Actualmente existen varias herramientas NER, sin embargo, es importante trabajar con herramientas que posean una comunidad de soporte activa y una documentación adecuada, esto con la finalidad de que pueda dar soporte y se agreguen características de mejora (Domino Data Lab, 2019). Para este estudio se realizó una revisión teórica de las distintas herramientas NER que están disponibles, se efectuó una evaluación teórica y se eligió la herramienta apropiada para realizar la experimentación. Se utilizaron las siguientes condiciones para la selección.

1. Herramientas de código abierto.
2. Uso ilimitado.
3. Reconocimiento de etiquetas de localización.
4. Soporte para el idioma español.

Las principales herramientas que cuentan con estas singularidades y las condiciones antes mencionadas son SpaCy, NLTK y Freelyng. No obstante, también existen otras herramientas como OpenNLP, Gensim, Stanford CoreNLP, entre otras (Domino Data Lab, 2019; Marrero, Sánchez-Cuadrado, Morato, & Andreadakis,

2009). Sin embargo, estas herramientas no han sido consideradas para este estudio debido a que no cumplen con las condiciones antes establecidas.

2.3.1 SpaCy

Es una herramienta de código abierto, su principal enfoque es la extracción de datos, también tiene otros ámbitos como el análisis de sentimiento y resumen de textos. Esta herramienta usualmente es utilizada para construir sistemas de extracción de información, comprensión de lenguaje y pre-procesamiento de texto. SpaCy es conocida por un rendimiento adecuado al realizar el reconocimiento de entidades nombradas, la ventaja más notable es su sistema de reconocimiento de entidades, pues estadísticamente es un sistema extremadamente rápido (SpaCy, 2019; TheAppSolutions, 2018).

SpaCy cuenta con el analizador sintáctico más rápido, también cuenta con modelos de redes neuronales para realizar etiquetado de un texto sin formato. Para que pueda etiquetar es necesario aplicar tokenización, esto permite que la herramienta realice un mejor análisis y aplique un modelo estadístico, con la finalidad de predecir etiquetas que sean similares al contexto que se está analizando. SpaCy realiza reconocimiento de entidades nombradas para diferentes idiomas e identifica varios tipos de etiquetas, ya sean numéricas y de nombre. Esto depende del corpus con el que se ha entrenado, sin embargo las principales etiquetas que identifica son persona, organización, localización y misceláneo (SpaCy, 2019).

a. Corpus

Para el entrenamiento de sus módulos, SpaCy utiliza un corpus de textos de Wikipedia y Google News, también cuenta con diccionarios propios. Por otro lado, esta herramienta busca almacenar información en un vocabulario común, el cual se denomina "*Vocab*". Este término es compartido por casi todos sus módulos con la finalidad de ahorrar memoria. Otro aspecto importante de SpaCy, es que codifica las cadenas en valores *hash*, el objetivo de realizar esta codificación es para trabajar internamente con los valores generados (SpaCy, 2019).

b. Modelos en idioma español

Para realizar el reconocimiento de entidades, SpaCy aplica modelos neuronales, estos modelos han sido creados específicamente para esta herramienta, al ser implementados desde cero brinda a la herramienta un mejor desempeño en cuanto a velocidad y precisión. Los modelos con los que cuenta SpaCy para español son

es_core_news_sm y *es_core_news_md.* Estos modelos son multi-tarea y se han capacitado con el corpus AnCora y WikiNER. Su funcionamiento se basa en asignar vectores de token específicos de contexto, etiquetado POS y análisis de dependencia. Las etiquetas que soportan estos modelos son PER (persona), LOC (localización), ORG (organización) y MISC (misceláneo) (SpaCy, 2019).

Al ser modelos pre-entrenados, se ha podido evidenciar que el modelo *es_core_news_sm,* en el reconocimiento de entidades nombradas puede alcanzar una precisión de 89.59%, *recall* de 89.31% y *f-measure* de 89.45%, por otro lado, el modelo *es_core_news_md,* puede alcanzar una precisión de 90.02%, *recall* equivalente a 89.70% y un *f-measure* de 89.86% (SpaCy, 2019).

c. Técnicas de pre-procesamiento

Esta librería incorpora tres técnicas de pre-procesamiento, las cuales son:

1. ***Tokenizer*:** es un algoritmo de tokenización que ofrece un equilibrio entre el rendimiento, definición y alineación con la cadena original, a continuación, se detalla el algoritmo general de tokenizer (SpaCy, 2019).
    1. Iterar subcadenas que estén separadas por espacios en blanco.
    2. Verificar si cuenta con una regla definida para la subcadena.
    3. Si no cuenta con una regla definida, consuma un prefijo y vuelva al paso dos.
    4. Si no encuentra un prefijo, intente consumir un sufijo y vuelva al paso dos.
    5. Si no encuentra un prefijo o sufijo, intente buscar un caso especial.
    6. Intente buscar una coincidencia simbólica.
    7. Busque infijos y divida la subcadena en cada infijo que encuentre.
    8. Si no pude consumir más la cadena, trátela como un token único.
2. ***Lemmatizer*:** asigna las formas básicas de las palabras y almacena los datos en un diccionario, en el cual cada dato asigna un término a su lema, para encontrar el lema de un token, únicamente se realiza una búsqueda en la tabla (SpaCy, 2019).
3. ***Lowercase*:** cuenta con tres tipos de parámetros de transformación de palabras minúsculas, estos parámetros son *lower* (int), *lower_* (Unicode) y *is_lower* (booleano).

2.3.2 NLTK (Natural Lenguage Toolkit)

Es un kit de herramientas que clasifica texto, etiqueta, extrae entidades, realiza tokenización y razonamiento semántico. Esta herramienta se encuentra disponible bajo una licencia de código abierto. NLTK se compone de varios módulos independientes organizados en forma jerárquica, cada módulo tiene su propia estructura y se enfoca en una tarea específica. El propósito de contar con estos módulos individuales es definir los tipos de datos y el sistema de procesamiento que se aplicará para procesar información (Loper & Bird, 2002; TheAppSolutions, 2018).

NLTK es una herramienta popular en el área de investigación y enseñanza debido a su algoritmo de aprendizaje automático supervisado. Esta herramienta trabaja con más de 50 recursos corporales y léxicos, también cuenta con varios corpus en distintos idiomas, sin embargo, no cuenta con un soporte multilenguaje. La principal característica que ofrece esta herramienta es la facilidad de uso, pues ofrece una interfaz simple y uniforme. Al mismo tiempo cuenta con una documentación extensa, lo que la vuelve una herramienta de fácil aprendizaje. Debido a su modelo de entrenamiento NLTK identifica etiquetas de tipo persona, ubicación y organización. (Bird, Klein, & Loper, 2009; Jiang et al., 2016).

a. Corpus

NLTK trabaja con más de 50 recursos corporales y léxicos, también contiene corpus que soportan varios idiomas, uno de los corpus contiene la declaración universal de derechos humanos la cual esta escita en más de 300 idiomas (Bird et al., 2009).

La herramienta NLTK contiene cuatro tipos de estructura de corpus los cuales son:

1. **Simple:** no contiene ninguna estructura, es una colección de textos.
2. **Categorización:** agrupa los textos en categorías de acuerdo a un género.
3. **Superpuesta:** es un tipo de agrupación en el cual un género es importante en más de una categoría.
4. **Temporales:** son colección de texto que utilizan el lenguaje a lo largo del tiempo.

b. Técnicas de pre-procesamiento

Esta herramienta proporciona algunas técnicas para el pre-procesamiento de información, a continuación, se detalla cada una.

1. ***Tokenizer*:** esta técnica de pre-procesamiento está dividida en tres módulos:
    a. ***Sent_tokenizer*:** retorna el texto con oraciones tokenizadas, para realizar la tokenización utiliza el método *PunktSentenceTokenizer*.
    b. ***Word_tokenizer:*** retorna los tokens que haya encontrado del texto, para encontrar los tokens utiliza dos métodos *TreebankWordTokenizer* y *PunktSentenceTokenizer.*
    c. ***TweetTokenizer:*** está dedicado únicamente para datos extraídos de Twitter, este módulo permite que se pueda mantener o quitar información como nombres de usuario, enlaces, entre otros.
2. ***Lemmatizer* y *Stemming*:** esta técnica cuenta con tres módulos *PorterStemmer*, *SnowballStemmer* y *WordNetLemmatizer*. Para el idioma español solo se encuentra disponible *SnowballStemmer*, este método se enfoca en la eliminación de afijos y sufijos de palabras, sin embargo, es deficiente debido a que elimina las palabras sin tomar en cuenta su morfología.
3. ***StopWordsRemoval*:** para aplicar este módulo, NLTK hace uso de un corpus en español, este corpus contiene alrededor de 313 palaras vacías tales como en, la, un, una, entre otras.

2.3.3 Freeling

Es una biblioteca de C++ dedicada al análisis de lenguaje, sus principales ramas son análisis morfológico, detección de entidades nombradas, desambiguación de sentidos de palabras, etiquetado PoS, entre otras. Esta herramienta trabaja con una variedad de idiomas tales como inglés, español, portugués, italiano, entre otros. También ofrece una interfaz de línea de comandos que permite analizar texto y configurar la salida en formato JSON, XML y CoNLL (Padró & Stanilovsky, 2012).

Para realizar el reconocimiento de entidades nombradas Freeling aplica un módulo basado en aprendizaje automático y asigna una clase a las entidades nombradas en el texto, las clases pueden ser cualquier cosa, ya que depende totalmente

de con que ha sido entrenado el modelo. Los modelos de esta herramienta distinguen cuatro clases tales como: persona (NP00SP0), ubicación geográfica (NP00G00), organización (NP00O00) y otros (NP00V00) (Padró & Stanilovsky, 2012).

a. Corpus

Esta herramienta cuenta con una gran variedad de diccionarios, los cuales soportan múltiples idiomas, su principal corpus es el diccionario morfológico. Estos diccionarios se caracterizan por su potente módulo de análisis de afijos y sufijos, también cuentan con un estimador de sufijos probabilístico para palabras desconocidas (Padró & Stanilovsky, 2012).

Los diccionarios más destacados con los que cuenta esta librería es el diccionario catalán, este contiene 71.000 combinaciones de lemas, mientras que el diccionario ingles cuenta con 37.000. Por otro lado, el diccionario español contiene 76.000 combinaciones de lemas y el diccionario portugués cuenta con 105.000 combinaciones (Padró & Stanilovsky, 2012).

b. Técnicas de pre-procesamiento

Freeling trata a las técnicas de pre-procesamiento como módulos, esto se basan en un análisis morfológico, sintáctico y semántico, está disponible para el idioma inglés, español, alemán, entre otros (Freeling, 2019). A continuación, se detalla sus principales módulos.

1. **Módulo tokenizer:** este módulo tiene la función de convertir un texto plano en un arreglo de objetos de palabras, para lograr esta conversión se aplican reglas de tokenización, estas reglas vienen definidas por expresiones regulares, la primera regla de coincidencia que encuentra extrae el token, elimina la subcadena y vuelve a repetir el proceso hasta que la oración quede vacía (Freeling, 2019).
2. **Modulo morfológico:** es un meta-módulo que no ejecuta ningún proceso autónomo, su única funcionalidad es crear instancias y llamar a otros módulos, se encarga de enviar la información sobre que submódulos se deben crear y que archivos deben ser usados (Freeling, 2019). Los submódulos con los que cuenta son detención de puntuación, detección de número, detección de fechas, reconocimiento de palabras múltiples, reconocimiento de entidades nombradas, entre otros (Freeling, 2019).

3. **Modulo identificador de idioma:** este módulo tiene como función comparar los datos de entrada con los distintos modelos que se encuentran disponibles para distintos idiomas, con la finalidad del retornar el idioma que más se asemeje al dato de entrada (Freeling, 2019).

2.4 Técnicas de pre-procesamiento de textos

Aplicar pre-procesamiento de texto es esencial cuando se va a trabajar con datos no estructurados y ruidosos, debido a que es una etapa de limpieza y normalización de información. Sin embargo, las distintas técnicas a aplicarse pueden variar dependiendo de la orientación que va a tener el estudio (Gupta & Joshi, 2018).

Otros factores que se deben tomar en cuenta para aplicar técnicas de pre-procesamiento de texto, son el idioma y dominio del texto. A continuación, se detalla las técnicas que se pueden aplicar de acuerdo al contexto de este trabajo de investigación.

2.4.1 Eliminar enlaces web

Generalmente los datos provenientes de redes sociales contienen enlaces web, este tipo de contenido agrega ruido al texto, por lo tanto, es necesario deshacernos de ellos para así eliminar el ruido y reducir la cantidad de texto. La aplicación de esta técnica es sencilla, usualmente se aplican el método tokenizer y donde encuentra coincidencias de enlaces web estos se eliminan (Jianqiang & Xiaolin, 2017).

2.4.2 Eliminar signos de puntuación

Para el análisis de reconocimiento de entidades nombradas, los signos de puntuación no intervienen en el procesamiento de texto, sin embargo, si afecta en el tiempo de procesamiento, por lo tanto, se aconseja eliminarlos. Esto con la finalidad de reducir la cantidad de caracteres dentro del texto y optimizar el tiempo de ejecución (Smailović, Kranjc, Grčar, Žnidaršič, & Mozetič, 2015).

2.4.3 Lowercase

Esta es una de las técnicas más básica dentro del campo de pre-procesamiento, su función consiste en transformar caracteres o palabras que estén en letras mayúsculas a letras minúsculas. Aplicar esta técnica es importante debido a que hace que

disminuya la dimensionalidad del texto (Symeonidis, Effrosynidis, & Arampatzis, 2018).

2.4.4 Tokenizer

Esta técnica de pre-procesamiento consiste en dividir un texto de entrada en palabras o frases significativas, cada división de un token debe delimitarse mediante caracteres especiales, estos pueden ser signos de puntuación (punto, coma, punto y coma, entre otros), espacios en blanco, entre otros. Aplicar esta técnica es bastante usual debido a que facilita el análisis al generar un set de palabras que se pueden analizar de forma individual (Kursat Uysal & Gunal, 2014).

2.5 Métricas de evaluación

Para el propósito de esta investigación es necesario aplicar métricas de evaluación, esto con la finalidad de evidenciar el rendimiento de lo que se está aplicando. Para ello, previamente se necesita realizar una validación a partir de los datos de Twitter que se obtuvieron. Esta validación implica realizar la extracción de etiquetas de dirección de los tweets de forma manual. Esto con la finalidad de evaluar las métricas de precisión, *recall* y *f-measure*.

A continuación, se detallan las métricas en las que se basa la evaluación:

Las métricas de precisión, *recall* y *f-measure* están basadas en una matriz de confusión. Esta matriz permite describir el rendimiento de un modelo de prueba frente a un modelo real basándose en cuatro parámetros: positivo verdadero, negativo verdadero, falso positivo y falso negativo (Renuka, 2016).

Tabla 2: Matriz de Confusión

	Clase predicha		
Clase real		***Clase = SI***	***Clase = NO***
	Clase = SI	Positivo Verdadero (TP)	Falso Negativo (FN)
	Clase = NO	Falso Positivo (FP)	Negativo Verdadero (TN)

- **Positivo verdadero (TP):** el valor de la clase real y pronosticada es sí.
- **Negativos verdaderos (TN):** el valor de la clase real y pronosticada es no.

- **Falsos positivos (FP):** el valor de la clase real es no y el valor de la clase pronosticada es sí.
- **Falsos negativos (FN):** el valor de la clase real es si y el valor de la clase pronosticada es no.

A partir de estos parámetros se puede calcular precisión, *recall* y *f-measure*.

Precisión: relación entre los valores positivos predichos correctos y el total de valores positivos predichos (Renuka, 2016). En nuestro contexto la precisión indica cuantas entidades extraídas son correctas.

$$\boldsymbol{precision} = \frac{TP}{TP + FP} \quad (1)$$

Recall: relación entre valores positivos predichos correctos y el total de valores de la clase real = SI. (Renuka, 2016). En nuestro contexto indica cuantas entidades se han extraído correctamente.

$$\boldsymbol{recall} = \frac{TP}{TP + FN} \quad (2)$$

F-measure: media armónica ponderada de precisión y recall (Shaalan & Raza, 2010). Es una medida que pesa recall y precisión por igual.

$$\boldsymbol{f - measure} = 2 * recall * \frac{presicion}{recall + presicion} \quad (3)$$

3. TRABAJOS RELACIONADOS

Obtener información de redes sociales (Facebook, Twitter, Instagram), se ha convertido en un tema activo de investigación, debido a que son la principal fuente de datos en tiempo real (Bontcheva et al., 2013; Derczynski, Maynard, Aswani, & Bontcheva, 2013; Hu et al., 2014). Sin embargo, la cantidad de información que se almacena en línea aumenta constantemente y limita la capacidad de los usuarios para encontrar información de un dominio de interés (Rodriquez et al., 2012). Por este motivo existen varias investigaciones encaminadas a la extracción de información, las cuales aplican distintos métodos, algoritmos o técnicas.

Uno de los métodos que se aplica en la extracción de información es el sistema de reconocimiento de entidades nombradas (NER). Entre las investigaciones que han aplicado este método está el trabajo de Inkpen et al. (2015), esta investigación ha

realizado la extracción de entidades nombradas de localizaciones, con la finalidad de clasificar las direcciones encontradas en países, provincias o estados, y ciudades. Para cumplir con su objetivo han utilizado un algoritmo CRF (Conditional Random Field), y trabajado con tweets en idioma inglés. Los resultados obtenidos mostraron un rendimiento eficiente en la detección de entidades de localización. Bouillot, Poncelet, & Roche (2012) se enfocan en la extracción automática de tweets que contengan localizaciones relacionados a problemas médicos, con las coincidencias encontradas se asignaron en forma jerárquica sus respectivos países, provincias y ciudades. Para realizar la extracción de direcciones utilizaron un diccionario geográfico, su fuente de datos son tweets de enero a mayo del 2011 escritos en idioma francés.

Por otro lado, Gelernter & Mushegian (2011) pretenden identificar el rendimiento en la extracción de entidades de localización aplicando Stanford NER. Para realizar esta evaluación se ha utilizado tweets del terremoto de Japón del 2011 escritos en inglés, también se ha realizado un conjunto de ubicaciones manuales para ser validadas frente a la herramienta. Con respecto a los resultados, Stanford NER obtuvo un rendimiento deficiente en la identificación de direcciones, debido a que únicamente encontró ubicaciones con nombre propio.

El estudio de Lingad, Karimi, & Yin (2013) está encaminada a evaluar el nivel de eficiencia de las herramientas NER para extraer entidades de localización que estén vinculadas a desastres. Las herramientas que se evaluaron fueron Stanford NER, OpenNLP, Yahoo! PlaceMaker y Twitter NLP, el conjunto de datos utilizado para la evaluación fue un conjunto de tweets de desastres que sucedieron desde el 2010 hasta el 2012. Los resultados obtenidos permitieron identificar que la herramienta Stanford NER logró ubicar la mayor parte de direcciones obteniendo una medida de *f-measure* de 0.87 frente a las otras herramientas.

Otro método que también se aplica para extraer entidades de textos es el algoritmo de *Levenshtein*. Varias investigaciones se han desarrollado en torno a esta herramienta, tal es el caso del trabajo de Zhañay et al. (2019), esta investigación tiene como propósito identificar direcciones aplicando el algoritmo de *levenshtein*. Para realizar el reconocimiento se utilizó un diccionario que contiene calles de la ciudad de Cuenca – Ecuador, los datos con los que trabaja son tweets de accidentes de tránsito, así mismo el conjunto de datos se ha dividido en n-gramas, esto con la finalidad de facilitar la comparación entre el diccionario y el conjunto de datos. Los resultados que

se han obtenido son muy prometedores, pues han alcanzado medidas de precisión de 0.95, *f-measure* de 0.85 y *recall* de 0.81.

La investigación de Sri Nandhini et al. (2015), está centrada en detectar ciberacoso en las redes sociales, los datos con los que se ha trabajado son conversaciones de Myspace y Formspring. Para detectar el ciberacoso se aplicó el algoritmo de *levenshtein*, el cual en base a un conjunto de palabras realiza una comparación con el conjunto de datos para identificar palabras intimidantes presentes en las conversaciones. Los resultados obtenidos son eficientes, pues alcanzan medidas de *f-measure* de 0.79 para Formspring y 0.77 para Myspace.

Otro trabajo que también aplica el algoritmo de *levenshtein* es el de Hossain et al. (2019), esta investigación propone desarrollar un sistema que verifique la ortografía de una palabra mediante el algoritmo de *levenshtein*, para realizar la corrección de ortografía se dividieron las palabras en unigramas lo que ayuda a comparar las palabras mal escritas con una lista de palabras de sugerencia, y cambiar a la palabra apropiada automáticamente. El nivel de precisión que se obtuvo en esta propuesta fue de 0.78.

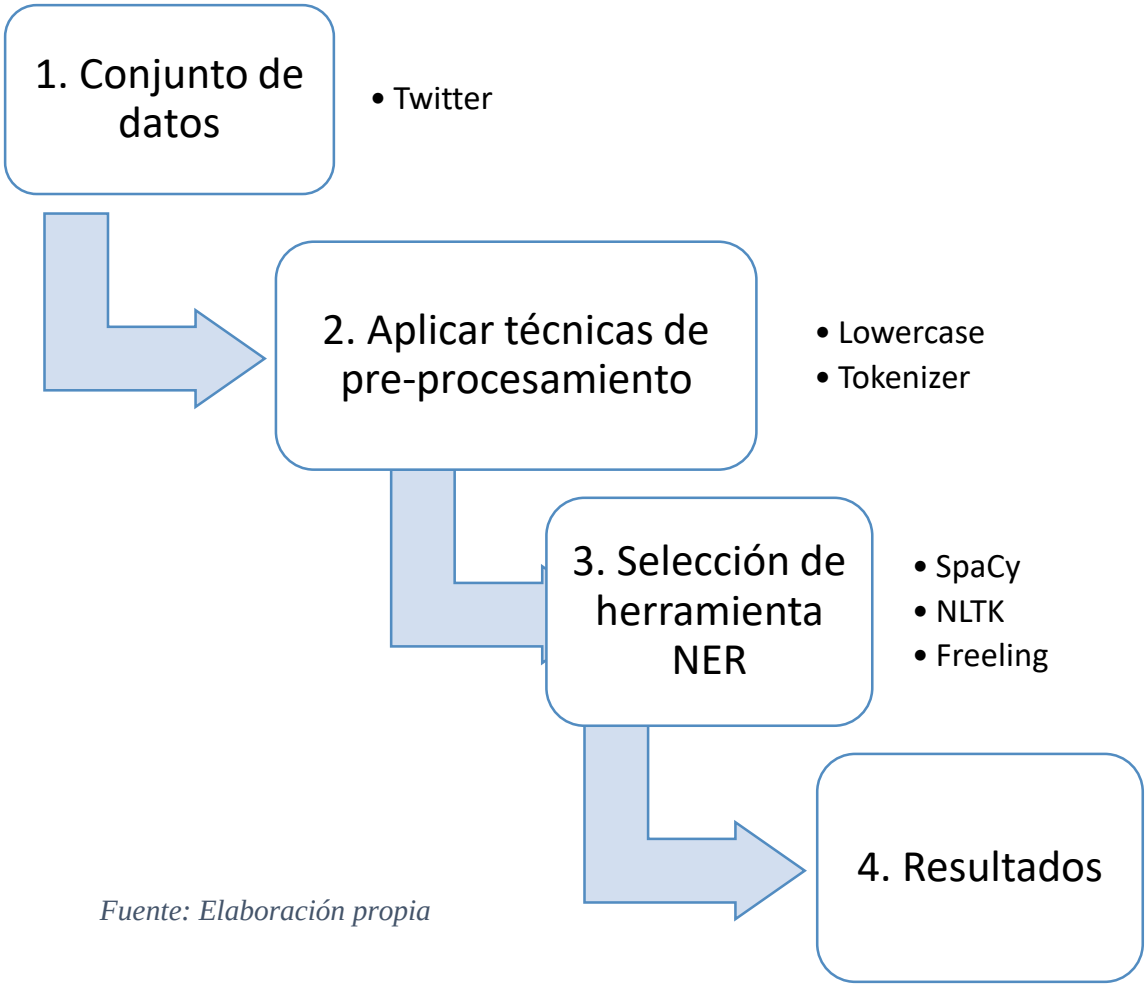
En la investigación de Jácome et al. (2019) han implementado el algoritmo de *levenshtein* para describir el nivel de confiabilidad de las páginas de fans en redes sociales. Para que se pueda aplicar esta investigación ha sido necesario contar con un diccionario de palabras positivas y negativas resultado de un análisis de tres páginas web. Para determinar el nivel de confiabilidad de las páginas se ha realizado una comparación entre los comentarios que se tiene en las páginas de fans frente al diccionario de palabras, los resultados que se obtuvieron fueron eficientes, ya que se pudieron identificar comentarios positivos y negativos entre las páginas analizadas.

4. MÉTODO Y MATERIALES

La Ilustración 1 muestra de forma general la metodología empleada, esta se basa en cinco pasos, los cuales han sido esenciales para la ejecución de este trabajo de investigación. Como paso inicial se tiene la extracción de un conjunto de datos de Twitter, consecuentemente se procede a realizar la limpieza de los datos aplicando técnicas de pre-procesamiento de texto. Como tercer paso se realiza la selección de la herramienta NER que será aplicada en la etapa de experimentación, luego de este paso se procede a la extracción de entidades de localización mediante la aplicación de la

herramienta NER y el algoritmo de *levenshtein* y finalmente se obtiene los resultados aplicando las métricas de evaluación.

Las siguientes subsecciones describen con más detalle cada uno de los pasos ejecutados.



Fuente: Elaboración propia

Ilustración 1: Proceso de experimentación

4.1 Conjunto de datos

Los métodos de extracción de entidades se evaluaron en un dominio de tweets de accidentes de tránsito que han sucedido en la ciudad de Cuenca – Ecuador, en un intervalo de tiempo de enero a diciembre del año 2017. El total de tweets que se obtuvieron fue 1.030.

Las principales cuentas de donde se extrajeron los tweets fueron de entidades de emergencia como el ECU 911 y Bomberos. Los tweets se obtuvieron mediante la API de Twitter y el lenguaje de programación R.

Las cadenas de búsqueda que se aplicaron fueron las siguientes:

- *"#Cuenca (transito OR tránsito OR calle OR calles OR cuenca OR Cuenca) (from:Ecu911Austro) until:2017-12-31 since:2017-01-01"*
- *"#Cuenca (transito OR tránsito OR calle OR calles OR cuenca OR Cuenca) (from:Bomberos_Cuenca) until:2017-12-31 since:2017-01-01"*

4.2 Técnicas de pre-procesamiento

Aplicar técnicas de pre-procesamiento de texto es importante debido a que ayuda a eliminar el ruido de los datos. Un punto de partida esencial en esta etapa fue la revisión de trabajos que apliquen dichas técnicas en el mismo contexto que se está desarrollando esta investigación, también se revisaron definiciones las cuales se describieron en la sección 2.4.

De acuerdo a las funcionalidades que ofrece cada una de las técnicas y al conjunto de datos con el que se está trabajando, se vio a necesidad de aplicar eliminación de enlaces web, eliminación de signos de puntuación, *lowercase* y *tokenizer*.

En la Tabla 3 se describe el orden y una breve descripción de la forma en cómo se aplicaron las técnicas al conjunto de datos.

Tabla 3: Técnicas de pre-procesamiento aplicadas

Orden	Técnica	Aplicación
1	Lowercase	Transformación de mayúsculas a minúsculas
2	Eliminar enlaces web	Eliminación de enlaces URL.
3	Eliminar signos de puntuación	Eliminación de caracteres especiales: @, #, '.', ´, " ",' '
4	Tokenizer	Separar oraciones en tokens

4.3 Selección de la herramienta NER

Existen varias herramientas de reconocimiento de entidades nombradas, sin embargo, para elegir una herramienta eficiente es importante considerar el contexto en

el que se va a aplicar, las entidades que se van a extraer y el idioma en el que se enfocará.

Para que una herramienta de extracción de entidades pueda ser considerada, la principal característica que debe satisfacer, es ser una herramienta de código abierto que permita un uso ilimitado y cuente con una comunidad activa para que pueda brindar soporte y mejoras a la herramienta. Otro factor que también influye fuertemente es que la herramienta permita extraer etiquetas de localización y soporte el idioma español.

Considerando estos aspectos, se ha realizado una revisión de las herramientas de reconocimiento de entidades nombradas de código abierto, en la Tabla 4 se detalla una descripción general de las principales herramientas que se encontraron.

Tabla 4: Herramientas de código abierto.

Herramienta	Código abierto	Idioma español	Extracción direcciones	Lenguaje	Comunidad activa
NLTK	Si	Si	Si	Python	Si
SpaCy	Si	Si	Si	Python	Si
AllenNLP	Si	No	No	Python	Si
Freeling	Si	Si	Si	C++	Si
OpenNLP	Si	No	Si	Java	No

Como se puede ver en la tabla anterior, las principales herramientas de código abierto que se encontraron fueron NLTK, SpaCy, AllenNLP, Freeling y OpenNLP. Las herramientas como AllenNLP y OpenNLP no soportan el idioma español, aunque si extraen direcciones y cuentan con una comunidad activa, no pueden ser consideradas para el análisis teórico que se requiere. Sin embargo, NLTK, SpaCy y Freeling sí podrían ser consideradas, ya que cuentan con las características antes descritas.

En la sección 2.3 se describe de manera teórica NLTK, SpaCy y Freeling, en esta descripción se hace una revisión del concepto general de la herramienta, el corpus que maneja, los modelos que trabajan en el idioma español y las técnicas de pre-procesamiento que realizan.

Tomando en cuenta sus definiciones y la ejecución de un análisis de características que las hacen apropiadas para el propósito de esta investigación, se eligió como herramienta para ser aplicada en la experimentación a SpaCy. La selección tiene varios justificativos:

- SpaCy cuenta con un sistema eficiente en el reconocimiento de entidades nombradas.
- Posee un analizador sintáctico y aplica tokenización para un análisis más profundo al momento de extraer entidades.
- La herramienta cuenta con dos modelos para el idioma español, los cuales permiten obtener medidas de precisión entre 89.59% - 90.02%, *recall* entre 89.315 - 89.70% y f-*measure* entre 89.45% - 89.86%.

4.4 Etiquetado manual

Para obtener las métricas de evaluación (precisión, *recall* y *f-measure*), fue necesario que el conjunto de datos este etiquetado manualmente. Para realizar este proceso de etiquetado se utilizó Doccano. La herramienta es de código abierto que ofrece la funcionalidad de clasificación y etiquetado de texto. Esta herramienta permite crear datos para reconocimiento de entidades nombradas, análisis de sentimiento, resumen de texto, entre otros (Doccano, 2019).

Doccano es una herramienta fácil de usar, únicamente de se debe cargar el conjunto de datos, elegir la funcionalidad y especificar la etiqueta con la que se va a trabajar. Para este trabajo de investigación, se ha definido la etiqueta de localización, como se puede observar en la Ilustración 2. El proceso de etiquetado que se ha realizado ha sido señalar la calle, y Doccano identifica como una etiqueta de localización.

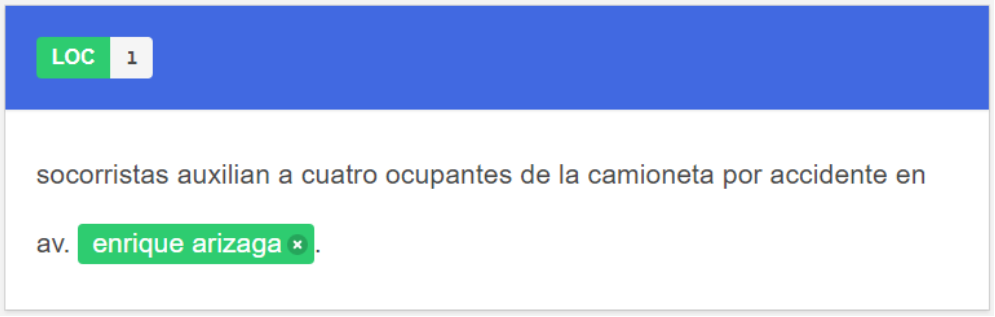


Ilustración 2: Etiquetado en Doccano.

En la Ilustración 3 se visualiza el resultado de haber etiquetado el conjunto de datos, como se puede observar se tiene al conjunto de datos seguido de la etiqueta con la que ha sido etiquetada y la posición en la que se encuentra dicha etiqueta.

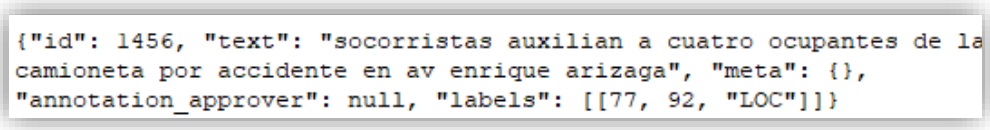


Ilustración 3: Resultado de etiquetado.

4.5 Diccionario de localizaciones

El recurso que se requirió para la aplicación de la herramienta SpaCy y el algoritmo de *levenshtein* es un diccionario, el cual contiene calles de la ciudad de Cuenca-Ecuador, así como también referencias de parques, barrios, entre otros. La información del diccionario se encuentra clasificada por un ID (identificador único), categoría, nombre y nombre corto.

El diccionario se encuentra diferenciado en cinco categorías:

1. Calles
2. Parques
3. Parroquias
4. Barrios

En la Tabla 5, se muestra un ejemplo de los datos que contiene el diccionario.

Tabla 5: Ejemplo del diccionario de calles de Cuenca-Ecuador.

Id	Categoría	Nombre	Nombre corto
1	calle	presidente antonio borrero	borrero
2	parque	parque plaza civica 9 de octubre	9 de octubre
3	parroquia	totoracocha	
4	barrio	barrio san antonio de gapal	

4.6 Parámetros para evaluar SpaCy y el Algoritmo de Levenshtein

4.6.1 SpaCy

Para aplicar esta herramienta al conjunto de datos fue necesario, determinar cuatro parámetros, el tamaño de evaluación y entrenamiento del conjunto de datos, *dropout*, diccionario y n-gramas.

a. Evaluación y entrenamiento.

Para el tamaño de entrenamiento y evaluación se determinó 80% y 20% respectivamente, esto significa que al momento de aplicar SpaCy se utilizará el 80% del conjunto de datos para realizar el entrenamiento y el 20% restante servirá para la evaluación y así obtener las medidas de evaluación descritas en la sección 2.5. El campo que indica este parámetro se denomina *size.*

b. Dropout

Como se mencionó en la sección 2.3.1, SpaCy cuenta con modelos de redes neuronales, los cuales permiten realizar el etiquetado de texto. Por esta razón es importante definir el valor del parámetro *dropout*.

Dropout es un método que permite inhabilitar aleatoriamente el número de neuronas de la red neuronal para la fase de entrenamiento. El parámetro permite que las neuronas tengan menor dependencia entre sí, trabajen de forma solidaria y no dependan tanto de las relaciones con neuronas vecinas (Pham, Bluche, Kermorvant, & Louradour, 2014).

Este parámetro toma valores en una escala de 0 a 1, 0.5 es su valor por defecto, el cual indica que la mitad de las neuronas están inactivas. Si se establecen valores cercanos a cero el *dropout* desactivará menos neuronas, mientras que si es cercano a uno desactivará más neuronas (Pham et al., 2014).

El campo que indica este parámetro se denomina *drop*

c. Diccionario

Este parámetro indica si se utiliza el diccionario de localizaciones para comparar las entidades encontradas y determinar si es o no una dirección, 0 → no aplica diccionario, 1 → si aplica diccionario. El campo que indica este parámetro es *dictionary.*

d. División en n-gramas

Un n-grama se denomina como una sub-secuencia de n elementos de una secuencia dada, en el contexto de este trabajo de investigación se refiere a la división del conjunto de datos en sub-secuencias, las cuales serán comparadas con el diccionario de localizaciones.

Este parámetro indica si es obligatorio la división del conjunto de datos en n-gramas, 0 → no divide en n-gramas, 1 → divide en n-gramas. El campo que indica este parámetro es *grams*.

En la Tabla 6, se visualizan las variaciones de los parámetros que se han aplicado en la fase de experimentación, para cada variación se asignó un nombre de test.

Tabla 6: Variaciones de parámetros SpaCy

Test	Size	Drop	Dictionary	Grams
T1	80	0,3	0	0
T2	80	0,3	0	1
T3	80	0,3	1	0
T4	80	0,3	1	1

4.6.2 Algoritmo de levenshtein

De acuerdo a lo que se describió en la sección 2.2.2 el algoritmo de *levenshtein* posee una métrica denominada *threshold*, esta métrica permite definir el grado de similitud que debe cumplir las palabras que se encuentran durante el proceso de aplicación. Por esta razón se tuvo que definir este parámetro, debido a que es indispensable para la aplicación del algoritmo.

En la Tabla 7, se detalla los valores de *threshold* que se utilizó para la experimentación, a cada variación realizada se asigna un nombre de test.

Tabla 7: Variación parámetros levenshtein

Test	Threshold
T1	0,9
T2	0,8
T3	0,7
T4	0,6

4.7 Software

En esta sección se describe los aplicativos que se utilizaron para aplicar la fase de experimentación.

SpaCy está diseñada para ser aplicada en el lenguaje Python, por lo tanto, es necesario contar con un software que soporte este lenguaje, permita la instalación de todos sus paquetes, y proporcione un entorno de desarrollo ágil para la programación, compilación y ejecución.

Por las razones mencionadas se utilizó PyCharm. Adicionalmente, el entorno de desarrollo proporciona todas las herramientas necesarias para un desarrollo productivo en Python (JetBrains, 2019). Para trabajar con el algoritmo de levenshtein también se utilizó PyCharm.

Otro software que se utilizó fue Doccano, como se mencionó en la sección 4.4 se tuvo la necesidad de realizar un etiquetado manual, esta herramienta permite realizar el etiquetado para análisis de sentimiento, reconocimiento de entidades con nombre, entre otros (Doccano, 2019).

El último software utilizado fue RStudio, el cual se utilizó para realizar los gráficos que permitan interpretar los resultados finales.

5. RESULTADOS

Esta sección se dividirá en tres subsecciones, en la primera subsección se detallará los resultados obtenidos de la aplicación de SpaCy, en la siguiente subsección los resultados de la aplicación del algoritmo de Levenshtein y finalmente la comparación entre SpaCy y el algoritmo de *Levenshtein*. A continuación, se detalla cada apartado.

5.1 SpaCy

De acuerdo a lo mencionado en la sección 4.6.1, para aplicar SpaCy fue necesario definir los parámetros *size*, *drop*, *dictionary* y *grams*. En la Tabla 8, se muestra los cuatro test resultantes, en cada test está definido los valores de los parámetros que se aplicaron y los resultados obtenidos de las medidas de precisión, *recall* y *f-measure*.

Tabla 8: Resultados SpaCy detallados

Test	Size	Drop	Dictionary	Grams	Precisión	Recall	F-measure
T1	80	0,3	0	0	96,2%	96,9%	96,5%
T2	80	0,3	0	1	96,4%	96,1%	96,2%
T3	80	0,3	1	0	78,0%	97,2%	86,5%
T4	80	0,3	1	1	78,0%	96,1%	86,1%

Para los cuatro test que se aplicaron, los valores de los parámetros de *size* y *drop* son constantes y los parámetros *dictionary* y *grams* varían. Para el parámetro *size* se tiene que el 80% del conjunto de datos se utilizó para entrenamiento y el 20% restante para evaluación, con respecto al parámetro *drop*, se ha definido un valor de 0.3, lo que significa que el 30% de las neuronas de la red neuronal están inhabilitadas.

Para el test uno, el valor del parámetro *dictionary* está en cero, lo que significa que no se realizó ninguna comparación de las direcciones encontradas frente al diccionario de localizaciones, con respecto al parámetro *grams*, este se encuentra en cero, denotando que el conjunto de datos no se ha dividido en n-gramas. Con respecto al test dos, el parámetro *dictionary* se mantiene en cero, sin embargo, el parámetro *grams* cambio a uno, señalando que el conjunto de datos se ha dividido en n-gramas durante el proceso de aplicación de la herramienta.

Por otra parte, en el test tres, se tiene al parámetro *dictionary* establecido en uno, lo que implica que las posibles direcciones encontradas en el conjunto de datos han sido comparadas frente al diccionario de localizaciones para determinar si en realidad es una dirección o no. Finalmente, para el test cuatro el parámetro *dictionary* y *grams* se encuentra en uno, lo que significa que el conjunto de datos ha sido dividido en n-gramas y cada n-grama ha sido cotejado con el diccionario de localizaciones.

Los resultados de las métricas de rendimiento que se obtuvo en el test uno, son eficientes, pues sus valores son superiores frente a los demás test, para esta experimentación no se utilizó un diccionario y no se dividió el conjunto de datos en n-gramas, denotando de esta manera que SpaCy logro identificar direcciones sin realizar ningún tipo de comparación extra, únicamente aplico redes neuronales para extraer direcciones del conjunto de datos.

Para el test dos, los resultados obtenidos son similares a los resultados del test uno, pero con la diferencia de que la experimentación se realizó con un conjunto de datos dividido en n-gramas. Sin embargo, realizar esta división no es relevante puesto a que no influye en el proceso propio de identificación de entidades de spaCy.

Con respecto al test tres, los resultados de las métricas continúan siendo eficientes, aunque es evidente que los valores han cambiado en relación con los test uno y dos, la variación de estas métricas está ligada directamente a la utilización de un diccionario, esto se debe a que durante la experimentación spaCy aplico redes neuronales y también tuvo que cotejar las posibles direcciones encontradas frente al diccionario, lo que significa que si una dirección estuvo mal escrita y al momento de ser comparada, no llego a concordar o no pudo ser considerada como tal.

En relación con el test cuatro, las medidas alcanzadas son similares al test tres, la diferencia de este test con el anterior, se debe a que el conjunto de datos ha sido dividió en n-gramas y cada uno de ellos fue comparado frente al diccionario de localizaciones.

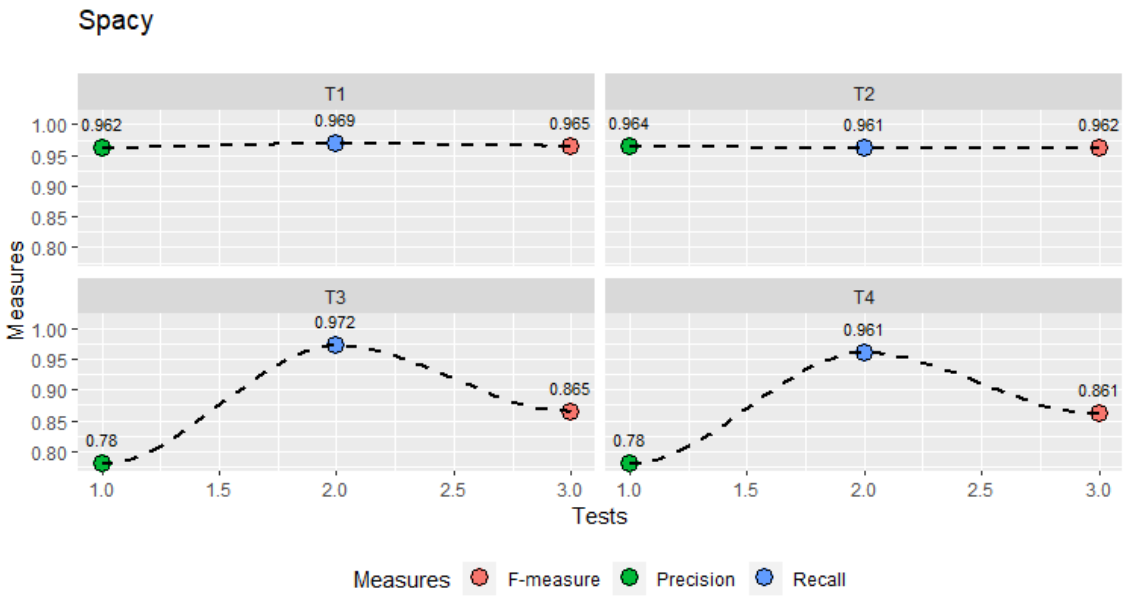


Ilustración 4: Gráfico de resultados SpaCy

En la ilustración 4, se puede visualizar que los cuatro test aplicados han obtenido resultados eficientes, el test uno y dos han alcanzado medidas similares, mientras que el test tres y cuatro, han sufrido un leve déficit en cuanto a precisión y *f-measure*, no obstante, el *recall* se mantiene similar en los cuatro test.

Estos resultados evidencian que spaCy es eficiente identificando por si solo direcciones y también cuando aplica un diccionario, no obstante, cuando está sujeto a un diccionario, la precisión puede llegar a tener un leve déficit, debido a que depende del resultado de la comparación para determinar cuándo es una dirección. Como se mencionó previamente en el caso de que una dirección del conjunto de datos que este mal escrita y al ser comparada con el diccionario no concuerde, está ya no se considera como tal, sin embargo, en la realidad si es una dirección.

5.2 Algoritmo de levenshtein

De acuerdo a lo mencionado en la sección 4.6.2, para aplicar el algoritmo de levenshtein fue necesario establecer el parámetro *threshold*, debido a que permite definir el grado de similitud que deben cumplir las direcciones encontradas en el conjunto de datos. En la Tabla 9, se muestra los cuatro test resultantes, en cada test está definido el valor de parámetro que se ha aplicado y los resultados obtenidos de las medidas de precisión, *recall* y *f-measure*.

Tabla 9: Resultados Levenshtein detallados

Test	Threshold	Precisión	Recall	F-measure
T1	0,9	63,0%	65,2%	65,5%
T2	0,8	60,0%	65,8%	64,8%
T3	0,7	49,3%	47,3%	42,4%
T4	0,6	39,1%	37,1%	38,1%

Para los cuatro test que se aplicaron el valor del parámetro *threshold* sufrió una variación de 0.6 a 0.9. Para el test uno se tiene un *threshold* de 0.9 lo que significa que las palabras encontradas deben ser similares en un 90% a una dirección del diccionario de localizaciones.

Por otra parte, para el test dos el valor del parámetro es de 0.8, lo que implica que las palabras deben ser similares en un 80% frente a las direcciones del diccionario. El test tres es de 0.7 y para el test de 0.6, denotando que las palabras deben ser similares en un 70% y 60% respectivamente.

Los resultados obtenidos en el test uno y test dos, han alcanzado medidas de rendimiento eficientes, sabiendo que para esta experimentación se aplicó un *threshold*

de 0.9, y 0.8, lo que significa que las direcciones encontradas son similares en un 90% y 80% a las direcciones del diccionario de localizaciones.

Con respecto al test tres y test cuatro, los resultados obtenidos son deficientes, es evidente que los valores han cambiado en relación con los test uno y dos, la variación de esas métricas está ligada directamente al valor del *threshold* que se aplicó, esto se debe a que mientras el valor de *threshold* va disminuyendo, también disminuye del grado de similitud.

Por ejemplo, si dentro del conjunto de datos encuentra la palabra "tranvía" y la coteja con la palabra "francia", y si el valor del *threshold* es igual a 0.7, el algoritmo identifica como una dirección a la palabra "tranvía" puesto a que cumple con el grado de similitud, no obstante, en la realidad esta no es una dirección y no debería ser considerada, por esta razón es que las medidas de precisión, *recall* y *f-measure* son muy bajas.

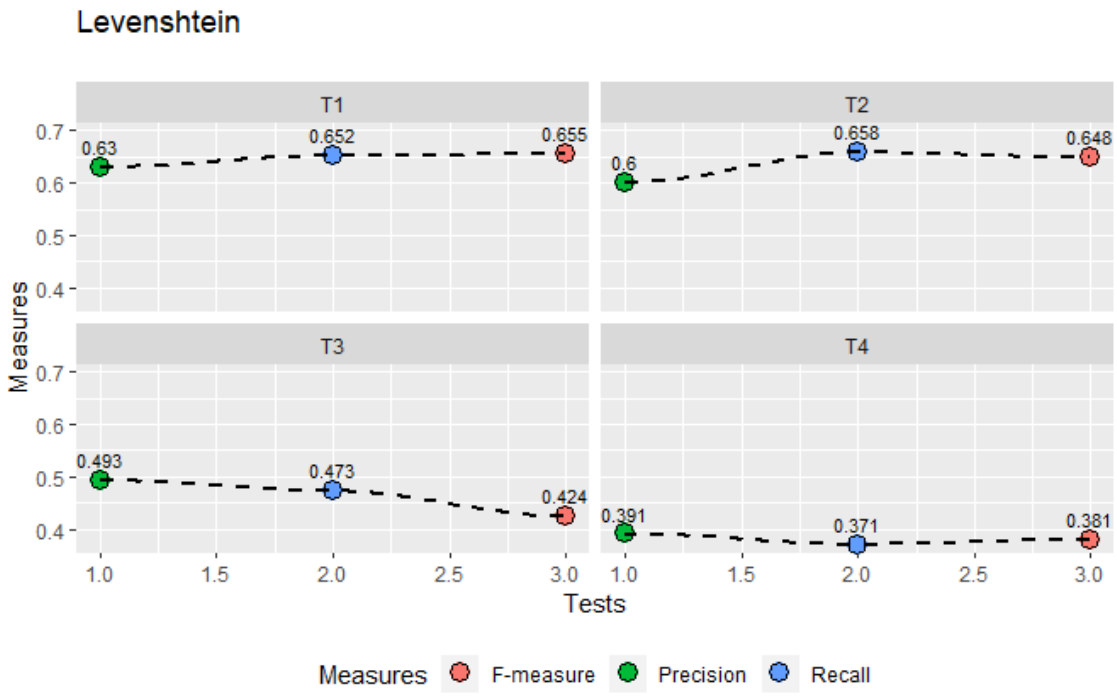


Ilustración 5: Gráfico de resultados levenshtein

En la ilustración 5, se puede visualizar que los test uno y dos han obtenido resultados eficientes, de igual manera sus medidas son similares. Por otra parte, los test tres y cuatro han alcanzado medidas de rendimiento muy bajas, han sufrido en déficit notable en precisión, *recall* y *f-measure*.

Estos resultados evidencian que el algoritmo de *levenshtein* depende en gran magnitud del valor de *threshold* que se establece y del diccionario con el que está trabajando, lo que significa que si se establece un valor de *threshold* menor a 0.8 no puede identificar correctamente las direcciones de un conjunto de datos, debido a que el grado de similitud es bajo, ocasionando que palabras que no son direcciones puedan ser identificadas como tal.

Por otro lado, si se establece un valor de *threshold* superior o igual a 0.8, se puede alcanzar resultados significativos, debido a que el grado de similitud es mayor, por lo que, al momento de cotejar las palabras del conjunto de datos con el diccionario, estas deberán cumplir con dicho grado de similitud para que puedan ser consideradas como direcciones.

5.3 SpaCy vs Algoritmo de Levenshtein

En los apartados 5.1 y 5.2 se detallaron los resultados de spaCy y el algoritmo de levenshtein respectivamente. Sin embargo, para que esta evaluación sea equilibrada, se vio la necesidad de elegir un resultado de cada método que aplique los mismos procedimientos. Por este motivo, para SpaCy se consideró el test cuatro, debido a que utiliza un diccionario y divide el conjunto de datos en n-gramas, estas son las condiciones que aplica el algoritmo de *levenshtein*. Por otro lado, el resultado seleccionado para el algoritmo de *levenshtein* es el test uno, ya que contiene los valores más altos en métricas de evaluación.

A continuación, se detalla los resultados obtenidos de cada método, y se añade el parámetro de velocidad, este parámetro indica el tiempo empleado en la extracción de direcciones del conjunto de datos.

Tabla 10: Resultados SpaCy vs Levenshtein

Método	Tiempo (seg)	Precisión	Recall	F-measure
SpaCy	108	78.0%	96.1%	86.1%
Levenshtein	101	63.0%	65.2%	65.5%

Los resultados obtenidos de aplicar SpaCy, han alcanzado una medida de precisión de 78.0%, un recall de 96.1% y un f-measure de 86.1%, con respecto al

tiempo de ejecución, esta herramienta se tardó 108 segundos en identificar direcciones del conjunto de datos.

Con respecto al algoritmo de levenshtein, los resultados han alcanzado una precisión de 63.8%, recall de 65.2%, f-measure de 65.5% y un tiempo de ejecución empleado para la extracción de direcciones del conjunto de 101 segundos.

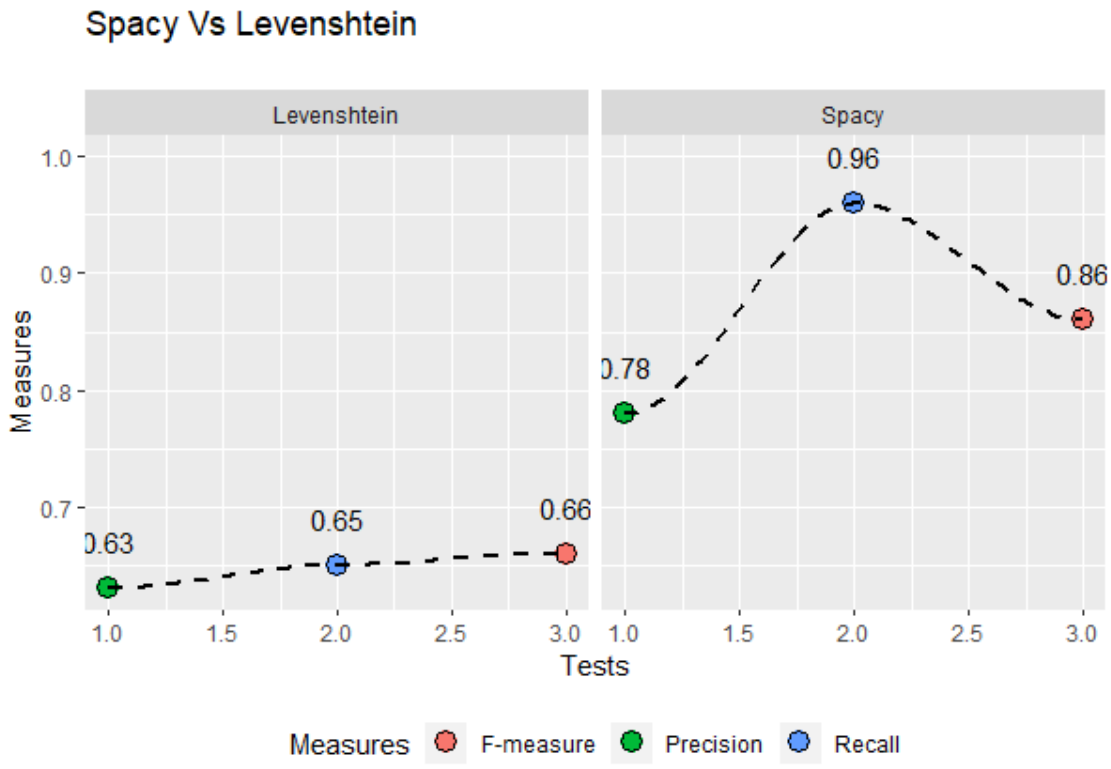


Ilustración 6: Resultados con métricas de rendimiento

La Ilustración 6, muestra los resultados de rendimiento entre los dos métodos propuestos en esta investigación. Al lado izquierdo se puede observar que el rendimiento de la aplicación del algoritmo de *levenshtein* es menor en cada una de las medidas evaluadas (precisión, *recall* y *f-measure*), en comparación de SpaCy. Es notable la diferencia existente entre los resultados con respecto a la presión, spaCy es mayor con aproximadamente un 30%, con relación a *recall*, el resultado de spaCy es mayor con un 20% y para *f-measure*, y spaCy es mayor con un 10%.

Estos resultados evidencian que el algoritmo de levenshtein ha identificado una menor cantidad de direcciones con respecto a spaCy, como se mencionó en la sección

5.2 este algoritmo depende en gran magnitud del diccionario con el que se está trabajando y el valor de *threshold*, por lo tanto, si las direcciones en el set de datos no están escritas correctamente o no coinciden con las direcciones del diccionario están no son consideradas como tal.

Por otro lado, spaCy ha logrado identificar mayor cantidad de direcciones del conjunto de datos, aunque está ligado al uso de un diccionario de localizaciones, spaCy también aplica redes neuronales para la extracción de etiquetas, lo que permite que esta herramienta no solo se base en el diccionario, sino además en un aprendizaje automático a través de las redes neuronales.

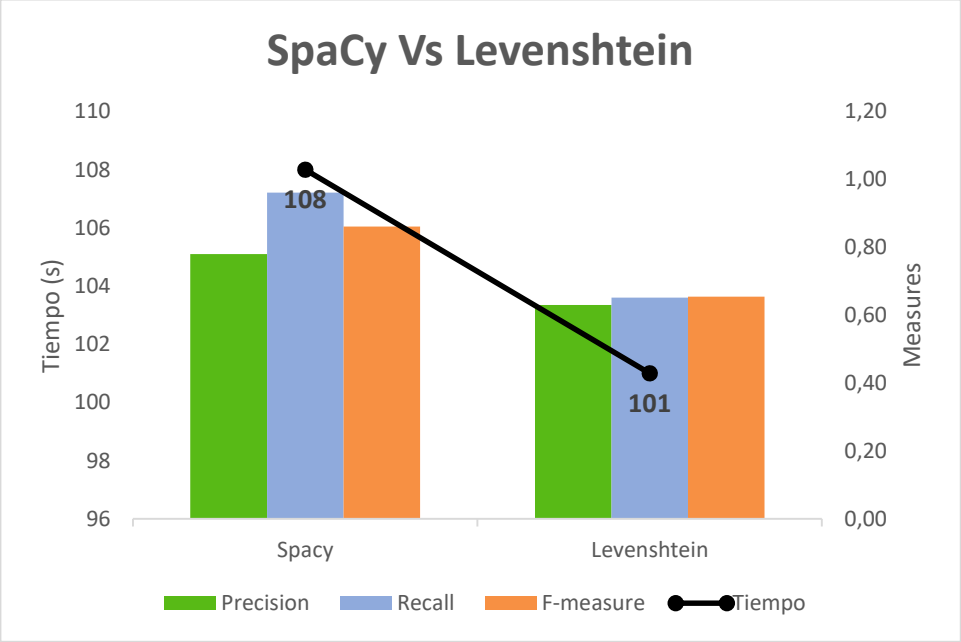


Ilustración 7: Resultado con medida de tiempo

En la Ilustración 7, se visualiza el tiempo que tardo el algoritmo de levenshtein y spaCy en la extracción de direcciones del conjunto de datos. Aunque spaCy haya tenido mejores resultados en cuanto a las métricas de precisión, recall y f-measure, su tiempo de ejecución es mayor con 7 segundos al tiempo de ejecución del algoritmo de *levenshtein*.

6. CONCLUSIONES

En este trabajo, se realizó una comparación de dos métodos para la extracción de entidades de direcciones, con la finalidad de determinar el método adecuado para realizar la extracción de entidades de localización de textos escritos en español. Las herramientas que se evaluaron fueron SpaCy y el Algoritmo de *Levenshtein*, estas técnicas se evaluaron con un conjunto de datos proveniente de twitter, el cual contiene datos de tweets de accidentes de tránsito que han sucedido en un intervalo de tiempo de enero a diciembre del año 2017.

Los aspectos que se consideraron en esta evaluación fueron el tiempo de procesamiento de cada método, y las medidas de precisión, *recall* y *f-measure*. El análisis experimental reveló que la herramienta que mejor extrae entidades de dirección de textos cortos escrito en español es Spacy, ya que el proceso de aprendizaje a través de la técnica de redes neuronales, supera la precisión que puede tener el algoritmo de levenshtein, que a diferencia de las redes neuronales, no posee un proceso de aprendizaje. Al contrario, este algoritmo se fundamenta en la configuración del parámetro de parentesco con respecto a un diccionario. Es resumen, si el diccionario no es lo suficientemente completo en cuanto a entidades nombradas, el rendimiento del algoritmo de *levenshtein* también se ve disminuido.

Estas herramientas han sido evaluadas en un mismo entorno, pues ambas han utilizado un diccionario para realizar la comparación y el conjunto de datos ha sido dividido en n-gramas. Sin embargo, se percibe la diferencia en la forma de procesamiento de cada método. SpaCy aparte de consultar el diccionario también aplica redes neuronales para la identificación de etiquetas.

Por otra parte, el algoritmo de levenshtein está sujeto directamente al diccionario y al valor de threshold ingresado, por lo tanto, si una dirección del conjunto de datos está mal escrita o no tiene el grado de similitud establecido no puede ser considerada como una dirección.

Con respecto al tiempo de ejecución de los métodos, el algoritmo de levenshtein es mucho más rápido que SpaCy, pues se tiene una diferencia de siete segundos entre los valores de ejecución de cada método.

BIBLIOGRAFÍA

Beijering, K., Gooskens, C., & Heeringa, W. (2008). Predicting intelligibility and perceived linguistic distance by means of the Levenshtein algorithm. *Linguistics in the NetherlandsLinguistics in the Netherlands. AVT Publications*, *25*, 13–24. https://doi.org/10.1075/avt.25.05bei

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. Retrieved from http://nltk.org/book

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., & Aswani, N. (2013). TwitIE: An open-source information extraction pipeline for microblog text. *International Conference Recent Advances in Natural Language Processing, RANLP*, (September), 83–90.

Borkar, V., Deshmukh, K., & Sarawagi, S. (2001). Automatic segmentation of text into structured records. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, *30*(2), 175–186. https://doi.org/10.1145/376284.375682

Bouillot, F., Poncelet, P., & Roche, M. (2012). *How and why exploit tweet ' s location information ?*

Cuningham, H. (2006). Information Extraction, Automatic. *Linguistics*, *5*, 665–677.

Derczynski, L., Maynard, D., Aswani, N., & Bontcheva, K. (2013). Microblog-genre noise and impact on semantic annotation accuracy. *HT 2013 - Proceedings of the 24th ACM Conference on Hypertext and Social Media*, (May), 21–30. https://doi.org/10.1145/2481492.2481495

Derczynski, L., Maynard, D., Rizzo, G., Van Erp, M., Gorrell, G., Troncy, R., … Bontcheva, K. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing and Management*, *51*(2), 32–49. https://doi.org/10.1016/j.ipm.2014.10.006

Doccano. (2019). Doccano. Retrieved from https://doccano.herokuapp.com/

Domino Data Lab, I. (2019). Comparing the Functionality of Open Source Natural Language Processing Libraries. Retrieved from https://blog.dominodatalab.com/

Freeling. (2019). *Freeling*.

Gelernter, J., & Mushegian, N. (2011). Geo-parsing messages from microtext. *Transactions in GIS*, *15*(6), 753–773. https://doi.org/10.1111/j.1467-9671.2011.01294.x

Getoor, L. (2003). Link Mining: A New Data Mining Challenge. *SIGKDD Explorations*, *4*(2), 1–6.

Grishman, R., Huttunen, S., & Yangarber, R. (2002). Information extraction for enhanced access to disease outbreak reports. *Journal of Biomedical Informatics*, *35*(4), 236–246. https://doi.org/10.1016/S1532-0464(03)00013-3

Gupta, I., & Joshi, N. (2018). Tweet normalization: A knowledge based approach. *2017 International Conference on Infocom Technologies and Unmanned Systems: Trends and Future Directions, ICTUS 2017*, *2018-Janua*, 157–162. https://doi.org/10.1109/ICTUS.2017.8285996

Hicham, G., Abdallah, Y., & Mostapha, B. (2012). Introduction of the weight edition errors in the Levenshtein distance. *International Journal of Advanced Research in Artificial Intelligence*, *1*(5), 1–3. https://doi.org/10.14569/ijarai.2012.010506

Hossain, M. M., Labib, M. F., Rifat, A. S., Das, A. K., & Mukta, M. (2019). Auto-correction of English to Bengali Transliteration System using Levenshtein Distance. *2019 7th International Conference on Smart Computing and Communications, ICSCC 2019*, (September), 1–5. https://doi.org/10.1109/ICSCC.2019.8843613

Hu, Y., Mao, H., Mckenzie, G., Science, G. I., Group, T., National, O. R., & Ridge, O. (2014). *Trajectory Generator for Animal Movement*. 0–23. https://doi.org/10.1080/1365881YYxxxxxxxx

Inkpen, D., Liu, J., Farzindar, A., Kazemi, F., & Ghazi, D. (2015). Detecting and disambiguating locations mentioned in twitter messages. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9042*, 321–332. https://doi.org/10.1007/978-3-319-18117-2_24

Jácome, D., Tapia, F., Lascano, J. E., & Fuertes, W. (2019). Contextual Analysis of Comments in B2C Facebook Fan Pages Based on the Levenshtein Algorithm. *Advances in Intelligent Systems and Computing*, *918*, 528–538. https://doi.org/10.1007/978-3-030-11890-7_51

Jansche, M., & Abney, S. P. (2001). *Information extraction from voicemail*. (July), 298–305. https://doi.org/10.3115/1073012.1073051

Jiang, R., Banchs, R. E., & Li, H. (2016). *Evaluating and Combining Name Entity Recognition Systems*. 21–27. https://doi.org/10.18653/v1/w16-2703

Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, *5*(c), 2870–2879. https://doi.org/10.1109/ACCESS.2017.2672677

Kursat Uysal, A., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, *50*(1), 104–112.

Lingad, J., Karimi, S., & Yin, J. (2013). Location extraction from disaster-related microblogs. *WWW 2013 Companion - Proceedings of the 22nd International Conference on World Wide Web*, 1017–1020. https://doi.org/10.1145/2487788.2488108

Loper, E., & Bird, S. (2002). *NLTK: The Natural Language Toolkit*. Retrieved from http://arxiv.org/abs/cs/0205028

Maedche, A., Neumann, G., & Staab, S. (2012). *Bootstrapping an Ontology-Based*

Information Extraction System. 345–359. https://doi.org/10.1007/978-3-7908-1772-0_21

Marrero, M., Sánchez-Cuadrado, S., Morato, J., & Andreadakis, Y. (2009). Evaluation of Named Entity Extraction Systems. *Research In Computer Science*, *41*, 47–58. Retrieved from http://pics.cicling.org/2009/RCS-41/047-058.pdf

Padró, L., & Stanilovsky, E. (2012). FreeLing 3 . 0 : Towards Wider Multilinguality. *Proceedings of the Language Resources and Evaluation Conference (LREC 2012) ELRA*, 2473–2479. Retrieved from http://nlp.lsi.upc.edu/publications/papers/padro12.pdf

Pham, V., Bluche, T., Kermorvant, C., & Louradour, J. (2014). Dropout Improves Recurrent Neural Networks for Handwriting Recognition. *Proceedings of International Conference on Frontiers in Handwriting Recognition, ICFHR, 2014-Decem*, 285–290. https://doi.org/10.1109/ICFHR.2014.55

Piskorski, J., & Yangarber, R. (2013). Information Extraction : Past , Present and Future. *Multi-Source, Multilingual Information Extraction and Summarization. Theory and Applications of Natural Language Processing*, 23–50. https://doi.org/10.1007/978-3-642-28569-1

Putera Utama Siahaan, A., Aryza, S., Hariyanto, E., . R., Hasudungan Lubis, A., Ikhwan, A., & Len Eh Kan, P. (2018). Combination of levenshtein distance and rabin-karp to improve the accuracy of document equivalence level. *International Journal of Engineering & Technology*, *7*(2.27), 17. https://doi.org/10.14419/ijet.v7i2.27.12084

Rodriquez, K. J., Bryant, M., Blanke, T., & Luszczynska, M. (2012). Comparison of Named Entity Recognition tools for raw OCR text. *11 Th Conference on Natural Language Processing (KONVENS)*, *5*, 410–414.

Santos, C. N. dos, & Guimarães, V. (2015). *Boosting Named Entity Recognition with Neural Character Embeddings*. https://doi.org/10.18653/v1/W15-3904

Sarawagi, S. (2007). Information Extraction. *Communications of the ACM*, *1*(3), 261–377. https://doi.org/10.1561/1500000003

Smailović, J., Kranjc, J., Grčar, M., Žnidaršič, M., & Mozetič, I. (2015). Monitoring the Twitter sentiment during the Bulgarian elections. *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015*. https://doi.org/10.1109/DSAA.2015.7344886

SpaCy. (2019). SpaCy. Retrieved from https://spacy.io

Sri Nandhini, B., & Sheeba, J. I. (2015). Cyberbullying detection and classification using information retrieval algorithm. *ACM International Conference Proceeding Series*, *06-07-Marc*(June). https://doi.org/10.1145/2743065.2743085

Srihari, R., & Li, W. (2000). *A question answering system supported by information extraction*. 166–172. https://doi.org/10.3115/974147.974170

Symeonidis, S., Effrosynidis, D., & Arampatzis, A. (2018). A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. *Expert Systems with Applications*, *110*, 298–310. https://doi.org/10.1016/j.eswa.2018.06.022

TheAppSolutions. (2018). NATURAL LANGUAGE PROCESSING TOOLS AND LIBRARIES. Retrieved from https://theappsolutions.com/blog/development/nlp-tools/#contents_9

Tkachenko, M., & Simanovsky, A. (2012). Named Entity Recognition: Exploring Features. *11 Th Conference on Natural Language Processing (KONVENS)*, *5*, 118–127.

Zhañay, B. A., Cordero, G. O., Cordero, M. O., & Urigüen, M. I. A. (2019). A Text Mining Approach to Discover Real-Time Transit Events from Twitter. *Advances in Intelligent Systems and Computing*, *884*, 155–169. https://doi.org/10.1007/978-3-030-02828-2_12

UNIVERSIDAD DEL AZUAY

Doctora María Elena Ramírez Aguilar, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay

C E R T I F I C A:

Que, el Consejo de Facultad de Ciencias de la Administración, en sesión del 31 de julio de 2019, conoció y aprobó la solicitud para la realización del trabajo de titulación y el respectivo protocolo presentado por:

Estudiante: Ruth Catalina Fárez Crespo con código 74965
Tema: **Herramientas NER frente a algoritmos adaptados a la extracción de entidades para identificar etiquetas de localización en textos cortos en español**
Previo a la obtención del título de **Ingeniera de Sistemas y Telemática**
Director: Ing. Marcos Orellana Cordero
Tribunal: Ing. Andrés Patiño León e Ing. Oswaldo Merchán Manzano

Plazo de presentación del trabajo de titulación: El Consejo de Facultad resolvió establecer el plazo de seis meses para la presentación del trabajo de titulación concluido y calificado por el Director; este plazo se contará desde la fecha de aprobación del protocolo, esto es hasta el 31 de enero de 2020.

Cuenca, 1 de agosto de 2019

Dra. María Elena Ramírez Aguilar
Secretaria de la Facultad de Ciencias de la Administración

UNIVERSIDAD DEL AZUAY
Facultad de Ciencias de la Administración
SECRETARÍA

UNIVERSIDAD
DEL AZUAY

CONVOCATORIA

Por disposición de la Junta Académica de la escuela de Ingeniería de Sistemas y Telemática se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación: **Herramientas NER frente a algoritmos adaptados a la extracción de entidades para identificar etiquetas de localización en textos cortos en español,** presentado por la estudiante Ruth Catalina Fárez Crespo con código 74965, previa a la obtención del título de Ingeniera de Sistemas y Telemática, para el día **Jueves, 27 de junio de 2019 a las 08:00**

Tomar en cuenta que posterior a la sustentación del Diseño del Trabajo de Titulación, por ningún concepto se puede realizar modificaciones ni cambios en los documentos; únicamente, en caso de diseño aprobado con modificación, el Director adjuntará al esquema un oficio indicando que se procede con los cambios sugeridos.

Cuenca, 24 de junio de 2019

Dra. María Elena Ramírez Aguilar
Secretaria de la Facultad

Ing. Marcos Orellana Cordero .....

Ing. Andrés Patiño León .....

Ing. Oswaldo Merchán Manzano .....

Oficio Nro. 054-2019-DIST-UDA

Cuenca, 18 de junio de 2019

Ingeniero,
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
UNIVERSIDAD DEL AZUAY

De nuestras consideraciones,

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 18 de junio del 2019, revisó la documentación del trabajo de titulación denominado"**HERRAMIENTAS NER FRENTE A ALGORITMOS ADAPTADOS A LA EXTRACCIÓN DE ENTIDADES PARA IDENTIFICAR ETIQUETAS DE LOCALIZACIÓN EN TEXTOS CORTOS EN ESPAÑOL**", por la/el estudiante **RUTH CATALINA FÁREZ CRESPO**, con código/s estudiantil **74965**, estudiante/s de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por **MARCOS ORELLANA**, previo a la obtención del título de Ingeniero de Sistemas y Telemática.

La Junta Académica considera que la documentación cumple con las normas legales y reglamentarias de la Universidad y de la Facultad de Ciencias de la Administración y designa como miembros del tribunal a Andrés Patiño y Oswaldo Merchán, así por su digno intermedio, el conocimiento y aprobación por parte del Consejo de Facultad.

Atentamente,

Marcos Orellana Cordero
Coordinador de la Escuela de Ingeniería de Sistemas y Telemática
Universidad del Azuay

1. FECHA DE RECEPCIÓN DE PROTOCOLO: 17-06-2019 FIRMA

2. REVISIÓN DE ESTADO ACADÉMICO DEL ALUMNO:

NOMBRE: Ruth Catalina Pérez Crespo

CÓDIGO: 74965

CARRERA: Ingeniería de Sistemas y Telemática

FECHA DE INICIO DE ESTUDIOS: 17 Marzo/2014

FECHA CULMINACIÓN DE ESTUDIOS: No Termina

HOMOLOGACIONES: NO CARRERA PROCEDENTE

CONVALIDACIONES: NO UNIVERSIDAD PROCEDENTE:

FECHA DE ESTA REVISIÓN: 17 Junio/2019 FIRMA:

DE: DRA. MARÍA ELENA RAMÍREZ, SECRETARIA

ASUNTO: ENVÍO DE PROTOCOLO DE TRABAJO DE TITULACIÓN

PARA: JUNTA ACADÉMICA DE LA CARRERA DE Sistemas

TÍTULO A OTORGARSE: ~~Ingeniero de Sistemas~~ y Telemática

Observación:

Fecha de revisión: 19/06/2019

FIRMA:

TÍTULO DEL TRABAJO: Herramientas NER frente a algoritmos adoptados a la extracción de entidades para identificar etiquetas de localización en textos cortos en español.

REALIZADO EN EL CURSO DE METODOLOGÍA: X SI ____ NO

FECHA DE APROBACIÓN DEL CONSEJO DE FACULTAD: 31 de julio de 2019

DIRECTOR: Ing. Marcos Orellana Cordero

TRIBUNAL: Ing. Andrés Patiño León e Ing. Oswaldo Merchán Manzano

UNIVERSIDAD
DEL AZUAY

ACTA
SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

1. Nombre del estudiante: Ruth Catalina Fárez Crespo
2. Código: 74965
3. Director sugerido: Ing. Marcos Orellana Cordero
4. Codirector (opcional): _____
5. Tribunal: Ing. Andrés Patiño León e Ing. Oswaldo Merchán Manzano
6. Título propuesto: **Herramientas NER frente a algoritmos adaptados a la extracción de entidades para identificar etiquetas de localización en textos cortos en español**
7. Aceptado sin modificaciones: _____
8. Aceptado con las siguientes modificaciones:

_____

_____

_____

9. No aceptado
10. Justificación:

_____

_____

_____

Tribunal

Ing. Marcos Orellana Cordero

Ing. Andrés Patiño León

Ing. Oswaldo Merchán Manzano

Srta. Ruth Catalina Fárez Crespo

Dra. María Elena Ramírez Aguilar
Secretaria de la Facultad

UNIVERSIDAD DEL AZUAY

RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN
(Tribunal)

1. Nombre del estudiante: Ruth Catalina Fárez Crespo
2. Código: 74965
3. Director sugerido: Ing. Marcos Orellana Cordero
4. Codirector (opcional):
5. Título propuesto: Herramientas NER frente a algoritmos adaptados a la extracción de entidades para identificar etiquetas de localización en textos cortos en español
6. Revisores tribunal: Ing. Andrés Patiño León e Ing. Oswaldo Merchán Manzano

	Cumple	No cumple
Problemática y/o pregunta de investigación		
1. ¿Presenta una descripción precisa y clara?	✓	
2. ¿Tiene relevancia profesional y social?	✓	
Objetivo general		
3. ¿Concuerda con el problema formulado?	✓	
4. ¿Se encuentra redactado en tiempo verbal infinitivo?	✓	
Objetivos específicos		
5. ¿Permiten cumplir con el objetivo general?	✓	
6. ¿Son comprobables cualitativa o cuantitativamente?	✓	
Metodología		
7. ¿Se encuentran disponibles los datos y materiales mencionados?	✓	
8. ¿Las actividades se presentan siguiendo una secuencia lógica?	✓	
9. ¿Las actividades permitirán la consecución de los objetivos específicos planteados?	✓	
10. ¿Las técnicas planteadas están de acuerdo con el tipo de investigación?	✓	
Resultados esperados		
11. ¿Son relevantes para resolver o contribuir con el problema formulado?	✓	
12. ¿Concuerdan con los objetivos específicos?	✓	
13. ¿Se detalla la forma de presentación de los resultados?	✓	
14. ¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	✓	

Nota sobre 10 puntos: : 10

Ing. Marcos Orellana Cordero

Ing. Andrés Patiño León

Ing. Oswaldo Merchán Manzano

Cuenca, 13 de junio de 2019

Ingeniero,
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
UNIVERSIDAD DEL AZUAY

De mi consideración,

Estimado Señor Decano, yo **Ruth Catalina Fárez Crespo** con C.I. **0106090574**, código estudiantil 74965; estudiante de la Carrera de Ingeniería de Sistemas y Telemática, solicito encarecidamente a usted, la aprobación del protocolo de trabajo de titulación con el tema **"HERRAMIENTAS NER FRENTE A ALGORITMOS ADAPTADOS A LA EXTRACCIÓN DE ENTIDADES PARA IDENTIFICAR ETIQUETAS DE LOCALIZACIÓN EN TEXTOS CORTOS EN ESPAÑOL"** previo a la obtención del título de Ingeniero de Sistemas y Telemática para lo cual adjunto la documentación respectiva.

Por la favorable acogida que brinde a la presente, anticipo mi agradecimiento/ anticipamos nuestro agradecimiento.

Atentamente

Catalina Fárez

Estudiante de la Escuela de Ingeniería de Sistemas y Telemática

Universidad del Azuay

Cuenca, 13 de junio de 2019

Ingeniero,
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
UNIVERSIDAD DEL AZUAY

De mi consideración,

Yo, **Marcos Patricio Orellana Cordero** informo que he revisado el protocolo de trabajo de titulación, elaborado previo a la obtención del título de Ingenier(o/a) de Sistemas y Telemática, **"HERRAMIENTAS NER FRENTE A ALGORITMOS ADAPTADOS A LA EXTRACCIÓN DE ENTIDADES PARA IDENTIFICAR ETIQUETAS DE LOCALIZACIÓN EN TEXTOS CORTOS EN ESPAÑOL"**, realizado por el estudiante **Fárez Crespo Ruth Catalina, con código estudiantil** 74965, protocolo que a mi criterio, cumple con los lineamientos y requerimientos establecidos por la carrera.

Por lo expuesto, me permito sugerir que sea considerado para la revisión y sustentación del mismo,

Sin otro particular, me suscribo.

Atentamente

Marcos Patricio Orellana Cordero

Universidad del Azuay

UNIVERSIDAD
DEL AZUAY

LA SECRETARIA DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACION DE LA UNIVERSIDAD DEL AZUAY

CERTIFICA:

Que, la señorita RUTH CATALINA FAREZ CRESPO, con número de cédula de identidad 0106090574, código de estudiante Nro. 74965, alumna de la carrera de INGENIERÍA DE SISTEMAS Y TELEMÁTICA, tiene aprobado el 91,11% de créditos de su malla curricular.

Cuenca, 28 de mayo de 2019

Dra. María Elena Ramírez Aguilar
SECRETARIA DE LA FACULTAD
DE CIENCIAS DE LA ADMINISTRACION

UNIVERSIDAD DEL AZUAY
Facultad de Ciencias de la Administración
SECRETARÍA

Derecho Nro. s/d

mjmr.-

Edición Autorizada
10.000 ejemplares
No 0904388

UNIVERSIDAD DEL AZUAY

UNIVERSIDAD DEL AZUAY

INGENIERIA EN SISTEMAS Y TELEMATICA

PROTOCOLO DE TRABAJO DE TITULACIÒN

1. DATOS GENERALES

1.1 Nombre del estudiante: Ruth Catalina Farez Crespo

1.1.1 Código: 74965

1.1.2 Contacto:

Teléfono convencional: 072275077

Celular: 0992298450

Correo electrónico: cfarez19@es.uazuay.edu.ec

1.2 Director sugerido: Orellana Cordero, Marcos Ing.

1.2.1 Contacto:

Teléfono convencional: 074128640

Celular:0999955611

Correo electrónico: marore@uazuay.edu.ec

1.3 Asesor metodológico: Daniela Elisabet Ballari

1.4 Tribunal designado:

1.5 Aprobación:

1.6 Línea de Investigación de la carrera:

1.6.1 Código UNESCO: 1203 Informática de computadores

1203.17 Informática

1.6.2 Tipo de trabajo: Investigación.

1.7 Área de estudio: Procesamiento de datos.

1.8 Título propuesto: Herramientas NER frente a algoritmos adaptados a la extracción de entidades para identificar etiquetas de localización en textos cortos en español.

1.9 Estado del proyecto: Nuevo

2. Contenido

2.1 Motivación de la investigacion.

En la actualidad existe cuantiosa información escrita en lenguaje natural proveniente de sitios web y redes sociales (Getoor, 2003). La información que se obtiene de estos medios es muy valiosa y representa un gran potencial en conocimiento (Derczynski et al., 2015). No obstante, esta información es trasmitida en textos no estructurados y resulta difícil analizarla (Eken & Tantug, 2010). Este inconveniente dio lugar a la necesidad de contar con técnicas que permitan analizar información escrita en lenguaje natural, lo que condujo a la aparición de técnicas de extracción de información (IE) (Piskorski & Yangarber, 2013). El proceso de IE consiste en identificar frases nominales que denotan personas, organizaciones y referencias geográficas a partir de un texto no estructurado (Maedche, Neumann, & Staab, 2012).

Considere el contexto de accidentes de tránsito. En este ejemplo la extracción de información sobre eventos de accidentes en redes sociales busca identificar a los principales actores involucrados, la ubicación del accidente y los afectados. Al obtener toda esta información de manera estructurada permite realizar cualquier tipo de análisis y generar conocimiento. A partir del ejemplo mencionado se puede observar la gran importancia que implica la extracción de información de textos no estructurados.

Existen varios entornos en los que se pueden aplicar las técnicas de extracción de información, uno de ellos es el análisis de aplicaciones de seguros de vida (Glasgow et al. 1998), para encontrar ubicaciones de eventos de tránsito en tiempo real (Zhañay, et al. 2019), y extracción automática de información de sitios web (Chang, Hsu, & Lui, 2003), entre otros.

2.2 Problemática:

En la sección 2.5 (estado del arte y marco teórico), se puede evidenciar que existen varias investigaciones que aplican la técnica NER, sin embargo, existen pocas investigaciones que apliquen el algoritmo de distancia de Levenshtein como un método aceptado para realizar extracción de identidades. Por lo general utilizan el algoritmo de distancia de Levenshtein para corregir errores de escritura. Otro aspecto importante que se puede destacar es la falta de investigaciones enfocadas a textos en idioma español, muchos de los trabajos revisados se enfocan en idioma inglés y portugués, entre otros.

Ninguna de las investigaciones previas propuso realizar una evaluación de las distintas técnicas disponibles para la extracción de entidades, como es la técnica

NER y los algoritmos adaptados a la extracción de entidades (algoritmo de distancia de Levenshtein). No obstante, tampoco se han realizado evaluaciones a textos en idioma español, así cómo tampoco se propone la extracción de entidades nombradas centradas en una sola categoría como pueden ser etiquetas de personas, localizaciones, organizaciones.

Frente a este vacío de conocimiento, se ve la necesidad de evaluar las técnicas de extracción de entidades y centrarse en una sola categoría debido a que se puede determinar la mejor técnica en una categoría específica. De igual manera es importante aplicar a textos en idioma español por la escasa cantidad de investigaciones existentes.

2.3 Pregunta de investigación:

¿Cuál es la técnica que mejor realiza el proceso de extracción de direcciones de un texto escrito en idioma español con herramientas NER y el algoritmo de Levenshtein?

2.4 Resumen

El reconocimiento de entidades nombradas (NER) es un área importante en el campo de la extracción de información. En los últimos años han surgido varias herramientas NER, estas emplean diferentes metodologías, identifican diferentes entidades y trabajan en varios idiomas. Estas circunstancias hacen difícil al usuario determinar la herramienta adecuada. Ante esta necesidad, proponemos evaluar herramientas NER frente a algoritmos adaptados para la extracción de entidades. Esta evaluación se enfocará en la identificación de etiquetas de localizaciones en textos en idioma español. A partir de los resultados obtenidos, se determinará el método adecuado y cuáles serán las librerías o algoritmos vinculados.

2.5 Estado del arte y marco teórico

Existen varias técnicas vinculadas a la extracción de información. Es el caso de la técnica de reconocimiento de entidades nombradas (NER). Esta técnica cuenta con herramientas que clasifican diferentes tipos de entidades nombradas (Persona, Ubicación, Organización, etc.) aplicando distintas metodologías en distintos dominios e idiomas (Jiang, Banchs, & Li, 2016). Otra técnica que también se vincula a la extracción de información son los algoritmos adaptados a la extracción de entidades, es el caso del algoritmo de distancia de Levenshtein. Este algoritmo mide

la distancia de las palabras y calcula los costos requeridos para cambiar una palabra mediante inserción o sustitución (Beijering, Gooskens, & Heeringa, 2008).

Contar con estas técnicas hace posible extraer información útil de un dominio de interés, para establecer la información de manera estructurada y realizar un mejor análisis de la información (Cardie, 1997). Sin embargo, al existir varias técnicas que trabajan con la extracción de información resulta difícil determinar la técnica con mejores características y que mejor se adapte a necesidades particulares como es el caso de la extracción de etiquetas de localización en idioma español.

Varias investigaciones se han desarrollado con respecto a la técnica de reconocimiento de entidades nombradas (NER). En ciertos casos se han evaluado herramientas de NER con el fin de determinar la herramienta con mejor desempeño en la identificación de etiquetas de tipo persona, ubicación y organización para textos en idioma ingles (Jiang et al., 2016) ; también se han evaluado herramientas de NER para determinar si retienen conceptos teóricos e identifica automáticamente reseñas verdaderas y falsas de hoteles posteadas en sitios web (Kleinberg, Mozes, Arntz, & Verschuere, 2018). Por otro lado, se han evaluado herramientas NER para determinar la herramienta con mejor rendimiento en la identificación de etiquetas (persona, localización y organización), esta evaluación está enfocada en un conjunto de datos en idioma portugués (Pires, Devezas, & Nunes, 2017).

Se han desarrollado también investigaciones que aplican algoritmos adaptados a la extracción de entidades. En una investigacion se propone descubrir la ubicación de eventos de tránsito en tiempo real, mediante la aplicación del algoritmo de distancia de Levenshtein (Zhañay et al., 2019); también proponen nuevos métodos para el reconocimiento de entidades y aplican el algoritmo de distancia de Levenshtein para corregir errores de escritura (Cheng, Zheng, & Li, 2013; Eken & Tantug, 2010). Otra investigación aplica el algoritmo de distancia de Levenshtein para la detección de plagio en textos académicos (Su et al., 2008).

2.6 Objetivo General

Realizar una evaluación experimental en la extracción de etiquetas de localización en textos cortos en idioma español utilizando herramientas NER y algoritmo de Levenshtein.

2.7 Objetivos específicos

a) Realizar una indagación inicial de investigaciones afines al trabajo planteado.

b) Realizar una indagación de herramientas NER que soporten el idioma español y etiquetas de localización para determinar la herramienta NER con mejor características.

c) Realizar una evaluación experimental de la herramienta NER, frente al algoritmo de Levenshtein.

d) Analizar los resultados obtenidos en la evaluación.

2.8 Metodología

Métodos

La Ilustración 1 muestra de forma general los pasos a realizar. Se revisará la base teórica del tema mediante una indagación bibliográfica en portales web académicos (Google Scholar, Scopus y IEEE). Se seleccionarán fuentes bibliográficas relevantes para realizar el estado del arte y trabajos relacionados. Se revisarán las distintas herramientas de reconocimiento de entidades nombradas (NER) que estén disponibles y que soporten el idioma español. A partir de esta información se seleccionará la herramienta con las mejores características. Posteriormente se realizará una evaluación experimental de la herramienta de reconocimiento de entidades nombradas seleccionada frente al algoritmo adaptado de Levenshtein para determinar la técnica que mejor realice la extracción de direcciones. La evaluación se realizará en base a métricas de factores de medida (*precision, recall, f-measure*), velocidad y rendimiento.

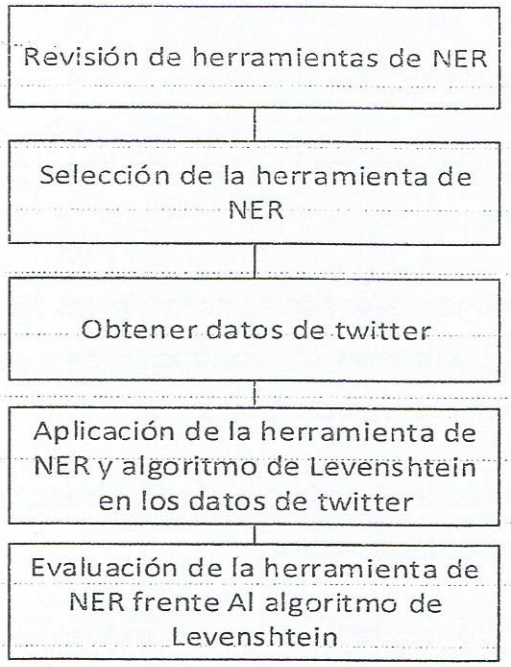


Ilustración 1: Pasos para la metodología.

Materiales

Para realizar la evaluación de la herramienta de NER frente al algoritmo de Levenshtein se utilizarán datos extraídos de la red social Twitter. Los datos provenientes de twitter deben contener información de dirección y estar en idioma español.

Pasos a realizar

a. **Revisión de las herramientas de reconocimiento de entidades nombradas (NER).**

La primera etapa a desarrollarse en el trabajo de investigacion es realizar una revisión teórica de las distintas herramientas de NER que están disponibles. Las herramientas deberán cumplir con las siguientes condiciones para ser consideradas en la selección de herramientas.

1. Debe permitir uso ilimitado y ser gratuita.
2. Debe ser capaz de reconocer la etiqueta de localización.
3. Debe soportar el idioma español.

b. **Selección de la herramienta de reconocimiento de entidades nombradas (NER).**

La segunda etapa de este trabajo es determinar la herramienta de NER que será evaluada posteriormente. Dentro de las herramientas que se pueden considerar para la selección están Stanford NER, spaCy y NLTK.

Stanford NER es una implementación de Java, incluye varias opciones para realizar la extracción de etiquetas, principalmente "PERSONA", "UBICACIÓN" y "ORGANIZACIÓN" (Finkel, Grenager, & Manning, 2007).

spaCy es una implementación en Python, reconoce varios tipos de entidades ya sean entidades de nombre o numéricas, también incluye etiquetas de empresas, ubicaciones, organizaciones y productos (spaCy, 2019).

NLTK es una biblioteca de código abierto de Python, esta herramienta se centra en un algoritmo de aprendizaje automático supervisado, de igual manera trabaja con etiquetas de persona, ubicación y organización (Bird, Klein, & Loper, 2009).

c. **Obtener datos de Twitter.**

Se utilizará la API de Twitter para extraer los tweets. Esta API se basa en una cadena de consulta para extraer datos de un dominio especifico (Zhañay et al., 2019). En este caso se requiere que los datos tengan información de direcciones y se encuentren en idioma español. Varios trabajos de investigacion han venido utilizando este método de extracción de tweets, como se puede ver en (Arias et al., 2014; Eleonora et al., 2015; Zhañay et al., 2019).

d. **Aplicación de la herramienta de NER y algoritmo de Levenshtein en los datos de Twitter.**

En esta cuarta etapa, se aplicará la herramienta de NER a los datos extraídos de twitter para que realice la extracción de etiquetas de localización. De igual manera se aplicará el algoritmo de Levenshtein para que realice la misma tarea.

Durante la aplicación de la herramienta de NER y el algoritmo es necesario medir la velocidad y el rendimiento de cada una de las técnicas para extraer etiquetas de ubicaciones. Esta información es de gran importancia para la

etapa final, pues proporcionará información relevante con respecto a la técnica que se está implementando.

e. **Evaluación de la herramienta de NER frente al algoritmo de Levenshtein.**

Previo a realizar la evaluación se requiere hacer una validación a partir de los datos de twitter que se obtuvieron. Esta validación implica realizar la extracción de etiquetas de dirección de los tweets de forma manual. Esto con la finalidad de evaluar las métricas de *precision, recall y f-measure.*

A continuación, se detallan las métricas en las que se basa la evaluación:

Las métricas de *precision, recall y f-measure* están basadas en una matriz de confusión. Esta matriz permite describir el rendimiento de un modelo de prueba frente a un modelo real en base a cuatro parámetros positivo verdadero, negativo verdadero, falso positivo y falso negativo (Renuka, 2016).

Tabla 1: Matriz de Confusión

		Clase Predicha	
Clase Real		Clase = SI	Clase = NO
	Clase = SI	Positivo Verdadero (TP)	Falso Negativo (FN)
	Clase = NO	Falso Positivo (FP)	Negativo Verdadero (TN)

- Positivo verdadero (TP): el valor de la clase real y pronosticada es sí.
- Negativos verdaderos (TN): el valor de la clase real y pronosticada es no.
- Falsos positivos (FP): el valor de la clase real es no y el valor de la clase pronosticada es sí.
- Falsos negativos (FN): el valor de la clase real es si y el valor de la clase pronosticada es no.

A partir de estos parámetros se pude calcular precisión, recall y f-measure.

Precisión: relación entre los valores positivos predichos correctos y el total de valores positivos predichos (Renuka, 2016). En nuestro contexto la precisión indica cuantas entidades extraídas son correctas.

$$presicion = TP/(TP + FP)$$

Recall: relación entre valores positivos predichos correctos y el total de valores de la clase real = SI (Renuka, 2016). En nuestro contexto indica cuantas entidades se han extraído correctamente.

$$recall = TP/(TP + FN)$$

F-measure: media armónica ponderada de precisión y recall (Shaalan & Raza, 2010). Es una medida que pesa recall y precisión por igual.

$$f - measure = 2 * recall * presicion/(recall + presicion)$$

Previamente se mencionó la necesidad de realizar una validación de los datos. Por lo tanto, a partir de esta evaluación se procede a aplicar las medidas de *precision, recall y f-measure*. Esto se realiza para determinar los valores de la herramienta de NER como del algoritmo de Levenshtein.

Con respecto a la evaluación de la herramienta de NER frente al algoritmo de Levenshtein. Se pretende elaborar una tabla en la cual se indiquen los valores de las métricas (*precision, recall, f-measure*) y los valores de velocidad y rendimiento que se obtuvieron.

A partir de esta información se realizará un análisis en base a los valores de las métricas, la velocidad y el rendimiento. Este análisis permitirá determinar la herramienta que mejor realiza la extracción de entidades de ubicaciones en textos cortos escritos en idioma español.

2.9 Alcances y resultados esperados.

Se espera identificar cuál es la técnica que mejor realiza la extracción de etiquetas de localización, en textos cortos en idioma español.

Se obtendrá una tabla resumen que contendrá la herramienta de NER, y el algoritmo adaptado a la extracción de entidades, junto con los valores de las métricas.

2.10 Supuestos y riesgos

Riesgos	Probabilidad	Alternativas de solución
Tweets escasos del dominio planteado.	Baja	Establecer un dominio de tweets en español con mayor alcance.
Escasas herramientas NER que trabajen con idioma español	Media	Análisis más amplio de las herramientas encontradas.
Tiempo insuficiente para la culminación del trabajo	Baja	Solicitar una prórroga para la entrega final del proyecto.

2.11 Presupuesto

Rubro-Denominación	Costo USD (detalle)	Cantidad	Subtotal	Justificación ¿para qué?
Horas de trabajo	8	480	3.840	Horas de trabajo involucradas en el proyecto
Documentación impresa	200	1	200	Impresiones del documento final.
Servicio de internet	35	4	140	Contratación del servicio para revisión de documentación e información para realizar el proyecto
		Total	4180	

2.12 Financiamiento: Propio

2.13 Esquema tentativo

Capítulo 1: Introducción.

Capítulo 2: Metodología.

Capítulo 3: Revisión teórica de las herramientas de reconocimiento de entidades nombradas.

Capítulo 4: Aplicación de la herramienta NER y el algoritmo de Levenshtein.

Capítulo 5: Evaluación y análisis de resultados.

Capítulo 6: Conclusiones y Recomendaciones.

Referencias

2.14 Cronograma

Objetivo	Actividad	Semanas (20 horas)															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Realizar una indagación inicial de investigaciones afines al trabajo planteado	Revisión de investigaciones a utilizar en el trabajo planteado.																
	Estado del arte																
Realizar una indagación de herramientas NER que soporten el idioma español y etiquetas de localización para determinar la herramienta NER con mejor características	Revisión de bibliografías y documentación enfocada en las herramientas NER																
	Evaluación teórica de las herramientas de NER que soportan el idioma español																
Evaluación experimental de la herramienta NER, frente al algoritmo de Levenshtein	Obtener conjunto de datos																
	Instalación de las librerías vinculadas a la herramienta NER y otras aplicaciones.																
	Aplicar las técnicas al conjunto de datos																
Análisis de los resultados obtenidos en la evaluación	Evaluar los resultados de la aplicación y plasmar los resultados en una tabla resumen																
	Realizar resumen con los resultados obtenidos																

2.15 Referencias

Arias, M., Arratia, A., & Xuriguera, R. (2014). Forecasting with twitter data. *ACM Transactions on Intelligent Systems and Technology*, *5*(1), 1–24. https://doi.org/10.1145/2542182.2542190

Beijering, K., Gooskens, C., & Heeringa, W. (2008). Predicting intelligibility and perceived linguistic distance by means of the Levenshtein algorithm. *Linguistics in the NetherlandsLinguistics in the Netherlands. AVT Publications*, *25*, 13–24. https://doi.org/10.1075/avt.25.05bei

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. Retrieved from http://nltk.org/book

Cardie, C. (1997). Empirical Methods in Information Extraction. *AI Magazine*, *39*(1), 65–79. https://doi.org/10.1007/s13398-014-0173-7.2

Chang, C. H., Hsu, C. N., & Lui, S. C. (2003). Automatic information extraction from semi-structured Web pages by pattern discovery. *Decision Support Systems*, *35*(1), 129–147. https://doi.org/10.1016/S0167-9236(02)00100-8

Cheng, Z., Zheng, D., & Li, S. (2013). *MULTI-PATTERN FUSION BASED SEMI-SUPERVISED NAME ENTITY RECOGNITION*. (10), 14–17.

Derczynski, L., Maynard, D., Rizzo, G., Van Erp, M., Gorrell, G., Troncy, R., … Bontcheva, K. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing and Management*, *51*(2), 32–49. https://doi.org/10.1016/j.ipm.2014.10.006

E., D., P., D., B., L., & F., M. (2015). Real-Time Detection of Traffic from Twitter Stream Analysis. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 2269–2283. https://doi.org/10.1109/TITS.2015.2404431

Eken, B., & Tantug, C. A. (2010). RECOGNIZING NAMED ENTITIES IN TURKISH TWEETS. *Child Development*, *8*(2005), 2010.

Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., & Dredze, M. (2010). Annotating named entities in Twitter data with crowdsourcing. *Human Language Technologies Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, *2010*(June), 80–88.

Finkel, J. R., Grenager, T., & Manning, C. (2007). *Incorporating non-local information into information extraction systems by Gibbs sampling*. (1995), 363–370. https://doi.org/10.3115/1219840.1219885

Getoor, L. (2003). Link Mining: A New Data Mining Challenge. *SIGKDD Explorations*, *4*(2), 1–6.

Glasgow, B., Mandell, A., Binney, D., Ghemri, L., & Fisher, D. (1998). MITA: An information-extraction approach to the analysis of free-form text in life insurance applications. *AI Magazine*, *19*(1), 59–71. https://doi.org/http://dx.doi.org/10.1609/aimag.v19i1.1354

Jiang, R., Banchs, R. E., & Li, H. (2016). *Evaluating and Combining Name Entity Recognition Systems*. 21–27. https://doi.org/10.18653/v1/w16-2703

Kleinberg, B., Mozes, M., Arntz, A., & Verschuere, B. (2018). Using Named Entities for Computer-Automated Verbal Deception Detection. *Journal of Forensic Sciences*, *63*(3), 714–723. https://doi.org/10.1111/1556-4029.13645

Maedche, A., Neumann, G., & Staab, S. (2012). *Bootstrapping an Ontology-Based Information Extraction System.* 345–359. https://doi.org/10.1007/978-3-7908-1772-0_21

Pires, A., Devezas, J., & Nunes, S. (2017). Benchmarking Named Entity Recognition Tools for Portuguese. *Arop.Github.Io*. Retrieved from https://arop.github.io/pubs/artigo-inforum-2017-04.pdf

Piskorski, J., & Yangarber, R. (2013). Information Extraction : Past , Present and Future. *Multi-Source, Multilingual Information Extraction and Summarization. Theory and Applications of Natural Language Processing*, 23–50. https://doi.org/10.1007/978-3-642-28569-1

Renuka, J. (2016). Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures. Retrieved from https://blog.exsilio.com

Shaalan, K., & Raza, H. (2010). *Person name entity recognition for Arabic.* (June), 17. https://doi.org/10.3115/1654576.1654581

spaCy. (2019). Retrieved from https://spacy.io

Su, Z., Ahn, B. R., Eom, K. Y., Kang, M. K., Kim, J. P., & Kim, M. K. (2008). Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm. *3rd International Conference on Innovative Computing Information and Control, ICICIC'08*, 0–3. https://doi.org/10.1109/ICICIC.2008.422

Zhañay, B. A., Cordero, G. O., Cordero, M. O., & Urigüen, M. I. A. (2019). A Text Mining Approach to Discover Real-Time Transit Events from Twitter. *Advances in Intelligent Systems and Computing*, *884*, 155–169. https://doi.org/10.1007/978-3-030-02828-2_12

2.16 Firma de responsabilidad (estudiante)

2.17 Firma de responsabilidad (director sugerido)

2.18 Fecha de entrega:

13 de junio de 2019