Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

EVALUACIÓN DE TÉCNICAS DE PROCESAMIENTO DE LENGUAJE NATURAL PARA TEXTOS CORTOS EN LENGUAJE ESPAÑOL

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

Autor:

Andrea Trujillo Orellana

Director:

Ing. Marcos Orellana Cordero

Cuenca – Ecuador

2018

DEDICATORIA

A Dios, por haberme permitido llegar hasta este punto, dándome salud y sabiduría para cumplir con mis metas.

A mis padres, por el apoyo, sus consejos, sus valores y la motivación constante que me ha permitido ser una persona de bien.

A mis hermanas y hermanos, Diana, Fernanda, Cristina y Erick, por ser un ejemplo de constancia y superación y un apoyo incondicional en mi vida y de manera especial a mi pequeño hermano Rommel David que ha sido mi ángel en cada paso que he dado.

A mis sobrinos Josué y Pablo que a la distancia me han sabido animar con su inocencia y amor.

A mis amigas Evelyn, Maricela, Shirley, Tania y Alex por convertirse en mi segunda familia en estos 5 años de estudio y de manera especial a Sergio quien me ayudó y motivó en cada día difícil para culminar este proyecto.

AGRADECIMIENTO

El primer agradecimiento es para Dios, quien me ha dado la fuerza para lograr este proyecto, a mis padres por la lucha diaria que hacen por verme superar y cumplir mis sueños, y a mi familia que ha sido incondicional.

Quiero agradecer de manera especial a mi director de tesis, Ing. Marcos Orellana, quien, con sus conocimientos, experiencia, y paciencia a permitido que pueda terminar mis estudios con éxito.

Me gustaría agradecer a mis maestros durante mi carrera profesional quienes han aportado con su conocimiento a mi formación tanto académica como personal.

También me gustaría agradecer, a personas que me han ayudado al desarrollo de este trabajo, Ing. María Belén Arias e Ing. Juan Fernando Lima, quienes compartieron sus conocimientos para que pueda finalizar con éxito este trabajo, además de compartir su amistad y consejos durante toda esta etapa.

RESUMEN

El Procesamiento de Lenguaje Natural identifica información clave, generando modelos predictivos y explicando eventos o tendencias mundiales, creando conocimiento, por ello, es importante aplicar técnicas de refinamiento en etapas principales como el pre-procesamiento, al producirse con frecuencia datos y procesamiento con resultados deficientes. Este documento analiza y mide el impacto de combinaciones de técnicas y librerías para textos cortos en español, mediante su aplicación en tweets aplicados al análisis de sentimiento, tomando en cuenta parámetros de evaluación en su análisis, el tiempo de procesamiento y características de las técnicas en cada librería. La experimentación demostró que elegir combinaciones de técnicas adecuadas en el pre-procesamiento, proporciona una mejora de hasta un 5% y 9% en el rendimiento de la clasificación.

Palabras clave: PLN, técnicas, pre-procesamiento, análisis de sentimiento.

ABSTRACT

Natural language processing identified key information and generated predictive models to explain global events or trends and create knowledge. For this reason, it was important to apply refinement techniques in major stages such as pre-processing where data and processing with poor results were often produced. This work analyzed and measured the impact of the combination of techniques and libraries for short texts in Spanish in tweets applied to sentiment analysis. It took into account evaluation parameters for the analysis, processing time and the characteristics of the techniques in each library. Experimentation showed that choosing combinations of appropriate techniques in the pre-processing provided an improvement of 5% to 9% in the performance of the classification.

UNIVERSIDAD DEL AZUAY
Dpto. Idiomas

Translated by
Ing. Paul Arpi

ÍNDICE

Índice de contenido

Índice de tablas y figuras

Tablas

Figuras

Índice de anexos

1. INTRODUCCIÓN

El Procesamiento de Lenguaje Natural (PLN) es una ciencia que se define formalmente como un campo de estudio que combina la informática, inteligencia artificial, lingüística y procesos de análisis de lenguaje natural, para generar conocimiento e inteligencia (Reese, 2015). La importancia de esta ciencia radica en la comprensión y procesamiento de información expresada en diferentes medios como la red social twitter, información que posteriormente puede ser utilizada en múltiples áreas como: búsqueda (Battistelli, Charnois, Minel, & Teissèdre, 2013), máquinas traductoras, reconocimiento de entidad nombrada (NER), agrupación de información, clasificación (Uysal & Gunal, 2014), análisis de sentimientos (Krouska, Troussas, & Virvou, 2016) y otros (Hidalgo, Jaimes, Gomez, & Luján-Mora, 2017).

PLN está conformado por etapas y técnicas aplicadas en áreas específicas como clasificación de texto, NER y análisis de sentimiento (Dubiau & Ale, 2013) y una de sus etapas más importantes es el pre-procesamiento. El pre-procesamiento, está definido como un proceso de limpieza y preparación del texto para su procesamiento final (clasificación, análisis, minería, etc) (Haddi, Liu, & Shi, 2013). Esta etapa a su vez tiene técnicas que se aplican en casi todos los casos al momento de trabajar en PLN, entre estas técnicas se encuentra: t*okenizer, stemming, lemmatizer, stop-words removal, lowercase,* etc. Existen técnicas más específicas que pueden depender del contenido del texto con el que se trabaja. Por ejemplo, a la hora de trabajar con Twitter se destacan técnicas de limpieza, como: eliminación de enlaces web, nombres de usuario, símbolos hashtag, espacios en blanco, nombres propios, palabras numéricas y signos de puntuación (Gupta & Joshi, 2017). La importancia de aplicar estas técnicas en PLN se debe a la frecuencia en que se producen datos de baja calidad generando un procesamiento con resultados muy pobres, por lo tanto, al emplearlas, el conjunto de datos a procesar y sus resultados son vistos con mayor consistencia y calidad.

Tomando en cuenta que para la aplicación de estas técnicas es necesario de librerías o herramientas, algunos investigadores han destacado librerías como: *NLTK, StanFord-NLP, SpaCy y FreeLing,* esta última es considerada una de las más potentes, por sus múltiples funcionalidades y su soporte para idioma español, además de proporcionar análisis del lenguaje para otros idiomas (inglés, portugués, italiano, francés, alemán, ruso,

catalán, etc.) (Henríquez, Guzmán, & Salcedo, 2016). Por otro lado, las librerías restantes se han enfocado en trabajos aplicados al idioma inglés, al ser el lenguaje universal.

Es grande el cambio que las librerías están tomando en la traducción a idiomas diferentes al inglés, en sus corpus, diccionarios y otras funciones (Prata, Soares, Silva, Trevisan, & Letouze, 2016); sin embargo para el español, aún están en etapa de maduración por su compleja sintaxis y semántica (Hidalgo et al., 2017).

En base a antecedentes preliminares es claro, que definir qué técnicas y librerías asociadas a PLN son las adecuada, es un desafío, y más aún si se aplica al lenguaje natural en español, al no existir información amplia en este idioma. La combinación de éstas puede significar un resultado relevante en el procesamiento final del texto; sin embargo, la mayoría de autores evalúa su impacto a breves rasgos, además de dejar a las librerías experimentales en un papel secundario, por lo que es necesario valorar de manera más detallada cada una de ellas, enfocándose en el principal problema: el análisis de datos en español. De esta manera, el objetivo de este trabajo, será analizar las técnicas más relevantes en PLN por medio de librerías que cuenten con soporte para idioma español, experimentando con conjuntos de tweets seleccionados previamente, aplicándolos al análisis de sentimiento, y de esta forma evaluar el impacto de las técnicas aplicadas al texto por medio de parámetros como es la precisión, exhaustividad y rendimiento.

2. ESTADO DEL ARTE

Las investigaciones en línea, la expresión libre en redes sociales, y en la web han despertado el interés en la búsqueda de técnicas de extracción automática de información, debido a que son fuentes de datos en tiempo real. Estas investigaciones, se encaminan especialmente en el análisis de sentimiento, detección y clasificación; áreas que se encuentran ligadas a herramientas, algoritmos y librerías para el Procesamiento de Lenguaje Natural (PLN). PLN al ser un campo de análisis de texto y datos procesados que se transforman en conocimiento, su estudio es constante, tomando en cuenta sus diferentes niveles de análisis tanto como semántico(Altszyler & Brusco, 2015), léxico (Pérez-guadarramas, Rodríguez-blanco, & Simón-cuevas, 2017), sintáctico (Antonio, Velásquez, Paul, Paz, & Guzmán, 2017) y pragmático (Soto Kiewit, 2017) . Otros investigadores enfocan sus estudios en las técnicas aplicadas en PLN principalmente en la etapa de pre-procesamiento, etapa cuyo objetivo es la limpieza y el refinamiento de los datos, con el fin de tener resultados óptimos en el procesamiento, entre estas técnicas de pre-procesamiento se encuentra: *lowercase*, *tokenizer* (He & Kayaalp, 2006), *stemming* (Alami, Meknassi, Ouatik, & Ennahnahi, 2017), *lemmatizer* (Singh & Kumari, 2016) *y stop-word removal* (Katariya & Chaudhari, 2015) como las más importantes, sin embargo, en el trabajo de Gupta & Joshi (2017) exploran técnicas más específicas aplicadas en texto procedente de la red social Twitter, éstas igualmente se aplican en la etapa de pre-procesamiento y comprenden dos segmentos: eliminación de ruido en el texto y normalización del texto por medio de la conversión de palabras no estándar a sus formas canónicas. Este trabajo, aplicó su experimentación en tweets en el idioma inglés haciendo uso de la librería NLTK. Dando una aplicación más específica Jianqiang y Xiaolin (2017) en su investigación comparativa de técnicas de pre-procesamiento de texto enfocada al análisis de opinión en Twitter, evaluaron el efecto de seis tipos de técnicas, entre ellas: eliminación de enlaces web, *stop-word removal*, sustitución de palabras negativas, eliminación de números, supresión de letras repetidas y ampliación de acrónimos a sus palabras originales. Estas técnicas fueron aplicadas con el fin de evaluar su efecto en el rendimiento de clasificación de sentimientos con dos tipos de métodos y experimentando con cinco conjuntos de datos diferentes en idioma inglés. Los experimentos mostraron que la precisión y el valor-F, parámetros evaluados del clasificador de sentimiento de Twitter, mejoran cuando se usan los métodos de pre-

procesamiento, para expandir acrónimos y reemplazar la negación, pero tienen un cambio menor si se eliminan los enlaces web y los números, o se aplica *stop-word removal.*

Otros idiomas con estructura compleja como el árabe también son estudiados en el área de PLN. El lenguaje árabe es considerado complejo cuando se aplica en resúmenes automáticos de texto y recuperación de información. Es un lenguaje altamente flexible, con palabras variables y morfología compleja, razón por la cual la aplicación de técnicas como stemming son las más idóneas para este lenguaje, al permitir reducir la longitud del texto y buscar información con rapidez (Alami, Meknassi, Ouatik, & Ennahnahi, 2017). Investigaciones como Galadanci, Muaz, & Mukhtar (2016) también destacan la importancia de las técnicas de procesamiento de lenguaje natural aplicadas al lenguaje árabe, experimentando con dos conjuntos de datos en los cuales se aplicó tres tipos de algoritmos de clasificación: *SVM, Naive Bayes, K-nearest.* En cuanto a técnicas se usaron correlación de características, *stemming* y n-grams. Sus resultados mostraron que la selección de técnicas de pre-procesamiento en las revisiones aumenta el rendimiento de los clasificadores. Tomando en cuenta que solo se aplicó la experimentación en la herramienta RapidMiner.

Dentro de la ciencia de PLN, autores destacan librerías que sirven como herramientas a la hora de procesar y pre-procesar texto, como *FreeLing* o *SpaCy*. *FreeLing* es considerada una de las librerías más potentes, al construir sistemas de minería de opiniones muy robustos, además de proporcionar funcionalidades de análisis del lenguaje para diferentes idiomas (inglés, español, portugués, italiano, francés, alemán, ruso, catalán, etc.) (Henríquez, Guzmán, & Salcedo, 2016). Por otro lado, *SpaCy* es una librería enfocada al procesamiento avanzado del lenguaje natural, la cual está dirigida a aplicaciones comerciales, y según Omran & Treude (2017) en su investigación en el análisis de documentación de software, la consideran una de las más rápidas y exactas, al lograr la mejor precisión en cuanto a *tokenizer*, con 90% de precisión en comparación con otras.

3. FUNDAMENTACIÓN TEÓRICA

3.1. Procesamiento del lenguaje natural

Para hablar de Procesamiento de lenguaje natural (PLN), es necesario definir como primera instancia el concepto de lenguaje natural. De acuerdo al libro de Quesada Moreno & Amo (2000) un lenguaje natural es difícil de describirlo por el gran debate que presentan varios autores en donde basan sus estudios en teoría de conjuntos desde un punto de vista matemático y teorías de lenguaje formales; sin embargo define de forma general al lenguaje como "un conjunto de cadenas de símbolos sobre un alfabeto, denominado vocabulario", a su vez recalca que los lenguajes están clasificados o delimitados de acuerdo a carácterísticas formales de los conjuntos de cadenas de símbolos que se generen. El mismo autor destaca características comunes que puede presentar un lenguaje para ser considerado natural, entre las más destacadas está:

- Tamaño del léxico
- Estructuras de conocimiento
- Ambigüedad léxica
- Ambigüedad sintáctica
- Etc.

El lenguaje natural, precedido por la palabra procesamiento, establece otro estudio al ir más allá de un lenguaje y sus características. PLN está definido de diferentes maneras, como herramienta de detección de características lingüísticas (Narmadha & Sreeja, 2016) junto con la identificación de patrones que resultan en el lenguaje natural (Agogo & Hess, 2018); sin embargo un concepto más amplio de éste es el de ser un área investigativa y de aplicación que estudia cómo las computadoras se podrían utilizar para comprender y manejar texto en lenguaje natural (Vijayarani, Ilamathi, & Nithya, 2015).

Niveles de análisis de PLN

El libro Indurkhya & Damerau (2010) considera cuatro niveles fundamentales o componentes en el proceso de PLN; sin embargo, considera que no siempre se debe utilizar todos, esto depende del tipo de procesamiento que se quiere obtener del texto. Entre estos niveles está:

Análisis semántico. – Lingüísticamente este enfoque hace referencia al análisis del significado de palabras, oraciones completas y expresiones de acuerdo al contexto, buscando siempre la coherencia y sentido en el texto.

Análisis léxico. – Este tipo de análisis centra su objetivo en el elemento básico del texto, la palabra, comprendiendo su composición y origen o derivación, es decir se basa en un análisis morfológico, teniendo como principal tarea relacionar las variantes morfológicas, por ejemplo, la palabra comprando se deriva del verbo comprar que a su vez puede generar variantes como compraste, compraron, compra, etc.

Análisis pragmático. – El enfoque pragmático se centra en la relación entre las palabras y oraciones, y el contexto donde se utiliza, para determinar su significado en el texto, tomando como base el análisis semántico para evaluar cómo se está usando dicha sentencia o palabra, al poder interpretarse de forma metafórica, irónica o literal.

Análisis sintáctico. – El objetivo de este análisis es determinar estructuras gramaticales presentes en componentes léxicos, verificando su construcción de acuerdo a reglas gramaticales del lenguaje. El árbol sintáctico es la forma general o estándar de representar la estructura sintáctica gramatical, ésta es una representación de todos los caminos en los que se puede derivar una oración desde el nodo raíz. Esto significa que cada nodo interno representa una aplicación de una regla gramatical.

Aplicaciones

De acuerdo a su funcionalidad PLN se encuentran en varias disciplinas como: informática, lingüística, matemáticas, inteligencia artificial, robótica, psicología, etc. (Vijayarani et al., 2015). Además, es un campo con múltiples aplicaciones o técnicas orientadas a un fin específico, incluyendo entre sus principales:

Traducción automática. – tiene como objetivo la traducción de un lenguaje natural a otro.

Reconocimiento de entidades nombradas (NER). – extrae información del texto como nombres de personas, lugares, organizaciones entre otros, clasificándolos en categorías predefinidas (Narmadha & Sreeja, 2016).

Agrupación y recuperación de información. – su objetivo es tomar los datos textualmente y generar categorías que expresen el contenido del documento (Reese, 2015).

Etiquetado gramatical (POS). – permite el análisis sintáctico y léxico separando al texto en elementos gramaticales y etiquetándolos de acuerdo a su función como verbos, sustantivos, artículos, etc (Sun, Luo, & Chen, 2017).

Generación de lenguaje natural. – éste genera texto a partir de una fuente de conocimiento como una base de datos automatizando la creación de reportes de cualquier tipo (Reese, 2015).

Análisis de sentimiento. – es un campo que abarca diferentes áreas que utilizan técnicas de procesamiento de PLN, minería de texto y lingüística computacional, su objetivo es identificar y extraer la información subjetiva de textos (Galadanci, Muaz, & Mukhtar, 2016), con el fin de clasificarlo de acuerdo a diferentes criterios, para obtener la opinión y el sentimiento de las personas ante un tema específico.

3.2. Técnicas PLN en la etapa de pre-procesamiento

El procesamiento de lenguaje natural está compuesto por etapas, estas pueden variar de acuerdo al fin del estudio, como se indicó en la sección anterior por medio de las aplicaciones; sin embargo, en todo procesamiento existe la etapa de pre-procesamiento que se aplican en casi todos los campos. Esta etapa tiene como fin la limpieza y la normalización de los datos no estructurados y ruidosos, con el objetivo de prepararlos para un procesamiento más específico (Gupta & Joshi, 2017). Además, cuenta con técnicas que pueden ser utilizadas de acuerdo a criterios como el idioma, dominio del texto u aplicación del procesamiento. De acuerdo a investigaciones previas, dentro de las técnicas más comunes aplicadas en PLN se tiene:

Eliminación de signos de puntuación y caracteres especiales

Los signos de puntuación y caracteres especiales como repetición de letras en palabras, hashtags, signos de interrogación y exclamación, no intervienen en el procesamiento del texto, lo recomendable es su eliminación o sustitución por tokens únicos, ya que su espacio dentro del texto puede generar mayor tiempo de procesamiento. Esta técnica es usada frecuentemente en análisis de sentimiento, clasificación de texto y análisis NER, al no generar un aporte lingüístico en estos campos de procesamiento, reducir el número de palabras y realizar un análisis más simple. (Smailović, Kranjc, Grčar, Žnidaršič, & Mozetič, 2015).

Eliminación de enlaces web y menciones de usuarios

La presencia de enlaces web es más común en conjuntos de datos cuyo contenido proviene de redes sociales como los tweets, los cuales son utilizados principalmente en análisis de sentimiento y clasificación de texto, si bien estos enlaces no afectan el sentimiento de los mismos, esta técnica permite eliminar ruido del texto y reducir el número de palabras. Para la aplicación de esta técnica suele aplicarse previamente los tokens, posteriormente los enlaces que coinciden con los tokens se eliminan de los tweets para limpiar su contenido (Jianqiang & Xiaolin, 2017). En el caso de menciones de usuario presentes en tweets como: *"La muy talentosa y profesional @manuelae..."*, se aplica el mismo método de *tokenizer* en búsqueda de coincidencias para ser eliminados y refinar el texto.

Lowercase

Lowercase es una de las técnicas más básicas dentro del pre-procesamiento y en ocasiones es obviada por su simplicidad. Esta técnica consiste en la transformación de todo el texto que se encuentra en mayúsculas a minúsculas, fusionando las palabras y disminuyendo la dimensionalidad del texto (Symeonidis, Effrosynidis, & Arampatzis, 2018). Es mayormente utilizado en la clasificación y análisis de sentimiento, campos que intentan establecer categorías sin centrar su objetivo en un análisis sintáctico o léxico como en el caso del etiquetado POS.

Ejemplo de aplicación:

Tweet original:

"*Mbappé tuvo un Mundial Rusia bárbaro Goles desequilibrios velocidad fútbol Para mí el Mejor del mundial"*

Tweet aplicado *lowercase*

"*mbappé tuvo un mundial rusia bárbaro goles desequilibrios velocidad fútbol para mí el mejor del mundial"*

El ejemplo presentado muestra como la palabra "mundial" es escrita de dos formas (Mundial, mundial), lo que produciría una mayor dimensionalidad del texto al ser consideradas palabras diferentes.

Tokenizer

Ésta es una de las técnicas básicas, cuyo objetivo es la división del texto en palabras o frases significativas, delimitándolas por medio de caracteres especiales como puntos, comas o espacios en blanco (Uysal & Gunal, 2014). La elección del delimitador en su mayoría son los espacios en blanco; sin embargo, es conveniente la configuración o establecimiento de parámetros de acuerdo a la herramienta de *tokenizer* utilizada en caso de requerir eliminación de signos de puntuación, consideraciones de palabras juntas u otros aspectos. La aplicación de esta técnica es común es todos los campos del procesamiento, desde la traducción del lenguaje natural hasta el análisis de sentimiento, su uso se debe particularmente a facilitar todo tipo de estudio al generar conjuntos de palabras que pueden ser analizadas de forma individual como el caso del análisis POS (Etiquetado POS).

Ejemplo de aplicación:

Tweet original:

"*Mbappé tuvo un Mundial Rusia bárbaro Goles desequilibrios velocidad fútbol Para mí el Mejor del mundial"*

Tweet aplicado *tokenizer* y análisis POS (etiqueta gramatical: [S] sustantivo)

["M*bappé"[S],"tuvo"[V],"un"[AR],"Mundial"[NOM],"Rusia"[S],"bárbaro"[ADJ],"goles"[S],"desequilibrios"[NOM],"velocidad"[NOM],"fútbol[NOM]", "para", "mí"[AD.POS], "el"[AR], "mejor"[ADJ], "del"[ART], "mundial"[NOM]]*

Stop-Word Removal

Una palabra de parada (*stop-word*) es una palabra de uso común (“un”, “la”, “en”) que es considerada menos importante dentro del texto, o que no aporta un significado preciso a éste, por lo cual un motor de búsqueda es programado para ignorarla, ya que ocupa un espacio en el conjunto de datos y requiere tiempo de procesamiento, lo que puede producir una baja eficiencia. La técnica *stop-word removal* tiene como objetivo la eliminación de estas palabras por medio de cuatro métodos:

Método clásico. – método usado para eliminar las palabras vacías o de parada mediante listas o diccionarios pre-compilados.

Métodos basados en la Ley de Zip. – utiliza tres métodos de creación de *stop-word.* Los métodos incluyen eliminar palabras más frecuentes, eliminar palabras de baja frecuencia, y eliminar palabras simples.

Método de Información Mutua (MI). – es un método supervisado, el cual utiliza la comprobación de información entre un término dado y el tipo de documento (Ejm. positivo, negativo) facilitando una solución de cuánta información el término puede generar sobre una clase del documento. La baja información apunta que el término tiene un bajo poder de distinción y consecuentemente se elimina.

Muestreo aleatorio basado en términos (TBRS). – este método propone la detección de palabras vacías manualmente por medio de documentos web. Su técnica es la iteración sobre fracciones de datos separados al azar, y la clasificación de los términos en cada fracción en función de sus valores utilizando la medida de divergencia de Kullback-Leibler como se muestra en la siguiente ecuación (Saif, He, & Alani, 2017):

$$d_x(t) = Px(t).log_2 Px(t)/P(t)$$

dónde:

Px(t) es la frecuencia normalizada de un término *t* dentro de un conjunto *x, y*

P (t) es la frecuencia normalizada de t en toda la colección.

Lemmatizer y Stemming

La técnica *lemmatizer* o lematización en español tiene como objetivo unificar los términos que aportan la misma información de un texto, reemplazando cada uno por su lema. El lema de una palabra es un término que por convención se acepta como representante de todas las formas de una palabra, y para encontrarlo, se eliminan todas sus flexiones (conjugaciones, género, número, etc) (Dubiau & Ale, 2013). Dentro de aplicaciones PLN esta técnica reduce la cantidad de términos de un texto que se analizan más tarde (Guzman & Maalej, 2014). La lematización necesita soporte adicional de diccionarios para buscar e indexar, lo cual mejora su precisión en aplicaciones de extracción de características (Asghar, Khan, Ahmad, & Kundi, 2014).

Por otra parte el stemming, al igual que la lematización se involucra en el análisis a nivel morfológico dentro de PLN, según Díaz Gómez (2001) la lematización y stemming llegan a ser la misma técnica; sin embargo, son consideradas diferentes por una variante, mientras la lematización busca encontrar la palabra raíz de un término generalmente por diccionarios, el *stemming* intenta eliminar o cortar automáticamente los sufijos y prefijos de los términos dentro del texto, su fin es el mismo que la lematización, es decir, busca obtener un mínimo de caracteres y el máximo de información de ésta, de tal manera se puede relacionar una palabra con otra de su misma familia. Esta técnica es muy útil en trabajos de recuperación de información (RI). Sin embargo, según Krovetz (1993), citado por Díaz Gómez (2001) la lematización y el stemming pierden sentido en algunas ocasiones, por ejemplo en el caso de la primera técnica, si se encuentra el término *"*Estado*",* que se refiere a un país y éste se transforma a "estar" como un verbo, se tendrían bajo un término conceptos distintos, cayendo en un problema que conlleva a analizar el contexto en el que surge este término. En el mismo libro Díaz Gómez (2001) describe algunos de los principales algoritmos de lematización que caen dentro del concepto de stemming, estos algoritmos integran dos grandes grupos, manuales y automáticos (Ver figura 1); sin embargo solo se describirán los algoritmos automáticos primordiales en trabajados en PLN, y entre ellos están:

Eliminación de afijos. - elimina afijos y/o sufijos de los términos para obtener su lema. Su implementación es iterativa, usando algoritmos de mayor coincidencia como Porter, el algoritmo más conocido y usado, aplica de 30 a 40 reglas sobre los términos; sin embargo, algunas de sus desventajas son que solo elimina sufijos en su mayoría y el

conjunto de reglas define la calidad de la lematización, por otro lado, ante nuevas palabras, la obtención de su lema es más sencillo.

Sucesor de variedad. – su algoritmo se basa en lingüística estructural, determinando los límites en los términos y los morfemas de una cadena. La variedad de sucesores de una cadena se define como el número de caracteres diferentes que siguen a ésta de acuerdo a un corpus. Calculado la variedad de sucesores se segmenta la palabra por medio de los siguientes métodos:

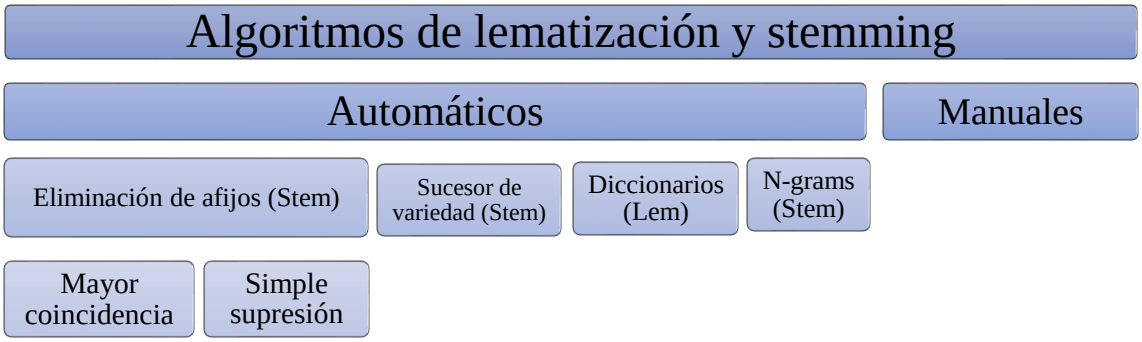
Método del valor de corte: se determina un valor de corte para las variedades de sucesores y un límite cada vez que se alcance el valor de corte.

Método de los picos y valles: se corta un segmento cuando se localiza un carácter cuya variedad de sucesores supera aquella del carácter que lo antepone.

Método de palabra completa: si éste es una palabra completa en el cuerpo de texto se hace el corte de un segmento.

Tabla de búsquedas o Diccionarios. – método que agrupa en una tabla los términos y sus lemas, éstos pueden ser buscados en dicha tabla para aplicarse la lematización. Para la búsqueda se utiliza el algoritmo de árbol-b o una dispersión. Es uno de los métodos más sencillos y rápidos de usar; sin embargo, requiere diccionarios de cada idioma para proporcionar ese tipo de análisis entre término y el lema correspondiente.

Figura 1
Algoritmos de lematización



Fuente: Elaboración propia

Etiquetado POS

Esta técnica lleva su nombre por su definición en inglés, *Part-Of-Speech Tagging*, además es también conocida como etiquetado gramatical, su objetivo es asignar a cada palabra de un texto una categoría gramatical (sustantivo, adjetivo, verbo, etc). Su proceso se realiza de dos maneras, conforme a la definición de la palabra o de acuerdo al contexto en el texto (Torres López, Sánchez-Sánchez, & Villatoro-Tello, 2014).

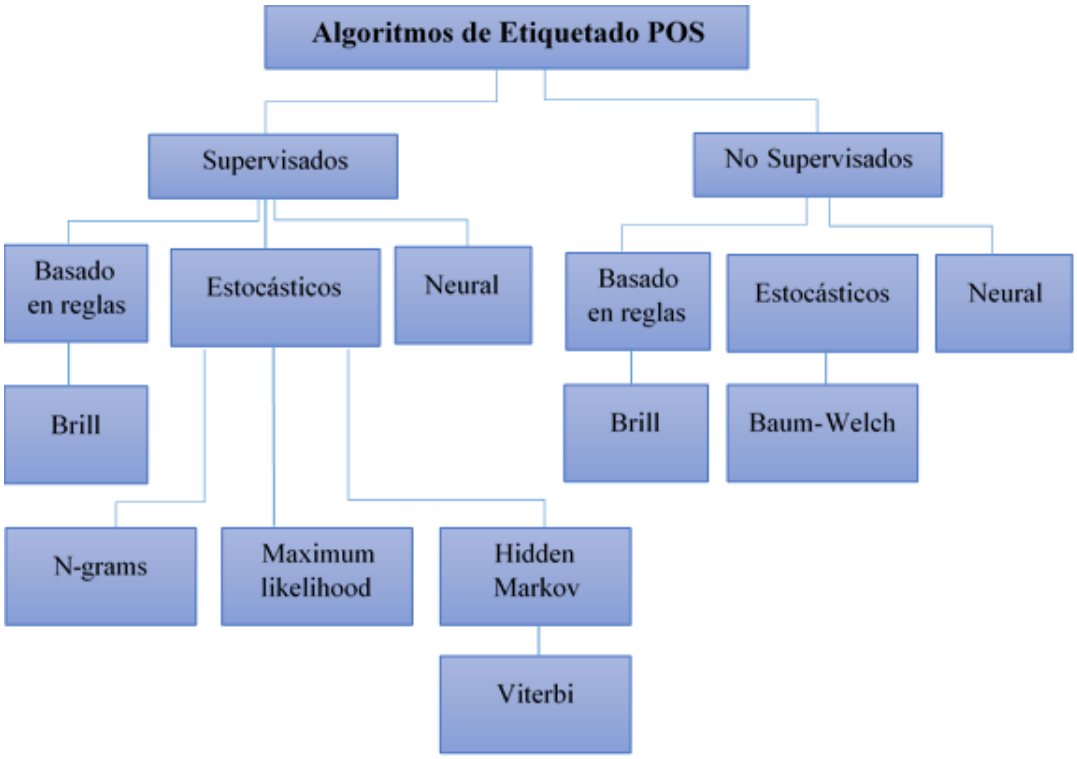
El etiquetado gramatical muestra mayor información sobre una palabra, dependiendo del tipo de problema que se quiera resolver, ya sea recuperación de información, clasificación, generación de lenguaje, etc, tomando en cuentas que existen investigaciones (Pang, Lee, & Vaithyanathan, 2002) (Go, Bhayani, & Huang, 2009) que demuestran que su uso no es útil cuando se aplica en SVM para el análisis de sentimiento. Por otro lado, realizar este tipo de etiquetado no es fácil, debido a que muchos etiquetadores pueden confundir etiquetas como un nombre con un verbo, por ejemplo, la palabra "dado" depende del contexto, ya que puede entenderse como el verbo "dar*"* o el objeto "dado". Esta técnica a su vez se basa en un análisis sintáctico, es decir utiliza un árbol gramatical para el proceso de etiquetado, además, de algoritmos de acuerdo a diversos enfoques que abarcan dos grandes grupos, algoritmos supervisados y no supervisados, los cuales son descritos a continuación:

Supervisados. – este modelo requiere un corpus previo, el cual es utilizado como entrenamiento para obtener información sobre el conjunto de etiquetas, conjuntos de reglas, frecuencia de etiquetas, diccionario del etiquetador, etc. Su rendimiento generalmente aumenta proporcionalmente con el tamaño del corpus (Hasan, UzZaman, & Khan, 2007).

No supervisados. – este modelo no requiere un corpus previo, usan técnicas computacionales avanzadas para inducir automáticamente conjuntos de etiquetas y reglas de transformación. A través de esta búsqueda, calculan la información probabilística para etiquetadores estocásticos o para inducir las reglas contextuales que necesitan los sistemas basados en reglas o transformación (Hasan et al., 2007).

Los modelos supervisados y no supervisados tienen una clasificación más amplia, la cual se puede observar en la Figura 2.

Figura 2
Algoritmos de Etiquetado POS



Fuente: Elaboración propia

3.3. Librerías para PLN

NLTK

Natural Language Toolkit (NLTK) es una plataforma líder de código abierto, escrito en el lenguaje de programación Python, siendo una de las más estables para trabajar con textos en estado natural. Proporciona interfaces fáciles de usar con más de 50 corpus y recursos léxicos con datos de entrenamiento para sus algoritmos (NLTK, 2018). Una de sus características, además de desventaja, es que no posee un soporte multilenguaje; sin embargo, puede ser entrenado a partir de sus corpus en distintos idiomas.

Entre las principales herramientas y recurso léxicos que cuenta NLTK están:

- *Tokenizer* mediante expresiones regulares.
- Etiquetado morfo sintáctico (*Penn Treeback*).
- Análisis morfo sintáctico (Corpus ACE)
- Reconocimiento de entidades nombradas (NER).

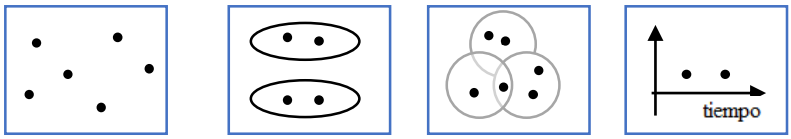
- WordNet.
- Etiquetado de texto.

Corpus en la librería NLTK

Un corpus puede definirse como una colección de textos representativos de un idioma, dialecto u otro subconjunto de un idioma, y es normalmente utilizado para análisis lingüístico (Svartvik, 1992).

Dentro de la librería NLTK y otras, existen un sin número de estructuras de corpus. En NLTK se diferencian cuatro tipos de estructuras, la primera es considerada la más simple al carecer de una estructura en sí, ésta es solo una colección de textos. La segunda agrupa los textos en categorías correspondientes a un género, fuente, idioma, etc. A continuación, se tienen las estructuras superpuestas, éstas se generan a raíz de categorías tópicas, es decir, cuando un texto es relevante en más de un tema. Por último, están las colecciones de texto que usan una estructura temporal, un ejemplo de ellas son las noticias (Ver figura 3).

Figura 3
Estructura de Corpus



Simple Categorías Superpuestos Temporal

Fuente: Elaboración Propia

Módulos y técnicas de pre-procesamiento para lenguaje español.

NLTK proporciona módulos de uso genérico para procesamiento del lenguaje, hasta módulos específicos como tokenización en twitter, con soporte para idioma inglés y español. Entre sus técnicas y módulos se tiene:

a) Tokenizer

Existen tres módulos utilizados para este tipo de técnica:

1. ***Sent_tokenizer.* –** opera a nivel de oraciones. Devuelve el texto con oraciones tokenizadas, utiliza el tokenizador de oraciones PunktSentenceTokenizer para el idioma especificado. Sus parámetros son: *nltk. tokenize. sent_tokenize (text, language='spanish')*

   text: texto a dividir.
   language: el nombre del idioma o modelo en el corpus Punk.

2. ***Word_tokenizer*. –** opera a nivel de términos. Devuelve los *tokens* encontrados en el texto, utilizando el *tokenizer* de palabras *TreebankWordTokenizer* junto con *PunktSentenceTokenizer* del idioma especificado. Sus parámetros son: *nltk. tokenize. Word_tokenize (text, language='spanish', preserve_line=False)*

   text: texto a dividir en palabras o tokens
   language: el nombre del idioma o modelo en el corpus Punk.
   preserve_line: opción que permite o no preservar la oración sin *tokens.*

3. ***TweetTokenize*r. -** éste es un tokenizador especial para texto extraído de Twitter, este texto conocido como tweets presenta características propias como contener nombres de usuario (@manueljt), enlaces web, entre otros términos. Mediante *tweet tokenizer* es posible hacer un pre-procesamiento al texto y eliminar dichas características si activamos los parámetros correctos, entre estos están:
   nltk. tokenize. Casual. TweetTokenizer (preserve_case=True, reduce_len=False, strip_handles=False)

   preserve_case: si no se actica (False) devuelve la cadena original
   reduce_len: normaliza la longitud de la palabra.
   strip_handles: elimina términos cuyas características cumplan con características de nombres de usuario de Twitter.

b) Stop Words Removal

Para este proceso NLTK utiliza su corpus en español de *stop-word*, donde se encuentran palabras vacías como: en, la, lo, las, un, una, etc. Para ver el corpus con el que se trabaja se utiliza las siguientes sentencias:

```
>>>import nltk
>>>from nltk. corpus import stopwords
>>>set (stopwords. words('spanish'))
```

Salida: muestra el corpus actual de palabras vacías de la librería

{'estaban', 'fuera', 'está', 'no', 'estado', 'tuviese', 'vuestras', 'hubieron', 'eran', 'fueron', 'estábamos', 'teniendo', 'nuestra', 'estuviste', 'al', 'qué', 'otro', 'habrás', 'e', 'tendríais', 'quien', 'estadas', 'serías', 'mucho', 'estuvimos', 'estad', 'vosotros', 'tuyas', 'tenemos', 'tuvieras', 'en', 'eso', 'hasta', 'seamos', 'tenían', 'habría', 'mi', 'tengas', 'ti', 'habré', 'sentida', 'tendrías', 'uno', 'hubo', 'estaríais', 'ante', 'nuestras', 'os', 'este', 'estamos', 'durante', 'seáis', 'estaremos', 'una', 'ya', 'tanto', 'su', 'tú', 'estuviesen', 'tienen', 'mío', 'tenida', 'habíamos', 'ella', 'con', 'hayan', 'tuvierais', 'tenía', 'habréis', 'serían', 'contra', 'tendrían', 'tu', 'también', 'tuyo', 'estuvo', 'son', 'nuestro', 'soy', 'tuviste', 'habríamos', 'un', 'tendrá', 'estuviésemos', 'ese', 'algo', 'hubieseis', 'tenga', 'estás', 'erais', 'fuerais', 'cuando', 'que', 'hubiera', 'esas', 'es', 'tienes', 'se', 'hubiesen', 'sentidos', 'hubiésemos', 'esa', 'fui', 'algunos', 'hemos', 'estéis', 'has', 'lo', 'hayas', 'hubieras', 'la', 'las', 'él', 'muchos', 'fuese', 'fueseis', 'hay'...}

Esta librería cuenta con 313 palabras vacías en su corpus, además permite la adición de palabras que se consideren convenientes en el análisis o proceso que se lleve a cabo.

c) Stemming y Lemmatizer

La librería NLTK cuenta con tres módulos principales dentro de las técnicas *Stemming* y *Lemmatizer*, entre estos módulos tenemos: *WordNetLemmatizer*, *SnowballStemmer* y *PorterStemmer*; sin embargo, solo *SnowBall Stemmer* cuenta con soporte para lenguaje español, aunque su diccionario y algoritmo aún es muy pobre y deficiente para este tipo de procesamiento. El algoritmo *Snowball* está dirigido a la eliminación de sufijos y afijos de palabras, técnica que es poco eficiente para el idioma español, ya que corta las palabras sin tomar en cuenta en su mayoría que en este lenguaje la morfología de las palabras es muy variada. Un ejemplo de la ineficiencia de este módulo aplicado al idioma español es presentado a continuación.

```
>>> from nltk.stem import SnowballStemmer
>>> snowball_stemmer = SnowballStemmer('spanish')
>>> snowball_stemmer.stem('cantaron')
```

Salida: mediante la técnica de *stemmer* para español, el algoritmo corta la palabra de acuerdo a su diccionario de sufijos; sin embargo; la palabra pierde sentido.

```
>>>'cant'
```

Para el caso del idioma inglés este algoritmo arroja mejores resultados, ya que la morfología y las reglas gramaticales de conjugación son menos complejas y comunes para muchos de los casos.

>>> from nltk.stem import SnowballStemmer
>>> snowball_stemmer = SnowballStemmer('english')
>>> snowball_stemmer.stem('playing')

Salida: mediante la técnica de *stemmer* para inglés, el algoritmo corta la palabra; pero para este idioma sus reglas están mejor definidas, logrando tener la palabra derivada correcta.

>>>'play'

FREELING

FreeLing es una librería de código abierto, escrita en C ++ que proporciona funcionalidades de análisis de lenguaje (análisis morfológico, etiquetado PoS, análisis sintáctico, análisis NER, etc.) para varios idiomas, siendo el más completo en módulos el idioma catalán. Además, proporciona un *front-end* de línea de comandos para analizar textos y lograr resultados en diferentes formatos (XML, JSON, CoNLL) (FreeLing, 2018).

Los principales servicios en FreeLing son:

- Tokenización de texto
- Análisis morfológico
- Tratamiento sufijo.
- Reconocimiento de palabras compuestas
- Análisis NER
- Reconocimiento de fechas, números, razones, moneda y magnitudes físicas (velocidad, peso, temperatura, densidad, etc.)
- Etiquetado PoS
- Extracción del gráfico semántico, etc.

Diccionarios o corpus en la librería FreeLing

FreeLing cuenta con una diversidad de diccionarios para múltiples idiomas, uno de los diccionarios principales en todos sus idiomas es el morfológico. Estos diccionarios

provienen de diferentes proyectos externos de código abierto, teniendo diferentes titulares y licencias en comparación de los paquetes de FreeLing (FreeLing, 2018).

Si bien los diccionarios para ciertos idiomas latinos parecen pequeños, se destaca que la librería permite el reconocimiento de muchas más formas que las del diccionario, gracias a su potente módulo de análisis de afijos y sufijos. Además, cuenta con un estimador basado en sufijos probabilísticos de categorías de palabras que no se encuentran en los diccionarios (FreeLing, 2018). Entre sus diccionarios más destacados se encuentran:

- El diccionario catalán con más de 520,000 formas correspondientes a 71,000 combinaciones de lemas.
- El diccionario inglés proviene de WSJ y otros corpus, con ciertas ediciones, éste contiene cerca de 68,000 formas correspondientes a unas 37,000 combinaciones diferentes de lemas.
- El diccionario español contiene más de 555,000 formas correspondientes a más de 76,000 combinaciones de lemas.
- El diccionario portugués contiene cerca de 909,000 formas correspondientes a 105,000 combinaciones de lemas.

Módulos y técnicas de pre-procesamiento para lenguaje español.

FreeLing es considerada una de las librerías más estables por sus módulos basados en el análisis morfológico, sintáctico y semántico de texto en idioma inglés, español, portugués, francés, italiano, alemán, ruso, catalán, entre otros. Algunos de sus módulos utilizados en lenguaje español son:

a) Módulo identificador de idioma

Este módulo no enriquece el texto dado con algún tipo de nivel de análisis, su función es comparar el texto de entrada con los modelos disponibles en la librería para diferentes idiomas, devolviendo el idioma más probable del texto. Regularmente es usado como un pre-proceso para determinar qué archivos de datos se usarán (FreeLing, 2018).

b) Módulo Tokenizer

Este módulo tiene como función principal la conversión de texto plano en un arreglo de objetos de palabras, por medio de un conjunto de reglas de Tokenización. Estas reglas

son expresiones regulares que coinciden con el inicio de la línea de texto. La primera regla de coincidencia se utiliza para obtener el token, la subcadena de coincidencia se elimina y el proceso se convierte en un bucle hasta que la línea esté vacía (FreeLing, 2018).

c) Módulo analizador morfológico

El analizador morfológico es un meta-módulo que no realiza ningún procesamiento propio. Es solo un módulo de conveniencia para simplificar la creación de instancias y llamar a los submódulos descritos en las siguientes secciones. En el momento de la instanciación, recibe un *maco_options* objeto que contiene información sobre qué submódulos deben crearse y qué archivos se deben usar para crearlos.

Cualquier submódulo cargado en el momento de instanciación, puede desactivarse y reactivarse más tarde utilizando el método *macro: set_active_options* descrito anteriormente. Se debe tener en cuenta que, si un submódulo no se cargó al crear una macro instancia, no se puede cargar ni activar más tarde. Una aplicación de llamada puede omitir este módulo y simplemente llamar directamente a los submódulos que son:

- Detección de puntuación.
- Detección de número.
- Módulo de mapa de usuario.
- Detección de fechas.
- Búsqueda de diccionario.
- Reconocimiento de palabras múltiples.
- Reconocimiento de entidad nombrada.
- Cantidad de reconocimiento.

SPACY

SpaCy es una librería de código abierto dedicada al procesamiento avanzado del lenguaje natural (PLN) en Python. Está diseñada específicamente para uso en producción lo que ayuda a crear aplicaciones que procesan y "entienden" grandes volúmenes de texto. Generalmente, es usada para construir sistemas de extracción de información o de comprensión del lenguaje natural o para pre-procesamiento de texto para aprendizaje

profundo. Cuenta con el analizador sintáctico más rápido, modelos de redes neuronales para etiquetado, y reconocimiento de entidad nombrada, para diferentes idiomas (spaCy, 2018).

Diccionarios o corpus en la librería SpaCy

SpaCy utiliza conjuntos de datos de entrenamiento para sus módulos o modelos, estos conjuntos incluyen todo el corpus de texto de Wikipedia y Google News; sin embargo, Wikipedia no cuenta con datos en aspectos particulares como doctrinas legales o religiosas, por lo que, si la aplicación de los modelos se centra en un dominio como estos, sus resultados pueden no ser óptimos.

La librería SpaCy además de corpus integrados para cada modelo, cuenta con vocabularios o diccionarios propios entrenados. Además, SpaCy intenta almacenar datos en un vocabulario común denominado *Vocab*. Éste es compartido por varios documentos para ahorrar memoria. SpaCy codifica todas las cadenas a valores hash; por ejemplo, "café" tiene el hash 3197928453018144401. Las etiquetas de entidad como "ORG" y etiquetas de parte de la voz como también están codificadas. Internamente, spaCy solo trabaja con los valores hash (spaCy, 2018).

Modelos y técnicas para idioma español

SpaCy presenta modelos neuronales para análisis sintáctico, reconocimiento de entidades y etiquetado. Los modelos fueron diseñados e implementado desde cero específicamente para SpaCy, brindando un equilibrio en cuanto a velocidad, tamaño y precisión. SpaCy cuenta con dos modelos para lenguaje español:

> ***es_core_news_sm* y *es_core_news_sm*. -** son modelos multi-tarea CNN de España que se crearon en el corpus AnCora y WikiNER. Su función es asignar vectores token específicos del contexto, etiquetas POS, análisis de dependencias y entidades con nombre. Admite la identificación de entidades PER, LOC, ORG y MISC.
> Los modelos están entrenados en Wikipedia, por lo que su funcionalidad puede ser inconsistente en muchos géneros, como en texto redes sociales.

Estos modelos además ofrecen técnicas de pre-procesamiento como *tokenizer*, *lowercase*, *lemmatizer* y *stop-word removal*, con las siguientes características.

Tokenizer

El algoritmo de Tokenización proporciona un mejor equilibrio entre el rendimiento, la facilidad de definición y la facilidad de alineación en el texto original. Los datos del tokenizador global y específico del idioma se suministran a través de los datos del idioma. Se presentan casos especiales como presencia de prefijos y sufijos, para ello se definen reglas de puntuación, por ejemplo, cuándo dividir términos y cuándo dejar intactos los tokens que contienen reglas o términos especiales como abreviaturas.

Algoritmo general de *tokenizer:*

1. Iteraciones sobre subcadenas separadas por espacios.
2. Comprobar si existe una regla definida explícitamente para la subcadena.
3. Si no existe una regla específica se intenta eliminar prefijos.
4. Si se eliminar un prefijo, regresa al principio del ciclo, para que los casos especiales siempre tengan prioridad.
5. Si no elimina un prefijo, intenta eliminar un sufijo.
6. Una vez que no podamos eliminar más de la cadena, se trata como un único token.

Lowercase

SpaCy cuenta con diferentes parámetros de transformación e identificación de palabras minúsculas como:

Tabla 1
Parámetros de transformación a minúscula

Nombre	Tipo	Descripción
lower	int	Forma en minúscula de la ficha.
lower_	unicode	Forma en minúsculas del texto del tokenEquivalente a Token.text.lower().
is_lower	booleano	¿Está el token en minúscula? Equivalente a token.text.islower().

Lemmatizer

SpaCy aplica reglas de lematización o una tabla basada en búsqueda para asignar formularios base, por ejemplo, para el verbo "ser"→"fue".

Este tipo de algoritmo se aplica a partir de la versión v2.0, spaCy, siendo la forma más rápida y fácil la búsqueda. Los datos se almacenan en un diccionario asignando un término a su lema. Para determinar el lema de un token, spaCy simplemente aplica una búsqueda en la tabla.

Stop-Word Removal

SpaCy maneja un corpus de palabras vacías. Para mejorar la legibilidad de su corpus, la librería los separa por espacios y líneas nuevas que se agregan como una cadena multilínea. SpaCy cuenta con un corpus de 551 palabras vacías en idioma español, que además puede ser ampliado de acuerdo a las necesidades que se tenga, algunas de ellas se muestran a continuación:

{'dias', 'varias', 'ese', 'consideró', 'conseguir', 'los', 'bajo', 'señaló', 'mias', 'sin', 'trabajamos', 'estados', 'hacemos', 'mio', 'agregó', 'menudo', 'algunas', 'nadie', 'tiempo', 'usa', 'ninguna', 'hacen', 'pudo', 'mientras', 'dicho', 'siete', 'último', 'hubo', 'adrede', 'hacer', 'temprano', 'las', 'cuánto', 'podeis', 'tan', 'día', 'vuestras', 'alguna', 'ésa', 'momento', 'estan', 'podrán', 'nuestro', 'cosas', 'habia', 'había', 'próximo', 'últimas', 'manera', 'grandes', 'ya', 'podriais', 'estado', 'yo', 'encuentra', 'explicó', 'nos', 'además', 'de', 'sea', 'mal', 'hecho', 'adelante', 'aunque', 'algún', 'ayer', 'sí', 'partir', 'ejemplo', 'veces', 'intento', 'tenía', 'alrededor', 'misma', 'tener', 'habrá', 'allí', 'nuevo', 'seis', 'empleo', 'cuantos', 'quizás', 'usar', 'usted', 'sois', 'claro', 'haces', 'soy', 'siendo', 'te', 'intentamos', 'tuyo', 'consigue', 'hemos', 'uno', 'días', 'primer', 'creo', 'indicó', 'entre', 'estos', 'pocas', 'decir', 'despacio', 'lejos', 'cerca', 'ser', 'siguiente', 'aquella', 'ciertas'...}

3.4. Análisis de sentimiento

El análisis del sentimiento es un campo interdisciplinario que recurre a técnicas de PLN, lingüística computacional y minería de texto con el fin de identificar y extraer información de fuentes de datos como redes sociales, noticias, mensajes, etc. Su objetivo es analizar las emociones y opiniones, reflejadas en el texto y clasificarlas de acuerdo a clases o también conocidas como polaridades que pueden ser negativa y positiva. A las polaridades positiva y negativa puede sumarse la neutra, la cual indica que el texto no tiene ningún aporte en cuanto a emociones. La importancia de este campo radica en la retroalimentación que se da gracias a usuarios de productos, películas, servicios, etc, mediante la recolección de información en tiempo real que se genera en redes sociales,

aportando a empresas de todo tipo a conseguir datos que se convierten en fuente de conocimiento, lo que implica mayores ingresos al proveedor. Esto demuestra que las opiniones se consideran una fuente de información muy valiosa. El sentimiento encontrado en estas fuentes puede expresarse en cuatro niveles diferentes ,fuertemente interconectados (Galadanci et al., 2016):

Nivel 0. Clasificación del sentimiento del documento: Es uno de los problemas más estudiados. Su objetivo es clasificar en positivo o negativo las opiniones recolectadas. En algunos casos, se considera la clase neutral. Toda oración o documento se asigna a las clases positiva, negativa o neutral. Tomando en cuenta que existen características muy específicas como la presencia de palabras de negación (“no”, “nunca”, “jamás”, etc), que pueden cambiar la dirección del sentimiento, es necesario realizar un análisis más profundo.

Nivel 1. Subjetividad de la oración y la clasificación del sentimiento: el objetivo es identificar oraciones subjetivas de las objetivas y clasificarlas en oraciones positivas o negativas; es decir, cuando tienen un mensaje que puede interpretarse con las dos polaridades.

Nivel 2. Análisis de sentimiento basado en el aspecto: el análisis del sentimiento toma un nivel más complejo en documentos y oraciones, al carecer en ocasiones de la capacidad de demostrar lo que a la gente le gusta o no; es decir, no encuentran los objetivos de las opiniones extraídas.

Nivel 3. Resumen de opinión basado en aspectos: Este nivel considera todas las opiniones sobre cada aspecto en los documentos u oraciones, intentando resumir los comentarios positivos o negativos. Por lo general, infiere y extrae información en porcentajes: "al 70% de las personas les gustó la película".

Algoritmos de clasificación en análisis de sentimiento

Los algoritmos de clasificación de texto se dividen en dos grupos de acuerdo a la categoría de aprendizaje automático, los algoritmos de aprendizaje automático supervisado que se caracterizan por tener un corpus manualmente clasificado, en donde el algoritmo lleva a cabo dos procesos: encontrar los mejores parámetros para el algoritmo, y evaluar el nivel de fiabilidad con esos parámetros, esta fase es llamada

entrenamiento (Baviera, 2016). Y los algoritmos de aprendizaje automático no supervisado, en donde no se encuentra un corpus previamente etiquetado o un resultado conocido. Su objetivo es deducir las estructuras presentes en los datos de entrada, a través de un proceso matemático para comprimir la redundancia.

Algoritmos de aprendizaje automático supervisados

Los algoritmos supervisados cuentan con 4 tipos de clasificadores: árboles de decisión, lineales, basados en reglas y probabilísticos. Dentro de estos clasificadores, los más utilizados son Naive Bayes y Support Vector Mahicne(SVM) RBF kernel.

Naive Bayes

Este algoritmo utiliza clasificación probabilística que requiere un pequeño conjunto de datos de entrenamiento para determinar la predicción de parámetros. Este clasificador utiliza la ecuación del teorema de Bayes descrita a continuación:

$$P\ (c \mid d) = P\ (d \mid c)\ P(c) / P(d)$$

d es la revisión y *c* es la clase o polaridad. Para una revisión textual dado *"d"* y para una clase *"c"* (positiva, negativa), la probabilidad condicional para cada clase dada de una revisión es *P (c | d).*

Este tipo de clasificador es regularmente utilizado en filtrado de correo basura o spam. Otra aplicación es la fusión de datos, combinando información indicada en términos de densidad de probabilidad procedente de sensores. Otras aplicaciones son:

- Diagnóstico de cáncer.
- Evaluación de probabilidades en el desarrollo de un juego.
- Probabilidades a priori y a posteriori.

Support Vector Machine RBF kernel

El clasificador SVM ha demostrado ser altamente efectivo en la categorización de texto tradicional, superando a Naive Bayes. Para los casos de dos categorías o clases, la idea básica del procedimiento de entrenamiento en SVM es encontrar un hiperplano, representado por el vector ~ w, que no solo separa los vectores del documento en una

clase y otra; si no intenta aplicar cuando la separación, o el margen de las clases, es lo más grande posible.

SVM es un clasificador basado en el modelo de espacio vectorial, es decir, requiere que los documentos de texto se transformen en vectores de características antes de ser clasificados. Regularmente los documentos de texto se transforman en vectores multidimensionales TF-IDF. El objetivo de la clasificación es categorizar cada documento de texto representado como un vector en una clase, aplicando el concepto de clasificación lineal, es decir toma una decisión al clasificar en base al valor de una combinación lineal de sus características. Sin embargo, el algoritmo SVM da la oportunidad de convertir algoritmos lineales (una sola dimensión) a versiones no lineales, esta función es conocida como métodos kernel como RBF. Este tipo de kernel surge por las limitaciones computacionales de las máquinas de clasificación lineal, que no son aplicadas comúnmente en la vida real. La clasificación por medio del kernel RBF da la solución a este problema, mostrando la información en un espacio de características de mayor dimensión, lo cual aumenta la capacidad computacional de las máquinas de clasificación lineal (Banados & Espinosa, 2014). En general, el kernel RBF es la opción más razonable en clasificación de texto. Este kernel mapea linealmente las muestras en un espacio dimensional más alto, a diferencia del kernel lineal que suele mostrar en una sola dimensión (Hsu, Chang, & Lin, 2010).

Existen dos parámetros importantes en un kernel RBF que son C y γ. Estos parámetros no pueden ser escogidos al azar y se desconoce cuál es el valor apropiado para cada caso; por lo tanto, es necesario realizar algún un procedimiento en el modelo, conocido como la "búsqueda de parámetros". El objetivo de este proceso es identificar el mejor valor para C y γ para que el clasificador pueda predecir datos desconocidos con precisión. La versión mejorada de este proceso de búsqueda se conoce como "validación cruzada". Esta validación divide el conjunto de entrenamiento en *v* subconjuntos de igual tamaño y secuencialmente se prueba un subconjunto usando el clasificador entrenado en el resto (*v -1 subconjuntos*). Cada instancia del conjunto de entrenamiento completo se predice una vez, por lo que la precisión de la validación cruzada es el porcentaje de datos que están clasificados correctamente (Chih-Wei Hsu, Chih-Chung Chang, 2016).

Investigaciones previas han demostrado que la optimización de valores de los parámetros del kernel RBF da mejores resultados en comparación con *Support Vector*

Machine lineal y el algoritmo de Naïve Bayes, teniendo resultados de clasificación de conjuntos de datos con más precisión (Jadav & Scholar, 2016).

3.5. Métricas de evaluación en algoritmos de clasificación

Las métricas de evaluación reconocidas en el campo de PLN y la clasificación de texto, son la precisión, exhaustividad (o *recall*) y *F_measure*. Estas métricas son calculadas en base a los valores de las clases asignadas de la matriz de confusión generada en el modelo de clasificación en el análisis de sentimiento: verdaderas positivas (TP), falsas positivas (FP), verdaderas negativas (TN) y falsas negativas (FN) (Haddi et al., 2013).

La precisión es una medida que certifica la exactitud de los métodos, y describe la cantidad de positivos verdaderos de todos los documentos asignados positivamente, y está dada por:

$$Precisión = \frac{TP}{TP+FP}$$

Recall es una medida más específica, mide la proporción de verdaderos positivos de los documentos positivos reales, y está dada por:

$$Precisión = \frac{TP}{TP+FN}$$

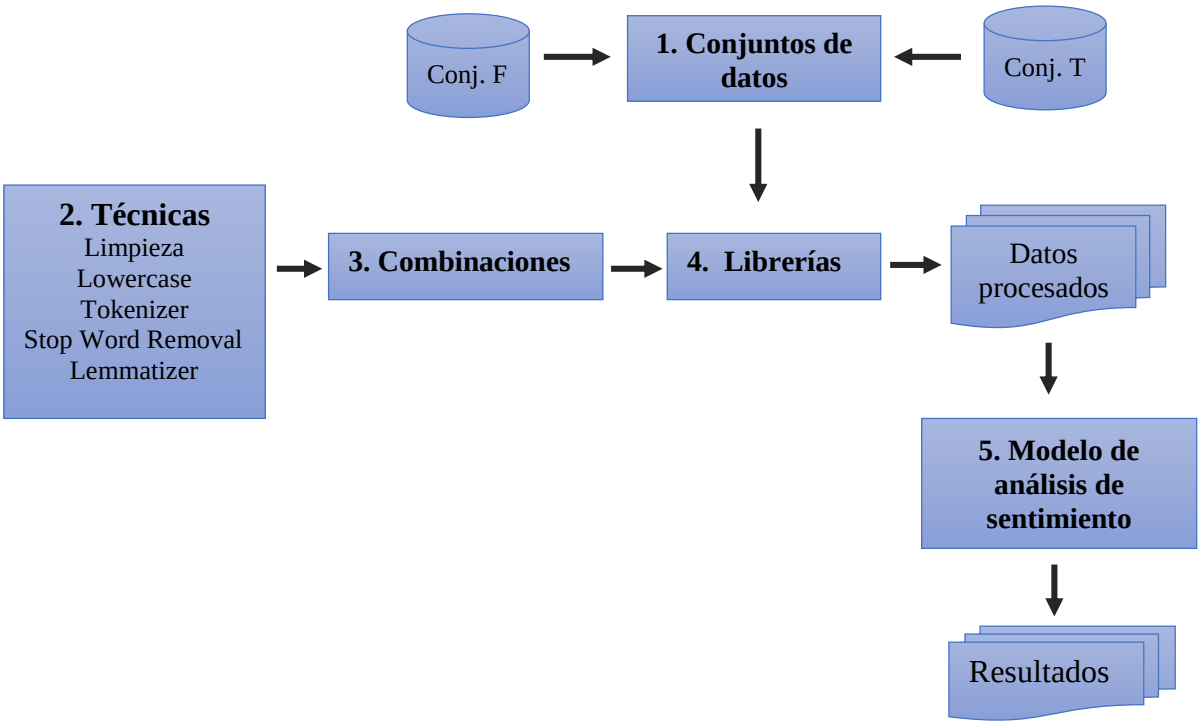
El rendimiento general se examina mediante el valor *F_measure*, que es la medida armónica entre la precisión y la exhaustividad. Este valor es aplicado en conjuntos de datos desbalanceados, es decir con una diferencia en cuanto a documentos en cada clase de gran proporción. Se calcula por:

$$F\_measure = \frac{2*precisión*recall}{precisión+recall}$$

4. MÉTODO Y MATERIALES

La metodología empleada se basa en cinco pasos, que han sido parte de la evaluación de técnicas PLN en la etapa de pre-procesamiento, entre estos se encuentran: la selección del conjunto de datos de experimentación como proceso de entrada, seguido del proceso de selección de técnicas. A continuación, se realizan las combinaciones posibles de técnicas, para seguir con la elección de librerías y sus respectivas combinaciones, por último, la evaluación de técnicas en el análisis de sentimiento; como método adicional se considera la selección y definición de los parámetros de evaluación aplicados a las técnicas y combinaciones de éstas en el proceso de análisis de sentimiento, y por último se describe una sección de materiales como softwares utilizados en diferentes procesos del trabajo investigativo. El proceso extendido de la experimentación se puede observar en la Figura 1, y se detalla a continuación.

Figura 4
Proceso extendido de experimentación



Fuente: Elaboración propia

4.1. Conjunto de datos.

Las técnicas de pre-procesamiento se evaluaron en dos dominios de tweets cerrados, Mundial de fútbol 2018 y Tránsito, para facilitar el entrenamiento previo de un segmento de datos. Estos tweets fueron extraídos por medio de la API de Twitter y el lenguaje de programación R.

Para una evaluación objetiva el número de tweets para cada dominio fue el mismo, 1,300 en total. El conjunto de datos del Mundial de fútbol 2018 (conjunto *F*) cuenta con tweets adquiridos de diferentes países como: Argentina, Colombia, España, México y Uruguay, tomados en tiempos diferentes, en los cuales se consideró mayor actividad de los usuarios en esta temática, como son: la semifinal y final. Posteriormente se clasificó de forma binomial, positivos y negativos, teniendo como resultado 912 positivos y 410 negativos. En el caso del conjunto de datos de tránsito (conjunto *T*) contiene tweets de diferentes ciudades del Ecuador del mes de mayo del 2018 y al igual que el conjunto *F* se lo clasificó de forma binomial, obteniendo 381 positivos y 919 negativos. Concluyendo que son conjuntos de datos desbalanceados, es decir, tienen una gran diferencia en cantidad entre clases (positivas, negativas). Los conjuntos de datos desbalanceados son frecuentes cuando su fuente son tweets ya que siempre marca la diferencia un tipo de opinión entre los usuarios; sin embargo, este tipo de datos pueden generar que el clasificador opte por minimizar el error en la clase predominante, lo que genera métricas de exactitud sesgada.

La elección de dos dominios diferentes dentro de la experimentación se planteó al presentarse ciertas características en el conjunto *F* con el cual se experimentó por primera vez. Este conjunto contiene tweets de diferentes procedencias, lo que implica una mayor dimensionalidad del lenguaje, al tener expresiones y formas de comunicación propias de cada país, esta característica produce una evaluación más detalla de las técnicas de pre-procesamiento, al no ir ligado a un conjunto homogéneo en cuanto a su diccionario o léxico, por otra parte, el conjunto *T* permite tener otra perspectiva, aplicando las mismas técnicas a un conjunto más cerrado en cuanto a lenguaje.

4.2. Técnicas

Las técnicas son los elementos base de la investigación, ya que serán el objeto de evaluación y estudio, con el fin de dar un criterio de su trabajo en PLN para texto corto en español. Como primer punto se analizó trabajos previos en el campo de PLN con la

finalidad de evaluar las técnicas más pertinentes en esta área, tomando en cuenta su definición y aplicaciones descritas en la sección 3.2. De acuerdo a sus funcionalidades las técnicas seleccionadas fueron: limpieza del texto, *tokenizer*, *stop-word removal*, l*emmatization* y *lowercase*; la forma en que se aplicaron estas técnicas en la investigación se describe en la Tabla 4.

Tabla 2
Técnicas evaluadas

Técnica	**¿Cómo se aplicó?**
Limpieza del texto	Eliminación de enlaces URL
	Eliminación de caracteres especiales: @, #
	Eliminación de nombres de usuarios
Tokenizer	División de palabras o tokens
Stop-Word Removal	Eliminación de palabras vacías por medio de corpus
Lemmatization	Transformación a su palabra raíz por medio de corpus
Lowercase	Transformación a minúsculas

4.3. Combinaciones de técnicas

Existen diferentes técnicas consideradas en PLN, por lo tanto, se generan múltiples combinaciones que pueden resultar de ellas, de acuerdo a lo que se quiere conseguir como resultado. En esta sección, se presentan todas las combinaciones posibles de técnicas de PLN en la etapa de pre-procesamiento, de modo que las interacciones con mejor resultado se descubran. La combinación de técnicas de pre-procesamiento se realizó en base al resultado de la investigación de Uysal & Gunal (2014), en su trabajo fundamentan que elegir combinaciones apropiadas en la etapa pre-procesamiento. En lugar de usar todas o solo una de ellas, puede proporcionar una mejora significativa en el procesamiento de lenguaje aplicado a la clasificación de texto. Además, se analizó el objetivo de aplicación de cada técnica seleccionada y el trabajo que ejerce en los diferentes campos de PLN, tomando como principal el análisis de sentimiento.

Las técnicas *lemmatization* (LEM), *stop-word removal* (STO), *tokenizer* (TOK) y *lowercase* (LOW) formaron parte de la evaluación, teniendo presente que la técnica *tokenizer* no fue evaluada en el análisis de sentimiento, por su participación obligatoria en el modelo de clasificación; por otra parte, las técnicas de limpieza de texto fueron aplicadas en todos los casos. Las técnicas se establecieron como 0 o 1, interpretándose como descartada o aplicada respectivamente en cada combinación. De acuerdo al número

de técnicas, se obtuvo 15 combinaciones diferentes (contando *tokenizer*), descritas en la Tabla 2, sin incluir la combinación que servirá como base de comparación en la evaluación (LM:0, SR:0, TK:0, LW:0).

Tabla 3
Combinaciones de técnicas de pre-procesamiento

	Combinaciones			
	LEM	**STO**	**TOK**	**LOW**
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	1	0	0
6	1	0	1	0
7	1	0	0	1
8	0	1	1	0
9	0	1	0	1
10	0	0	1	1
11	1	1	1	0
12	1	0	1	1
13	0	1	1	1
14	1	1	0	1
15	1	1	1	1

4.4. Librerías

En esta sección se presentan las librerías que han sido parte de la experimentación, en base a características comunes y sus funcionalidades de acuerdo al objetivo de la investigación.

De acuerdo a la sección 3.3 existen múltiples librerías que enfocan su trabajo para brindar herramientas en diferentes ámbitos de PLN como: clasificación de texto, etiquetado automático, análisis de sentimiento, análisis NER y aplicación de técnicas de pre-procesamiento; sin embargo, aún es limitada su funcionalidad en cuanto a idiomas como el español, al ser un lenguaje tan extendido.

Las librerías seleccionadas NLTK, SpaCy y FreeLing fueron el resultado de un análisis de características que las hacen competentes para el objetivo de la investigación, además de cumplir con la mayoría de combinaciones de técnicas establecidas; si bien no

cumplen con las 15 combinaciones, permiten un análisis con las más importantes (Ver Tabla 4).

Tabla 4
Librerías y técnicas que soportan

	Combinaciones				Librerías		
	LEM	STO	TOK	LOW	NLTK	SpaCy	Freeling
1	1	0	0	0		✓	✓
2	0	1	0	0	✓	✓	
3	0	0	1	0	✓	✓	✓
4	0	0	0	1		✓	✓
5	1	1	0	0		✓	
6	1	0	1	0		✓	✓
7	1	0	0	1		✓	
8	0	1	1	0	✓	✓	
9	0	1	0	1		✓	
10	0	0	1	1		✓	✓
11	1	1	1	0		✓	
12	1	0	1	1		✓	
13	0	1	1	1		✓	
14	1	1	0	1		✓	
15	1	1	1	1		✓	

NLTK y *FreeLing* son más utilizadas en PLN, *NLTK* por su fácil implementación y antigüedad en el campo investigativo, aunque su funcionalidad es mejor para texto en idioma inglés. FreeLing por otra parte es considerada una de las más potentes al permitir la construcción de sistemas de minería de opiniones muy fuertes (Henríquez et al., 2016). Spacy, aunque es una de las menos explotadas en PLN para lenguaje español; se consideró, por presentar un modelo completo para este lenguaje, con amplios corpus y múltiples técnicas básicas en pre-procesamiento de acuerdo a sus módulos en español.

Las librerías establecidas tienen como lenguaje de programación en común, el lenguaje Python, por lo que se decidió trabajar en éste, donde se realizó un script para cada librería, los cuales cuentan con funciones para la ejecución de cada combinación de técnicas que se aplicaron en los dos conjuntos de datos.

4.5. Modelo de análisis de sentimiento

La etapa de evaluación por medio del análisis de sentimiento, campo de PLN, fue fundamental dentro de la investigación, ya que su análisis sirvió como resultado final en la evaluación de las combinaciones de técnicas aplicadas a los conjuntos de datos. Para ello, fue necesario la construcción de un modelo de análisis (Ver figura 5) en la herramienta *RapidMiner*.

1. Fase

La primera fase cuenta con tres operadores básicos:

Read csv. – este operador recibe como entrada un archivo *csv* con un conjunto de datos aplicados una combinación de técnicas y clasificado manualmente de acuerdo a las dos clases establecidas (positiva, negativa),

Filter examples. – filtra los datos que corresponden al tipo de combinación a evaluar.

Process documents. – genera vectores de palabras a partir de atributos del texto. El esquema creado para el vector fue TF-IDF

2. Fase

La segunda fase trabaja con operadores para entrenamiento y clasificación del modelo, como:

Split data. – establece el número de tweets que sirvieron como entrenamiento y aplicación del modelo, quedando 70% y 30 % respectivamente,

Optimize Parameters. – este operador encuentra los valores óptimos de los parámetros elegidos para los operadores en su subproceso, en este caso para el algoritmo SVM con kernel RBF y sus parámetros *C* y *gamma*. Mediante este operador se configuró lo valores de las constantes *C* y γ tomando como base la investigación de Hsu, Chang, & Lin (2010) donde recomienda usar una "*grid-search*" o red de búsqueda por medio del método de validación cruzada, el cual consiste en probar diferentes pares de los valores de C y γ y seleccionar el que obtenga una mejor precisión u otro parámetro de evaluación, que para este caso fue el valor *F_measure,* al trabajar con texto desbalanceado. Tomando en cuenta que las secuencias de crecimiento exponencial en los parámetros (C y γ) es un

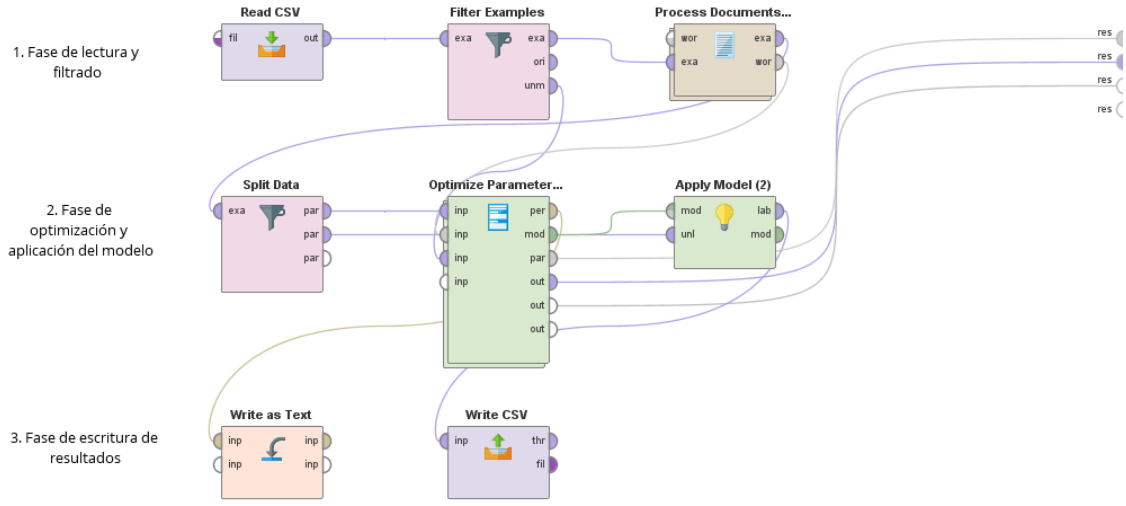
método más práctico, se usó la secuencia en donde C toma valores de: 2^{-5}, 2^{-3}, …, 2^{15} y γ valores de: 2^{-15}, 2^{-13}, …, 2^{3}. Este operador dio como resultado la matriz de confusión con los valores respectivos de los parámetros evaluados, entre ellos la precisión, recall y *F_measure*, siendo *F_measure* el valor base como medida de impacto de las técnicas y combinaciones aplicadas a los conjuntos.

Apply Model. – este operador aplica el modelo que fue entrenado para la clasificación.

3. Fase

La última fase fue la escritura de la matriz de confusión generada y los resultados de clasificación del modelo. Esta escritura se generó por medio de los operadores *Write as text* y *Write csv*.

Figura 5
Modelo de análisis de sentimiento



Fuente: Elaboración propia

4.6. Software

El software utilizado como sistema operativo, librerías, entornos de desarrollo de programación y software para análisis y minería de datos, se describen en la presente sección.

El sistema operativo, base de la instalación de las diferentes librerías y entornos de desarrollo de programación fue un ambiente Linux, Ubuntu 18.0 con una memoria de 4.0 GB y disco de 70 GB. Este sistema fue utilizado al considerarse uno de los más estables y con mayor facilidad de instalación al momento de trabajar con librerías de PLN. En el

caso de Freeling presenta versiones de paquetes y arquitectura estándar de Linux, mientras NLTK y SpaCy muestran mayor compatibilidad y estabilidad en sus nuevas versiones, con este sistema operativo.

Las librerías que fueron empleadas en la investigación utilizaron el lenguaje de programación Python 3. El lenguaje fue común para cada librería. NLTK es una librería de código abierto propio de Python, al igual que SpaCy; si bien Freeling es una librería en C++, sus nuevas versiones han sido adaptadas para trabajar en el lenguaje Python, en el caso de NLTK se utilizó la versión 3.3, en Spacy se implementó la versión 3.6 y para Freeling la versión 4.1.

La utilización de librerías y programación de scripts requiere de la instalación de paquetes y módulos, para lo cual es útil un entorno de desarrollo integrado, el cual facilita la programación, compilación y ejecución de los diferentes scripts desarrollados. Para esta investigación, en la programación en cada librería se optó por el entorno de desarrollo integrado PyCharm, un entorno dirigido específicamente para el lenguaje Python, con un alto nivel en cuanto a sus funcionalidades y servicios en este lenguaje.

El último software utilizado, RapidMiner, versión 8.2, es un software para análisis y minería de datos, cuyo entorno gráfico permite el desarrollo de procesos de análisis de datos por encadenamiento de operadores, desarrollados en lenguaje de programación Java. Por medio de este programa se generó el modelo de análisis de sentimiento que permitió la evaluación y medición del impacto de las combinaciones de técnicas aplicadas a los dos conjuntos de datos, por medio de los parámetros de evaluación establecidos.

5. RESULTADOS

Una vez aplicado el algoritmo de clasificación *SVM* en los conjuntos *F* y *T* procesados con las diferentes técnicas y sus combinaciones en cada librería detalladas en la Tabla 5 y 6 respectivamente, se obtuvieron los siguientes resultados que serán analizados desde diferentes puntos de vista.

5.1. Análisis de técnicas en la etapa de pre-procesamiento

La evaluación de técnicas en la etapa de pre-procesamiento estuvo ligada a parámetros cualitativos como número de palabras vacías eliminadas en cada librería y características encontradas en la lematización del texto, además de un parámetro cuantitativo como es el tiempo de procesamiento para cada técnica en las diferentes librerías. Los resultados de estos parámetros se reflejan en los siguientes gráficos.

Tiempo de procesamiento:

Figura 6
Tokenizer

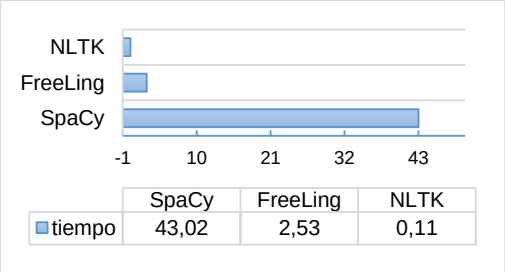


Figura 7
Lowercase

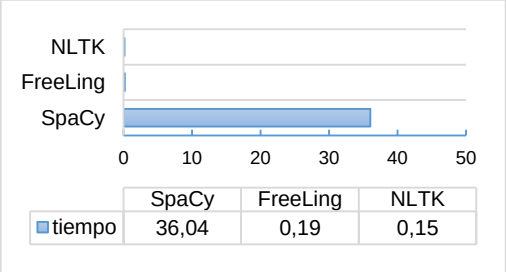


Figura 8
Stop-Word Removal

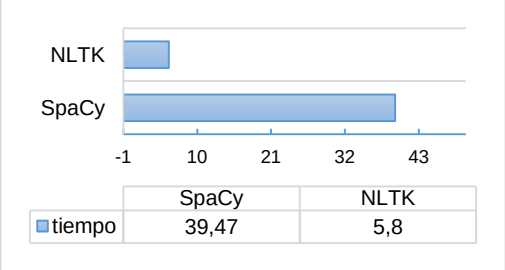
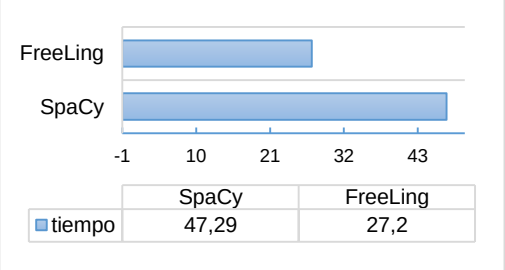


Figura 9
Lemmatizer



Palabras vacías eliminadas:

Figura 10
Conjunto F

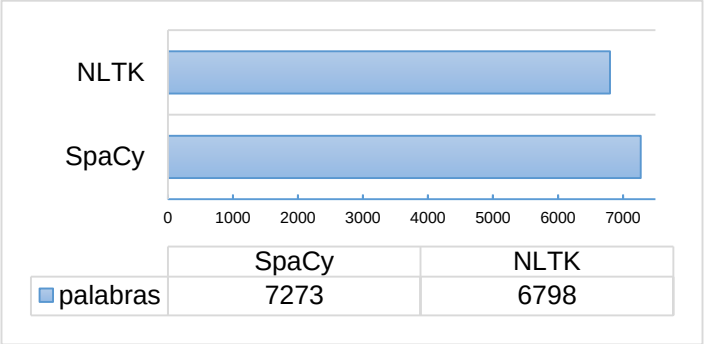
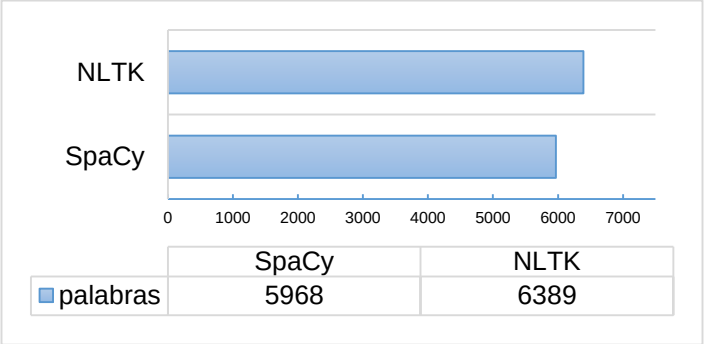


Figura 11
Conjunto T



Características de librerías evaluadas para cada técnica:

Tabla 5
Análisis cualitativo de librerías

NLTK	Tokenizer	Incluye transformación de lowercase
	Stop-Word Removal	Requiere transformación a lowercase previamente.
SpaCy	Lemmatizer	Lematización de números No existe un análisis semántico y pragmático dentro del algoritmo. No existe transformación de plural a singular.
	Stop-Word Removal	Requiere transformación a lowercase previamente El diccionario no cuenta con palabras vacías comunes en el lenguaje español : y, a.
FreeLing	Lemmatizer	Existe un análisis semántico y pragmático No existe transformación de plural a singular

5.2. Análisis de técnicas aplicadas al análisis de sentimiento.

Para la evaluación de técnicas de PLN aplicadas al análisis de sentimiento en los dos conjuntos de datos, se tomó como parámetros de evaluación, la precisión, recall y *F_measure* como medida de rendimiento general y valor a considerar en la determinación de las mejore técnicas a usar en PLN para texto corto en español.

Tomando el valor *F_measure* como parámetro principal, se extrajo las combinaciones con mayor valor, de esta forma se evidenció que al aplicar la técnica *lemmatizer* combinada con técnicas como *lowercase*, se alcanzó el *F_measure* máximo, el cual supera en un 5% el valor obtenido al no aplicar ninguna técnica de pre-procesamiento, sobre el conjunto de datos *F*. Para el conjunto de datos *T*, el valor de *F_measure* mejora en un 9% al aplicar la técnica *lemmatizer* de forma individual y combinada con *lowercase.* Demostrando así que *lemmatizer* como técnica de pre-procesamiento en PLN, beneficia a la clasificación de textos cortos en lenguaje español. A su vez se evidencia que técnicas como *lowercase* y *stop-word removal*, si bien no alcanzan un *F_measure* máximo, generan una mejora significativa en la clasificación. Se recalca que la técnica *tokenizer* no se evaluó en la clasificación de texto al ser una técnica necesaria en todos los casos y que el modelo requiere por defecto internamente. Los resultados de forma resumida se pueden observar en la Tabla 6, los resultados completos se muestran en el Anexo 1 y 2, con los detalles de precisión, *recall* y *F_measure* para cada combinación en las librerías.

Tabla 6
Mejores resultados F_measure

	Precisión	Recall	F_measure	Técnicas	Librerías
Conjunto F	77%	99%	86%	LW:0 | TK:0 | LM:0 | STO:0	
	87%	95%	91%	LW:1 | LM:1	SpaCy
Conjunto T	82%	98%	87%	LW:0 | TK:0 | LM:0 | STO:0	
	93,9%	97,2%	96%	LM:1	SpaCy
	95%	97%	96%	LW:1 | LM:1	SpaCy
	94%	97%	96%	LM:1	FreeLing

6. CONCLUSIONES

En este trabajo, se evaluó el impacto de técnicas de pre-procesamiento en PLN, tomando como caso de estudio el campo de análisis de sentimiento, uno de los más experimentados en PLN. Las técnicas fueron evaluadas en dos diferentes dominios de tweets, Mundial de Fútbol 2018 y Tránsito en el Ecuador de mayo del 2018, utilizando todas las combinaciones posibles de las cuatro técnicas de pre-procesamiento seleccionadas, además de ser evaluadas en diferentes librerías de PLN. Los aspectos a considerar en la evaluación, fue el tiempo de procesamiento de las librerías en cada técnica, características de procesamiento y capacidad de sus corpus y algoritmos aplicados para el idioma español, en cuanto al análisis de sentimiento se midió el impacto de cada una de las combinaciones obtenidas considerando su precisión, recall y *F_measure* como medida de rendimiento general.

El análisis experimental reveló que las combinaciones apropiadas de técnicas como *lemmatizer* y *lowercase* en la etapa de pre-procesamiento según el *F_measure* obtenido después de aplicar el modelo de análisis de sentimiento proporcionan una mejora significativa en la clasificación. Aunque hay técnicas de pre-procesamiento, como *lowercase* o *stop-word removal* que no alcanzan la mejora máxima en el proceso de análisis de sentimiento, si existe una mejora demostrativa para éste. Por lo tanto, para un problema de clasificación de texto como el análisis de sentimiento para texto en español en cualquier dominio, es recomendable analizar cuidadosamente todas las combinaciones posibles de técnicas, en lugar de utilizar una o todas en la etapa de pre-procesamiento. En caso de trabajar con otros campos de PLN es necesario examinar el fin del estudio y analizar la función de cada técnica en el texto, para realizar un procesamiento más eficiente y con mayor precisión en los resultados finales.

7. BIBLIOGRAFÍA

Agogo, D., & Hess, T. J. (2018). 110. Scale Development Using Twitter Data: Applying Contemporary Natural Language Processing Methods in IS Research. *Springer International Publishing AG 2018*. https://doi.org/10.1007/978-3-319-58097-5

Alami, N., Meknassi, M., Ouatik, S. A., & Ennahnahi, N. (2017). 117. Impact of stemming on Arabic text summarization. *Colloquium in Information Science and Technology, CIST*, 338–343. https://doi.org/10.1109/CIST.2016.7805067

Altszyler, E., & Brusco, P. (2015). semantico. Análisis de la dinámica del contenido semántico de textos. *Argentine Symposium on Artificial Intelligence*, 256–263.

Antonio, F., Velásquez, C., Paul, J., Paz, Z. De, & Guzmán, J. F. (2017). sintactico. Aplicación del análisis sintáctico automático en la atribución de autoría de mensajes en redes sociales Authorship Attribution of Social Media Messages, *137*, 109–119.

Asghar, M. Z., Khan, A., Ahmad, S., & Kundi, F. M. (2014). lemma2. A Review of Feature Extraction in Sentiment Analysis, *4*(3), 181–186.

Banados, J. A., & Espinosa, K. J. (2014). 136. Optimizing Support Vector Machine in classifying sentiments on product brands from Twitter. *IISA 2014 - 5th International Conference on Information, Intelligence, Systems and Applications*, 75–80. https://doi.org/10.1109/IISA.2014.6878768

Battistelli, D., Charnois, T., Minel, J. L., & Teissèdre, C. (2013). 128. Detecting salient events in large corpora by a combination of NLP and data mining techniques. *Computacion y Sistemas*, *17*(2), 229–237.

Baviera, T. (2016). 111. Técnicas para el análisis del sentimiento en Twitter : Aprendizaje Automático Supervisado y SentiStrength Tecnhiques for sentiment analysis in Twitter : Supervised Learning and SentiStrength. *Revista Dígitos 1.3*, 33–50.

Chih-Wei Hsu, Chih-Chung Chang, and C.-J. L. (2016). 137. A Practical Guide to Support Vector Classification. *BJU International*, *101*(1), 1396–1400. https://doi.org/10.1177/02632760022050997

Díaz Gómez, R. (2001). *lemalibro. Estudio de la incidencia del conocimiento linguistico en los sistemas de recuperación de la informacion para el español*.

Dubiau, L., & Ale, J. M. (2013). 102. Análisis de Sentimientos sobre un Corpus en Español: Experimentación con un Caso de Estudio. *Asai 2013*, 36–47. Retrieved from http://42jaiio.sadio.org.ar/proceedings/simposios/Trabajos/ASAI/04.pdf

Galadanci, B. S., Muaz, S. A., & Mukhtar, M. I. (2016). 115. Comparing research outputs of Nigeria Federal Universities based on the scopus database. *CEUR Workshop Proceedings*, *1755*, 79–84. https://doi.org/10.1177/0165551510000000

Go, A., Bhayani, R., & Huang, L. (2009). 127-18. Twitter Sentiment Classification using Distant Supervision. *Processing*, *150*(12), 1–6. https://doi.org/10.1016/j.sedgeo.2006.07.004

Gupta, I., & Joshi, N. (2017). 116. Tweet Normalization : A Knowledge Based Approach. *Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), 2017 International Conference On*, 1–6.

Guzman, E., & Maalej, W. (2014). lemma1. How do users like this feature? A fine grained sentiment analysis of App reviews. *2014 IEEE 22nd International Requirements Engineering Conference, RE 2014 - Proceedings*, 153–162. https://doi.org/10.1109/RE.2014.6912257

Haddi, E., Liu, X., & Shi, Y. (2013). 112. The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, *17*, 26–32. https://doi.org/10.1016/j.procs.2013.05.005

Hasan, F. M., UzZaman, N., & Khan, M. (2007). POS2. Comparison of Unigram, Bigram, HMM and Brill's POS Tagging Approaches for some South Asian Languages. *Machine Learning*, (June).

He, Y., & Kayaalp, M. (2006). Tokenizers on MEDLINE December 2006. *Building*. https://doi.org/10.13140/2.1.1133.1206

Henríquez, C., Guzmán, J., & Salcedo, D. (2016). 130. Minería de Opiniones basado en la adaptación al español de ANEW sobre opiniones acerca de hoteles Opinion Mining based on the spanish adaptation of ANEW on hotel customer comments. *Procesamiento Del Lenguaje Natural*, *41*, 25–32.

Hidalgo, O., Jaimes, R., Gomez, E., & Luján-mora, S. (2017). 103. Análisis de sentimiento aplicado al nivel de popularidad del líder político ecuatoriano Rafael Correa Sentiment Analysis applied to the popularity level of the Ecuadorian political leader Rafael Correa. *Information Systems and Computer Science (INCISCOS), 2017 International Conference On*. https://doi.org/10.1109/INCISCOS.2017.64

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2010). 137. A Practical Guide to Support Vector Classificatio. *Department of Computer Science National*, (2). https://doi.org/10.1007/s11119-014-9370-9

Jadav, B. M., & Scholar, M. E. (2016). 135. Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis. *International Journal of Computer Applications*, *146*(13), 26–30.

Jianqiang, Z., & Xiaolin, G. (2017). 120. Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, *5*, 2870–2879. https://doi.org/10.1109/ACCESS.2017.2672677

Katariya, N. P., & Chaudhari, M. S. (2015). 126. Text Preprocessing for Text Mining Using Side Information. *International Journal of Computer Science and Mobile Applications*, *3*, 3–7.

Krouska, A., Troussas, C., & Virvou, M. (2016). 119. The effect of preprocessing techniques on Twitter Sentiment Analysis. *Information, Intelligence, Systems & Applications (IISA), 2016 7th International Conference On*, 1–5. https://doi.org/10.1109/IISA.2016.7785373

Krovetz. (1993). lemmalibro1. 1993 TOC Search Viewing Morphology as an Inference Process, *118*(October 1998), 191–202.

Narmadha, R. P., & Sreeja, G. G. (2016). 101. A survey on online tweet segmentation for linguistic features. *2016 International Conference on Computer Communication and Informatics, ICCCI 2016*. https://doi.org/10.1109/ICCCI.2016.7479955

Pang, B., Lee, L., & Vaithyanathan, S. (2002). 127-31. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP '02*, *10*(July), 79–86. https://doi.org/10.3115/1118693.1118704

Pérez-guadarramas, Y., Rodríguez-blanco, A., & Simón-cuevas, A. (2017). lexico. Combinando patrones léxico - sintácticos y análisis de tópicos para la extracción automática de frases relevantes en textos Combining lexical - syntactic patterns and topic analysis for automatic keyphrase extraction from texts.

Prata, D. N., Soares, K. P., Silva, M. A., Trevisan, D. Q., & Letouze, P. (2016). 103-1. Social Data Analysis of Brazilian's Mood from Twitter. *International Journal of Social Science and Humanity*, *6*(3), 179–183. https://doi.org/10.7763/IJSSH.2016.V6.640

Singh, T., & Kumari, M. (2016). 118. Role of Text Pre-processing in Twitter Sentiment Analysis. *Procedia Computer Science*, *89*, 549–554. https://doi.org/10.1016/j.procs.2016.06.095

Soto Kiewit, L. D. (2017). pragmatico. Un acercamiento a la concepción de gobernabilidad en los discursos presidenciales de José María Figueres Olsen. *Revista Rupturas*, *7*(1), 1–27. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=120099102&site=ehost-live&scope=site

Sun, S., Luo, C., & Chen, J. (2017). *106. A review of natural language processing techniques for opinion mining systems*. *Information Fusion* (Vol. 36). Elsevier B.V. https://doi.org/10.1016/j.inffus.2016.10.004

Symeonidis, S., Effrosynidis, D., & Arampatzis, A. (2018). 140. 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

Torres López, J. C., Sánchez-Sánchez, C., & Villatoro-Tello, E. (2014). POS1. Laboratorio en línea para el procesamiento automático de documentos. *23 Research in Computing Science*, *72*, 23–36. Retrieved from https://pdfs.semanticscholar.org/6877/6baebce8835f31595a0fe5289fe106bf6bfa.pd

f?_ga=2.255859913.331449770.1528969293-1865361036.1525084730

Uysal, A. K., & Gunal, S. (2014). 109. The impact of preprocessing on text classification. *Information Processing and Management*, *50*(1), 104–112. https://doi.org/10.1016/j.ipm.2013.08.006

Vijayarani, S., Ilamathi, J., & Nithya, M. (2015). 114. Preprocessing Techniques for Text Mining - An Overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7–16. Retrieved from http://www.ijcscn.com/Documents/Volumes/vol5issue1/ijcscn2015050102.pdf

8. ANEXOS

Anexo 1. Tabla de resultados: Conjunto F

	Combinación	Tiempo-técnicas(seg)	Tiempo-clasificación	Precisión	Recall	F_measure
NLTK	STO	5,80	6:17	80,5%	98,7%	88%
	TOK-STO	0,07	-	-	-	-
	TOK	0,11	9:53	-	-	-
SPACY	TOK	43,02	-	-	-	-
	LOW	36,04	9:02	84,8%	94,2%	89%
	LOW-TOK	35,51	-	-	-	-
	LOW-STO	40,67	6:21	84,4%	95,2%	89%
	LOW-TOK-STO	36,53	-	-	-	-
	TOK-STO	20,55	-	-	-	-
	STO	39,47	7:46	82,6%	98,0%	90%
	LOW-TOK-STO-LEM	36,04	-	-	-	-
	LEM	47,29	8:41	86,5%	93,6%	90%
	LOW-STO-LEM	35,72	6:40	83,1%	97,8%	90%
	STO-LEM	21,84	5:45	84,4%	96,4%	90%
	TOK-STO-LEM	20,00	-	-	-	-
	TOK-LEM	22,80	-	-	-	-
	LOW-TOK-LEM	34,12	-	-	-	-
	LOW-LEM	38,11	5:58	86,5%	95,0%	91%
FREELING	LOW-TOK	0,17	-	-	-	-
	LOW	0,18	10:44	83,8%	95,6%	89%
	TOK	2,53	-	-	-	-
	LEM	27,20	8:49	84,7%	95,9%	90%

Anexo 2. Tabla de resultados: Conjunto T

	Combinación	Tiempo-técnicas(seg)	Tiempo-clasificación	Precisión	Recall	F_measure
NLTK	STO	4,20	5:22	93,1%	96,7%	95%
	TOK-STO	4,19	-	-	-	-
	TOK	0,11	-	-	-	-
SPACY	LOW-STO-LEM	20,87	6:14	93,4%	94,9%	94%
	TOK-STO	21,07	-	-	-	-
	LOW-LEM	20,59	5:11	92,8%	96,1%	94%
	LOW-TOK-STO	20,69	-	-	-	-
	STO	21,67	5:36	92,8%	96,2%	94%
	TOK	22,46	-	-	-	-
	LOW	22,87	5:54	93,7%	96,7%	95%
	LOW-TOK	20,37	-	-	-	-
	STO-LEM	20,23	6:49	94,8%	95,9%	95%
	TOK-STO-LEM	20,97	-	-	-	-
	LOW-TOK-STO-LEM	20,94	-	-	-	-
	LEM	23,30	4:48	95,1%	96,6%	96%
	TOK-LEM	21,10	-	-	-	-
	LOW-LEM	21,05	5:09	95,1%	96,6%	96%
	LOW-TOK-LEM	20,46	-	-	-	-
FREELING	TOK	10,96	-	-	-	-
	LOW	8,86	4:40	92,6%	97,4%	95%
	LOW-TOK	9,26	-	-	-	-
	LEM	18,31	5:49	93,9%	97,2%	96%

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 en sesión del 11 de mayo de 2018, conoció y aprobó la solicitud para realización del trabajo de titulación, presentada por :

Estudiante: Andrea Alexandra Trujillo Orellana (cód. 73189)
Tema: **"Evaluación de técnicas de procesamiento de lenguaje natural para textos cortos en lenguaje español"**
Previo a la obtención del título de Ingeniero de Sistemas y Telemática
Director: Ing. Marcos Orellana
Tribunal: Ing. Lenin Erazo
Ing. María Inés Acosta

Plazo de presentación del trabajo de titulación, con la calificación del Director: seis meses a partir de la fecha de aprobación, esto es hasta el 11 de noviembre de 2018.

E INFORMA:

Que en aplicación de la Disposición General Cuarta del Reglamento de Régimen Académico vigente, en caso de que la estudiante no culmine y apruebe el trabajo de titulación luego de dos períodos académicos contados a partir de la fecha de culminación de estudios, deberá realizar la actualización de conocimientos previa a su titulación.

Cuenca, 14 de mayo de 2018

UNIVERSIDAD DEL AZUAY | Facultad Ciencias de la Administración

Dra. María Elena Ramírez Aguilar
Secretaria - Abogada

Oficio Nro. 015-2018-DIST-UDA

Cuenca, 2 de mayo de 2018

Señor Ingeniero
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 2 de mayo del 2018, recibió el proyecto de tesis titulado "Evaluación de técnicas de procesamiento de lenguaje natural para textos cortos en lenguaje español", presentado por Andrea Alexandra Trujillo Orellana estudiante de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Marcos Orellana, previo a la obtención del título de Ingeniero de Sistemas y Telemática.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomendamos como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Marcos Orellana, como co-directora y miembro de tribunal a la Ing.Belén Arias, y como miembro de tribunal a la Ing. María Ines Acosta.

Atentamente,

Ing. Catalina Astudillo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
Universidad del Azuay

Ing. Esteban Crespo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
Universidad del Azuay

Oficio Nro. 015-2018-DIST-UDA

Cuenca, 2 de mayo de 2018

Señor Ingeniero
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 2 de mayo del 2018, recibió el proyecto de tesis titulado "Evaluación de técnicas de procesamiento de lenguaje natural para textos cortos en lenguaje español", presentado por Andrea Alexandra Trujillo Orellana estudiante de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Marcos Orellana, previo a la obtención del título de Ingeniero de Sistemas y Telemática.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomendamos como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Marcos Orellana, como co-directora y miembro de tribunal a la Ing.Belén Arias, y como miembro de tribunal a la Ing. María Ines Acosta.

Atentamente,

Ing. Catalina Astudillo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
Universidad del Azuay

Ing. Esteban Crespo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
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: **"Evaluación de técnicas de procesamiento de lenguaje natural para textos cortos en lenguaje español"**, presentado por la estudiante Andrea Alexandra Trujillo Orellana con código 73189 , previa a la obtención del título de Ingeniero de Sistemas y Telemática, para el día **Miércoles, 09 de mayo de 2018 a las 07:00.**

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, 07 de mayo de 2018

Dra. María Elena Ramírez Aguilar
Secretaria de la Facultad

Ing. Marcos Orellana .....

Ing. Belén Arias .....

Ing. María Inés Acosta .....

ACTA
SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

Fecha de sustentación: **Miércoles, 09 de mayo de 2018 a las 07:00**

1.1. Nombre del estudiante: Andrea Alexandra Trujillo Orellana
1.2. Código: 73189
1.3. Director sugerido: Ing. Marcos Orellana
1.4. Codirector (opcional): _____
1.4.1. Tribunal: Ing. Belén Arias e Ing. María Inés Acosta
1.4.2. Título propuesto: **"Evaluación de técnicas de procesamiento de lenguaje natural para textos cortos en lenguaje español"**

1.4.3. Aceptado sin modificaciones : X

1.4.4. Aceptado con las siguientes modificaciones:

_____

_____

_____

1.4.5. No aceptado
1.4.6. Justificación:

_____

_____

_____

Tribunal

Ing. Marcos Orellana

Ing. Belén Arias

Ing. María Inés Acosta

Srta. Andrea Alexandra Trujillo Orellana

Dra. María Elena Ramírez Aguilar
Secretaria de la Facultad

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

1.1. Nombre del estudiante: Andrea Alexandra Trujillo Orellana
1.2. Código : 73189
1.3. Director sugerido:
1.3.1. Codirector (opcional): Ing. Marcos Orellana
1.3.2. Título propuesto: **"Evaluación de técnicas de procesamiento de lenguaje natural para textos cortos en lenguaje español"**
1.3.3. Revisores tribunal: Ing. Belén Arias e Ing. María Inés Acosta
1.4. Recomendaciones generales de la revisión:

	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?	✓	

Ing. Marcos Orellana

Ing. Belén Arias

Ing. María Inés Acosta

UNIVERSIDAD DEL AZUAY

Cuenca, 04 de Mayo del 2018

Ingeniero

Oswaldo Merchán Manzano

DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

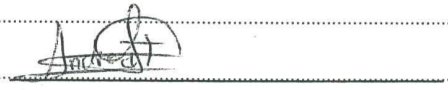
UNIVERSIDAD DEL AZUAY

De mi consideración,

Estimado Señor Decano, yo Andrea Alexandra Trujillo Orellana con C.I. 1401328180, código estudiantil 73189, estudiante de la Carrera de Sistema y Telemática, solicito muy comedidamente a usted y por su intermedio al Consejo de Facultad, la aprobación del protocolo de trabajo de titulación con el tema: **"Evaluación de técnicas de procesamiento de lenguaje natural para textos cortos en lenguaje español"** previo a la obtención del título de Ingeniería en Sistema y Telemática, para lo cual adjunto la documentación respectiva.

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

Atentamente:



Andrea Alexandra Trujillo Orellana

UNIVERSIDAD DEL AZUAY

UNIVERSIDAD DEL AZUAY
INGENIERIA EN SISTEMAS Y TELEMATICA
DISEÑO DE TESIS

1. DATOS GENERALES

1.1 Nombre del estudiante: Andrea Alexandra Trujillo Orellana

1.1.1 Código: 73189

1.1.2 Contacto:

Teléfono convencional: 2740331

Celular: 0960174885

Correo electrónico: andre.t.2294@gmail.com

1.3Director sugerido: Orellana Cordero, Marcos Ing.

1.2.1 Contacto:

Teléfono convencional:428640

Celular:0999955611

Correo electrónico: marore@uazuay.edu.ec

1.3 Co-director sugerido: Ing. María Belén Arias

1.4 Asesor metodológico:

1.5 Tribunal designado: Ing.

1.6 Aprobación: Junta Académica:

Consejo de Facultad:

1.7 Línea de Investigación de la carrera:

1.7.1 Código UNESCO: 1203 Informática de computadores

1203.17 Informática

1.7.2 Tipo de trabajo: Investigación.

Edición autorizada de 15.000 ejemplares
Del 835.501 al 850.500
Nº 0848336

1.8 Área de estudio: Procesamiento de datos.

1.9 Título propuesto: Evaluación de técnicas de procesamiento de lenguaje natural para textos cortos en lenguaje español.

1.10 Estado del proyecto: Existen investigaciones que proyectan estudios de técnicas y librerías para PLN (Procesamiento de Lenguaje Natural); sin embargo, solo se enfocan en textos para lenguaje inglés u otros. No existe una investigación o trabajo previo al proyecto planteado. El trabajo propuesto será una base de apoyo para investigaciones futuras en el campo de análisis de datos. Este proyecto tiene una nueva perspectiva de investigación para PLN en lenguaje español basado en técnicas encontradas aplicadas al lenguaje inglés, teniendo como objetivo la evaluación de estas técnicas implementadas en librerías para textos cortos en lenguaje español. En función de los resultados obtenidos, se determinará las mejores técnicas a aplicar en el procesamiento de textos cortos en español, y cuáles serán las librerías vinculadas. Se tomará en cuenta aspectos como: precisión, exhaustividad y rendimiento.

2. CONTENIDO

2.1 Motivación de la investigación:

En la actualidad existen múltiples librerías con diferentes técnicas para PLN. Sin embargo, la mayoría de investigaciones se enfocan en el lenguaje inglés, por ello, la importancia y el interés de la investigación. El presente trabajo buscará evaluar las técnicas y librerías asociadas, enfocándose en una característica específica: la compatibilidad con textos en lenguaje español. Para esto, es necesario analizar una combinación de técnicas y sus resultados aplicados al análisis de sentimiento en textos cortos.

2.2 Problemática:

PLN es un campo de las ciencias computacionales muy amplio, por ello, la información sobre herramientas, bibliotecas, técnicas y metodologías existen en gran cantidad. Sin embargo, es poca la información o la importancia que se le da al uso de este campo en textos en lenguaje español. Artículos, libros e informes presentan una gran variedad de recursos para el procesamiento de texto principalmente enfocados a lenguajes como el inglés, turco y árabe. Las investigaciones más relevantes evalúan a breves rasgos librerías en general, o técnicas de forma independiente, por lo que es necesario valorar de manera más detallada cada una de ellas, enfocándose en el principal problema: el análisis de datos en español. De esta forma se colaborará a futuras investigaciones en la decisión sobre la elección de librerías y sus técnicas en el procesamiento de textos cortos en este idioma.

2.3 Pregunta de investigación:

¿Cuáles son las técnicas y librerías relevantes para el procesamiento de lenguaje natural en textos cortos del lenguaje español?

2.4 Resumen:

Las técnicas aplicadas en el Procesamiento de Lenguaje Natural (PLN) son componentes claves en el resultado del procesamiento de texto. Sin embargo, la escaza información de técnicas de procesamiento para lenguaje español dificulta la aplicación de PLN para textos en este idioma. El objetivo de este trabajo es analizar y recomendar las técnicas de PLN para lenguaje español, en el análisis de textos cortos recopilados y clasificados previamente en la extracción de comentarios de redes sociales para el dominio establecido en twitter. Se evaluará la precisión y rendimiento de librerías asociadas a las técnicas seleccionadas, por medio de una base de datos(tweets) común para cada librería con sus respectivas técnicas. La evaluación de técnicas descubrirá librerías compatibles y con mejor rendimiento. Será necesario experimentar con

varias librerías y combinar las técnicas elegidas, lo que generará conocimiento para futuras investigaciones.

2.5 Indagación Exploratoria:

Las investigaciones en línea, la expresión libre en redes sociales, y en la web han despertado el interés en técnicas de extracción automática de información, debido a que son fuentes de datos en tiempo real. Estas investigaciones se encaminan especialmente en el análisis de sentimiento, detección y clasificación, técnicas que se encuentran ligadas a herramientas, algoritmos y librerías para PLN (Procesamiento de Lenguaje Natural). Una de las aplicaciones web que en los últimos años requiere librerías para este trabajo es Twitter, dado que es una fuente que permite analizar el estado de ánimo del público sobre un problema social particular (Kanakaraj & Mohana, 2015). Herramientas como Weka son útiles en el análisis de algoritmos, métodos y técnicas de PLN (Fernández & Otros, 2013). En cuanto a librerías, otros autores proponen NLTK (Hernández, 2016) y SpaCy (Omran & Treude, 2017).

PLN es definido por Reese (2015) como un área de estudio que utiliza campos como la informática e inteligencia artificial para analizar los lenguajes por medio de conocimientos lingüísticos; lo explica también como un grupo de herramientas que permiten obtener información relevante de documentos de texto, oraciones, páginas en línea u otras fuentes. Destaca la utilización de PLN al ser un campo de análisis de textos que pueden ser cortos o largos, y que se encuentran especialmente en la red, como blogs y redes sociales (especialmente Twitter). Reese en su libro define áreas de aplicación de PLN, entre ellas:

- Búsquedas.
- Agrupación de información.
- Reconocimiento de voz.

- Análisis de sentimiento.
- Traducciones automáticas.
- Reconocimiento de entidades nombradas (NER), entre otras.

Estudios sobre PLN señalan que la lengua española es uno de los idiomas que estadísticamente se encuentra por debajo de la inglesa por la cantidad de países que la utilizan. Sin embargo, es el segundo idioma más hablado en el mundo, de ello, la importancia de estudiar librerías de PLN para lenguaje español. El análisis y la comparación de la precisión y exhaustividad, que será nombrada de ahora en adelante como recall, son los parámetros evaluados frecuentemente. Valverde y Martínez (2016) destacan librerías populares de código abierto como: NLTK, OpenNLP, OpenNER y Patter.ES. En esta investigación fueron evaluadas en su rendimiento general por medio del valor-F; esta medida es una cohesión entre la precisión y exhaustividad. Sin embargo, se limita a aplicaciones como NER y análisis de sentimiento. Además, se señala a breves rasgos técnicas como: *tokenizer, stemmer*, *POS tagger, tree tagger*, análisis sintáctico, análisis NER y análisis de sentimiento (Valverde & Martínez, 2016).

Kursat Uysal & Gunal (2014) presentan una investigación a profundidad, midiendo el impacto de técnicas de pre-procesamiento enfocado en textos en lenguaje turco e inglés; entre éstas incluye: *tokenizer, stop-word removal, lowercase y stemming*. La metodología aplicada es una comparativa entre los dos idiomas con dos tipos de dominios de texto, noticias y correos electrónicos, para ello, se utilizó la combinación de las diferentes técnicas de pre-procesamiento seleccionadas tomando en cuenta *tokenizer* alfanumérica o alfabética, *stop-word removal* activada o desactivada, al igual que el *lowercase y stemming*. A diferencia de Valverde & Martínez (2016) no se estudia la comparación entre librerías PLN, su precisión,

recall y rendimiento. Kursat Uysal & Gunal (2014) evaluaron técnicas de pre-procesamiento que sean útiles para un procesamiento completo orientado al análisis de sentimiento.

Existen múltiples librerías enfocadas a PLN, entre ellas: CoreNLP, NLTK (Paramkusham, 2014), OpenNLP y OpenNer. Destacando NLTK Y CoreNLP. Según Hernández (2016) NLTK (Kit de Herramienta de Lenguaje Natural) tiene un mayor soporte debido a su comunidad de usuarios, además afirma que ésta tiene mayor aceptación en el ámbito científico. Esta biblioteca incluye tipos de datos como: *tokens*, etiquetas, árboles, estructuras de características y definiciones de interfaz; que se aprovechan de técnicas de *tokenizer, stemming, taggers*, *clusters,* y clasificadores (Paramkusham, 2014).

Estudios mencionados evalúan o utilizan preferentemente bibliotecas de código abierto, un ejemplo de ello es "AraNLP" basada en Java, cuenta con varias técnicas de pre-procesamiento de texto en árabe; esta biblioteca reúne la mayoría de las metodologías o técnicas básicas de pre-procesamiento de texto, entre sus características está: acceso fácil y adaptable (Althobaiti, Kruschwitz, & Poesio, 2014). Adicionalmente, se tienen librerías como SpaCy enfocadas al procesamiento avanzado del lenguaje natural, la cual está dirigida a aplicaciones comerciales, y en particular es considerada una de las más rápidas entre otras bibliotecas (Omran & Treude, 2017).

Los investigadores mencionados anteriormente, señalan librerías para textos fuera de la web. Las nuevas tecnologías PLN intentan dotar de un servicio web, este es el caso de "ITU Turco Servicio web NLP ". Esta plataforma trabaja como un SaaS (Software as a Service) y proporciona a investigadores y estudiantes lo último en herramientas PLN multi-capas tales como: pre-procesamiento, morfología, sintaxis y reconocimiento de la entidad. Los usuarios pueden comunicarse con la plataforma por medio de tres canales: a través de una interfaz web

amigable para el usuario, por archivos cargados y por medio de la API web dentro de sus propios códigos para construir aplicaciones de nivel superior (Eryigit, 2014).

Los trabajos anteriormente descritos han establecido qué técnicas de procesamiento son las más adecuadas para el análisis de texto, sin especificar cuáles son las librerías que cuenten con estas funcionalidades. Otros estudios realizan un análisis comparativo entre librerías de código abierto con escasa información de las técnicas relevantes que tienen dichas bibliotecas y sin enfocarse en textos en lenguaje español. Este proyecto aportará con un análisis detallado de técnicas relevantes de procesamiento de texto corto en lenguaje español y sus librerías asociadas. El resultado será una tabla resumen que contenga la precisión, recall y rendimiento de cada librería, experimentando las técnicas seleccionadas. La experimentación será llevada a cabo por un dominio de texto, específicamente tweets de un programa televisivo de entretenimiento, al ser un dominio de alto tráfico de datos, además de existir una variedad de expresiones al ser de índole internacional.

2.6 Objetivo general:

Evaluar técnicas relevantes de procesamiento del lenguaje natural para textos cortos en lenguaje español y las librerías asociadas.

2.7 Objetivos específicos:

1. Realizar una indagación inicial de proyectos paralelos o afines al trabajo planteado.
2. Elaborar un marco teórico sobre las técnicas de procesamiento de PLN relevantes y las librerías asociadas
3. Experimentar con las técnicas y librerías identificadas, por medio de pruebas con el dominio de texto planteado, identificando las técnicas relevantes y las librerías asociadas.

4. Analizar los resultados obtenidos de la evaluación de técnicas seleccionadas.

2.8 Metodología:

Se estudiará la base teórica del tema a través de una búsqueda bibliográfica en diferentes portales web: Google Académico, IEEE y repositorios de universidades. Se seleccionarán las fuentes bibliográficas relevantes y se procederá a crear un documento que refleje el estado del arte y una evaluación sobre técnicas y sus librerías.

Todos los experimentos serán registrados para construir un informe técnico de los resultados obtenidos; determinando, que técnicas son las más apropiadas para procesar texto corto en español, y que librerías son compatibles y con mejor rendimiento. Se obtendrá datos de la red social Twitter, para luego experimentar sobre un conjunto de datos con técnicas y sus librerías asociadas.

2.9 Alcances y resultados esperados:

Se espera identificar cuáles son las técnicas relevantes, e implementarlas en librerías asociadas para el análisis de textos cortos en lenguaje español.

Se obtendrá un informe técnico, tablas y documentos de evaluación resultantes de la experimentación de las diferentes técnicas con sus librerías.

2.10 Supuestos y riesgos:

Riesgos	**Probabilidad**	**Alternativas de solución**
Tweets escasos del dominio planteado.	Baja	Establecer un dominio de tweets en español que sea tendencia mundial.
Acceso limitado a herramientas de las librerías a evaluar	Media	Seleccionar librerías de código abierto.
Tiempo insuficiente para la culminación del trabajo	Baja	Solicitar una prórroga para la entrega final del proyecto.

UNIVERSIDAD DEL AZUAY

2.11 Esquema tentativo:

Capítulo 1: Introducción.

Capítulo 2: Metodología de experimentación de técnicas y librerías.

Capítulo 3: Resultados de precisión y rendimiento.

Capítulo 4: Conclusiones y Recomendaciones.

Referencias

2.12 Cronograma:

OBJETIVO	ACTIVIDAD	MAYO Semana/15 horas				JUNIO Semana/15 horas				JULIO Semana/15 horas				AGOSTO Semana/15 horas				SEPTIEMBRE Semana/15 horas			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Realizar una indagación inicial de proyectos paralelos o afines al trabajo planteado.	Revisión de documentación y conceptos a utilizarse dentro del proyecto.(Búsqueda)																				
	Estado del arte																				
Elaborar un marco teórico sobre las técnicas de procesamiento de PLN relevantes y las librerías asociadas.	Revisión sobre bibliografía enfocada a librerías y técnicas de PLN.(Escritura del marco teórico)																				
Experimentar con las técnicas y librerías identificadas, por medio de pruebas con el dominio de texto planteado, identificando las técnicas relevantes y las librerías asociadas.	Obtener el conjunto de datos (extracción de tweets).																				
	Descarga e instalación de librerías																				
	Experimentar con diferentes librerías y técnicas identificadas sobre un conjunto de datos.																				
Analizar los resultados obtenidos de la evaluación de técnicas seleccionadas.	Evaluar resultados. Obtener tablas resumen de acuerdo a las métricas.																				
	Sintetizar los resultados obtenidos de cada librería.(Escritura)																				

Edición autorizada de 15.000 ejemplares
Del 835.501 al 850.500

2.13 Referencias:

Fernández, A.; Morere, P.; Nuñez, L.; & Santos, A. (2013). Sentiment Analysis and Topic Detection of Spanish Tweets: A. *Procesamiento del Lenguaje Natural*, , 45-52.

Rodrigues, M., & Teixeira, A. (2015). *Advanced Applications of Natural Language Processing for Performing Information Extraction.* Springer.

Althobaiti, M., Kruschwitz, U., & Poesio, M. (2014). AraNLP: A Java-based Library for the Processing of Arabic Text.

Eryigit, G. (2014). ITU Turkish NLP Web Service.

Hernández, J. M. (2016). *Análisis automático de textos en español.* San Critóbal de la Laguna.

Kanakaraj, M., & Mohana , R. (2015). Performance Analysis of Ensemble Methods on Twitter Sentiment Analysis using NLP Techniques. *International Conference on Semantic Computing*.

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

Omran, F., & Treude, C. (2017). Choosing an NLP Library for Analyzing Software.Documentation: A Systematic Literature Review and. *International Conference on Mining Software Repositories*.

Paramkusham, S. (2014). NLTK: THE NATURAL LANGUAGE TOOLKIT. *International Journal For Technological Research In Engineering*.

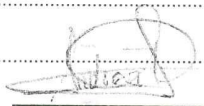
Reese, R. M. (2015). *Natural Language Processing with Java}*, Packt Publishing Ltd,.

Valverde, B., & Martínez, R. (2016). Herramientas de código abierto para el procesamiento del español.

UNIVERSIDAD DEL AZUAY

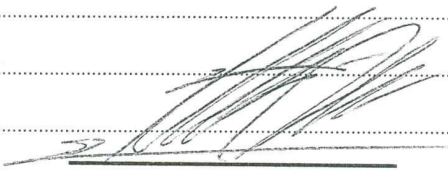
2.14 Anexos: para casos en los que se requiera respaldar el proyecto.

2.15 Firma de responsabilidad (estudiante)



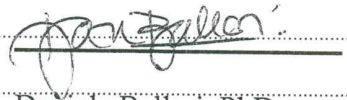
Andrea Trujillo

2.16 Firma de responsabilidad (director sugerido)



Ing. Marcos Orellana

2.17 Firma de responsabilidad (Asesor Metodológico)



Daniela Ballari, PhD

2.18 Fecha de entrega: 09/05/2018 n