ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Clasificación de artículos científicos relacionados a la agricultura de precisión con imágenes de UAV por medio de técnicas de aprendizaje profundo

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Ciencias de la Computación

Presentado por:

César Alberto León Galárraga

Pabelco Gabriel Zambrano Vélez

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

El presente proyecto va dedicado a mi madre, Yrina Gálarraga, por anteponer siempre las necesidades de sus hijos ante las suyas, por su apoyo y amor incondicional. A la Srta. Lidia Stefania Vargas por ser una guía, a través de sus consejos, de su amor y paciencia para concluir esta meta.

César Alberto León Galárraga

Este proyecto de titulación está dedicado a mi familia, pero principalmente a mis padres: Pabelco Zambrano y Paquita Vélez, que me apoyaron siempre, recordándome ser perseverante. A mi hermano: Jean Pabelco Zambrano, ya que no he sido el hermano mayor confidente y consejero que me hubiera gustado ser, por estar lejos del hogar. A la Srta. María Isabel Zambrano, quien incontables veces me ayudó a superar grandes dificultades, mediante sus escuchas, consejos, ánimos y sobre todo amor.

Pabelco Gabriel Zambrano Vélez

AGRADECIMIENTOS

Este proyecto lleva un enorme agradecimiento a cada una de las personas que aportaron con su guía y experiencia durante todos estos años de estudios universitarios y un especial agradecimiento al PhD. Miguel Andrés Realpe por su invaluable colaboración como director de este proyecto.

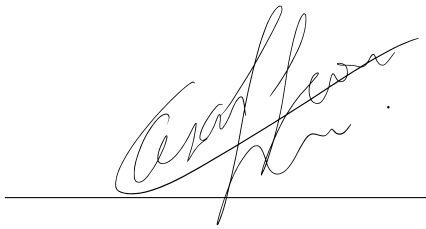
César Alberto León Galárraga

A la ESPOL y sus docentes que aportaron la mejor educación de grado que pude haber recibido, en especial al PhD. Miguel Andrés Realpe por su gran guía en el presente proyecto de titulación, y al PhD. Boris Xavier Vintimilla por la retroalimentación durante el desarrollo del presente documento. A César, mi compañero de proyecto por siempre mantener una buena relación de trabajo.

Pabelco Gabriel Zambrano Vélez

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; César Alberto León Galárraga y Pabelco Gabriel Zambrano Vélez damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



César Alberto León Galárraga

Firmado electrónicamente por:
PABELCO GABRIEL
ZAMBRANO VELEZ

Pabelco Gabriel Zambrano Vélez

EVALUADORES

Firmado electrónicamente por:
BORIS XAVIER VINTIMILLA BURGOS

PhD. Boris Xavier Vintimilla

PROFESOR DE LA MATERIA

MIGUEL ANDRES REALPE ROBALINO

Digitally signed by MIGUEL ANDRES REALPE ROBALINO
DN: c=EC, o=BANCO CENTRAL DEL ECUADOR, ou=ENTIDAD DE CERTIFICACION DE INFORMACION-ECIBCE, l=QUITO, serialNumber=0000535317, cn=MIGUEL ANDRES REALPE ROBALINO
Date: 2022.09.21 22:58:14 -05'00'

PhD. Miguel Andrés Realpe

PROFESOR TUTOR

RESUMEN

La revisión literaria consiste en encontrar trabajos similares al que se desea elaborar y discriminar si pueden ser de utilidad. Esta tarea puede tornarse lenta y tediosa, debido a que suele ser manual y extensa. Este proyecto tiene como fin, automatizar este proceso mediante la clasificación de artículos científicos relacionados a la agricultura de precisión con imágenes UAV, en tres niveles de relevancia: alto, medio y nulo; mediante un modelo clasificador de aprendizaje profundo. La solución planteada contó de cuatro módulos: la adquisición de los datos para el entrenamiento del modelo como primer módulo, en la cual el etiquetado de la data se dio de forma manual y sintética con una métrica *Ad hoc*; el segundo módulo se refiere a la minería de metadata para el cálculo de esta métrica; el tercer módulo detalla la representación del texto realizada por el modelo preentrenado de *FastText* y el último módulo corresponde a la clasificación del texto en la que se implementó una CNN 1D. Estos módulos se integraron en un prototipo de aplicación web, con la finalidad de crear una herramienta en la que se pueda realizar la clasificación de los trabajos académicos y poder visualizar los resultados. El modelo clasificador llegó a tener un *F1 score* del 87 % en pruebas con ejemplos inéditos, lo que concluye que, el modelo es capaz de realizar una preselección de alta fidelidad, optimizando el proceso de revisión literaria para el investigador sobre esta temática en particular.

Palabras Clave: Revisión literaria, Agricultura de precisión, Aprendizaje profundo, Minería de datos, Análisis de datos

ABSTRACT

The literary review consists of finding previous papers on the desired topic to be treated and discriminating if they can be useful. This task can become slow and tedious, since it's often manual and large. The purpose of this project is to automate this process by classifying scientific papers related to precision agriculture with UAV images, in three levels of relevance: high, medium and null; using a deep learning classifier model. The proposed solution had four modules: the acquisition of the data for the training of the model as the first module, in which the labeling of the data was done manually and synthetically with an Ad hoc metric; the second module refers to metadata mining for the calculation of this metric; the third module details the text representation performed by the pretrained model of FastText and the last module corresponds to the text classification in which a 1D CNN was implemented. These modules were integrated into a web application prototype, with the purpose of creating a tool in which the classification of academic works can be carried out and the results can be visualized. The classifier model reached an F1 score of 87 % in tests with new examples, which concludes that the model is capable of performing a high-fidelity preselection, optimizing the literature review process for the researcher about this particular topic.

Keywords: *Literary review, Precision agriculture, Deep learning, Data mining, Data analytic*

ÍNDICE GENERAL

ABREVIATURAS

ESPOL	Escuela Superior Politecnica del Litoral
CIB	Centro de Información Bibliotecario
CIDIS	Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales
UAV	Unnamed Aerial Vehicle
CNN	Convolutional Neural Network Application Programing
API	Interfaces
SOTA	State Of The Art
TDM	Text and Data Mining
PDF	Portable Document Format
HTML	Hyper Text Markup Language
NLP	Natural Language Processing
FAIR	Facebook Artificial Intelligence Research
DL	Deep Learning
ML	Machine Learning
JSON	Java Script Object Notation
AWS	Amazon Web Services
EC2	Elastic Cloud Computing
S3	Simple Storage Service
HTTPS	Hyper Text Transfer Protocol Secure
SSL	Secure Sockets Layer

SIMBOLOGÍA

1D	Una dimensión
MB	MegaByte
GB	GigaByte

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1 INTRODUCCIÓN

El internet es una basta fuente de información, por ende, también de una enorme cantidad de artículos científicos que se publican día a día. Dentro de esta abundante fuente de datos, lograr encontrar información relacionada y relevante a utilizar en una investigación o trabajo científico, puede ser una tarea extenuante y que consuma muchos recursos, por lo que es de vital importancia el contar con la información adecuada en la menor cantidad de tiempo posible.

Dentro del área de la agricultura de precisión, ha surgido el uso de drones o UAV (vehículos aéreos no identificados) para realizar actividades de medición, seguridad o fumigación en los campos agrícolas. Esta nueva forma de agricultura ha tenido un auge desde el 2017, lo cual convierte a este tema en un tópico de investigación relativamente nuevo. Por lo que, recolectar información mediante la búsqueda de trabajos previos relacionados con este tipo de agricultura llega a ser poco productivo.

Al acudir a un repositorio digital de artículos científicos para revisar trabajos previos sobre este tema, hay poca certeza de encontrar exactamente lo que se busca, en su mayoría de veces, se encuentran trabajos relacionados con agricultura tradicional en comparación a la de precisión.

Consecuentemente, revisar si todos los artículos científicos orientados a la agricultura de precisión son relevantes para la investigación sin ayuda alguna, tomaría mucho más tiempo de lo previsto. En su defecto, optar por herramientas computacionales que puedan identificar que artículos científicos, contienen la información más relevante para el proceso de investigación es de cuantiosa valía.

Por lo tanto, este trabajo se centra en crear una herramienta computacional que determine la relevancia de textos relacionados a la agricultura de precisión con imágenes de UAV, de una gran cantidad de documentos provistos por la biblioteca de la ESPOL.

En el capítulo I, se da a conocer en detalle la descripción general del problema y justificación, además de dar a conocer los objetivos tanto generales como específicos. Además, una breve introducción acerca de cómo está dividido la propuesta de solución.

En el capítulo II, se da a conocer a más detalle sobre las soluciones existentes de los distintos modelos computacionales para identificar textos e indicar cuál de dichos modelos presentados se ajusta para resolver el problema de la investigación.

En el capítulo III, se establece el modelo más acorde para el desarrollo de la investigación, con el fin de satisfacer con los objetivos descritos en el capítulo I. Así como también, se presentan los resultados obtenidos.

En el capítulo IV, se establece de forma definitiva si la investigación logró satisfacer los objetivos, así como también, las conclusiones y las recomendaciones para futuros trabajos.

1.1 Descripción del problema

La selección de artículos científicos de interés en una revisión del estado del arte (SOTA) en inglés sobre una temática en particular, como lo es la agricultura de precisión con imágenes de UAV, tiene como fin ser el punto de partida en el desarrollo de una investigación y/o proyecto académico.

Esta tarea se ha vuelto cada vez más extenuante debido al crecimiento de la información disponible, en consecuencia, se consumen más recursos de tiempo y personal. Por lo que, se desea realizar una clasificación de dichos artículos científicos que representen

un gran valor de relevancia, obteniendo así una tarea previa y automatizada con lo cual se ayude a optimizar el proceso de revisión bibliográfica que un investigador o estudiante realiza.

1.2 Justificación del problema

Se plantea la creación de una herramienta de procesamiento de documentos, la cual tiene como propósito principal, el clasificar un listado de artículos científicos en las siguientes categorías: alta, media y nula relevancia. La finalidad de esta clasificación es sugerir a los investigadores qué información puede ser relevante para el trabajo académico a efectuar sobre la agricultura de precisión con imágenes de UAV.

Mediante el uso de técnicas de aprendizaje profundo se desarrolla un modelo clasificación, que cuenta de dos partes: la representación de palabras a vectores, como capa de entrada para realizar una clasificación multicategórica de artículos que prediga el nivel de interés relevancia de entre los definidos anteriormente. Obteniendo así, una selección eficiente de artículos científicos relacionados a la agricultura de precisión con imágenes UAV.

1.3 Objetivos

1.3.1 Objetivo general

Crear un modelo que clasifique artículos científicos en inglés relacionados a la agricultura de precisión con imágenes UAV.

1.3.2 Objetivos específicos

- ❒ Generar una metodología para el etiquetado sintético de artículos científicos, a fin de complementar el etiquetado manual del conjunto de datos para el entrenamiento supervisado.

- ❒ Aplicar un modelo de representación de palabras, a través de la transferencia de aprendizaje, a fin de generar los vectores de la capa de entrada para el modelo de clasificación.

- ❒ Generar un modelo de clasificación de aprendizaje profundo, para categorizar los artículos científicos relacionados a la agricultura de precisión con imágenes UAV, en las clases de alta, media o nula relevancia.

1.4 Módulos del proyecto

Con la finalidad de proporcionar más información sobre cómo está compuesta la herramienta de análisis de texto, a continuación, se provee un breve resumen sobre la división de módulos establecidos.

1.4.1 Adquisición de los datos

En esta sección se da a conocer los distintos repositorios, información para el preprocesamiento de los datos y conformación del conjunto de datos inicial para del modelo de representación de palabras.

1.4.2 Minería de datos

En esta sección se da a una breve explicación de lo que supone la búsqueda de información acerca de piezas científicas en formato PDF, ofreciendo así indicadores cualitativos y cuantitativos que sirven para la conformación de la métrica *Ad hoc*.

1.4.3 Representación del texto

En esta sección se da a conocer algunos de los modelos de representación de palabras mayormente conocidos, los cuales interpretan los corpus de un documento y los convierten a vectores numéricos. Se usan estos vectorizadores de texto mediante el llamamiento de librerías, es decir, se aplica una transferencia de conocimiento, puesto que,

estos modelos ya han sido adecuadamente entrenados, así sus vectores resultantes son las entradas para el modelo de clasificación de texto.

1.4.4 Clasificación del texto

En esta sección se da a conocer algunos de los modelos de clasificación de datos mayormente conocidos, los cuales discriminan los datos de entrada y los agrupan en clases. Se usan estos clasificadores mediante el llamamiento de librerías, es decir, se aplica una transferencia de conocimiento y se realiza un ajuste fino sobre estos para acomodarse a las categorías establecidas y a las particularidades del conjunto de datos de entrada.

1.5 Marco teórico

El presente capítulo da a conocer los distintos módulos en los cuales está dividida la investigación, enfocándose en cuáles serán los datos utilizados y de donde se obtuvieron. También se describe los principales modelos de aprendizaje profundo que existen y su respectivo flujo de trabajo entre los distintos módulos.

1.5.1 Adquisición de los datos

Los datos y/o artículos científicos que fueron utilizados para los entrenamientos de los módulos a desarrollar son obtenidos a partir de la amplia base de datos provista por el CIB – ESPOL, la cual cuenta con acceso a alrededor de 7 repositorios digitales entre los cuales están: *ACM digital, IEEE Xplore, Science Direct, Scopus, Springer, Web of Science y Taylor & Francis.* En la que todas las búsquedas son en el idioma inglés y contienen al menos una de las siguientes palabras claves: *agriculture, precision y UAV.*

Puesto que los artículos científicos provienen de repositorios de gran renombre, los cuales ya cuentan con un riguroso sistema de verificación y poseen diversos certificados sobre la información que alojan, se puede asegurar que la información utilizada es 99 %

confiable, por lo que no se evalúa si dichos artículos pueden ser apócrifos.

El etiquetado de los artículos que son obtenidos desde las fuentes mencionadas se dio de manera manual y sintética para determinar su nivel de relevancia en la escala de clasificación previamente establecida. Dicha recolección de los artículos arroja una muestra semi-uniformemente distribuida para las categorías de alta, media y nula relevancia.

Se delimita a una muestra final de 84 artículos científicos asociados a la agricultura de precisión, los cuales van a ser sometidos a un preprocesamiento, que se compone de dos partes: la tokenización del texto y su posterior limpieza; extrayendo fórmulas, ecuaciones, figuras, tablas y las *stopwords* propias del idioma inglés. De forma que se consolidan arreglos de cadenas de caracteres que pasan a ser el conjunto de datos inicial para el modelo de representación de palabras escogido.

1.5.2 Minería de datos

La minería de datos es un proceso por el cual grandes volúmenes de datos son analizados, con la finalidad de encontrar patrones o relaciones entre los elementos [1], dichos patrones son recolectados y representados de forma estadística al observador.

Existen muchos algoritmos que proveen análisis de datos, pero en su mayoría son mediante la utilización de bases de datos llamadas *Werehouse*, o una base de datos relacional. Para la elaboración de este proyecto, se hace uso de analizadores de texto plano, conocidos como *Text and Data Mining* (TDM) [2]

Los TDM, son analizadores de texto en documentos de gran magnitud, la cual es mucho más rápida y eficiente en sus búsquedas de texto en documentos de extensión PDF o HTML que una técnica de minado de datos tradicional. Por lo que realizar la búsqueda de una palabra en específico es mucho más eficiente en este tipo de analizadores.

1.5.3 Representación del texto

La representación de palabras es una componente importante de los modelos de clasificación de aprendizaje profundo debido a que proporcionan la entrada para las siguientes capas en el procesamiento natural del lenguaje (NLP), como lo es el etiquetado de secuencias y en específico la clasificación de texto. En la última década, con el avance a pasos agigantados que ha tenido esta subárea de la inteligencia artificial, se han propuesto un gran número de técnicas para la representación de textos [3].

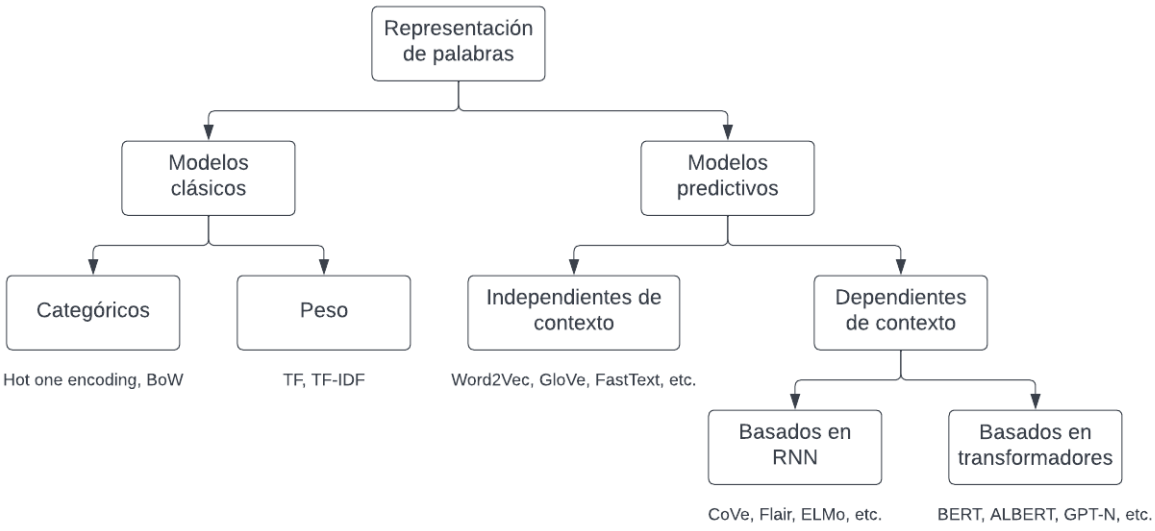


Figura 1.1: Taxonomía de los modelos de representación de palabras [4], [5]

Generalizando, las técnicas de representación de texto se pueden agrupar en dos modelos, por medio de la diferenciación en las metodologías que utilizan para asignar un vector a una palabra, como se denota en la Figura 1.1. El primer grupo corresponde a las técnicas de conteo o frecuencia conocidas como modelos clásicos, que se subdividen en categóricas y de peso, donde se encuentran métodos como *One hot encoding* y bolsa de palabras (BoW), y para la segunda están la frecuencia de términos (TF) y *term frequency-inverse document frequency* (TF-IDF) [4].

El segundo grupo de técnicas de representación de palabras son de clase predictiva,

que son los modelos que descubren las características que vectorizan a un texto por sí mismos. A su vez, estos se dividen de dos tipos, que son los independientes del contexto y los dependientes del contexto. Las primeras conocidas como clásicos, ya que aprenden representar el texto mediante modelos de lenguaje basados en redes neuronales superficiales o factorización de matrices de co-ocurrencia [3]. De forma que, estos modelos de representación al no estar sujetos a un contexto, producen vectores únicos para cada palabra, entre sus modelos más prominentes están *Word2Vec*, *GloVe* y *FastText* [3], [4] .

En su defecto, los modelos de representación de palabras dependientes del contexto aprenden de manera distinta a como vectorizar una palabra, por tal motivo, una palabra puede tener múltiples representaciones de acuerdo con el contexto. En los años recientes este tipo de representaciones han ganado popularidad, lo que ha llevado a que naturalmente sigan la tendencia de las redes neuronales y también se dividan en dos: basadas en redes neuronales recurrentes como *CoVe*, *Flair* y *ELMo*, o en transformadores como *BERT*, *ALBERT* y *GPT-N* [4], [5].

La gran mayoría de estos modelos predictivos de representación ya se encuentran preentrenados y/o implementados en librerías, paquetes o APIs (como *GPT-N*) listos para su aplicación a una tarea en particular. Lo que corresponde a una transferencia de aprendizaje que tiene como objetivo el desarrollar el conocimiento en específico para otra tarea posterior.

En la transferencia de aprendizaje se pueden distinguir tres enfoques [6]:

- ❒ Extracción de características: El modelo preentrenado solamente devuelve las representaciones compactas en función de los datos específicos del dominio.
- ❒ Ajuste fino: El conocimiento obtenido por el modelo preentrenado se ajusta a una tarea o conjunto de datos en específico.
- ❒ Cálculo pérdidas de características: Las representaciones obtenidas, a través de un modelo previamente entrenado, se utilizan para calcular pérdidas al entrenar sistemas de aprendizaje profundo.

La revisión literaria recomienda que para la tarea de representar palabras se utilice

la transferencia de aprendizaje enfocada a la extracción de características. Es así que, teniendo a artículos académicos que corresponden a una temática como conjunto de datos de entrada, se apunte al uso de un modelo de una representación no contextual como lo es *FastText* [7].

FastText es una librería creada por el laboratorio *Facebook AI Research* (FAIR), en el que se dispone del modelo preentrenado de representación de palabras con el corpus de Wikipedia, con un enfoque basado en el modelo *skipgram*, donde cada palabra se representa como una bolsa de n-gramas de caracteres. A cada n-grama de carácter se le asocia una representación vectorial; las palabras se representan como la suma de estas representaciones. Obteniendo representaciones de palabras que toman en cuenta la morfología de las mismas [8].

1.5.4 Clasificación del texto

La clasificación de texto cuenta con un rico SOTA, en el que sobresalen los algoritmos de aprendizaje de máquina, que se pueden clasificar en: de aprendizaje superficial y de aprendizaje profundo. Siendo que, algoritmos comunes para el primer grupo son: naïve bayes (NB), *support vector machine* (SVM), regresión logística (LR), basados en árboles de decisión (DT) y bosque aleatorio (RF). En contraparte, los algoritmos de aprendizaje profundo (DL) como: redes recurrentes neuronales (RNN) en forma de *long short-term memory* (LSTM) y *gated recurrent unit* (GRU) y redes neuronales convolucionales (CNN)[4].

Los modelos basados en DL han alcanzado resultados SOTA en muchas áreas diferentes, incluido NLP, pero requieren de una buena representación semántica de los datos textuales. Las principales arquitecturas de DL que se utilizan comúnmente en cualquier tarea de clasificación de texto, donde se analizan brevemente a continuación:

- ❒ RNN: Es un modelo en donde los datos anteriores que representan los puntos de una secuencia, se les asigna más peso en un modelo RNN, lo que lo hace óptimo para cualquier texto, cadena o clasificación de datos secuenciales. Obteniendo

resultados superiores para la semántica en el análisis de un corpus.

- ✧ LSTM: Tiene como fin el abordar los problemas de descenso de gradiente de RNN, manteniendo a lo largo de la red la dependencia de un término de mejor manera que con respecto a RNN. Aunque los LSTM tienen una arquitectura de cadena, se diferencian de RNN al tener diferentes compuertas que manejan el volumen de información.
- ✧ GRU: Es la forma más simple de arquitectura LSTM. Sin embargo, incluye dos compuertas y no contiene memoria interna lo que lo diferencia de LSTM. Además, en GRU no se aplica una segunda no linealidad (tanh) en la red.

❒ CNN: Inicialmente fueron construidas y utilizadas para clasificación de imágenes, pero también han mostrado excelentes resultados para la clasificación de datos secuenciales como lo es el texto. En la clasificación de imágenes, un tensor de imagen se convoluciona con un conjunto de núcleos de tamaño *dxd*. Las capas de convolución en la CNN se conocen como mapas de características que se pueden apilar para tener varios filtros. Para superar el problema computacional debido al tamaño de dimensionalidad, las CNN utilizan una capa de agrupación (*pooling*) para reducir el tamaño de una capa a otra [4].

Las CNN han mostrado resultados excepcionales para encontrar los patrones en la examinación de datos secuenciales como texto y series temporales, por medio de operaciones de convolución *1D*. Más específicamente, en tareas como la detección de tópicos en corpus de artículos académicos provenientes de la base de arXiv. Es así que, la combinación de CNN como modelo de clasificación, donde se usa capas convolucionales *1D*, dos capas ocultas con pérdida de entropía cruzada y un optimizador Adam, y *FastText* como modelo de representación; se obtienen los mejores resultados con una precisión de 96 %, en comparación con todos los modelos de clasificación de aprendizaje profundo descritos anteriormente, utilizando la misma técnica de representación de palabras [7].

En el siguiente capítulo se da a conocer a mayor detalle cuales fueron los requerimientos funcionales y no funcionales, además de los criterios tomados en cuenta para el desarrollo de la herramienta y la metodología utilizada.

CAPÍTULO 2

2 METODOLOGÍA

2.1 Requerimientos del sistema

En esta sección se detalla el proceso llevado a cabo para el desarrollo de la herramienta de clasificación de los artículos científicos relacionados con la agricultura de precisión, de forma que, se identifican las principales necesidades de los usuarios involucrados, es decir, los requerimientos funcionales y no funcionales. Además, se da a conocer el diseño de solución y la metodología utilizada para la elaboración de este proyecto.

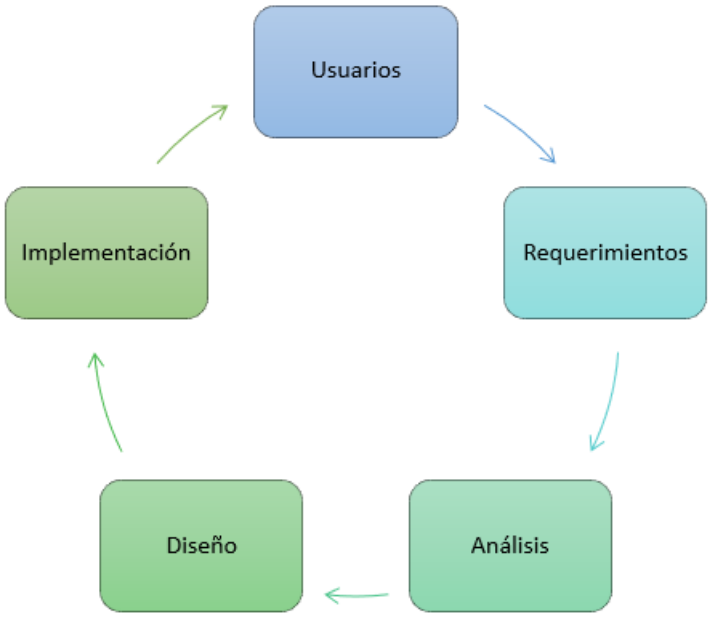


Figura 2.1: Ciclo del análisis y diseño del sistema [Elaboración propia]

En la Figura 2.1, se aprecia el ciclo de vida del desarrollo de la investigación, donde se indica a grandes rasgos como se ha llevado a cabo el proceso de diseño e implementación de la solución de la herramienta.

Para el desarrollo de este proyecto, se utilizó la metodología de trabajo *Scrum* [9], mediante la herramienta *ClickUp*; la que tiene como eje central, la constante comunicación entre los actores del proyecto, por medio de reuniones recurrentes entre los estudiantes, cliente (CIDIS - ESPOL) y tutor para brindar el soporte y retroalimentación por cada uno de los avances planificados durante todo el desarrollo del proyecto.

Debido a esto se identificaron, los principales requerimientos alineados a los objetivos planteados, además de obtener la información necesaria para seleccionar cuáles serían las herramientas de desarrollo y los recursos computacionales más adecuados para cumplir con las necesidades del cliente.

2.1.1 Requerimientos funcionales

Estos requerimientos son aquellas características que debe presentar la herramienta clasificadora para cumplir con el objetivo general del proyecto y por ende satisfacer la necesidad principal del cliente. De forma que, para establecer dichas funcionalidades, se vinieron realizando reuniones a fin de determinar cómo debería funcionar la herramienta.

Se establecieron cuáles serían los principales involucrados al momento de utilizar la herramienta, tal como se muestra en la siguiente tabla:

Tabla 2.1: Roles del sistema [Elaboración propia]

Usuario	Descripción
Investigador	Los usuarios de la herramienta pueden colocar un listado de archivos en PDF, lista para ser clasificada.
Desarrollador	Los estudiantes se encargan de desarrollar, mantener y mejorar la herramienta.

En la Tabla 2.1, se muestra que los 2 principales actores son los investigadores o estudiantes, los cuales deben de seleccionar su base de datos de artículos científicos, la cual será evaluada, y los desarrolladores evalúan y analizan el desempeño de la herramienta para seguir mejorando el producto a medida que pasa el tiempo.

2.1.2 Requerimientos no funcionales

Son las prestaciones que se espera que tenga la herramienta, no como un elemento para resolver el problema, sino más bien que tan eficiente sea esta, por ejemplo, la mejora en el rendimiento y la productividad de los usuarios del sistema de clasificación. Para esto se utilizó *clusters* en la nube para alojar y ejecutar la herramienta, con el fin de optimizar recursos, por tal motivo no fue necesario el uso *hardware* local, reduciendo así costos de mantenimiento para un futuro cercano.

Sin duda, el aspecto más importante es la eficacia de la herramienta, la cual es el valor agregado con el que contribuye esta investigación, por tal motivo, los modelos deben llegar a una métrica de *F1 Score* de al menos el 85 % en pruebas.

$$F_1 Score = \frac{\text{Precisión} \times \text{Exhaustividad}}{\text{Precisión} + \text{Exhaustividad}} \tag{2.1}$$

Donde la precisión es:

$$Precisión = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Verdaderos Negativos}} \tag{2.2}$$

la exhaustividad es:

$$Exhaustividad = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}} \tag{2.3}$$

Otro aspecto para tener en cuenta es el funcionamiento de la herramienta clasificadora, la cual debe ser lo más rápida posible, teniendo en cuenta que el procesamiento de datos mediante el uso de estas técnicas puede llegar a ser extensivo.

2.2 Diseño del sistema

El prototipo de solución se dividió en dos grandes secciones dentro de una misma aplicación, la primera sección se centra en analizar cada uno de los archivos recibidos,

utilizando los modelos de representación y de clasificación de texto, y la segunda sección es la presentación de los resultados en orden según la valoración obtenida, tal como se aprecia en la Figura 2.2.

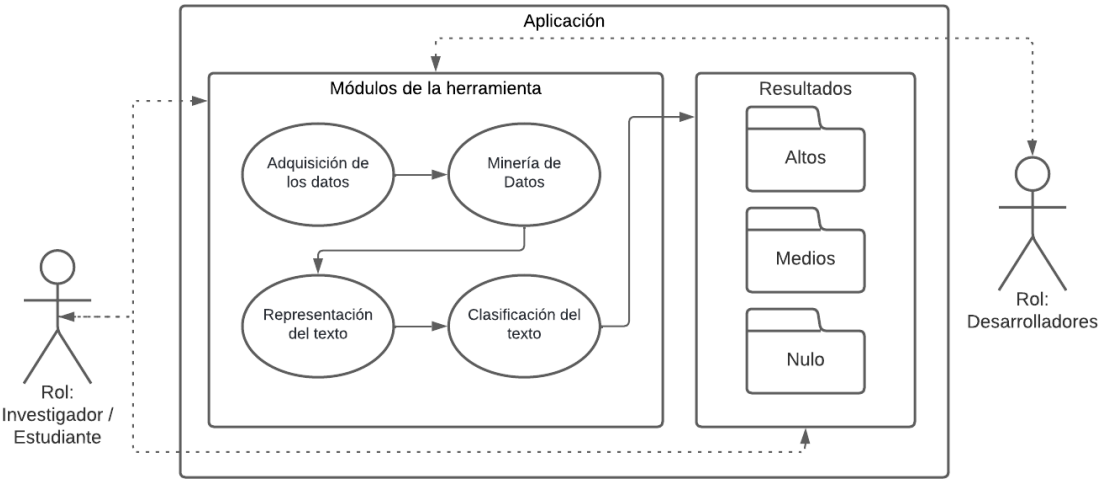


Figura 2.2: Diagrama de la herramienta diseñada [Elaboración propia]

Para esta primera sección de la representación de los datos, se va a utilizar técnicas de aprendizaje profundo para elaborar un sistema que logre determinar según ciertas características previamente establecidas, ordenar los distintos artículos científicos a la clasificación correspondiente. A continuación, se detalla cómo se llevaron a cabo cada uno de los módulos del proyecto.

2.2.1 Criterios de diseño

La clasificación de artículos científicos por medio de técnicas DL se enmarca en la subárea de clasificación de propósito de NLP. El SOTA en este subcampo establece que el ML debe ser supervisado, es decir, que el modelo DL que se implemente aprenda a realizar clasificaciones basadas en ejemplos etiquetados, que cumplan los datos de entrenamiento, de tal forma que, la etiqueta dada a los textos sea la clase o categoría predeterminada en la que podría caer cualquier artículo científicos que fuera analizado por el clasificador.

Sin embargo, para llegar a obtener modelos que analicen la data de entrada, en este

caso artículos científicos relacionados a la agricultura de precisión con imágenes UAV, es necesario la implementación de un diseño que detalle la consecución del objetivo general que este trabajo plantea.

Este desarrollo se da por etapas, los cuales corresponden con los módulos definidos en este proyecto.

Tabla 2.2: Fases de desarrollo del proyecto según sus módulos [Elaboración propia]

Sprint	Fase	Módulo
1	Consolidación del conjunto de datos a partir de la entrada etiquetada.	Adquisición de los datos
1	Preprocesamiento del conjunto de datos consolidados.	Adquisición de los datos
2	Extracción de identificador para consulta de metadata	Minería de datos
3	Extracción de los vectores a partir del conjunto de datos preprocesados.	Representación del texto
4	Clasificación de los vectores acorde a las clases definidas.	Clasificación del texto

En la Tabla 2.2, se muestra la correspondencia de las fases o etapas en el desarrollo de la propuesta de solución con los módulos del proyecto y *sprints*, según la metodología de trabajo de *Scrum*. En las cuales, las duraciones de los dos primeros *sprints* son de una semana y los otros de dos semanas.

2.2.1.1 Consolidación del conjunto de datos a partir de la entrada etiquetada

El conjunto datos inicial con el que se ha trabajado es de 10 artículos en inglés por cada categoría de relevancia definida, en total 40 artículos científicos, todos provenientes de la base especializada de *Springer*, por lo que su formalidad y rigurosidad es alta. Este etiquetado, se ha hecho manualmente a partir de los juicios de valor del cliente (experto en la agricultura de precisión).

- ❒ **Construcción de la tabla delta inicial:** La tarea manual de etiquetado es de naturaleza incremental, por lo que, debido a la facilidad que brindan las tablas delta para llevar el control de los cambios a lo largo del tiempo, se optó por la utilización

de estas tablas, para el almacenamiento estructurado de los artículos científicos etiquetados, siguiendo el esquema mostrado en la Tabla 2.3

Tabla 2.3: Esquema que poseen las tablas delta [Elaboración propia]

ID	*Digital Object Identifier* (DOI)
Category	Nivel de relevancia: alta, media o nula
Title	Texto respectivo
Abstract	Texto respectivo
Introduction	Texto respectivo
Methodology	Texto respectivo
Results	Texto respectivo
Conclusions	Texto respectivo
Acknowlodgements	Texto respectivo
References	Texto respectivo

La Tabla 2.3, muestran en conjunto las cabeceras de todas las tablas delta que se han utilizado para manejar las versiones del conjunto de datos. Además, desde la columna de *Title* a *References*, mapean la estructura bien definida que tiene cada artículo científico.

La fragmentación del artículo científico tiene por objetivo el que cada sección de análisis sea tratada como un corpus, en particular para que la etapa de preprocesamiento se realice en paralelo, optimizando el tiempo que conlleva esta.

- ❒ **Construcción de la tabla delta tokenizada:** El esquema de la tabla descrita en la Tabla 2.3 se mantiene, pero se crea una copia de esta a la que se tokeniza, desde la columna de *Title* hasta *Acknowlodgements*, de forma que se mantiene la data inicial cruda, o sea, sin ningún afinamiento, mientras se tiene otra tabla que sirve para no repetir el proceso de tokenización sobre todos las filas de la tabla delta inicial.

2.2.1.2 Preprocesamiento de la tabla delta tokenizada

Sobre esta tabla se sigue un flujo de pasos que tiene como objetivo el estandarizar el corpus de la data, a fin de lograr una óptima representación del texto por parte del modelo preentrenado de vectorización. Este flujo sobre los datos se ilustra en la Figura

2.3 descrita a continuación.

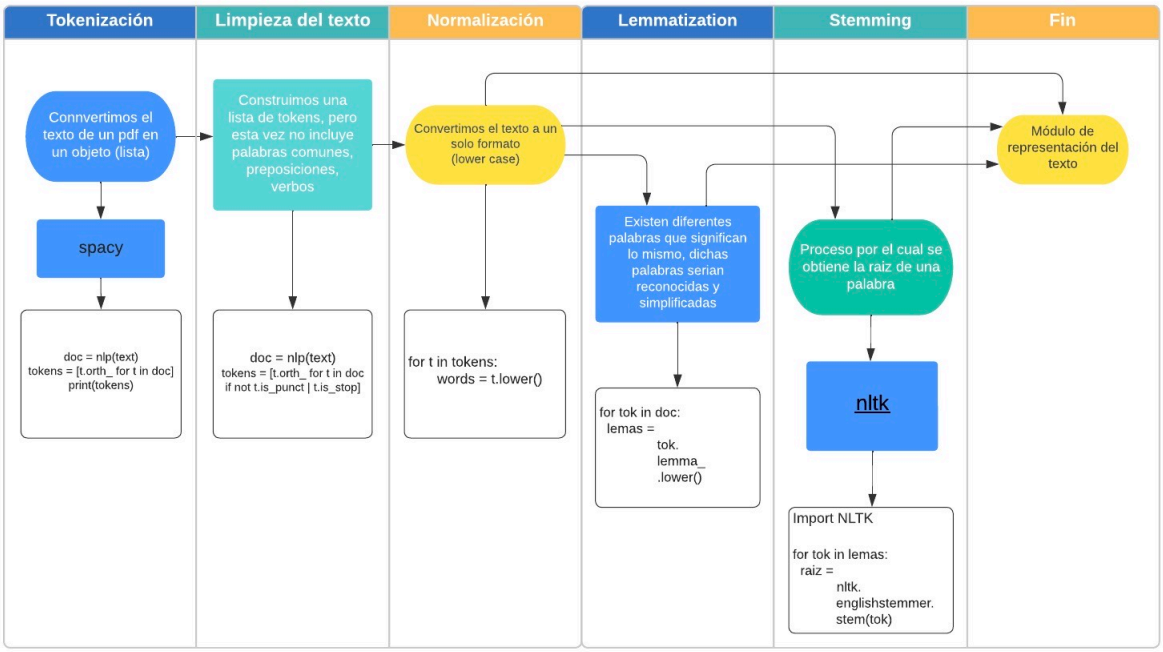


Figura 2.3: Flujo del preprocesamiento de la tabla delta tokenizada [Elaboración propia]

En la Figura 2.3 se muestra que en el flujo hay una triple bifurcación a partir de la etapa de normalización, esto es debido a que se evaluaron los niveles de efectividad de las técnicas de reducción de caracteres en las palabras, en comparación con los tokens completos. Lo que derivó en la creación de dos tablas adicionales asociadas a cada uno de los procesos respectivamente.

2.2.1.3 Extracción de identificador para consulta de metadata

Una vez construida la tabla delta tokenizada, tal coetadao se indica en la Tabla 2.3, se extrae el identificador (DOI), el cual pasa a ser un parámetro de referencia a la API de *Crossref*, para minar toda la *metadata* disponible.

Esta API es provista la organización de *Crossref*, de la que forman parte los principales repositorios con los cuales se trabajó. Los distintos repositorios proveen información sobre la cantidad de veces descargada, referencias emitidas, editoriales, entre otros

datos, los cuales son registrados para proveer *metadata* a aquellos investigadores que buscan información adicional sobre los documentos a los cuales revisan.

Para la elaboración de este proyecto se utilizó una librería más cómoda, la cual tiene métodos muchos más sencillos y legibles para su funcionamiento. Dicha librería es provista por habanero desarrollada para *Python*.

La finalidad de utilizar estos datos es crear una métrica que indique el peso que tiene un documento con relación a los años publicados. Es decir, un documento publicado más atrás en el tiempo, no debería ser un factor que indique mayor relevancia, y así mismo, un documento más actual no debería ser un factor que indique baja relevancia para una investigación.

2.2.1.4 Extracción de vectores a partir de las tablas tokenizadas

En esta fase todas las secciones tokenizadas se vuelven a unir para formar un solo documento que pasa por el modelo de extracción de *FastText* [7] que vectorizó los tokens como se muestra en la Figura 2.4.

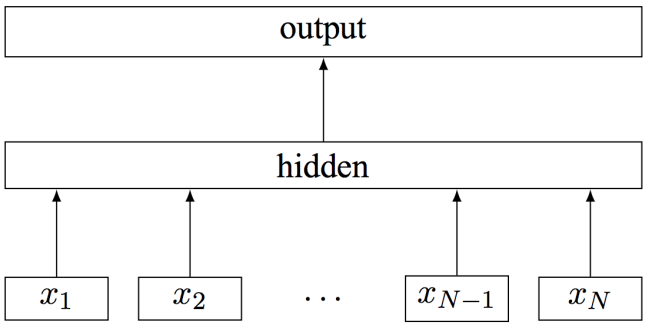


Figura 2.4: Arquitectura de FastText [8]

2.2.1.5 Clasificación de los textos a partir de los niveles de relevancia

En esta fase se creó la CNN 1D, de la forma descrita en la Figura 2.5, la cual se entrenó y llevo el control del modelo de a través de *MLFlow* del entorno de *Databricks*. El

entrenamiento se dio con los siguientes hiperparámetros: *Learning Rate* (LR) de 0,001, Adam como optimizador y 22 épocas, los cuales se definieron a partir de la convergencia del aprendizaje del modelo.

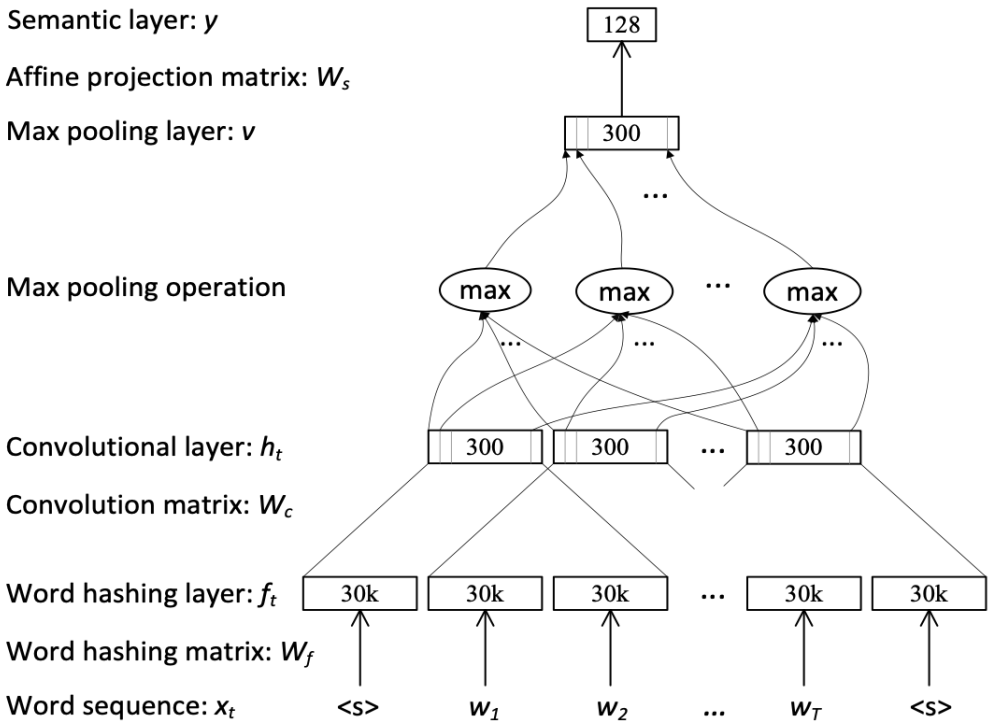


Figura 2.5: Arquitectura de la CNN 1D [8]

2.2.2 Selección de recursos

Para el preentrenamiento de las técnicas de aprendizaje profundo, se seleccionó la plataforma de *Azure Databricks* de *Microsoft*, la cual es una plataforma que permite realizar procesamiento de datos a gran escala, de igual manera, permite realizar análisis de datos según los criterios que los desarrolladores necesiten.

Se utilizó el lenguaje de programación de *Python*, debido a que existen numerosas librerías de aprendizaje profundo previamente entrenadas y de código abierto, haciendo uso de las mismas mediante llamamiento a dichas librerías.

Para convertir todos los artículos científicos a un formato de representación más sen-

cillo, se procedió a utilizar la librería de *Spacy* [10], que es una librería de procesamiento de lenguaje natural, la cual ya cuenta con una amplia documentación sobre su uso y eficiencia con alrededor de 60 lenguajes.

Los resultados obtenidos por *Spacy* fueron transmitidos a un modelo de representación de palabras provisto por la librería de *FastText* [7], la cual cuenta con una amplia variedad de modelos de representación de texto en vectores, en varios idiomas y cuenta con bastante documentación sobre su uso. Para finalizar el modelo de clasificación se implementó ML del tipo CNN con capas de convoluciones 1D, creándolas con *Tensorflow Keras*.

En el siguiente capítulo se da a conocer los resultados y los respectivos análisis obtenidos durante el desarrollo de la herramienta de clasificación utilizando la metodología descrita en este capítulo.

CAPÍTULO 3

3 RESULTADOS Y ANÁLISIS

En este capítulo se presentan los resultados obtenidos luego del desarrollo de la propuesta de solución, descrita en la metodología. Además, se detalla el prototipo de la herramienta de clasificación de artículos científicos de agricultura de precisión, que integra todos los módulos del proyecto y su despliegue en un ambiente de cuasi-producción.

3.1 Módulos de la solución

El proyecto fue modularizado para que el desarrollo se ajustara de la mejor manera al marco de trabajo *Scrum*; y por consiguiente, se logró que la solución se implementara con un enfoque de independencia modular, para que los cambios o ajustes en parámetros y/o subprocesos no requieran una refactorización de todo el flujo de entrenamiento o inferencia para el modelo de clasificación. Del mismo modo, este enfoque posibilitó la efectiva reutilización de grandes secciones de código para la integración del modelo de clasificación para con la aplicación web, que finalmente conformaron el prototipo funcional.

3.1.1 Adquisición de los datos

En el módulo descrito a continuación, se establece la cantidad de pruebas realizadas y con cuantos documentos utilizados de muestras y sus respectivos resultados.

3.1.1.1 Consolidación del conjunto de datos

En la consolidación del conjunto de datos de entrada, se llevó a cabo un proceso iterativo, debido al gradual aumento de nuevos artículos que iban siendo etiquetados de forma manual en 4 niveles de relevancia (alto, medio, bajo y nulo); hasta que se comenzó con

un etiquetado sintético a partir de la métrica *Ad Hoc* (propuesta en la metodología) para 3 clases (alta, media y nula relevancia). Cabe destacar, que esta evolución del conjunto de datos se fue dando a medida que los entrenamientos del modelo de clasificación se iban afinando con el fin de llegar a la métrica de *F1 score* propuesta como requerimiento no funcional.

Tabla 3.1: Iteraciones del conjunto de datos del proyecto [Elaboración propia]

Iteración	Descripción	Alta	Media	Baja	Nula	Total
1.0	Artículos etiquetados manualmente	5	5	5	5	20
2.0	Artículos etiquetados manualmente	10	10	10	10	40
3.0	Artículos etiquetados manualmente y sintéticamente a través de la métrica *Ad hoc*	24	20	21	23	88
4.0	Artículos etiquetados manualmente, sintéticamente a través de la métrica *Ad hoc* y unión de las categorías 'Media' y 'Baja'	30	27	0	27	84

En la Tabla 3.1, se observan 4 iteraciones del conjunto de datos consolidado. La primera iteración tiene un total de 20 artículos científicos etiquetados manualmente y distribuidos uniformemente entre los 4 niveles de relevancia (alta, media, baja y nula). En la segunda iteración se duplica la cantidad artículos pasando a un total de 40 y también uniformemente entre sus clases. Para la tercera iteración, el aumento de 48 ejemplares por sobre la segunda iteración, se da por medio del etiquetado sintético, a través de la definición de límites arbitrarios con base en la distribución que presentaban aquellas piezas con la métrica *Ad hoc*.

Por último, la cuarta iteración responde al re-etiquetado por la unión de las clases media y baja, ya que el mejor modelo entrenado para la iteración anterior tendía a confundirse entre las dos categorías mencionadas. Por tal motivo, en la Tabla 3.1 se evidencia un valor de cero para la categoría 'baja', debido a que los artículos previamente etiquetados en los 4 niveles definidos fueron reducidos a 3, y los artículos de esta clase fueron distribuidos entre las categorías media y nula, llegando a tener en total 84 documentos, 30 muestras para alta, 27 para media y 27 para nula relevancia.

3.1.1.2 Preprocesamiento del texto

Los corpus que fueron preprocesados pasaron por el mismo flujo de tokenización y su posterior limpieza para cada una de las secciones en las que fueron divididos los documentos. Sin embargo fueron sometidos a una prueba extra donde se extrajo una copia de los *tokens* generados para pasar a una serie de técnicas de normalización como lo son: lematización, *stemming* y una copia sin alteración alguna de los *tokens*, descrita mediante la Figura 2.3, para ser evaluados de forma independiente tal como se muestra en la siguiente Tabla 3.2.

Tabla 3.2: Iteraciones de los mejores entrenamientos según las técnicas de normalización de los *tokens* [Elaboración propia]

Entrenamiento	Iteración del conjunto de datos (tabla 3.1)	Técnica	*F1 score*
1.a	1.0	Ninguna	0,57
1.b	1.0	Lematización	0,52
1.c	1.0	*Stemming*	0,54
2.a	2.0	Ninguna	0,66
2.b	2.0	Lematización	0,67
2.c	2.0	*Stemming*	0,64
3.a	3.0	Ninguna	0,79
3.b	3.0	Lematización	0,78
3.c	3.0	*Stemming*	0,79
4.a	4.0	Ninguna	0,94
4.b	4.0	Lematización	0,89
4.c	4.0	*Stemming*	0,90

En la Tabla 3.2, la primera columna hace referencia al código del mejor entrenamiento efectuado para el modelo de clasificación, a partir de la iteración del conjunto de datos (segunda columna) y la técnica utilizada en la normalización de los *tokens*. De manera que, este código se conforma por el primer número de iteración (descrita en la Tabla 3.1) y la asignación de una letra según la técnica.

Además, se puede evidenciar que el valor de la métrica *F1 score* es mayor cuando no se utiliza ninguna técnica de normalización. Por tal motivo, empíricamente el aplicar lematización o *stemming* a los *tokens* no aporta significativamente al mejoramiento del modelo en entrenamiento. Esto se debe en gran parte, a que el vectorizador de *FastText* es bastante robusto y no necesita de que las palabras pasen por un proceso previo a su

representación.

3.1.2 Minería de datos

En el proceso de re-etiquetado del conjunto de datos final, se estableció la utilización de una métrica *Ad hoc*, donde se evaluó su distribución (Figuras 3.1 y 3.2) para asentar los límites a tomar, a fin de lograr una distribución aproximadamente uniforme para los 3 niveles finales. Teniendo como consideración de que los artículos etiquetados de forma manual permanezcan en sus categorías, dado que en el caso en que su respectiva métrica *Ad hoc* cayera dentro del intervalo designado para una categoría diferente.

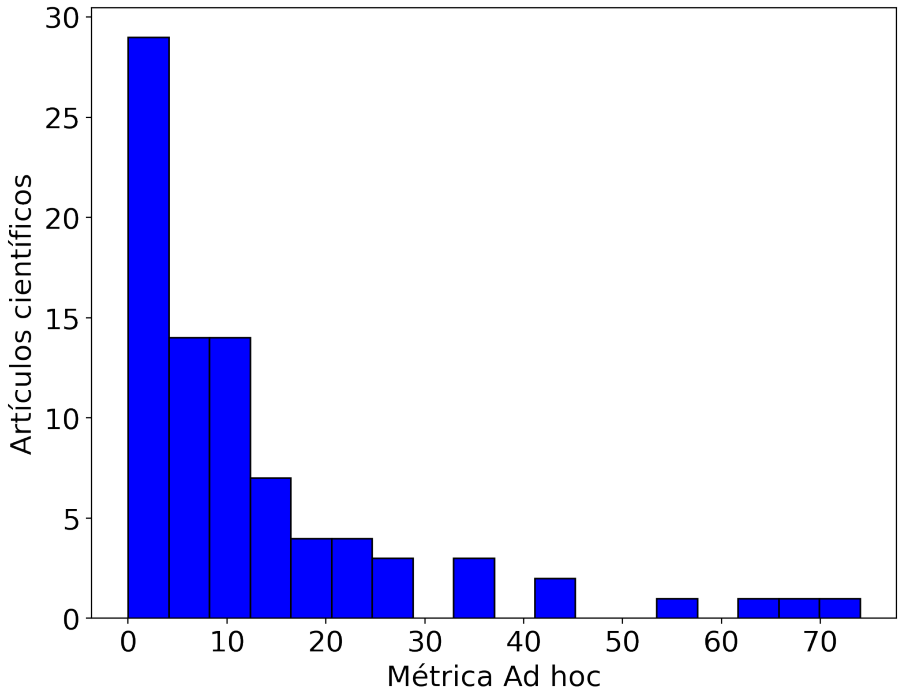


Figura 3.1: Histograma de métrica Ad hoc de los artículos científicos [Elaboración propia]

En la Figura 3.1, se muestra la distribución de los 84 artículos científicos con base en la métrica *Ad hoc*, calculada a partir de la minería de la *metadata* de los documentos (descrita en la sección de la metodología). Donde se puede observar que algo menos de 30 son los trabajos académicos que tienen un valor por debajo de 4; de la misma forma, se ve que alrededor de 30 documentos están entre valores de 4 y 11; y para concluir

existen aproximadamente 25 piezas de información desde el valor de 11 en la métrica.

En síntesis, la distribución es la esperada puesto que esta métrica tiene como consigna el clasificar a los artículos en relación con su relevancia académica, por lo que se ajusta a la realidad, ya que la mayoría de los trabajos publicados no llegan a la cima de la comunidad científica en términos de veces referenciados, vigencia a lo largo de los años e impacto en el *journal* donde fueron publicados.

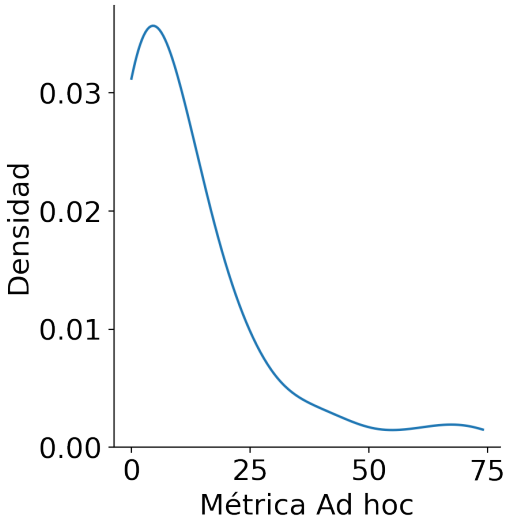


Figura 3.2: Densidad de la métrica Ad hoc en los artículos científicos [Elaboración propia]

En la Figura 3.2, se denota que la aproximación continua de la función de densidad para la distribución discreta de la métrica *Ad hoc*, tiene características de ser leptocúrtica y asimétrica positivamente, por lo que considerando el teorema de límite central, tendería a la *Normal*; a primera instancia se podría inferir que es una aproximación a la distribución de *Poisson* con una lambda (λ) de 12, puesto que su media (μ) y su varianza (σ) también es 12.

3.1.3 Representación del texto

En la representación del texto y más específicamente la vectorización de los *tokens* inalterados por medio de *FastText*, este proceso se tuvo que volver una capa de *embeddings* para CNN clasificadora. Esto con el objetivo de que sea fácilmente exportable el

modelo para la tarea de inferencia en la integración con el prototipo.

En la exploración descriptiva de los *tokens*, el modelo de representación logró identificar un vocabulario de 53246 *items* únicos, que al ser transformados en secuencias de caracteres para cada uno de los corpus denotaron la siguiente distribución en el tamaño de estas secuencias para su vectorización y posterior transformación a tensor.

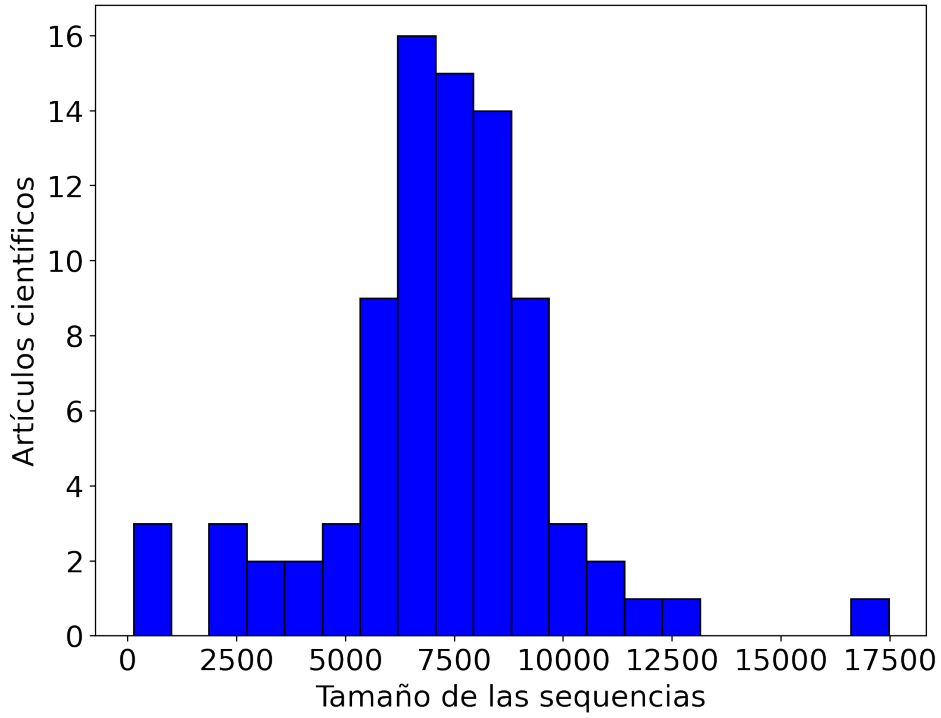


Figura 3.3: Histograma del tamaño de las secuencias de corpus de los artículos científicos [Elaboración propia]

En la Figura 3.3, al igual que en la distribución de la métrica *Ad hoc*, se puede aproximar a una distribución *Normal*, pero en este caso, se puede inferir un límite superior en los tamaños de las secuencias de 13000, con el que limita y por consecuente se estandariza el tamaño de todas las secuencias de *tokens*. Puesto que, existe en valor de 17000 que fue considerado como un dato aberrante, el cual no aportaría a nuestro modelo de distribución, bien sea por *padding* a las que no llegan aquel tamaño o por la reducción de secuencias atípicamente largas hasta esa cota. Esta estandarización hizo que se optimizara el procesamiento de las secuencias de *tokens* hasta su transformación a tensores de dimensiones: 84 como número de documentos y 13000 que corresponde

al tamaño convenido mediante el análisis de su distribución.

3.1.4 Clasificación del texto

Los resultados del modelo de clasificación, al igual que los módulos anteriores, tuvo una evolución descrita en la Tabla 3.2; donde se muestra una mejora sustancial en la métrica de evaluación escogida de *F1 score*, a medida que, la cantidad de documentos etiquetados iba aumentando. Demostrando la existencia de una relación entre estas variables discretas que se podrían aproximar a una función lineal, como se denota en la Figura 3.4.

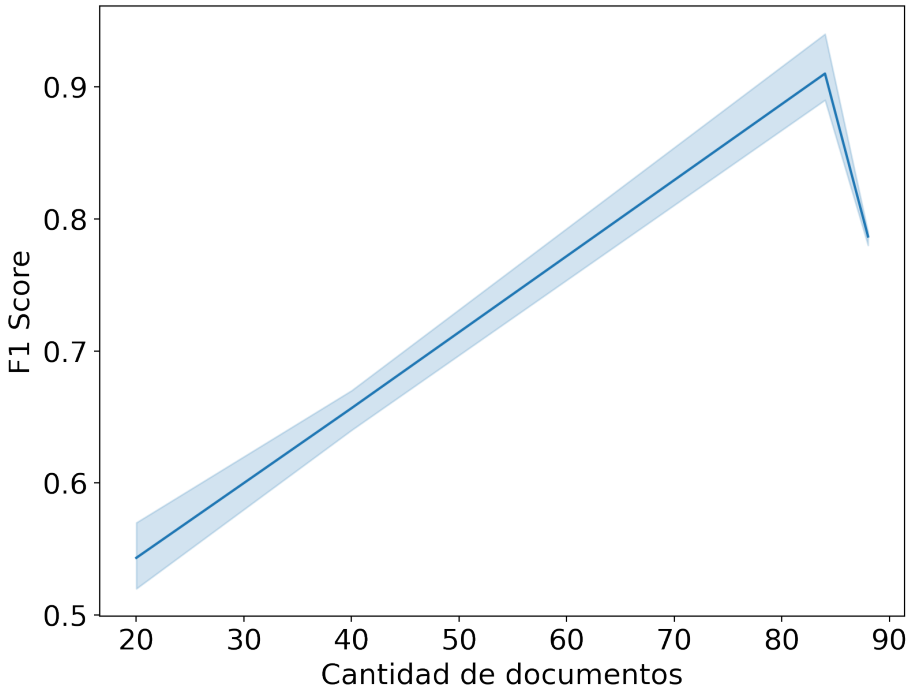


Figura 3.4: Curva de evolución de la métrica *F1 score* en función del tamaño del conjunto de datos [Elaboración propia]

En la Figura 3.4 se puede hacer una observación particular, es que hay un punto máximo donde en el eje de las ordenadas (métrica *F1 score*) decae, a pesar de seguir creciendo en las abscisas (tamaño del conjunto de datos). Esto se produce, debido a que en la versión final del conjunto de datos se redujo la cantidad de piezas de información, gracias al re-etiquetado de las clases, que pasaron de 4 a 3 como se detalló en la Tabla 3.1; esto como consecuencia al gran traslape en la validación del modelo que suponía

las clases de media y baja relevancia.

Esta superposición entre estas dos categorías se muestra contundentemente, a través de la matriz de confusión en la Tabla 3.3, efectuada para los datos inéditos probados que se enmarcan en los 4 niveles de relevancia correspondientes al etiquetado en la iteración 3.0 del conjunto de datos.

Tabla 3.3: Matriz de confusión para el entrenamiento 3.a de la tabla 3.2 [Elaboración propia]

		Predicción			
		Alta	Media	Baja	Nula
Real	Alta	**14,77 %**	3,41 %	0,00 %	1,14 %
	Media	3,41 %	**18,18 %**	5,68 %	0,00 %
	Baja	4,55 %	15,91 %	**4,55 %**	4,55 %
	Nula	1,14 %	5,68 %	2,27 %	**14,77 %**

La matriz de confusión que se corresponde con la Tabla 3.3, se aprecia que la diagonal presenta valores desde intermedios a bajos, considerando que en el caso ideal esta matriz debería ser una de tipo diagonal con el 25 % como valor en sus celdas. Por otro lado, se muestra la confusión del mejor modelo entrenado a partir de la iteración 3.0 del conjunto de datos, para con las clases de media y baja de relevancia, lo que desencadenó en un re-etiquetado del conjunto de datos detallada anteriormente.

La conformación de un nuevo conjunto de datos con solo 3 etiquetas, hizo que el mejor modelo (4.a) elevara sustancialmente la métrica de *F1 score* para el conjunto de datos de pruebas, en el que se logró un valor de 0,87 (87 %) que cumple con el requerimiento no funcional del 85 % establecido en la metodología.

La matriz de confusión representada por la Tabla 3.4, se tiene que en la diagonal posee valores altos, pues rondan al valor ideal de 33 %, si esta fuera una matriz diagonal. Por consecuente, el traslape entre clases es mínimo para el modelo 4.a que fue entrenado a partir de la iteración 4.0 del conjunto de datos de 3 etiquetas.

Tabla 3.4: Matriz de confusión para el entrenamiento 4.a de la tabla 3.2 [Elaboración propia]

		Predicción		
		Alta	Media	Nula
Real	Alta	**27,38 %**	5,95 %	2,38 %
	Media	2,38 %	**27,39 %**	2,38 %
	Nula	1,19 %	2,38 %	**28,57 %**

Tabla 3.5: Métricas para el modelo proveniente del entrenamiento 4.a de la tabla 3.2 [Elaboración propia]

	Precisión	Exhaustividad	*F1 score*	
Alta	0,88	0,86	0,87	
Media	0,80	0,89	0,84	
Nula	0,86	0,94	0,90	
	0,84	**0,89**	**0,87**	**Media**

Con respecto a la Tabla 3.5, se muestran las métricas del mejor modelo entrenado a partir del conjunto de datos de pruebas, donde se evidencia que la mayor precisión del modelo se da para la clase alta y tanto la exhaustividad como el *F1 score* son máximos para la etiqueta de nula relevancia.

3.2 Prototipo

Para tener una mejor perspectiva de cómo está implementado el demo de este proyecto, se desarrolló un prototipo mediante el uso de la arquitectura cliente-servidor, donde el cliente son los investigadores/estudiantes que pueden acceder directamente desde su navegador.

El servidor consta a su vez de tres partes, los cuales son descritos a grandes rasgos a continuación:

- El **modelo de clasificación de aprendizaje profundo**, el cual ha sido descrito en la metodología de este documento.
- El **proveedor de servicios** , que es el encargado de responder a las peticiones de los clientes, orquestando la ejecución los módulos y comunicación entre ellos.

❒ El ***Spawn de Python***, que es el encargado de hacer el llamamiento tanto del modelo, como al hacer uso de la API *Crossref*, encargado de la minería de datos.

El servidor está construido de forma modular, de esta forma se optimiza el tiempo, en el caso de que se deba realizar algún cambio, ya sea en el diseño del prototipo, como en el modelo o el *script* encargado de ejecutar la minería de datos.

3.2.1 Implementación

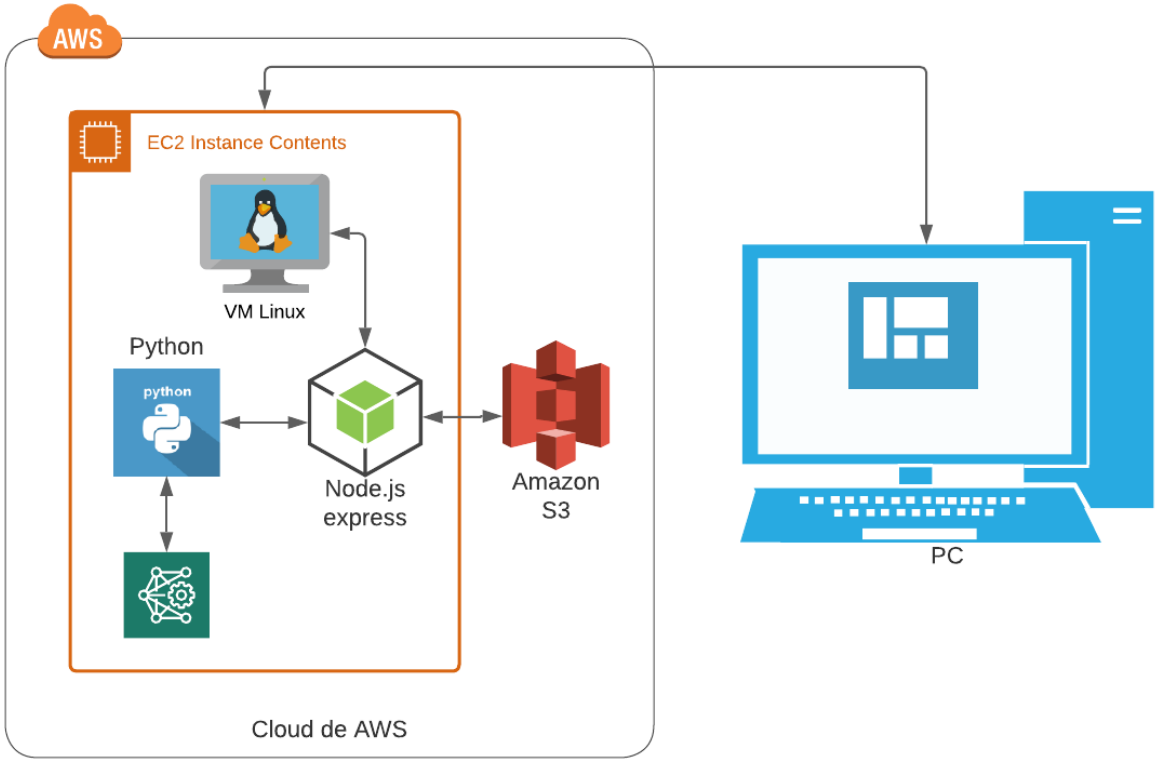


Figura 3.5: Diagrama del prototipo ejecutado en producción [Elaboración propia]

Como se puede apreciar en la Figura 3.5, el servidor está alojado en un contenedor de *Amazon EC2*, el cual provee de una instancia o máquina virtual con una imagen del sistema operativo de Linux con características esenciales para hacer de servidor de cuasi-producción.

El servidor fue creado enteramente con *NodeJS*, utilizando *express* como *framework*

de desarrollo para el levantamiento de funciones web, a su vez, *NodeJS* provee de funciones para ejecutar instrucciones de *Python* mediante un método llamado *Spawn*.

Es requerido un *bucket* de *Amazon S3*, para alojar todos los archivos que el investigador desea subir para su análisis. Así mismo, los resultados son alojados dentro del mismo *bucket* en directorios separados de acceso público, con la finalidad de poder acceder a ellos desde el lado del cliente.

Una vez que es requerido el funcionamiento del *script de Python*, se procede a ejecutar en forma secuencial la lectura de cada uno de los documentos alojados en el directorio de archivos del *bucket*, para su posterior procesamiento mediante la ejecución del *script*.

Una vez el texto ha sido tokenizado, se extrae el identificador (DOI) para ser enviado como parámetro a la API de *Crossref*, y lograr extraer metadata. En el caso de que esta extracción falle los resultados de tipo texto se les asignó el valor 'API FAILED' y los resultados numéricos el valor '0'.

Los documentos pasan a ser analizados por el modelo clasificador de aprendizaje profundo de manera secuencial y finalmente se establece a qué niveles de relevancia pertenecen.

Toda esta información es recopilada y alojada nuevamente en el *bucket* de *Amazon S3*, en un directorio definido para ser leída por los clientes al momento de recibir el acuse de respuesta por parte del servidor.

3.2.2 Funcionamiento

A continuación, se revisa el correcto flujo de trabajo para la utilización del prototipo realizado en la elaboración de este proyecto.

Como el prototipo se encuentra levantado en un servidor en la nube, se puede acceder fácilmente desde cualquier navegador por medio de la siguiente ruta:

$$URL = http://18.212.231.17/ \tag{3.1}$$

Mediante esta ruta se puede acceder a la pantalla principal del prototipo, la cual está en un estado 'pasivo', ya que no cuenta con ningún elemento seleccionado, tal como se aprecia en la Figura 3.6.

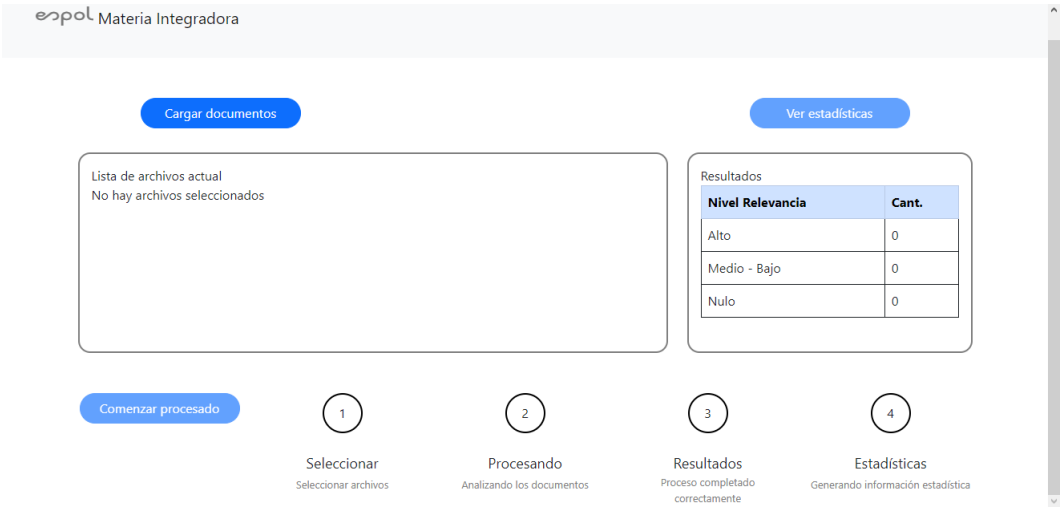


Figura 3.6: Pantalla principal del prototipo [Elaboración propia]

Una vez cargada la página principal, se pueden cargar los archivos a analizar, con un *click* sobre el botón de **'Cargar documentos'**, tras la cual aparecerá una pantalla de selección de archivos tal como se aprecia en la Figura 3.7.

Posterior a la selección de los archivos, se regresa a la pantalla principal nuevamente, pero esta vez, con un estado 'activo', ya que cuenta con la información básica de los archivos seleccionados, como lo es el nombre del archivo y el tamaño de este en MegaBytes (MB), tal como se muestra en la Figura 3.8

Luego, es posible comenzar con el procesado de los documentos seleccionados, el cual se lleva a cabo luego de borrar los artículos subidos anteriormente y su posterior subida al servidor. Una vez que este procesamiento haya terminado, el servidor le envía

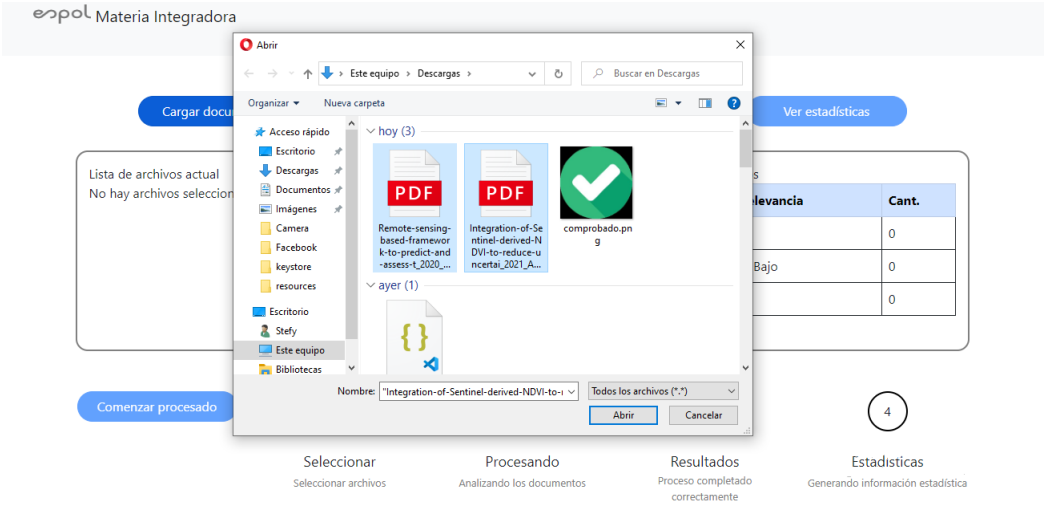


Figura 3.7: Pantalla de selección de archivos [Elaboración propia]

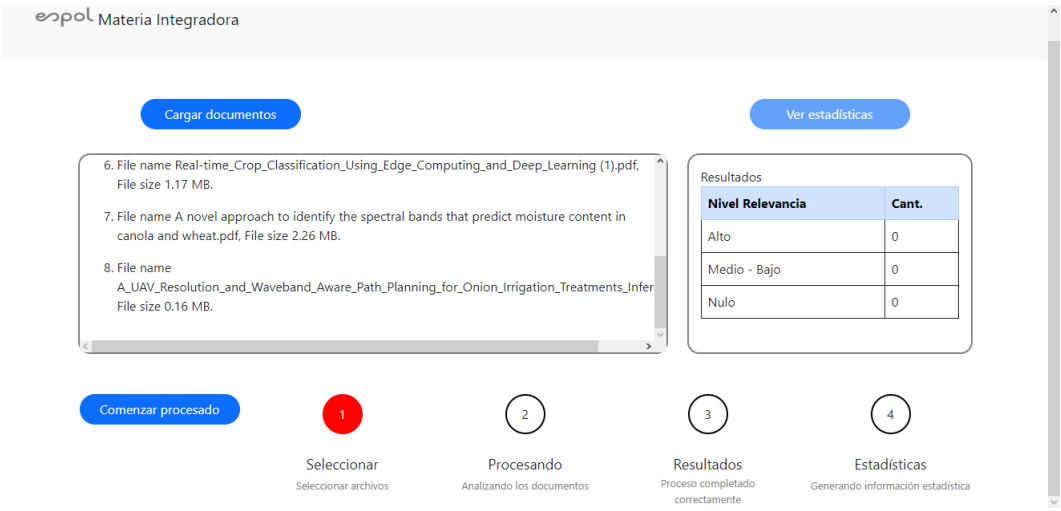


Figura 3.8: Pantalla principal del prototipo con datos [Elaboración propia]

un *response* indicando que el archivo de resultados se ha generado correctamente tal como se indica en la Figura 3.9.

Figura 3.9: Modal del mensaje de estado del proceso [Elaboración propia]

Si el proceso se ha completado con total normalidad, se puede revisar la información generada mediante un *dashboard* en el cual se presenta los artículos seleccionados con la respectiva clasificación asignada por el modelo creado para este proyecto como se muestra en la Figura 3.10, e información adicional para que el investigador puede utilizar según sea el caso tal como se muestra en la Figura 3.11.

3.2.3 Limitaciones

Con la finalidad de realizar pruebas de funcionamiento y de rendimiento, se tomó la decisión de realizar el levantamiento del servidor de pruebas en una instancia de *Amazon EC2*, como se puede ver en la Figura 3.5. Sin embargo, dicho servidor cuenta con las siguientes limitaciones:

3.2.3.1 Limitaciones de hardware

Al ser una cuenta básica de *Amazon Web Services* (AWS), sólo se puede acceder a una instancia mínima de Linux con 1 solo núcleo, 2 GB de RAM y 8 GB de almacena-

Regresar

#	Titulo	Año	Referencias	F. Impacto	Métrica	Clase
1	Assessment of maize yield and phenology by drone-mounted superspectral camera	2019	48	5.767	69	C
2	Precision land leveling for sustainable rice production: case studies in Cambodia, Thailand, Philippines, Vietnam, and India	2022	0	5.767	0	C
3	Intelligent robots for fruit harvesting: recent developments and future challenges	2022	1	5.767	5	C
4	Coupling ground-level panoramas and aerial imagery for change detection	2016	18	4.278	11	C
5	Post flash flood survey: the 14th and 15th October 2015 event in the Paupisi-Solopaca area (Southern Italy)	2016	30	2.657	11	C
6	Assessing expected utility and profitability to support decision-making for disease control strategies in ornamental heather production	2022	0	5.767	0	C
7	Detection of Colchicum autumnale in drone images, using a machine-learning approach	2020	6	5.767	11	C
8	Features and applications of a field imaging chlorophyll fluorometer to measure stress in agricultural plants	2020	3	5.767	5	C
9	Lindblom2017 Article PromotingSustainableIntensific	0	0	0	0	C

Figura 3.10: Pantalla del *dashboard* de la clasificación otorgada. [Elaboración propia]

miento.

Por otra parte tenemos que la librería de *TensorFlow* de *Keras* utiliza un optimizador de rendimiento mediante aceleración gráfica, lo cual tampoco consta en esta instancia de Linux.

3.2.3.2 Limitaciones de software

Una limitante en cuanto a rendimiento es la cantidad de archivos que pueden ser subidos al *bucket* de *Amazon S3*, el cual tiene una limitante de máximo 20 archivos con un peso máximo de 100 MB en total.

Otra limitante que influye en el rendimiento es el contar con 1 solo núcleo lógico, el cual, al momento de realizar la clasificación de varios documentos mediante el modelo de aprendizaje profundo, puede tomar un tiempo considerable.

Además, es preciso recalcar que una limitante importante, en cuanto a rendimiento, es la calidad del internet que pueda contar el dispositivo que acceda a la aplicación web.

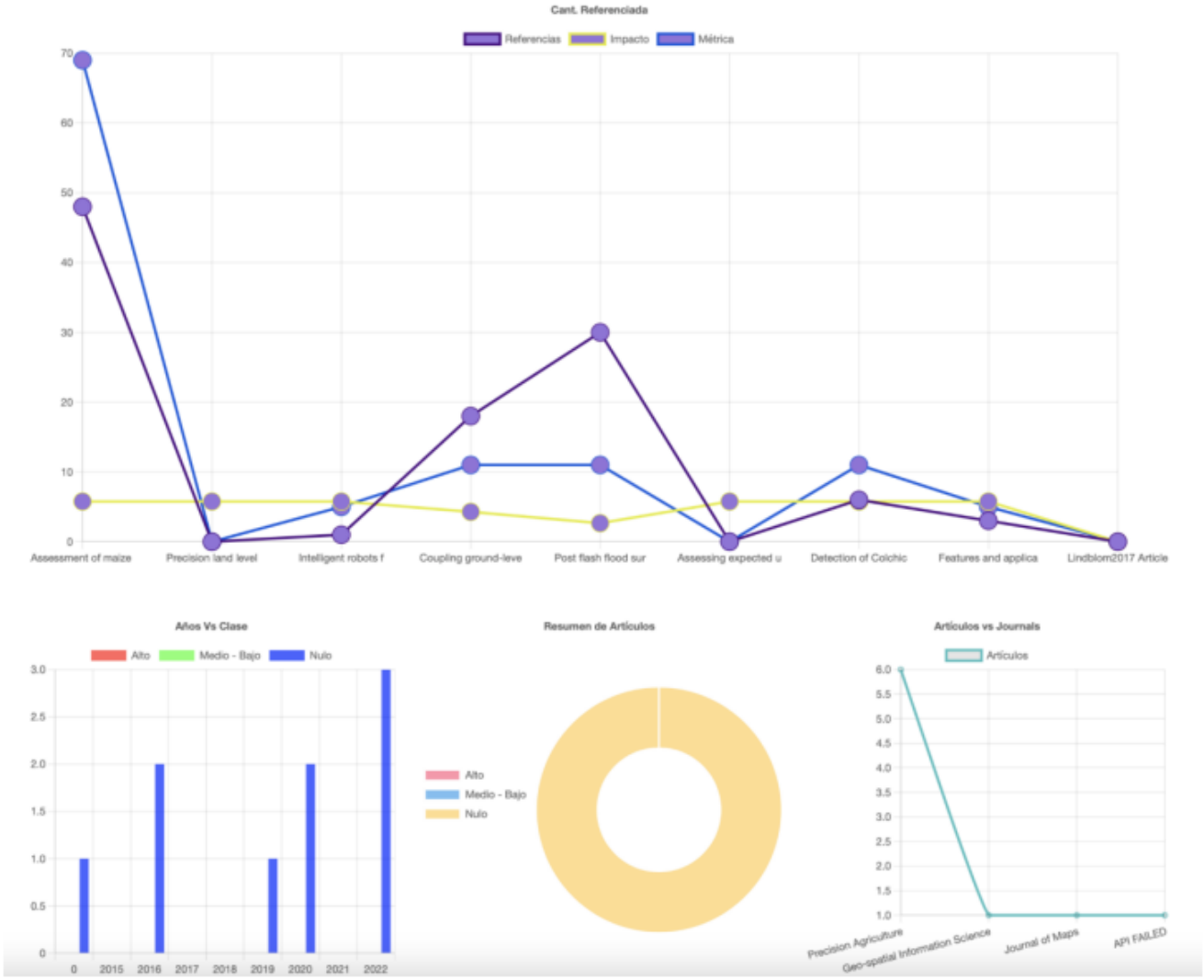


Figura 3.11: Pantalla del *dashboard* de las métricas obtenidas de la minería de datos. [Elaboración propia]

Esto es considerable, porque dependiendo de la velocidad de subida, la carga de los archivos al *bucket* de *Amazon S3* puede ser rápida o lenta.

En el siguiente capítulo, se describen las conclusiones, recomendaciones y trabajos futuros a partir de los resultados presentados en este capítulo.

CAPÍTULO 4

4 CONCLUSIONES Y RECOMENDACIONES

En este último capítulo, se describen las conclusiones, recomendaciones y trabajos futuros a partir de los resultados presentados en el capítulo anterior, por medio de la finalización de los módulos del proyecto, los cuales permitieron alcanzar los objetivos descritos en el primer capítulo.

4.1 Conclusiones

A continuación, se describen las conclusiones que se pudieron identificar a partir del desarrollo de la propuesta de solución planteada a lo largo del proyecto:

- Con el objetivo de crear un modelo de aprendizaje profundo que clasifique artículos científicos en inglés relacionados a la agricultura de precisión con imágenes UAV, se logró entrenar un modelo que los discriminará en 3 niveles acorde a su relevancia, que son alto, medio y nulo. Este modelo que es el núcleo de la propuesta de solución, obtuvo métricas de 94 % en entrenamiento para *F1 score* como promedio para todas sus categorías; y para prueba obtuvo un 87 % en la métrica indicada.

- A partir del conjunto de datos etiquetados, se diseñó un flujo de procesos que permitan la transformación de los corpus de los documentos, a fin de que sean estandarizados, en orden para su óptima vectorización y posterior clasificación.

- Se creó una métrica *Ad hoc* para completar el etiquetado manual, de forma que sintéticamente se aumentó la cantidad de artículos científicos etiquetados; mejorando la calidad en los entrenamientos del modelo.

- Se implementó un prototipo el cual puede ser ejecutado desde cualquier parte y corroborar los resultados obtenidos por cuenta propia.

4.2 Recomendaciones

Por consiguiente, se presentan las recomendaciones que se dan al lector para implementar o tomar como trabajo previo, en el diseño de una propuesta de solución similar a la descrita en este proyecto:

- ❒ Para mejorar el modelo que clasifique artículos científicos en inglés relacionados a la agricultura de precisión con imágenes UAV, deberá incrementar las muestras en los 3 niveles de relevancia, que son alto, medio y nulo. Aunque llegamos a una métrica de 87 % en pruebas, el modelo demostró no ser infalible cuando se realiza pruebas con documentos inéditos.

- ❒ Una recomendación para mejorar el flujo de procesos estandarizado para su vectorización y clasificación del texto, estaría en realizar un análisis solo de secciones concretas del documento, como lo puede ser, solo el '*abstract*' o las 'conclusiones' del documento, para reducir posible ruido.

- ❒ Se puede modificar la métrica *Ad hoc* para aumentar o disminuir el factor de proporcionalidad según sea necesario para ajustar los niveles de relevancia del documento.

- ❒ En cuento al prototipo se deberá mejorar el hardware, debido a las limitaciones presentadas por el servicio que ofrece alojar un servidor en la nube.

- ❒ Se deberá mejorar la extracción del identificador de los documentos (DOI), puesto que existen documentos que no lograron ser detectados por el API de *Crossref*, dando como resultado un *dashboard* con un gran número de documentos sin métricas a explorar.

4.3 Trabajos futuros

Para finalizar, se detallan los trabajos a realizar en iteraciones futuras de la herramienta que mejoren y robustezcan la propuesta de solución:

- ❒ Con respecto al conjunto de datos final, este se puede seguir aumentando, bien sea con el etiquetado manual o sintético descrito en la metodología; de modo que este conjunto de datos pueda asegurar ser una muestra representativa del universo de datos en la temática de agricultura de precisión con imágenes UAV.

- ❒ De la mano con el punto anterior, se puede seguir haciendo entrenamientos según como el conjunto de datos vaya mejorando; en pro de seguir retroalimentando el modelo final.

- ❒ Del lado del prototipo para una puesta a producción, podría ser instanciado en una infraestructura con más recursos. Además de hacer el trabajo de *networking* necesario para que se tenga la comunicación con el cliente o navegador, por medio del puerto seguro 443; a fin de que al adquirir el dominio se pueda adjuntar el certificado SSL, y quede desplegada la herramienta para un uso constante y seguro.

BIBLIOGRAFÍA

[1] C. Perez y D. Santin, *Minería de Datos, técnicas y herramientas*, 1.ª ed. Madrid, España: Innternational Thomson Paranninfo S.A, 2007.

[2] P. Verhaar y P. Verhaar, *Text and Data Mining: The Theory and Practice of Using TDM for Scholarship in the Humanities*. Facet Publishing, 2025, ISBN: 9781783304196. dirección: https://books.google.com.ec/books?id=kjY2uAEACAAJ.

[3] C. Wang, P. Nulty y D. Lillis, «A Comparative Study on Word Embeddings in Deep Learning for Text Classification,» dic. de 2020, págs. 37-46. DOI: 10.1145/3443279.3443304.

[4] U. Naseem, I. Razzak, S. K. Khan y M. Prasad, «A Comprehensive Survey on Word Representation Models: From Classical to State-Of-The-Art Word Representation Language Models,» *CoRR*, vol. abs/2010.15036, 2020. arXiv: 2010.15036. dirección: https://arxiv.org/abs/2010.15036.

[5] L. Zhang, S. Wang y B. Liu, «Deep Learning for Sentiment Analysis : A Survey,» *CoRR*, vol. abs/1801.07883, 2018. arXiv: 1801.07883. dirección: http://arxiv.org/abs/1801.07883.

[6] P. Beckmann, M. Kegler y M. Cernak, «Word-Level Embeddings for Cross-Task Transfer Learning in Speech Processing,» en *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, págs. 446-450. DOI: 10.23919/EUSIPCO54536.2021.9616254.

[7] Ö. Köksal y Ö. Akgül, «A Comparative Text Classification Study with Deep Learning-Based Algorithms,» en *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE)*, 2022, págs. 387-391. DOI: 10.1109/ICEEE55327.2022.9772587.

[8] P. Bojanowski, E. Grave, A. Joulin y T. Mikolov, «Enriching Word Vectors with Subword Information,» *CoRR*, vol. abs/1607.04606, 2016. arXiv: 1607.04606. dirección: http://arxiv.org/abs/1607.04606.

[9] P. Deemer y G. Benefield, «The Scrum Primer. An Introduction to Agile Project Management with Scrum,» 2007. dirección: http://www.rallydev.com/documents/scrumprimer.pdf.

[10] M. Honnibal e I. Montani, «spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing,» To appear, 2017.