UNIVERSIDAD NACIONAL DEL ALTIPLANO

ESCUELA DE POSGRADO

MAESTRÍA EN INGENIERÍA DE SISTEMAS

TESIS

ALGORITMOS DE APRENDIZAJE AUTOMÁTICO NO SUPERVISADO PARA LA EXTRACCIÓN DE PALABRAS CLAVE EN TRABAJOS DE INVESTIGACIÓN DE PREGRADO

PRESENTADA POR:

FRED TORRES CRUZ

PARA OPTAR EL GRADO ACADÉMICO DE:

MAGÍSTER SCIENTIAE EN INGENIERÍA DE SISTEMAS

PUNO, PERÚ

2022

DEDICATORIA

A mis padres por su abnegada labor en mi formación profesional.

A mi hermano Edward por su apoyo incondicional para la obtención de este grado.

AGRADECIMIENTOS

A los docentes de la Maestría en Ingeniería de Sistemas quienes impartieron sus conocimientos y experiencia.

A los miembros del jurado Dr. Edelfré Flores, M.Sc. Wiliam Arcaya, M.Sc. Irenio Chagua, y en particular a la M.Sc. Marga Ingaluque, quienes han contribuido con la culminación de este estudio de manera satisfactoria.

Al Vicerrectorado de Investigación por haberme facilitado la información necesaria para la ejecución del estudio presentado.

ÍNDICE GENERAL

CAPÍTULO III

MATERIALES Y MÉTODOS

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

Puno, 4 de marzo de 2022

ÁREA: Ciencias de la Ingeniería.
TEMA: Recuperación de Información.
LÍNEA: Sistemas, Computación e Informática.

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

La información que administra la Universidad Nacional del Altiplano de Puno, en los últimos años se ha visto incrementada sobre todo trabajos de investigación realizados por estudiantes y egresados de pregrado, para los que se usan técnicas empíricas para la selección de palabras clave, existiendo a la fecha métodos técnicos que ayuden en este proceso, en tanto el uso de tecnologías de información y comunicación han tomado relevancia e importancia en la administración y seguimiento de trabajos de investigación como la Plataforma de Investigación Integrada a la Labor Académica con Responsabilidad (PILAR), donde registra información de los proyectos de investigación como (Título, Resumen, Palabras Clave), en sus diferentes modalidades. En el presente trabajo de investigación se ha analizado 7430 registros de proyectos de investigación, a los cuales se realizaron predicciones con cada uno de los 09 modelos de aprendizaje automático no supervisado implementados. Los resultados nos muestran que el modelo TF-IDF, es el más eficiente en tiempo y en precisión de extracción de palabras clave, obteniendo un 72 % de precisión y en un tiempo de extracción entre [0.4786 ,SD 0.0501], por cada documento procesado por este modelo.

Palabras clave: Aprendizaje automático, investigación, modelos, palabras clave, precisión.[1]

[1] Palabras Clave generadas con el modelo TF-IDF, implementado en el presente trabajo de Investigación

ABSTRACT

The management of information at *Universidad Nacional del Altiplano de Puno*, in recent years has been increased, especially research work made by undergraduate students and graduates, so these existing empirical techniques are used for the selection of keywords over time. Technical methods that help in this process, as the use of information and communication technologies have become relevant and important in the management and monitoring of research works such as the integrated research platform for academic work with responsibility (PILAR), where registers information on research projects as (title, abstract, keywords), in its different ways. in the present research work, 7,430 records of research projects have been analyzed, that predictions were made for each of the nine (09) unsupervised machine learning models implemented. The results show us that the TF-IDF model is the most efficient from this group in time and precision in keyword extraction, obtaining 72% precision and an extraction time between [0.4786, sd, 0.0501], for each document processed by this model.

Keywords: Accuracy, keywords, machine learning, models, research.

INTRODUCCIÓN

Es innegable el incremento de los volúmenes de información que se vienen generando a través de la implementación sistemas de información del conocimiento en las organizaciones, con el objetivo de lograr una eficiente gestión y servicio, por otro lado, las universidades vienen implementando servicios con ayuda de la de tecnología de diferentes tipos que permiten realizar transacciones a distintos niveles de gestión institucional almacenando información en todos sus procesos(Martín-Mora *et al.*, 2020; Ranguelov, 2012). Todo el volumen de información existente en internet viene creciendo permanentemente y adquiere diferentes formas de representación, desde simples archivos de texto en una computadora personal o un periódico electrónico hasta librerías digitales y espacios mucho más grandes y complejos como la web, esta información en hordas se acrecentó aún más con el uso de medios digitales (Jayawardene *et al.*, 2021; Tolosa & Bordignon, 2008), la Universidad Nacional del Altiplano de Puno no se encuentra ajena a este crecimiento acaudalado, por lo que a partir de la implementación de sistemas de información como la Plataforma de Investigación Integrada a la Labor Académica con Responsabilidad (PILAR), así como la Plataforma de Investigación para el Fondo Especial para Docentes Universitarios (FEDU), mediante estos aplicativos se viene recabando información así como se realiza la gestión del conocimiento, generado por estudiantes, egresados, pero notándose en sus procedimientos ningún tratamiento al momento de elegir las palabras clave.

Los enfoques de selección automática de palabras clave son cada vez más importantes para clasificar grandes volúmenes de documentos, este proceso se ha convertido esencial para hacer estos documentos más manejables y obtener información valiosa (Ahadh *et al.*, 2021). La extracción automática de palabras clave ayudan a filtrar, encontrar, mejorar la recomendación y la recuperación de información basada en el contenido del mismo texto, por ende tiene una representación basada en el contenido que se está evaluando (Ahadh *et al.*, 2021). El objetivo de la extracción automática de palabras clave es la aplicación del poder y la velocidad de las capacidades computacionales e informáticas actuales para el problema del acceso y la recuperación, haciendo hincapié en la organización de la información (Santosh *et al.*, 2017). Así mismo para la extracción de información y conocimiento es objeto de un considerable interés de investigación en los campos de aprendizaje automático y minería de datos. la minería de datos de texto y en particular la minería de texto se ha convertido en uno de los sub campos de investigación más activos en la minería de datos (Xuezhong *et al.*, 2010).

Por lo expuesto en el presente trabajo se aborda el uso de las técnicas más representativas de la extracción automática de palabras clave, mediante los modelos de aprendizaje automático no supervisado, siendo estas técnicas unas de las que son de rápida implementación , así mismo sentar las bases para otros estudios similares y que esta área del conocimiento puede seguir siendo estudiada, en se sentido este trabajo de investigación se estructura en cuatro capítulos con el siguiente detalle en el Primer Capítulo la revisión de conceptos, bases teóricas y antecedentes. En el Segundo Capítulo una recapitulación del planteamiento problema planteado en el proyecto de tesis. En el Tercer Capítulo los métodos aplicados. En el cuarto Capítulo los resultados obtenidos y discusión con otras fuentes de trabajos relacionados el tema objetivo. Finalmente presentamos las conclusiones del trabajo y recomendaciones de la presente tesis de grado.

CAPÍTULO I

REVISIÓN DE LITERATURA

1.1. Marco teórico

1.1.1. Descubrimiento del conocimiento

El descubrimiento de conocimiento en base de datos por sus siglas en inglés *Knowledge Discovery in Databases (KDD)*, es un campo de investigación que se ocupa de obtener conocimientos de alto nivel a partir de los datos. Las tareas que se llevan a cabo en este campo requieren muchos conocimientos y, a menudo, pueden beneficiarse de la utilización de conocimientos adicionales procedentes de diversas son comúnmente utilizados para detectar patrones que nos permiten explicar la naturaleza de los datos, para así contar con una interpretación o predicción de eventos futuros (Ristoski & Paulheim, 2016).

1.1.2. Minería de datos

Como parte del descubrimiento de conocimiento, la minería de datos se define como el proceso de descubrir patrones, correlaciones y anomalías en el caso de ser necesario a partir de grandes cantidades de datos (Campos *et al.*, 2020). Las fuentes de datos pueden incluir bases de datos, depósitos de datos, la web, otros repositorios de información o datos heterogéneos que son administrados y transmitidos dinámicamente por un sistema o recolectados desde algún tipo de fuente (Sourav *et al.*, 2022).

1.1.3. Minería de texto

La minería de textos, es también conocida como análisis de textos, es una técnica de inteligencia artificial que convierte los datos no estructurados en datos estructurados mediante algoritmos de aprendizaje automático. La minería de textos es un conjunto de técnicas muy conocidas entre las ciencias de la computación, las ciencias de la información, las matemáticas y los campos de gestión para extraer inteligencia de los grandes datos (Kumar *et al.*, 2021)

La minería de texto contiene una gran cantidad de enfoques, métodos y técnicas, que tienen como fuente primaria de información texto. Esto permite dar múltiples definiciones, que van desde una extensión de la minería de datos clásica hasta textos y formulaciones más sofisticadas como el uso de grandes colecciones de texto en la red para descubrir nuevos hitos y predicciones sobre lo mismo (Kumar *et al.*, 2021).

De manera análoga a la minería de datos, la minería de texto busca extraer información útil de las fuentes de datos a través de la identificación y exploración de patrones interesantes. La fuente de análisis de información está formada por colecciones de documentos, y no se encuentran patrones interesantes entre los registros de bases de datos formalizados sino en los datos textuales no estructurados en los documentos de estas colecciones. Mostrando así similitudes arquitectónicas de alto nivel como algoritmos de descubrimiento de patrones y elementos de redacción y/o presentación, herramientas de visualización y exploración de respuestas (Feldman & Sanger, 2007).

En resumen, la minería de texto es un área de investigación aún emergente y que contribuye a la ciencia de la computación al tratar de resolver la crisis de sobrecarga de información combinando técnicas y algoritmos en este proceso (Mahata *et al.*, 2018).

1.1.4. Extracción de información

El objetivo de los métodos de extracción de información es la extracción de información específica de los documentos de texto, esta información se almacena en patrones tipo base de datos y queda disponible para su uso posterior, en el análisis o predicción (Hotho *et al.*, 1978; Tintinago *et al.*, 2018).

1.1.5. Pre procesamiento de texto

En el trabajo de investigación de Korde & Mahender (2012) explican que como primer paso se debe de realizar el pre procesamiento para presentar el texto en un formato clara. Los documentos preparados para el siguiente paso en la clasificación de texto están representados por una gran cantidad de características por lo general acompañado de expresiones regulares. Para a continuación poder realizar la tokenización al documento al cual se procesa como una cadena, finalmente una técnica que se deberá evaluar es la eliminación de palabras de detención o stopwords tales como "el", "la", "y", "los" etc., que están ocurriendo con frecuencia, Este conjunto de técnicas es denominado pre procesamiento de datos (Duari y Bhatnagar, 2020).

1.1.5.1. Expresiones regulares (ER)

Las expresiones regulares son utilizadas para realizar búsquedas contextuales y modificaciones sobre textos, sin embargo, no existe un lenguaje estándar de expresiones regulares, al contrario, podemos decir que existen diferentes dialectos que según la configuración del texto en análisis nos ayudará a encontrar preprocesar el contenido del texto (Viloria, 2015).

1.1.5.2. Normalización de texto

La normalización de texto, concierne a la evaluación del contenido registrado con la finalidad de mantener la misma estructura y lograr un análisis homogéneo, sin embargo, es importante no perder de vista el contenido original del texto (Škrlj *et al.*, 2019).

Tokenización

“Un documento se trata como una cadena, y luego se divide en una lista de tokens. Eliminación de las palabras de parada: Las palabras de parada como "el", "a", "y", etc. aparecen con frecuencia, por lo que es necesario eliminar las palabras insignificantes” (Korde & Mahender, 2012).

Reducción de palabras o Stemming

"Aplicar el algoritmo de desplazamiento de palabras que convierte las diferentes formas de las palabras en formas canónicas similares. Este paso es el proceso de confluir los tokens a su forma raíz, por ejemplo, conexión a conectar." (Korde & Mahender, 2012)

1.1.6. Aprendizaje automático

El aprendizaje automático es el área de investigación que estudia como las computadoras pueden aprender tomando como base patrones de datos (Agarwal, 2014, p. 24), los cuales se toman como referencia al momento de analizarlos. El aprendizaje automático está muy de la mano de la minería de datos y en sus formas de aplicación podemos clasificarlos en tipos:

1.1.6.1. Aprendizaje supervisado

Kretschmann *et al.* (2020) El aprendizaje supervisado en palabras más simples se puede definir como un sinónimo de clasificación de información, y la supervisión corresponde al proceso de contraste de la información con valores previamente calculados. Análogamente en el proceso de extracción de palabras clave de manera supervisada supone un corpus de entrenamiento para convertir la extracción de palabras clave en un problema de clasificación (Guan *et al.*, 2019). Dado esta característica los ámbitos de aplicación son limitados debido a la limitación de los corpus, sin embargo, este enfoque no es adecuado para textos sin una distribución temática evidente (Xu & Zhang, 2021).

1.1.6.2. Aprendizaje no supervisado

El aprendizaje no supervisado, es el equivalente a la agrupación y este proceso no se encuentra sujeto a un análisis de contraste, por lo que esta agrupación no se encuentra previamente identificada, si vamos a un apartado específico en la extracción de palabras clave con el enfoque no supervisado no es necesario necesita un corpus de etiquetado manual o automático (Aquino & Lanzarini, 2015). Para realizar estas tareas existen diferentes técnicas que en las que se analizan las características del texto para obtener un mejor efecto de extracción (Xu & Zhang, 2021).

1.1.7. Palabras clave

Las palabras clave son términos o frases cortas (lexemas) que permiten clasificar y etiquetar las entradas en diferentes sistemas de indexación y de recuperación de la información en las bases de datos de un manuscrito o área temática en particular. Las palabras clave se convierten entonces en una herramienta esencial de doble vía, es decir, de quienes escriben y de quienes buscan la información de manuscritos o áreas temáticas relacionadas. En consecuencia, no se debe subvalorar o menospreciar su Importancia a la hora de considerarlas, pues se podría dificultar la difusión de un manuscrito o trabajo académico e incluso no detectar la relación del mismo con otros similares, justamente por el uso inadecuado de las palabras clave (Gonzáles & Mattar, 2012).

Extrayendo las palabras clave, podemos entender los acontecimientos noticiosos de forma clara y concisa no solamente es útil para la búsqueda bibliográfica. También podemos comparar las similitudes y diferencias de diferentes eventos, o mediante el análisis de la relación entre la importancia de las palabras clave a lo largo del tiempo, podemos seguir la tendencia de los cambios en el mismo evento y estudiar los temas de mayor relevancia en un documento o de la opinión pública, estas herramienta podría ayudar eficazmente a las personas a adaptarse activamente a las características de la difusión de información en este nuevo contexto de los medios de comunicación digitales, entre otros como ayudar a las instituciones pertinentes a establecer mecanismos de emergencia de la opinión pública, científicos y eficientes (Ding *et al.*, 2022; Xu & Zhang, 2021).

1.1.8. Selección de palabras clave

Para la selección de palabras clave para trabajos de investigación, artículos y documentos académicos el autor debe de elegir entre 3 a 10 palabras (Gonzáles & Mattar, 2012), las mismas que deberán representar la idea que en su mayoría de veces se presentan en el resumen y título del trabajo reiteradamente, para una fácil ubicación y trazabilidad del mismo (Mack, 2012).

1.1.9. ISO 25964

El estándar internacional de los tesauros norma el proceso de Información y documentación (International Organization for Standardization, 2013).

Tesauros e interoperabilidad con otros vocabularios en dos partes:

a) Parte1: Tesauro para la recuperación de información (2011).

b) Parte2: Interoperabilidad con otros vocabularios (2013).

1.1.10. Extracción de palabras clave

La extracción automática de palabras clave es el proceso de seleccionar palabras y frases del documento de texto que, en el mejor de los casos, puede proyectar la idea central del documento sin ninguna intervención humana, dependiendo del modelo (Zhang *et al.*, 2008).

1.2. Antecedentes

La extracción de palabras es ampliamente estudiada desde las diferentes vertientes de la ciencia de la computación en particular desde el aprendizaje automático, para lo cual presentamos los siguientes trabajos que fundamentan apoyan el desarrollo del presente trabajo de investigación, además los antecedentes que se presentan son los más representativos aprobados por los miembros de jurado:

Xu & Zhang (2021), en su artículo de investigación proponen un nuevo enfoque de extracción de palabras clave a partir de un texto que combina características como la frecuencia de palabras y la asociación entre estas. Los resultados de los experimentos muestran que la tasa de precisión, la tasa de recuperación y la medida F son mejores que las de TextRank y TF-IDF.

Kretschmann, Fischer & Elser (2020), presentan un sistema automatizado de asignación de palabras clave para resúmenes científicos. Ese sistema se aplica a los resúmenes en papel recopilados en una base de datos de publicaciones locales y se utiliza para impulsar un sistema de recomendación de investigadores. Para la remediación, la capacitación se realiza en un conjunto de datos extendido basado en grandes bases de datos de publicaciones en línea. Además, se observa más de cerca el desequilibrio de etiquetas en el conjunto de datos. Se comparan diez algoritmos de clasificación de etiquetas múltiples para asignar palabras clave de un catálogo dado a un resumen científico. El sobre muestreo aleatorio antes de la fase de entrenamiento aumenta adicionalmente la puntuación F1 en un 5 - 6 % (Kretschmann *et al.*, 2020).

YAKE es un método de extracción automática de palabras clave sin supervisión que se basa en características de texto estadístico extraídas de documentos individuales para seleccionar las palabras clave más relevantes de un texto, se ha comparado este método con diez enfoques no supervisados de última generación y un método supervisado. ¡Los resultados experimentales realizados sobre veinte conjuntos de datos muestran que YAKE! supera significativamente a otros métodos no supervisados en textos de diferentes tamaños, idiomas y dominios (Campos *et al.*, 2020).

Duari y Bhatnagar (2020), presentan un framework supervisado para la extracción automática de palabras clave de un solo documento. En que se modela un texto como una red compleja y se extraen propiedades de nodos seleccionados a partir del mismo texto. El artículo denominado "Complex Network based Supervised Keyword Extractor", evidencia un método mejorado que es estadísticamente significativo para el idioma inglés, pero al mismo tiempo se puede evidenciar el mismo rendimiento en el idioma indí y asamés (Duari & Bhatnagar, 2020).

Semantic connectivity aware keyword extraction method SCAKE, es un método propuesto por Duari y Bhatnagar en el año 2019, quienes lo refieren como una combinación de construcción de gráficos y métodos de puntuación basado en la conectividad semántica de palabras en un documento. Mostrando que el método resultante de su trabajo es una solución competente para extraer palabras clave de documentos de idiomas que carecen de soporte sofisticado de PNL (Duari & Bhatnagar, 2019).

Guan *et al.* (2019), en su artículo "Improved TF-IDF for We Media Article Keywords Extraction", detalla la mejora el algoritmo TF-IDF tradicional, agrega la parte del discurso y el comentario del lector como factor de impacto, y recalcula el peso de TF-IDF, para que se mejore la precisión del algoritmo. Concluye que ha mejorado significativamente en comparación con el TF-IDF tradicional, en términos de precisión, tasa de recuperación (Guan *et al.*, 2019).

RAKUN el acrónimo de "Rank-based Keyword Extraction Via Unsupervised learning and Meta vertex agregación", artículo en el que exploran la centralidad de carga, una medida teórica de gráficos aplicada a gráficos derivados de un texto dado, puede usarse para identificar y clasificar palabras clave de manera eficiente. El método propuesto no está supervisado, es interpretable y también se puede utilizar para la visualización de documentos (Škrlj *et al.*, 2019).

Mahata *et al.* (2018), realizó artículo sobre la extracción de palabras clave con métodos no supervisados aprovechando la formación de incrustaciones de frases de varias palabras que se utilizan para la representación temática de artículos científicos y la clasificación de las frases clave extraídas de ellos usando PageRank ponderado por tema. Las evaluaciones se realizan en conjuntos de datos de referencia que producen resultados de vanguardia (Mahata *et al.,* 2018).

Aquino & Lanzarini (2015), en su artículo presentan un algoritmo para la extracción de palabras clave de documentos escritos en español. Este algoritmo combina los autoencoders; que son adecuados para problemas de clasificación altamente desequilibrados; con el poder discriminativo de los clasificadores binarios convencionales. Para mejorar su rendimienTO en conjuntos de datos más grandes y diversos (Aquino & Lanzarini, 2015).

Chahine *et al.* (2008) planteó un sistema de soporte de indexación que toma como entrada una ontología y un documento de texto sin formato y proporciona como salida palabras clave contextualizadas del documento. Este artículo fue evaluado explotando los enlaces de categoría de Wikipedia como recursos con términos ontológicos (Chahine *et al.,* 2008).

Los repositorios de objetos no son ajenos a la extracción de palabras clave por lo que Coursey *et al.* (2008), publicó un artículo titulado "Automatic Keyword Extraction for Learning Object Repositories" En el que se analiza un repositorio de recursos de aprendizaje de un curso de historio en pregrado, y describe experimentos en la generación de metadatos para el aprendizaje de repositorios de objetos. Específicamente, analiza varios métodos para la extracción automática de palabras clave (Coursey *et al.,* 2009).

Los recursos analizados no solo pueden ser evaluados desde un texto, sino que en el artículo de Van Der Plas *et al.* (2004), hace uso de WordNet y EDR, ambos recursos en la misma tarea para hacer posible una comparación. La tarea estudiada fue la asignación automática de palabras clave a episodios de diálogo de múltiples partes (es decir, tramos temáticamente coherentes de texto hablado). En el que se muestra que el uso de recursos léxicos en una tarea de este tipo da como resultado rendimientos ligeramente más altos que el uso de un método basado puramente en estadísticas (Van Der Plas *et al.*, 2004)

Chahine *et al.* (2008), en su artículo propone un método para un sistema de soporte de indexación. Este sistema toma como entrada el texto y su ontología y tiene como salida palabras clave contextualizadas para el documento. Esta herramienta fue evaluada con los enlaces de categorías de Wikipedia como recurso ontológico (Chahine *et al.,* 2008)

El procesamiento del Lenguaje Natural no está exento a este tipo de análisis para extraer palabras clave de documentos, por ello Vega-Oliveros *et al.* (2019), en su publicación "A multi-centrality index for graph-based keyword extraction", presentan el enfoque del índice de multi-centralidad (MCI), cuyo objetivo es encontrar la combinación óptima de clasificaciones de palabras de acuerdo con la selección de medidas de centralidad. Analizamos nueve medidas de centralidad (intermediación, coeficiente de agrupamiento, cercanía, grado, excentricidad, vector propio, K-Core, PageRank, agujeros estructurales) para identificar palabras clave en la representación de documentos de gráficos de palabras de coincidencia (Vega-Oliveros *et al.,* 2019).

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1. Identificación del problema

Hoy en día, la web y las redes son los medios elegidos para difundir información que luego se utiliza para resolver una amplia gama de problemas. sin embargo, a medida que aumenta la cantidad de datos almacenados, su administración se hace más difícil y los usuarios comienzan a sufrir la llamada sobrecarga de información (Villa, 2019). Esto es considerado como una explosión de información en donde la asimilación y utilización intensiva del conocimiento ha conducido a lo que se denomina sociedad del conocimiento, en la que la gestión de la información, la documentación y el conocimiento se perfila como un componente estratégico de primer orden (Botta-Ferret & Cabrera-Gato, 2007).

La detección de las tendencias en la investigación ayuda a los investigadores y a los responsables de tomar decisiones a identificar y analizar rápidamente los temas de investigación (Lu *et al.*, 2021). sin embargo, debido a la demora en la citación y la publicación, los estudios anteriores sobre el análisis de tendencias son más propensos a identificar las tendencias a posteriori, este apoyo a los investigadores en la búsqueda, filtrado y utilización de la información en un entorno electrónico es un proceso extremadamente complejo (Botta-Ferret & Cabrera-Gato, 2007). Más aún que hoy en día donde más del 80 % de los datos disponibles en el mundo se almacena en formato texto, en especial en los trabajos de investigación en donde los metadatos son recolectados previos a su publicación, el procesamiento automático termina siendo una tarea crucial (Miner *et al.*, 2012).

En este contexto la Universidad Nacional del Altiplano – Puno (UNA Puno) , no se encuentra ajena al procesamiento de información mediante procesos electrónicos en documentos la dependencia encargada de este proceso es el Vicerrectorado de Investigación en donde se registra información correspondiente a los trabajos de Investigación de estudiantes y egresados en sus distintas modalidad en un sistemas web como son : a) Plataforma Integrada a la Labor Académica con Responsabilidad (PILAR) (Torres, 2016) y b) Plataforma del Fondo Especial de Desarrollo Universitario (FEDU), los que almacenan en sus registros de base de datos información en formato de texto sin tratamiento alguno para la optimización de los procesos que realizan cada uno de los sistemas de información.

La extracción de palabras clave no fue aplicada antes del presente trabajo a los documentos de investigación que administra la Universidad Nacional del Altiplano de Puno en sus distintos procedimientos de almacenamiento y tratamiento de la información. En especial técnicas de aprendizaje automático no supervisado para dar la rigurosidad al proceso de extracción de palabras clave.

2.2. Enunciados del problema

a) Pregunta general

¿Serán eficaces los algoritmos de aprendizaje no supervisado para la extracción de palabras clave en trabajos de investigación de pregrado en la Universidad Nacional del Altiplano de Puno?

2.3. Justificación

La necesidad de realizar verificación de la eficacia de los algoritmos de aprendizaje automático no supervisado para la generación de palabras clave de los trabajos de investigación en la Universidad Nacional del Altiplano, nace de la practicidad que se requiere para la implementación de estos en entornos de trabajo que se encuentran en ejecución, estos algoritmos utilizados en las distintas disciplinas computacionales así como la minería de datos con la minería de texto e inteligencia artificial con el aprendizaje automático ambos dedicados a la generación de conocimiento. Sin embargo, a pesar de la existencia de estos métodos de generación de conocimiento, en la actualidad no se hace uso de estos algoritmos, en tal sentido será de vital Importancia proporcionar la evaluación sobre los métodos más eficaces, para tener una implementación exitosa de los mismos.

a) Resultados Esperados de la Investigación

Este trabajo de investigación pretendió sentar las bases para el uso de la tecnología y algoritmos adecuados en la extracción de palabras clave utilizando técnicas de aprendizaje automático no supervisado, verificando su eficacia, realizando la comparación y evaluación respectiva.

b) Alcance de la Investigación

Desde la documentación bibliográfica sobre el estado del arte de la minería de textos y en particular sobre aprendizaje automático no supervisado, extracción de palabras clave, identificación de las técnicas aplicables al contexto e implementación de los modelos propuestos.

2.4. Objetivos

2.4.1. Objetivo general

El objetivo general del presente trabajo fue extraer palabras clave de manera eficaz utilizando modelos de aprendizaje automático no supervisado en trabajos de investigación de la Universidad Nacional del Altiplano de Puno.

2.4.2. Objetivos específicos

- Identificar los modelos de aprendizaje automático no supervisado para generar palabras clave en los trabajos de Investigación.
- Implementar modelos de aprendizaje automático no supervisado para generar palabras clave en los trabajos de Investigación.
- Comparar los modelos de aprendizaje automático no supervisado para generar palabras clave en los trabajos de Investigación.

2.5. Hipótesis

2.5.1. Hipótesis general

La implementación de los modelos de aprendizaje automático no supervisado permitirá extraer de manera eficaz las palabras clave en trabajos de investigación de pregrado de la Universidad Nacional del Altiplano de Puno.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. Lugar de estudio

La investigación "Algoritmos de Aprendizaje Automático no Supervisados para la Extracción de Palabras Clave en Trabajos de Investigación de Pregrado", se ejecutó durante seis meses, en el Vicerrectorado de Investigación de la Universidad Nacional del Altiplano de Puno, con la debida autorización de acceso a la información para realizar el procesamiento y análisis de datos correspondiente.

3.2. Población

La población de estudio corresponde a todos los trabajos de investigación de la Universidad Nacional del Altiplano de Puno, trabajos de investigación realizado por estudiantes y egresados de pregrado, trabajos de investigación que se encuentran registrados en sistemas como la Plataforma de Investigación Integrada a la Labor Académica con Responsabilidad (PILAR), que es administrado por el Vicerrectorado de Investigación, del cual se extrajo la información para poder realizar la presente investigación, los cuales se presentan con mayor detalle en la Tabla N° 1.

Tabla 1

Distribución Poblacional

Área del Conocimiento	Estado de los Trabajo de Investigación				
	Archivado	Borrador	Proyecto	Rechazado	Sustentado
Biomédicas	17	133	320	5	973
Económico Empresariales	32	162	286	10	465
Ingenierías	60	368	847	24	1211
Sociales	34	291	891	20	1281
Total, Parcial	143	954	2344	59	3930
Total, General	7430				

Fuente: Base de datos de la Plataforma de Investigación Integrada a la Labor Académica con Responsabilidad (PILAR) – UNA PUNO 2016-2021. Exportado el 01-09-2021

3.3. Muestra

La selección de la muestra estratificada se caracteriza por la división de la población en subgrupos y/o estratos que tienen alguna característica común; además, interesa mantener estos estratos en la muestra (Icart-Isern *et al.*, 2000), puesto que es esencial conservar las características del universo, sin embargo para el desarrollo del presente trabajo no se utilizó muestreo y se utilizó la cantidad de datos disponibles, al no ser este un problema de clasificación de documentos y con la finalidad de contar un resultados más precisos y realizar pruebas de rendimiento a los algoritmos de extracción de palabras clave.

3.4. Método de investigación

3.4.1. Metodología

En el presente apartado especificaremos las características teóricas técnicas que se utilizaron para llevar a cabo este trabajo de investigación.

3.4.2. Operacionalización de Variables

Para tener clara la metodología a seguir es importante tener en cuenta la Operacionalización de variables.

Tabla 2

Operacionalización de Variables

Variable	Dimensión	Indicador	Escala
Variable Independiente Modelos de Extracción de Palabras Clave	Algoritmos de Aprendizaje Automático No Supervisados	Tiempo de Ejecución	Tiempo
Variable Dependiente Palabras clave	Número de Palabras Clave	Score	Índice de Acierto

Fuente: El presente trabajo de Investigación.

3.4.3. Modelos de extracción de palabras clave

Para la extracción de palabras clave a partir de los datos almacenados en una base de datos es necesario tener una visión general sobre los métodos más utilizados para este fin, conforme a lo definido en nuestro marco conceptual y según la revisión de literatura realizada por Santosh *et al.* (2017), sobre las metodologías y algoritmos utilizados para la extracción de palabras clave tenemos la estructura definida en la siguiente figura.

Dado este precedente se realizó la revisión teórica y técnica para la implementación de estos algoritmos en los metadatos obtenidos de los Proyectos de Investigación de la Universidad Nacional del Altiplano, al mismo tiempo se confrontó los resultados presentados de (Godoy, 2017; Hult, 2004; Santosh *et al.*, 2017). Para los algoritmos propuestos en la Figura N° 2 se ha seleccionado las siguientes técnicas:

3.4.4. Desarrollo e implementación algorítmica

El uso de las diferentes técnicas para resolver este tipo de problemas de extracción supone la elección, prueba y evaluación de los diferentes tipos de algoritmos

3.4.5. Pruebas unitarias

En 1957 se conoce la prueba del Debugging y Dijkstra en 1970 presenta una afirmación: “La prueba de software puede ser usada para mostrar la presencia de bugs, pero nunca su ausencia" (Dijkstra, 1969). Según Swebook: "Es una actividad realizada para evaluar la calidad del producto y mejorarla, identificando defectos y problemas" (Bourque & Fairley, 2014). Prueba de software: "Es la verificación dinámica del comportamiento de un programa contra el comportamiento esperado, usando un conjunto finito de casos de prueba, seleccionados de manera adecuada" (ISTQB, 2018).

Las pruebas unitarias supondrán la evaluación específica de cada uno de los casos y volúmenes de información.

3.4.6. Tipo y diseño de la investigación

a) Tipo de Investigación

Para el presente trabajo de investigación es de corte transversal, descriptivo comparativo, porque se pretende la evaluación de la variable, y su rendimiento, para posteriormente emitir un juicio respecto a esta medición y obtener un resultado en el presente trabajo de investigación. "Por lo general, se aplican pruebas existentes, con lo que se caracteriza a los grupos específicos, a través de los cuales se revelarán las virtudes y defectos, los aciertos y errores de forma cuantitativa. Para analizar los datos recolectados en los estudios de caso, estos deben clasificarse, categorizarse e interpretarse" (Thomas & Nelson, 1986)

Hernández *et al.* (2017), sostiene que, al hacer uso del método lógico hipotético deductivo, el investigador primero formula una hipótesis y después a partir de inferencias lógicas deductivas, logra a conclusiones particulares que posteriormente son corroborables.

b) Diseño de investigación

La investigación es de tipo no experimental, de corte transversal ya que se recolecta datos en un solo momento, en un tiempo determinado. Con la finalidad de describir los resultados en función de la información recopilada de cada una de las pruebas realizadas en el presente estudio.

3.5. Descripción detallada de métodos por objetivos específicos

3.5.1. Técnicas e instrumentos de recolección y procesamiento de datos

3.5.1.1. Técnica

Se utilizó estadística descriptiva para el análisis de frecuencias y porcentajes por otro lado estadística inferencial para probar las hipótesis, por otro lado, las técnicas de evaluación serán a través de pruebas de rendimiento en ejecución con distintos volúmenes de datos y según el enfoque aplicado.

3.5.1.2. Instrumento

El instrumento está referido a una evaluación de los métodos que es definida como el proceso de recolección de información y juicio significado de la información (Disch y Moor, 2003). Para juzgar un fenómeno es necesario contar con un parámetro de referencia a partir del cual es posible tomar una decisión (Canales, 2001), por cuanto el instrumento de evaluación del presente trabajo se encuentra basado en los registros técnicos de ejecución de los modelos evaluados.

3.5.1.3. Procesamiento de datos

La evaluación de los algoritmos utilizados para la extracción de palabras clave que se encuentran al analizar información almacenada mediante sistemas de información que administran los trabajos de investigación, hace que tengamos información válida y confiable con respecto a la calidad de la misma dado que son procedimientos reglamentados dentro de la Universidad Nacional del Altiplano de Puno. Para el procesamiento requerido y detallado en el primer capítulo, se utilizó Python como leguaje de programación e interprete para todo el proceso de implementación algorítmica y de extracción de palabras clave.

3.5.1.4. Análisis de datos

Para el análisis de datos del presente trabajo de investigación se utilizó el programa informático R y hojas de cálculo para que, con los registros obtenidos de la ejecución los modelos aplicados en el procesamiento de registros, se obtengan los resultados presentados en el siguiente capítulo.

3.5.1.5. Frecuencia de términos

Consiste en una lista de términos normalizados, lo cual es acompañado por su frecuencia de aparición adicionalmente, los términos pertenecientes al índice pueden estar en su forma original o lematizados y pueden ser palabras simples, compuestas, siglas o nombres propios, en síntesis, todo lo que contenga el texto (Tolosa & Bordignon, 2008). Además de ver la Frecuencia de Términos es importante complementar indicadores basados en este los cuales describimos en las siguientes ecuaciones.

$$FT_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Ecuación N° 1

Frecuencia de Términos

$$FDI_{(wt)} = \log\left(\frac{N}{ft_i}\right)$$

Ecuación N° 2

Frecuencia de Documentos Inversa

$$FT - FDI_{(wt)} = FT_{i,j} x\ FDI_{(wt)}$$

Ecuación N° 3

Frecuencia de Términos - Frecuencia de Documentos Inversa

3.5.1.6. Matriz de documentos-términos

Una técnica común para la extracción de palabras clave y en general procesamiento del lenguaje natural es la creación de la matriz de documentos-términos, donde las filas corresponden a los documentos y las columnas a los términos. Donde los términos de mayor frecuencia es decir con mayor número de ocurrencias en un texto son catalogados como más importantes, según el cómputo general del documento procesado (Torres, 2017).

$$\begin{bmatrix} & Term1 & Term2 & \dots & TermN \\ Doc1 & C_{1,1} & C_{2,1} & \dots & C_{n,1} \\ Doc2 & C_{1,2} & C_{2,2} & \dots & C_{n,2} \\ \vdots & \vdots & \vdots & \ddots & \dots \\ DocN & C_{1,n} & C_{2,n} & \dots & C_{n,n} \end{bmatrix}$$

Figura 1. Matriz Documentos-Términos

3.5.1.7. Fuente primaria de datos

Para el desarrollo de esta investigación se ha preparado un registro de los proyectos de investigación de estudiantes y egresados, que se administran en el Vicerrectorado de Investigación de la Universidad Nacional del Altiplano de Puno, tomando como fuente inicial la estructura de datos, con 7430 registros que se detalla en la Tabla 3, estos registros fueron exportados y procesados de manera individual en archivos de texto individuales en formato (.txt) debido al coste que representaría el almacenamiento en memoria para el procesamiento en bloque.

Tabla 3

Formato de Datos de Análisis

N°	Nombre	Detalle
01	Tipo	*Tipo de trabajo de investigación*
02	Código	*Código de Trabajo en Base de Datos*
03	Título	*Título del trabajo registrado por el autor.*
04	Resumen	*Resumen registrado por el autor.*
05	Palabras Clave	*Resumen registrado por el autor. [,]*

3.5.2. Tratamiento de datos

Para desarrollar el presente trabajo de investigación se utilizó la información registrada por el Vicerrectorado de Investigación de la Universidad Nacional del Altiplano de Puno, mediante la Plataforma de Investigación Integrada a la Labor Académica con Responsabilidad (PILAR), que ascienden a 7430 registros de títulos detallados en la Tabla 2 , cuya estructura se encuentra compuesta por títulos, resúmenes y palabras clave en la estructura declarados en la Tabla 3, estos datos corresponden a los trabajos de investigación de estudiantes y egresados que optaron por la modalidad de tesis para la obtención de su título profesional.

Esta información que se ha procesado se encuentra en formato de texto plano, inició su tratamiento utilizando métodos de expresiones regulares con la finalidad de normalizar el texto y excluir los caracteres especiales y quedarnos únicamente con el análisis del texto, para posteriormente realizar la segmentación y procedimientos necesarios para mantener los datos un formato operable.

3.5.3. Recurso informático

Para realizar una evaluación uniforme de los datos, se realizó la evaluación en un mismo equipo informático con las características técnicas siguientes:

Tabla 4

Características Técnicas del Equipo Informático

Nombre	**Detalle**
Procesador	*Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz 2.60 GHz*
Memoria RAM	*16.0 GB*
Disco Duro	*1 TB.*
Sistema Operativo	*Ubuntu Linux*

3.5.4. Implementación de modelos

Para el presente trabajo se han utilizado los modelos de Python Keyword Extraction PKE (Boudin, 2016), estos modelos fueron codificados con la siguiente estructura de equivalencia de nombres de archivo para el presente trabajo de investigación TF-IDF (M1), KPMiner (M2), YAKE (M3), TextRank (M4), SingleRank (M5), TopicRank (M6), TopicPageRank (M7), PositionRank (M8), MultipartRank (M9), de estos presentamos la evaluación e implementación en el siguiente capítulo conforme a los objetivos planteados en el presente trabajo de investigación.

3.5.5. Evaluación de modelos

Para la evaluación de los modelos implementados aplicaron las siguientes métricas de evaluación para lo cual se utilizó la librería sklearn, como una de las más utilizadas en este tipo de evaluaciones así mismo se ha considerado su adaptabilidad a los resultados obtenidos en la predicción de cada grupo de palabras clave.

Tabla 5

Métricas de Evaluación de Modelos

Nombre	**Detalle**
Test F1 *F1-Score*	La puntuación F1 se puede interpretar como una media armónica de la precisión y la recuperación, donde una puntuación F1 alcanza su mejor valor en 1 y la peor puntuación en 0.
Test de recordación *Recall Score*	El retiro es la proporción donde está el número de verdaderos positivos y el número de falsos negativos. El retiro es intuitivamente la capacidad del clasificador de encontrar todas las muestras positivas.
Test de Precisión *Precision Score*	La precisión es la razón donde está el número de verdaderos positivos y el número de falsos positivos. La precisión es intuitivamente la capacidad del clasificador de no etiquetar como positiva una muestra que es negativa
Puntuación de Precisión *Acuracy Score*	Calcula la precisión del subconjunto: el conjunto de etiquetas predichas para una muestra debe coincidir exactamente con el conjunto de etiquetas correspondiente en la lista de etiquetas originales
Pérdida Promedio *Hamming Score*	La pérdida de *hamming* es el número de etiquetas que se predice incorrectamente.
Tiempo *Time*	El tiempo se encuentra detallado desde la lectura del recurso fuente hasta la predicción final.

Fuente: (Scikit learn, 2021) Métricas y puntuación: cuantificación de la calidad de las predicciones. Modules. https://scikit-learn.org/stable/modules/model_evaluation.html

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

Al realizar la implementación de los métodos de extracción de palabras clave fue necesario codificar individualmente los nueve (09) modelos descritos en el punto 3.3 del capítulo anterior, para los cuales se realizó una evaluación de tiempo y precisión conforme a las métricas detalladas en la Tabla 5 para cada uno de los 7430 documentos estas métricas de predicción nos permitieron determinar la eficacia de la extracción de palabras clave computado para cada modelo. En la Tabla 6 presentamos un cuadro comparativo de las predicciones realizadas para cada modelo con un documento de ejemplo, obteniendo diferentes resultados como podemos ver a continuación.

Tabla 6

Ejemplo de Extracción de Palabras Clave

Nombre Modelo	Palabras Clave
Palabras Clave	materiales audiovisuales, aprendizaje, investigación, rendimiento
M1 *TF-IDF*	materiales audiovisuales, audiovisuales, educativa politécnico, politécnico, Huáscar
M2 *KPMiner*	materiales audiovisuales, primer grado, primer, institución educativa, institución educativa politécnico
M3 *YAKE*	materiales audiovisuales, educativa politécnico, institución educativa, politécnico Huáscar, estudiantes
M4 *TextRank*	estudiantes primer grado, primer grado, estudiantes, investigación, trabajo

M5	*SingleRank*	estudiantes primer grado, estudiantes, materiales, aprendizaje, grado
M6	To*picRank*	estudiantes, aprendizaje, materiales audiovisuales, institución, año
M7	To*picPageRank*	estudiantes primer grado, institución educativa politécnico Huáscar, educativa politécnico Huáscar, institución educativa politécnico, estudiantes
M8	*PositionRank*	estudiantes primer grado, materiales audiovisuales, institución educativa politécnico, primer grado b, educativa politécnico Huáscar
M9	*MultipartRank*	estudiantes, materiales audiovisuales, aprendizaje, institución educativa politécnico Huáscar, año, investigación, rendimiento escolar, puno, medida, área

Las palabras clave resultantes de la implementación de cada uno de estos modelos se almacenaron en vectores incluyendo incluyéndose el cómputo del tiempo en el cálculo del procesamiento de cada modelo implementado estos vectores fueron almacenados en archivos de texto individuales primigeniamente, por otro lado, el cálculo de las métricas de evaluación de cada uno los modelos implementados se realizó con la librería Scikit Learn, pasando como argumentos los vectores originales y de extracción respectivamente.

4.1. Extracción de palabras clave

Al combinar el tiempo y las métricas podemos evaluar la eficiencia al extraer palabras clave, en la Figura 3 tomando como referencia el valor calculado del F1- Score podremos observar el detalle de evaluación del modelo y su estimación de tiempo de manera individual, por lo que de las combinaciones de todos los modelos podemos decir que no se tiene una diferencia visualmente significativa entre los modelos implementados con respecto al tiempo y a la precisión de la implementación a excepción de algunos modelos.

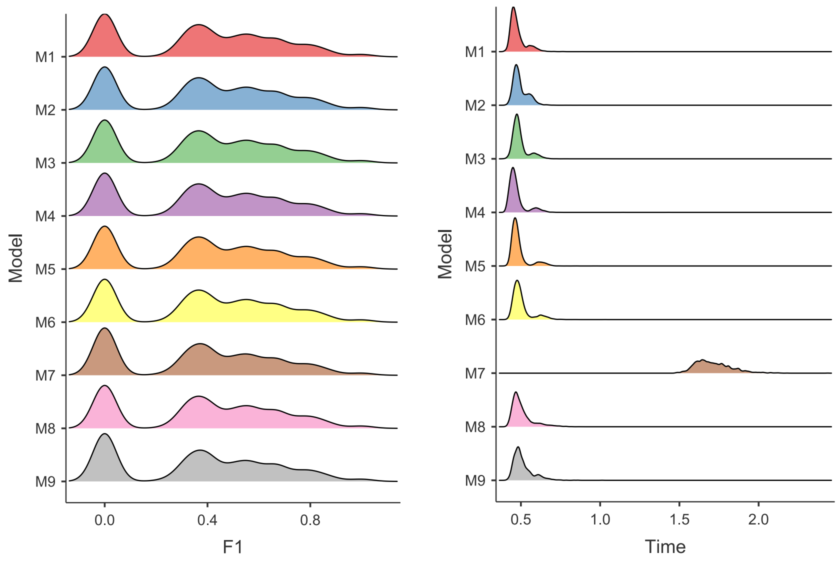

Figura 2. Densidad de Precisión F1-Score y Tiempo.

A simple vista parece no haber una diferencia significativa entre la implementación de los modelos, no obstante, si empezamos a desglosar el análisis individualmente empezaremos a observar las diferencias entre cada uno de estos modelos, que desde ya nos muestran una respuesta rápida en la implementación de extracción de palabras clave de manera individual.

De manera inicial y desde una inspección gráfica de la implementación de los modelos no se ha podido determinar verdadera la diferencia de este análisis, es por cuanto procedemos a realizar el análisis de varianza (ANOVA) para identificar la diferencia de medias para cada modelo implementado, en la Tabla 7 donde se puede confirmar como única diferencia significativa el tiempo de ejecución tomando un valor de $p < 0.005$ y dejando a la precisión de cálculo F1 relegada por una diferencia de 0.007 para alcanzar a ser significativo lo que nos sugiere una similitud en la implementación de este tipo de modelos debido a su agilidad y nivel de precisión promedio.

Tabla 7

One-Way ANOVA (Welch's)

	F	df1	df2	p
Time	29330.1788	8	16197.4216	< .00001
F1	1.8896	8	16460.2705	0.05699

La prueba t de Welch se ha utilizado para comparar las medias entre dos grupos independientes que vienen a ser los nueve 09 modelos de implementados para la extracción de palabras clave, así mismo por la gran cantidad de datos analizados se realizó el análisis de varianza de entre los índices de precisión y tiempo respectivamente como se observa en la Tabla 7, así mismo se rectificó el análisis previo con la homogeneidad de varianzas de Levene en la Tabla 8.

Tabla 8

Test de Homogeneidad de Varianzas (Levene)

	F	df1	df2	p
Time	455.4304	8	56087	< .00001
F1	0.7050	8	56087	0.68750

Con este análisis se explicó cómo se determinará el mejor modelo en una posible implantación de un extractor de palabras clave utilizando algoritmos de aprendizaje automático no supervisado, quedando descartado la precisión, dado que no existe diferencia significativa entre este grupo por tener un valor [p > 0.05] = [0.05699 > 0.05], es por cuanto se procedió a realizar un análisis detallado de la implementación de cada uno de los modelos, los valores de tiempo estimados y el cálculo del indicador de precisión en la extracción de palabras clave, explicado por la diferencia de medias calculado.

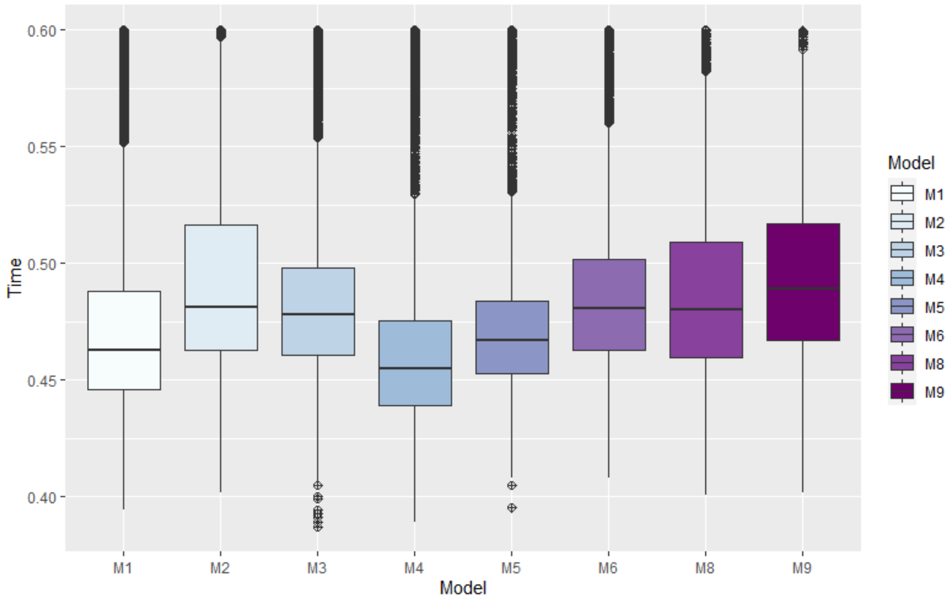

Figura 3. Visualización Diferencia de Medias.

Tabla 9

Descriptivos ANOVA Welch's

	Modelo	N	Mean	SD	SE
	TF-IDF, [M1]	7430	**0.4786**	0.0501	5.8174e-4
	KPMiner, [M2]	7430	0.4952	0.0463	5.3712e-4
	YAKE, [M3]	7430	0.4919	0.0482	5.5899e-4
	TextRank, [M4]	7430	**0.4725**	0.0557	6.4612e-4
Tiempo	SingleRank, [M5]	7430	0.4863	0.0569	6.5992e-4
	TopicRank, [M6]	7430	0.5015	0.0596	6.9102e-4
	To*picPageRank*, [M7]	2007	1.7140	0.1132	0.0025
	PositionRank, [M8]	7430	0.5037	0.0655	7.6006e-4
	MultipartRank, [M9]	2079	0.5114	0.0626	0.0014

Realizando un análisis visual y observando los valores del ANOVA de Welch se puede remarcar los valores más pequeños en el promedio de por lo que podemos que una diferencia significativa entre la implementación de modelos TF-IDF [M1] (0.4786) y TextRank [M4] (0.4725), con respecto a los demás modelos por su optimización en cuanto al tiempo de ejecución lo que nos ayudó a comprobar la hipótesis de este trabajo de investigación, siendo estos los modelos más óptimos de implementar en un posible despliegue del extractor de palabras clave.

Los modelos basados en TF-IDF, han venido siendo estudiados a lo largo de muchos años mostrando siempre su eficiencia, así sea con las mejoras como las que se desarrollaron el trabajo de Guan *et al.* (2019), en el que se realiza el recalculo de los scores de predicción optimizando hasta en un 20 % los indicadores evaluados. Lo mismo sucede con el modelo TextRank pero en menor rango en el que en varios estudios y en especial en el que definen su implementación se demuestra su precisión comparando su implementación con diferentes grupos de datos (Mihalcea & Tarau, 2004), a lo largo de tiempo varios estudios vienen demostrando que la implementación de estos modelos que son teóricamente sencillos demuestran sus grandes capacidades de procesamiento.

Es importante mencionar también en este análisis de eficiencia que los modelos TopicPageRank [M7]y MultipartRank[M9], no han presentado un buen desempeño de manera general con la cantidad de datos usados en la ejecución del presente trabajo de investigación lo que generó un error en el tiempo de ejecución en tanto solo se ha logrado procesar el 28 % de la información total, no por ello deben de dejar de ser aplicados si no estos modelos pueden ser aplicados en otros contextos y con diferentes conjuntos de datos, dado esta justificación con fines de visualización se ha extraído de la siguiente figura el Modelo TopicPageRank [M7].

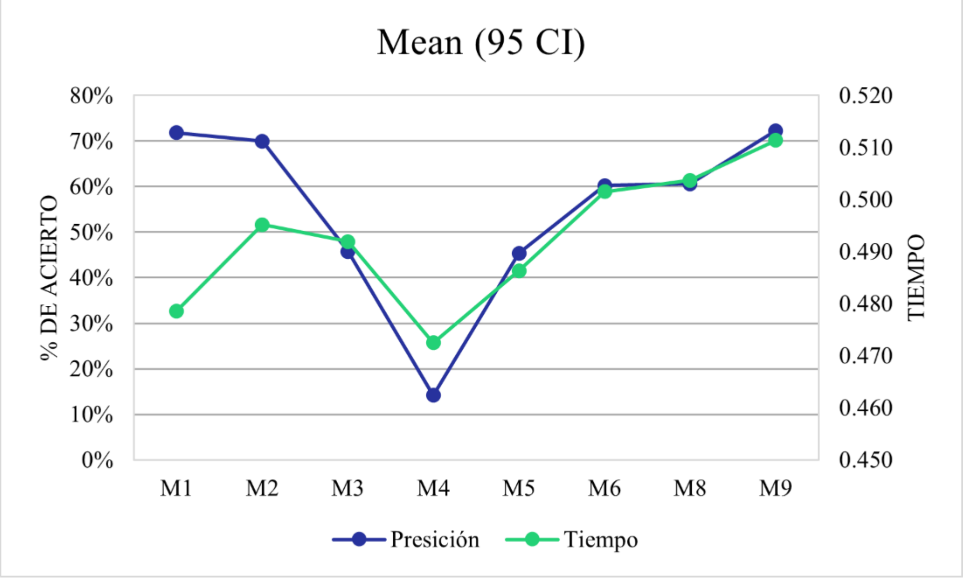

Figura 4. Comparación Tiempo, Acierto.

Con la descripción presentada dejamos por comprobada la hipótesis en el presente trabajo de investigación, confirmando la eficacia de los modelos simples tal como es el modelo TF-IDF denotando su gran potencia para realizar tareas de extracción de palabras clave en grandes volúmenes, así mismo es importante continuar realizando evaluaciones técnicas conjuntamente con el incremento de datos, de este modo mantener actualizados estos registros, finalmente también sería importante determinar la eficiencia de los modelos utilizando otro tipo de equipamiento de hardware, para continuar evaluando este modelo pero siempre garantizando la igualdad de condiciones al ejecutar cada modelo.

4.2. Identificación

Para el desarrollo del presente trabajo de investigación se realizó la búsqueda de literatura científica y académica de trabajos de investigación que pretendieron extraer palabras clave utilizando técnicas de aprendizaje automático, resaltando entre todas las técnicas más estudiadas los modelos de aprendizaje no supervisado, visto el contexto de una aplicación de uso en este grupo se tendría un manejo de información óptimo, sin la necesidad de reentrenar los modelos para ajustar la precisión al momento cálculo, es por ello que cada uno de los modelos implementados fueron evaluados y se tomó como fuente de codificación se utilizó la herramienta de código abierto Python Keyphrase Extraction, PKE (Boudin, 2016), en la que nos ofrece los distintos modelos, revisando así la literatura de cada uno de los modelos implementados, y la documentación prevista por el autor de estos módulos.

Tabla 10

Identificación de Modelos de Aprendizaje Automático No Supervisados

Modelo	Abrev.	Nombre	Autor
Modelos No Supervisados	M1	*TF-IDF*	-
	M2	*KPMiner*	(El-Beltagy & Rafea, 2010)
	M3	*YAKE*	(Campos et al., 2020)
	M4	TextRank	(Mihalcea & Tarau, 2004)
	M5	SingleRank	(Wan & Xiao, 2008)
	M6	TopicRank	(Bougouin et al., 2013)
	M7	To*picPageRank*	(Sterckx et al., 2015)
	M8	PositionRank	(Florescu & Caragea, 2017)
	M9	MultipartRank	(Boudin, 2018)

4.3. Implementación

Para la implementación de cada uno de los métodos, se utilizó el lenguaje de programación Python, tomando como fuente primaria de información se tuvo los archivos almacenados en formato de texto plano (.txt), se inició el procedimiento de implementación con la estandarización de contenido utilizando el ISO 8859-1 que es una norma de la ISO que define la codificación del alfabeto latino, incluyendo los diacríticos, y letras especiales, conocido también como latin-1 (ISO 8859-1, 1999), para así garantizar la compatibilidad en el tratamiento de datos. Luego con ayuda de las técnicas de expresiones regulares se realizó el proceso de limpieza de caracteres especiales como tabulaciones ("\t"), espacios ("") y cualquier otro símbolo que sea texto, para finalmente iterar en cada uno de los archivos y obtener un nuevo archivo de texto esta vez procesable.

La extracción de palabras clave ha venido siendo estudiado a lo largo de todos estos años, sin embargo la mayoría de estudios realizados a la actualidad y que son de referencia con grandes cantidad de información analizadas son estudios que se realizaron en el idioma inglés, siendo esta implementación un referente para los estudios posteriores en el ámbito de la extracción de palabras clave, a razón de esto la librería que nos fue de gran ayuda fue *spacy* y su lista de *stopwords* en español, lo que nos permitió limpiar bien las cadenas a procesar.

Luego de ello se realizó la adaptación de cada uno de los modelos descritos en el apartado 4.2, para lo cual presentamos a continuación las particularidades de cada implementación, de la cual también se obtuvo un análisis individual con respecto al tiempo y las métricas de precisión, en las siguientes figuras presentamos cada uno de los modelos implementados para la extracción de palabras clave

En la implementación de algunos de estos modelos, se han utilizado cálculos como la frecuencia de términos del documento y la asignación latente de Dirchlet (Blei *et al.*, 2003) los cuales fueron calculados previamente con una estructura similar y con las consideraciones de limpieza de texto correspondiente, con la finalidad de evaluar los pesos de manera general y tener una referencia de palabras que podrían causar redundancia en el análisis, así mismo es importante precisar que en estos modelos se referenció a múltiples herramientas, conceptos básicos y librerías para lograr el procesamiento de los resultados obtenidos.

Además de ello debemos indicar que la implementación no fue igual para cada uno de los modelos, en vista de que cada uno de ellos tiene sus propias características, en tal caso presentamos los puntos más importantes para la implementación y desarrollo de este trabajo de investigación.

Pseudo código por cada modelo puesto a prueba:

a) Modelo M1

```
IMPORT time, sys, os, pke, numpy, spacy, punctuation
SET documentos TO os.listdir('/home/datadir /')
SET df TO load_document_frequency_file ( INPUT ‘file.tsv.gz' )
SET stopWords TO list(stopWords) + list(punctuation)
FOR file IN documentos:
        SET starTime TO time.time()
        SET extractTfidf TO pke.unsupervised.TfIdf()
        extractTfidf.load_document(
                INPUT='/home/data/'+file,
                encoding='latin1',
                language='es',
                normalization='none'
        )
        extractTfidf.candidate_selection(n=2, stoplist=stopWords)
        extractTfidf.candidate_weighting(df=df)
        SET keyWords TO extractTfidf.get_n_best(n=5)
        SET endTime TO time.time()
        timess=endTime - starTime
        keyWords.append(('time',timess))
        np.savetxt('/home/'+file, keyWords , delimiter="\n", fmt="%s")
        OUTPUT ("Archivo Generado "+file+"!")
```

b) Modelo M2

```
IMPORT time, sys, os, pke, numpy, spacy, punctuation
SET documentos TO os.listdir('Directorio/Data1/')
SET df TO
pke.load_document_frequency_file(INPUT_file='Directorio/DocumentFrecuency/D
ocumentFrecuency1.1.tsv.gz')
FOR file IN documentos:
   SET starTime TO time.time()
   SET keyWords TO pke.unsupervised.KPMiner()
   keyWords.load_document(
      INPUT='Directorio/Data1/'+file,
      encoding='latin1',
      language='es',
      normalization='none'   )
   keyWords.candidate_selection(lasf=3, cutoff=400)
   SET alpha TO 3.3
   SET sigma TO 2.3
   keyWords.candidate_weighting(df=df, alpha=alpha, sigma=sigma)
   SET keyWords TO keyWords.get_n_best(n=5)
   SET endTime TO time.time()
```

```
    timess=endTime - starTime
    keyWords.append(('time',timess))
    np.savetxt('/d/models/M2/'+file, keyWords , delimiter="\n", fmt="%s")
    OUTPUT ("Archivo Generado "+file+" !")
```

c) Modelo M3

```
IMPORT time, sys, os, pke, numpy, spacy, punctuation
from string IMPORT punctuation
from spacy.lang.es.stop_words IMPORT stop_words
SET documentos TO os.listdir('Directorio/Data1/')
stopWords=list(STOP_WORDS)+list(punctuation)
FOR file IN documentos:
   SET starTime TO time.time()
   SET extractor TO pke.unsupervised.YAKE()
   extractor.load_document(
               INPUT='Directorio/Data1/'+file,
               encoding='latin1',
               language='es',
               normalization='none'    )
   extractor.candidate_selection(n=2, stoplist=stopWords)
   SET window TO 1
   extractor.candidate_weighting(window=window,
                          stoplist=stopWords,
                          use_stems=use_stems)
   SET threshold TO 0.8
   SET keyWords TO extractor.get_n_best(n=5, threshold=threshold)
   SET endTime TO time.time()
   timess=endTime - starTime
   keyWords.append(('time',timess))
   np.savetxt('/d/models/M3/'+file, keyWords , delimiter="\n", fmt="%s")
   OUTPUT("Archivo Generado "+file+"!")
```

d) Modelo M4

```
IMPORT time, sys, os, pke, numpy, spacy, punctuation
SET documentos TO os.listdir('Directorio/Data1/')
SET pos TO {'NOUN'}
FOR file IN documentos:
   SET starTime TO time.time()
   SET kwTR TO pke.unsupervised.TextRank()
   kwTR.load_document(
               INPUT='Directorio/Data1/'+file,
               encoding='latin1',
               language='es',
               normalization='none'  )
kwTR.candidate_weighting(window=1,pos=pos,top_percent=0.33,normalized=False)
   keyWords =kwTR.get_n_best(n=5)
   SET endTime TO time.time()
   timess=endTime - starTime
   keyWords.append(('time',timess))
   np.savetxt('/d/models/M4/'+file, keyWords , delimiter="\n", fmt="%s")
   OUTPUT("Archivo Generado "+file+"!")
```

e) Modelo M5

```
IMPORT time, sys, os, pke, numpy, spacy, punctuation
SET pos TO {'NOUN'}
SET documentos TO os.listdir('Directorio/Data1/')
FOR file IN documentos:
    SET starTime TO time.time()
    SET kwSingler TO pke.unsupervised.SingleRank()
    kwSingler.load_document(
    INPUT='Directorio/Data1/'+file,
    encoding='latin1',
    language='es',
    normalization='none'
    )
    kwSingler.candidate_selection(pos=pos)
    kwSingler.candidate_weighting(window=10, pos=pos)
    SET keyWords TO kwSingler.get_n_best(n=5)
   SET endTime TO time.time()
    timess=endTime - starTime
    keyWords.append(('time',timess))
    np.savetxt('/d/models/M5/'+file, keyWords , delimiter="\n", fmt="%s")
    OUTPUT("Archivo Generado "+file+"!")
            np.savetxt('/home/'+file, keyWords , delimiter="\n", fmt="%s")
            OUTPUT ("Archivo Generado "+file+"!")
```

f) Modelo M6

```
IMPORT time, sys, os, pke,nltk ,numpy, spacy, punctuation
SET pos TO {'NOUN', 'PROPN'}
SET documentos TO os.listdir('Directorio/Data1/')
SET stoplist TO ['-lrb-', '-rrb-', '-lcb-', '-rcb-', '-lsb-', '-rsb-']
stoplist += stopwords.words('spanish')
stopW=list(STOP_WORDS)+list(punctuation)+list(stoplist)
FOR file IN documentos:
    SET starTime TO time.time()
    SET kwTR TO pke.unsupervised.TopicRank()
    kwTR.load_document(
          INPUT='Directorio/Data1/'+file,
          encoding='latin1',
          language='es',
          normalization='none'    )
    kwTR.candidate_selection(pos=pos, stoplist=stopW)
    kwTR.candidate_weighting(threshold=0.75, method='average')
    SET keyWords TO kwTR.get_n_best(n=5)
    SET endTime TO time.time()
    timess=endTime - starTime
    keyWords.append(('time',timess))
    np.savetxt('/d/models/M6/'+file, keyWords , delimiter="\n", fmt="%s")
    OUTPUT("Archivo Generado "+file+"!")
```

g) Modelo M7

```
IMPORT time, sys, os, pke,nltk ,numpy, spacy, punctuation
SET documentos TO os.listdir('Directorio/Data1/')
SET stopW TO list(STOP_WORDS)+list(punctuation)+list(stoplist)
SET grammar TO "NP: {<ADJ>*<NOUN|PROPN>+}"
SET pos TO {'NOUN', 'PROPN', 'ADJ'}
FOR file IN documentos:
   SET starTime TO time.time()
   SET kwTPR TO pke.unsupervised.TopicalPageRank()
            kwTPR.load_document(
            INPUT='Directorio/Data1/'+file,
            encoding='latin1',
            language='es',
            normalization='none'
            )
   kwTPR.candidate_selection(grammar=grammar)
   kwTPR.candidate_weighting(window=20,
                      pos=pos,
                      lda_model='Directorio/DocumentFrecuency/LDA-Model.gz')
   SET keyWords TO kwTPR.get_n_best(n=5)
   SET endTime TO time.time()
   timess=endTime - starTime
   keyWords.append(('time',timess))
   np.savetxt('/d/models/M7/'+file, keyWords , delimiter="\n", fmt="%s")
   OUTPUT("Archivo Generado "+file+"!")
```

h) Modelo M8

```
IMPORT time, sys, os, pke,nltk ,numpy, spacy, punctuation
SET documentos TO os.listdir('Directorio/Data1/')
SET pos TO {'NOUN', 'PROPN', 'ADJ'}
SET grammar TO "NP: {<ADJ>*<NOUN|PROPN>+}"
FOR file IN documentos:
   SET starTime TO time.time()
   SET kwPR TO pke.unsupervised.PositionRank()
   kwPR.load_document(
      INPUT='Directorio/Data1/'+file,
      encoding='latin1',
      language='es',
      normalization='none'   )
   kwPR.candidate_selection(grammar=grammar,maximum_word_number=3)
   kwPR.candidate_weighting(window=10, pos=pos)
   SET keyWords TO kwPR.get_n_best(n=5)
   SET endTime TO time.time()
   timess=endTime - starTime
   keyWords.append(('time',timess))
   np.savetxt('/d/models/M8/'+file, keyWords , delimiter="\n", fmt="%s")
   OUTPUT("Archivo Generado "+file+"!")
```

i) Modelo M9

```
IMPORT time, sys, os, pke,nltk ,numpy, spacy, punctuation
SET documentos TO os.listdir('Directorio/Data1/')
SET pos TO {'NOUN', 'PROPN', 'ADJ'}
SET stoplist TO list(string.punctuation)
stoplist += ['-lrb-', '-rrb-', '-lcb-', '-rcb-', '-lsb-', '-rsb-']
stoplist += stopwords.words('spanish')
stopWords=list(STOP_WORDS)+list(punctuation)+list(stoplist)
FOR file IN documentos:
    SET starTime TO time.time()
    SET kwMtp TO pke.unsupervised.MultipartiteRank()
    kwMtp.load_document(
            INPUT='Directorio/Data1/'+file,
            encoding='latin1',
            language='es',
            normalization='none'    )
    kwMtp.candidate_selection(pos=pos, stoplist=stopWords)
    kwMtp.candidate_weighting(alpha=1.1,
                    threshold=0.74,
                    method='average')
    SET keyWords TO kwMtp.get_n_best(n=10)
    SET endTime TO time.time()
    timess=endTime - starTime
    keyWords.append(('time',timess))
    np.savetxt('/d/models/M9/'+file, keyWords , delimiter="\n", fmt="%s")
    OUTPUT ("Archivo Generado "+file+"!")
```

En los Anexos 1 al 9 al presente trabajo de investigación presentamos el código fuente de la implementación de los modelos.

4.4. Comparación

Como se ha evidenciado en el punto 4.3 se ha realizado la adaptación para la implementación individual de cada modelo según los requerimientos de cada uno de los modelos implementados, así mismo se realizó la evaluación de manera individual de cada uno de estos en función al tiempo de ejecución en cada una de las extracciones ejecutadas para complementar la evaluación e identificación individual de cada uno de los modelos, es importante señalar también que para una correcta visualización de tiempos se excluido los valores atípicos superiores en el análisis realizado.

Tabla 11

Métricas de Evaluación Individual de Modelos

Modelo		M1	M2	M3	M4	M5	M6	M7	M8	M9
N		**7424**	**7430**	**7430**	**7430**	**7430**	**7430**	**2007**	**7430**	**2079**
Acuracy	*Mean*	0.2775	0.2673	0.1478	0.0369	0.1472	0.2243	0.1523	0.2219	0.338
	SD	0.2354	0.2356	0.1924	0.0978	0.1917	0.2317	0.2033	0.2292	0.282
Recall	*Mean*	0.2775	0.2673	0.1478	0.0369	0.1472	0.2243	0.1523	0.2219	0.338
	SD	0.2354	0.2356	0.1924	0.0978	0.1917	0.2317	0.2033	0.2292	0.282
Precisión	*Mean*	0.7179	0.6993	0.4576	0.1419	0.4532	0.6024	0.4459	0.6061	0.723
	SD	0.4500	0.4586	0.4982	0.3489	0.4978	0.4894	0.4972	0.4887	0.448
F1 Score	*Mean*	0.3831	0.3697	0.2152	0.0574	0.2142	0.3126	0.2181	0.3106	0.439
	SD	0.2841	0.2855	0.2570	0.1469	0.2577	0.2891	0.2666	0.2859	0.320
Hamming	*Mean*	0.7225	0.7327	0.8522	0.9631	0.8528	0.7757	0.8477	0.7781	0.662
	SD	0.2354	0.2356	0.1924	0.0978	0.1917	0.2317	0.2033	0.2292	0.282

Lo característico de esta comparación de estos modelos es que se presentan altos índices en el Haming Score, lo que nos indica que para el presente caso de estudio a lo más se cuenta con una precisión un valor de perdida promedio mínimo de hasta el 60 % en la extracción de palabras clave, esto se debe a las reglas de implementación que se han dado.

Estar reglas obedecen a que los trabajos de investigación deben tener entre uno (01) y cinco (05) palabras clave elegidas por lo que para cada predicción tal y como es requerido en los trabajos de investigación se ha extraído este número máximo admitido de palabras clave. Estos resultados no deberían sorprendernos en este tipo de análisis por ser este un número reducido de palabras clave se requiere una alta precisión sin embargo esto está en función al modelo y la cantidad de texto introducido como parámetros para la evaluación.

En contraparte se ha realizado pruebas con extracción de palabras clave de al menos diez (10) palabras clave y el índice de acierto incrementa drásticamente pudiendo obtener hasta un 80 % de acierto y 20 % de pérdida respectivamente, justificándose los resultados del presente estudio por la gran cantidad de información y la variabilidad de los datos que están siendo sometidos a análisis.

Es importante resaltar en este análisis que ninguno de los trabajos citados en los antecedentes del presente trabajo de investigación ha considerado tal cantidad de registros como lo realizamos en este trabajo, presentando un nuevo hito para que se pueda seguir desarrollando más investigación, poniendo como reto el incremento del dataset y quizá se pueda trabajar con documentos completos, lo que nos podría ayudar en el posicionamiento de estos trabajos de investigación en repositorios y agilizar los procedimientos de las plataformas ya existentes.

Así mismo se realizó una comparación de los modelos que tuvieron al menos un acierto en la extracción de palabras clave, de esta manera tener una perspectiva de diferencia no solo desde el tiempo sino también con la precisión de los modelos, dejando claro que esta sería una precisión subjetiva puesto que solo hace referencia a un (01) acierto de cinco (05) que sería lo ideal, sin embargo, nos ofrece otro punto de vista en el análisis de resultados.

Tabla 12

Precisión de Extracción de Palabras Clave

Modelo	***nE***	***nA***	**Total**	**% E**	**% A**
M1	2094	5330	7424	28 %	72 %
M2	2234	5196	7430	30 %	70 %
M3	4030	3400	7430	54 %	46 %
M4	6376	1054	7430	86 %	14 %
M5	4063	3367	7430	55 %	45 %
M6	2954	4476	7430	40 %	60 %
M7	1112	895	2007	55 %	45 %
M8	2927	4503	7430	39 %	61 %
M9	576	1503	2079	28 %	72 %

Donde:

nE: Numero de Errores

nA: Número de Aciertos

% E: Porcentaje de Error

% A: Porcentaje de Acierto

Si realizamos el análisis de precisión respectivo y lo consideramos conjuntamente con nuestro primer análisis de extracción de palabras clave donde identificamos en cuestión de tiempo los modelos TF-IDF [M1] y TextRank [M4], sin embargo al incluir este último análisis en el proceso de decisión podemos decir que el modelo más eficiente para la extracción de palabras clave es el modelo basado en la frecuencia de términos y la frecuencia inversa de documentos TF-IDF [M1], debido que adicionalmente de tener un buen rendimiento en tiempo de ejecución, también la precisión de extracción de palabras clave se realiza en al menos un 72 % por lo que confirmamos el análisis realizado.

CONCLUSIONES

Concluimos el presente trabajo de investigación realizando los siguientes aportes al conocimiento en el área de procesamiento de texto utilizando modelos de aprendizaje automático:

PRIMERO: Objetivo General

La manera más eficiente de extracción de palabras clave para este conjunto de datos fue el método TF-IDF, debido a su relación entre el tiempo y la precisión del análisis, porque nos garantiza un menor tiempo de procesamiento y alto puntaje de precisión en la extracción de palabras clave por lo que lo podemos considerar un modelo simple y de carácter general para este y otros trabajos relacionados.

SEGUNDO: Objetivo Específico 1

El presente trabajo nos permitió identificar, explorar y conocer más a fondo los ocho (08) modelos restantes que se implementaron en el presente trabajo permitiéndonos conocer otros usos y aplicaciones alternativas de estos métodos que son diferentes a la extracción de palabras clave en tesis de estudiantes y egresados universitarios.

TERCERO: Objetivo Específico 2

La implementación de estos modelos estudiados, nos permitió explotar los recursos computacionales en el procesamiento de texto, debido al reto que supuso la administración y el análisis de más de cincuenta y seis mil registros (> 56000).

CUARTO: Objetivo Específico 3

La comparación de múltiples scores de precisión nos ayudó a tomar la decisión de descartar el modelo TextRank [M4] y quedarnos solo con el modelo TF-IDF [M1], debido que además del menor tiempo también obtuvo un alto índice de precisión.

RECOMENDACIONES

El presente trabajo de investigación, tuvo la intención de marcar un hito en cuanto al número de registros analizados ascendiendo a un total de 7430 registros y 56090 extracciones realizadas, con la implementación de los nueve (09) modelos no obstante es importante tener las siguientes consideraciones en futuros trabajos de investigación:

Para Investigadores del área:

- La exploración de las técnicas del procesamiento de lenguaje natural, nos ayudarán tener extracciones más precisas, por lo que debe de considerarse una opción para futuros estudios.
- El uso de modelos de lenguaje autorregresivo como GPT-3, es otra vertiente que es importante explorar debido a su gran volumen de información, no obstante, se podría contribuir con más registros en el idioma español.
- La comprobación de las predicciones nos ayudar a mejorar la precisión en la implementación de estos modelos.

Para estudiantes y egresados:

- Consideren evaluar y revisar las palabras clave que seleccionan al momento de presentar sus trabajos de investigación, porque en este estudio se pudo evaluar buenas predicciones de palabras clave, pero deficiencias en los documentos originales.
- La selección de las palabras clave adecuadas, ayudarán a implementar estos algoritmos en los sistemas y por ende mejorar la precisión.

En ambos casos estas recomendaciones ayudarán a la Universidad Nacional del Altiplano a continuar con el proceso de generación de conocimiento y en lo que concierne a las palabras clave estas son la principal fuente de indexación y visualización de trabajos de investigación.

BIBLIOGRAFÍA

Agarwal, S. (2014). Data mining: Data mining concepts and techniques. In *Proceedings - 2013 International Conference on Machine Intelligence Research and Advancement, ICMIRA 2013*. https://doi.org/10.1109/ICMIRA.2013.45

Ahadh, A., Binish, G. V., & Srinivasan, R. (2021). Text mining of accident reports using semi-supervised keyword extraction and topic modeling. *Process Safety and Environmental Protection*, *155*, 455–465. https://doi.org/10.1016/j.psep.2021.09.022

Aquino, G., & Lanzarini, L. (2015). Keyword Identification in Spanish Documents using Neural Networks. *Journal of Computer Science and Technology*, *15*(2), 55–60.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*(4–5), 993–1022. https://doi.org/10.1016/b978-0-12-411519-4.00006-9

Botta-Ferret, E., & Cabrera-Gato, J. E. (2007). Minería de textos: Una herramienta útil para mejorar la gestión del bibliotecario en el entorno digital. *Acimed*, *16*(4).

Boudin, F. (2018). Unsupervised keyphrase extraction with multipartite graphs. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *2*, 667–672. https://doi.org/10.18653/v1/n18-2105

Boudin, F. (2016). pke: an open source python-based keyphrase extraction toolkit. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, 69–73. Recuperado de http://aclweb.org/anthology/C16-2015

Bougouin, A., Boudin, F., & Daille, B. (2013). TopicRank: Topic ranking for automatic keyphrase extraction. *Revue Traitement Automatique Des Langues*, *55*(1), 45–69.

Bourque, P., & Fairley, R. E. (2014). *Guide to the Software Engineering Body of Knowledge* (Vol. 3). Recuperado de www.swebok.org.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features.

Information Sciences, *509*, 257–289. https://doi.org/10.1016/j.ins.2019.09.013

Chahine, C. A., Chaignaud, N., Kotowicz, J. P., & Pécuchet, J. P. (2008). Context and keyword extraction in plain text using a graph representation. *SITIS 2008 - Proceedings of the 4th International Conference on Signal Image Technology and Internet Based Systems*, 692–696. https://doi.org/10.1109/SITIS.2008.47

Coursey, K. H., Mihalcea, R., & Moen, W. E. (2009). Automatic keyword extraction for learning object repositories. *Proceedings of the American Society for Information Science and Technology*, *45*(1), 1–10. https://doi.org/10.1002/meet.2008.1450450274

Dijkstra, E. W. (1969). On the reliability of mechanisms. In *Structured Programming* (pp. 3–6).

Ding, T., Yang, W., Wei, F., Ding, C., Kang, P., & Bu, W. (2022). Chinese keyword extraction model with distributed computing. *Computers and Electrical Engineering*, *97*(November 2021), 107639. https://doi.org/10.1016/j.compeleceng.2021.107639

Duari, S., & Bhatnagar, V. (2019). sCAKE: Semantic Connectivity Aware Keyword Extraction. *Information Sciences*, *477*, 100–117. https://doi.org/10.1016/j.ins.2018.10.034

Duari, S., & Bhatnagar, V. (2020). Complex Network based Supervised Keyword Extractor. *Expert Systems with Applications*, *140*, 112876. https://doi.org/10.1016/j.eswa.2019.112876

El-Beltagy, S. R., & Rafea, A. (2010). KP-miner: Participation in SemEval-2. *ACL 2010 - SemEval 2010 - 5th International Workshop on Semantic Evaluation, Proceedings*, *July*, 190–193.

Feldman, R. (, & Sanger, J. (. (2007). *The Text mining handbook : advanced approaches in analyzing unstructured data / Ronen Feldman, James Sanger.*

Florescu, C., & Caragea, C. (2017). PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*

(Long Papers), *1*, 1105–1115. https://doi.org/10.18653/v1/P17-1102

Godoy Viera, A. F. (2017). Técnicas de aprendizaje de máquina utilizadas para la minería de texto. *Investigacion Bibliotecologica*, *31*(71), 103–126. https://doi.org/10.22201/iibi.0187358xp.2017.71.57812

Gonzáles Tous, M., & Mattar V, S. (2012). Las claves de las palabras clave en los artículos científicos. *Revista MVZ Cordoba*, *17*(2), 2955–2956.

Guan, X., Li, Y., & Gong, H. (2019). Improved TF-IDF for We Media Article Keywords Extraction. *Journal of Physics: Conference Series*, *1302*(3), 032003. https://doi.org/10.1088/1742-6596/1302/3/032003

Hernández, R., Mendez, S., Paulina, C. y Cuevas, A. (2017). *Fundamentos de Investigación* (McGraw-Hil (ed.); Primera). México.

Hotho, A. (KDE G., Nurnberger, A. (Information R. group), & PassB, G. (Knowledge D. G. (1978). Abrief Survey of Text Mining. *Journal of Food Science*, *43*(1), 211–214. https://doi.org/10.1111/j.1365-2621.1978.tb09773.x

Hult, A. (2004). *Combining Machine Learning and Natural Languaje Processing for Automatic Keyword Extraction*. Stockholm University.

International Organization for Standardization. (2013). *ISO 25964 : International Standard for Thesauri*. https://www.niso.org/schemas/iso25964

ISO 8859-1. (1999). Recuperado de https://www.mendeley.com/search/?dgcid=md_homepage&query=ISO 8859-1

ISTQB, I. S. T. Q. B. (2018). *Oficinas Principals*. Recuperado de https://www.istqb.org/downloads/category/2-foundation-level-documents.html

Jayawardene, V., Huggins, T. J., Prasanna, R., & Fakhruddin, B. (2021). The role of data and information quality during disaster response decision-making. *Progress in Disaster Science*, *12*, 100202. https://doi.org/10.1016/j.pdisas.2021.100202

Korde, V., & Mahender, C. N. (2012). Text Classification and Classifiers:A Survey. *International Journal of Artificial Intelligence & Applications*, *3*(2), 85–99. https://doi.org/10.5121/ijaia.2012.3208

Kretschmann, M., Fischer, A., & Elser, B. (2020). Extracting Keywords from Publication Abstracts for an Automated Researcher Recommendation System. *Digitale Welt*, *4*(1), 20–25. https://doi.org/10.1007/s42354-019-0227-2

Kumar, S., Kar, A. K., & Ilavarasan, P. V. (2021). Applications of text mining in services management: A systematic literature review. *International Journal of Information Management Data Insights*, *1*(1), 100008. https://doi.org/10.1016/j.jjimei.2021.100008

Lu, W., Huang, S., Yang, J., Bu, Y., Cheng, Q., & Huang, Y. (2021). Detecting research topic trends by author-defined keyword frequency. *Information Processing & Management*, *58*(4), 102594. https://doi.org/https://doi.org/10.1016/j.ipm.2021.102594

Mack, C. (2012). How to write a good scientific paper: title, abstract, and keywords. *Journal of Micro/Nanolithography, MEMS, and MOEMS*, *11*(2), 020101. https://doi.org/10.1117/1.jmm.11.2.020101

Mahata, D., Kuriakose, J., Shah, R. R., & Zimmermann, R. (2018). *Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings*. 634–639. https://doi.org/10.18653/v1/n18-2100

Martín-Mora, E., Ellis, S., & Page, L. M. (2020). Use of web-based species occurrence information systems by academics and government professionals. *PLoS ONE*, *15*(7 July). https://doi.org/10.1371/journal.pone.0236556

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 - A Meeting of SIGDAT, a Special Interest Group of the ACL Held in Conjunction with ACL 2004*, *85*, 404–411.

Miner, G. D., Elder, J., & Nisbet, R. A. (2012). Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications. In *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. https://doi.org/10.1016/C2010-0-66188-8

Ranguelov, S. (2012). Gestión de la Información y el Conocimiento en las Organizaciones. *Biblios*, *12*(1), 1–7.

Ristoski, P., & Paulheim, H. (2016). Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Journal of Web Semantics*, *36*, 1–22. https://doi.org/10.1016/j.websem.2016.01.001

Santosh Kumar, B., Korra Sathya, B., & Sanjay Kumar, J. (2017). *Automatic Keyword Extraction for Text Summarization: A Survey*. http://arxiv.org/abs/1704.03242

Scikit learn. (2021). *Métricas y puntuación: cuantificación de la calidad de las predicciones*. Modules. https://scikit-learn.org/stable/modules/model_evaluation.html

Škrlj, B., Repar, A., & Pollak, S. (2019). *RaKUn: Rank-based Keyword extraction via Unsupervised learning and Meta vertex aggregation*. 1–12. Recuperado de http://arxiv.org/abs/1907.06458

Sourav, D., Sandip, D., Bhatia, S., & Bhattacharyya, S. (2022). An introduction to data mining in social networks. *Advanced Data Mining Tools and Methods for Social Computing*, 1–25. https://doi.org/10.1016/B978-0-32-385708-6.00008-4

Sterckx, L., Demeester, T., Deleu, J., & Develder, C. (2015). Topical word importance for fast keyphrase extraction. *WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web*, *2*, 121–122. https://doi.org/10.1145/2740908.2742730

Tintinago, A., Muñoz, Y., Uribe, G. A., & Álvarez, P. H. (2018). Assisted labeling of research documents through natural language processing and semantic web technologies. *Scientia et Technica Año XXIII*, *23*(04).

Tolosa, G. H., & Bordignon, F. R. a. (2008). Introducción a la Recuperación de Información Conceptos , modelos y algoritmos básicos. In Universidad Nacional de Luján (Ed.), *Universidad Nacional de Luján, Argentina* (Pre-Edició). Laboratorio de Redes de Datos.

Torres Calvo, M. (2017). *Text Analytics para Procesado Semántico*. 64. http://files/1514/Torres - Text Analytics para Procesado Semántico.pdf

Torres, F. (2016). Plataforma web basada en cloud computing para el seguimiento de proyectos de tesis de pregrado UNA Puno 2016 [Universidad Nacional del

Altiplano]. In *Universidad Nacional del Altiplano*. Recuperado de http://repositorio.unap.edu.pe/handle/UNAP/4848

Van Der Plas, L., Pallotta, V., Rajman, M., & Ghorbel, H. (2004). Automatic keyword extraction from spoken text. A comparison of two lexical resources: The EDR and WordNet. *Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC 2004*, 2205–2208.

Vega-Oliveros, D. A., Gomes, P. S., E. Milios, E., & Berton, L. (2019). A multi-centrality index for graph-based keyword extraction. *Information Processing and Management*, *56*(6), 102063. https://doi.org/10.1016/j.ipm.2019.102063

Villa Monte, A. (2019). *Generación automática inteligente de resúmenes de textos con técnicas de Soft Computing*. 162.

Viloria, A. (2015). *Introducción a las Expresiones Regulares 2 Lenguajes y Expresiones Regulares*.

Wan, X., & Xiao, J. (2008). CollabRank: Towards a collaborative approach to single-document keyphrase extraction. *Coling 2008 - 22nd International Conference on Computational Linguistics, Proceedings of the Conference*, *1*(August), 969–976.

Xu, Z., & Zhang, J. (2021). Extracting Keywords from Texts based on Word Frequency and Association Features. *Procedia Computer Science*, *187*, 77–82. https://doi.org/10.1016/j.procs.2021.04.035

Xuezhong, Z., Yonghong, P., & Baoyan, L. (2010). Text mining for traditional Chinese medical knowledge discovery: A survey. *Journal of Biomedical Informatics*, *43*(4), 650–660. https://doi.org/10.1016/j.jbi.2010.01.002

Zhang, C., Wang, H., Liu, Y., Wu, D., Liao, Y., & Wang, B. (2008). Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, *4*(3), 1169–1180.

ANEXOS

Anexo 1.

Código Fuente Implementación *TF-IDF*

```
#  ****************************************
#  *    Project         : KEYWORD EXTRACTION [Thesis]
#  *    Model           : TF-IDF
#  *    Code Date       : 30-09-2020
#  ******************************************
# Llamado a Librerías
IMPORT time
IMPORT sys
IMPORT os
IMPORT pke
IMPORT numpy as np
from string IMPORT punctuation
from spacy.lang.es.stop_words IMPORT STOP_WORDS
#Lista de documentos a analizar
documentos = os.listdir('/home/Data1/')
# Cargamos el archivo de Frecuencia de Documentos
df = pke.load_document_frequency_file(
        input_file='/home/DocumentFrecuency1.1.tsv.gz')
# PreprocesamienTO  : Definimos las los caracteres de salto
stopwords=list(STOP_WORDS)+list(punctuation)
# Declaramos la Iteración a los documentos
FOR file IN documentos
        # Iniciamos el tiempo de inicio del proceso
        starTime = time.time()
        # Definimos el Método de Extracción
        extractTfidf = pke.unsupervised.TfIdf()
        # Invocamos al archivo de referencia
        extractTfidf.load_document(
                input='/home/Data1/'+file,
                encoding='latin1',
                language='es',
                normalization='none')
        # Definimos el número máximo
        extractTfidf.candidate_selection(n=2,
                stoplist=stopwords)
        # Calculamos los pesos FT-FDI
        extractTfidf.candidate_weighting(df=df)
        # Extraemos los 5 candidatos
        keyWords = extractTfidf.get_n_best(n=5)
        # FIN del tiempo del proceso
        endTime = time.time()
        #Cálculo del tiempo de proceso
        timess=endTime - starTime
        # Rresultado el tiempo de ejecución
        keyWords.append(('time',timess))
        #Creamos el archivo y almacenamos
        np.savetxt('/home/Modelos/M1/'+file, keyWords ,
                delimiter="\n", fmt=" %s")
        # Mensaje de Mapeo
        print("Archivo Generado "+file+"!")
```

Fuente: Elaboración Propia

Anexo 2.

Código Fuente Implementación KP-Miner

```
#  ***************************************
#  *    Project     : KEYWORD EXTRACTION  [Thesis]
#  *    Model       : KP-Miner
#  *    Code Date   : 15-10-2020
#  *****************************************
# Llamado a Librerías
IMPORT time
IMPORT sys
IMPORT os
IMPORT pke
IMPORT numpy as np
#Enlistamos el directorio raiz
documentos = os.listdir('/home/Data1/')
# Cargamos el archivo de Frecuencia de Documentos
df = pke.load_document_frequency_file(
    input_file='/home/DocumentFrecuency1.1.tsv.gz')
FOR file IN documentos:
    # Iniciamos el tiempo de inicio del proceso
    starTime = time.time()
    # Definimos el Método de Extracción
    keyWords = pke.unsupervised.KPMiner()
   # Invocamos al archivo
    keyWords.load_document(
        input='C://Data1/'+file,
        encoding='latin1',
        language='es',
        normalization='none'
    )
    # Seleccionamos los 5 primeros cantidatos
    keyWords.candidate_selection(lasf=3, cutoff=400)
    #Definimos alfa y sigma
    alpha = 3.3
    sigma = 2.3
    # Se calcula los pesos
    keyWords.candidate_weighting(df=df,
        alpha=alpha, sigma=sigma)
    # Obtenemos las palabras clave
    keyWords = keyWords.get_n_best(n=5)
    # FIN del tiempo del proceso
    endTime = time.time()
    timess=endTime - starTime
    # Añadimos  el tiempo de ejecución
    keyWords.append(('time',timess))
    #Creamos el archivo y almacenamos
    np.savetxt('/homeModelos/M2/'+file, keyWords ,
        delimiter="\n", fmt=" %s")
    # Mensaje de Mapeo
    print("Archivo Generado "+file+" !")
```

Fuente: Elaboración Propia

Anexo 3.

Código Fuente Implementación YAKE (Campos et.al. 2020)

```
#  ************************************************
#  *    Project      : KEYWORD EXTRACTION  [Thesis]
#  *    Model        : YAKE  (Campos et al., 2020)
#  *    Code Date    : 01-11-2020
#  **************************************************
# Llamado a Librerías
IMPORT time
IMPORT sys
IMPORT os
IMPORT pke
IMPORT numpy as np
from string IMPORT punctuation
from spacy.lang.es.stop_words IMPORT STOP_WORDS
#Lista de documentos a analizar
documentos = os.listdir('/home/Data1/')
# Preprocesamiento
stopwords=list(STOP_WORDS)+list(punctuation)
FOR file IN documentos:
    # Iniciamos el tiempo de inicio del proceso
    starTime = time.time()
    # Definimos el Método de Extracción
    extractor = pke.unsupervised.YAKE()
    # Invocamos al archivo de referencia
    extractor.load_document(
        input='/home/Data1/'+file,
        encoding='latin1',
        language='es',
        normalization='none' )
    # Definimos el número máximo de palabras
    extractor.candidate_selection(n=2, stoplist=stopwords)
    #Cálculo de pesos de las palabras
    window = 1
    use_stems = True #para el cálculo de pesos
    extractor.candidate_weighting(window=window,
                                  stoplist=stopwords,
                                  use_stems=use_stems)
    # Configuramos los parámetros
    threshold = 0.8
    keyWords = extractor.get_n_best(n=5, threshold=threshold)
    # FIN del tiempo del proceso
    endTime = time.time()
    #Cálculo del tiempo de proceso
    timess=endTime - starTime
    # Añadimos a la cadena de resultado el tiempo de ejecución
    keyWords.append(('time',timess))
    #Creamos el archivo y almacenamos
    np.savetxt('/home/Modelos/M3/'+file,
                     keyWords , delimiter="\n", fmt=" %s")
    # Mensaje de Mapeo
    print("Archivo Generado "+file+"!")
```

Fuente: Elaboración Propia

Anexo 4.

Código Fuente Implementación TextRank (Mihalcea & Tarau, 2004)

```
#  ***************************************************
#  *    Project     : KEYWORD EXTRACTION  [Thesis]
#  *    Model       : TextRank  (Mihalcea & Tarau, 2004)
#  *    Code Date   : 20-12-2020
#  *****************************************************
# Llamado a Librerías
IMPORT time
IMPORT sys
IMPORT os
IMPORT pke
IMPORT numpy as np
#Lista de documentos a analizar
documentos = os.listdir('/home/Data1/')
# Defenimos las partes válidas del texto
pos = {'NOUN'}
# Declaramos la Iteración a los documentos
FOR file IN documentos:
    # Iniciamos el tiempo de inicio del proceso
    starTime = time.time()
    # Definimos el Método de Extracción
    kwTR = pke.unsupervised.TextRank()
    # Invocamos al archivo de referencia
    kwTR.load_document(
        input='/home/Data1/'+file,
        encoding='latin1', language='es',
        normalization='none'
        )
    # Calculamos los pesos y definimos el porcentaje admitido.
    kwTR.candidate_weighting(window=1,
        pos=pos,top_percent=0.33,normalized=False)
    # Obtenemos las Key words.
    keyWords =kwTR.get_n_best(n=5)
    # FIN del tiempo del proceso
    endTime = time.time()
    #Cálculo del tiempo de proceso
    timess=endTime - starTime
    # Añadimos a la cadena de resultado el tiempo de ejecución
    keyWords.append(('time',timess))
    #Creamos el archivo y almacenamos
    np.savetxt('/home/Modelos/M4/'+file, keyWords , delimiter="\n",
        fmt=" %s")
    # Mensaje de Mapeo
    print("Archivo Generado "+file+"!")
```

Fuente: Elaboración Propia

Anexo 5.

Código Fuente Implementación SingleRank (Wan & Xiao, 2008)

```
#  ***************************************************
#  *    Project     : KEYWORD EXTRACTION  [Thesis]
#  *    Model       : SingleRank    (Wan & Xiao, 2008)
#  *    Code Date   : 06-01-2021
#  *****************************************************
# Llamado a Librerías
IMPORT time
IMPORT sys
IMPORT os
IMPORT pke
IMPORT numpy as np
# Defenimos las partes válidas del texto
pos = {'NOUN'}
#Lista de documentos a analizar
documentos = os.listdir('/home/Data1/')
# Declaramos la Iteración a los documentos
FOR file IN documentos:
    # Iniciamos el tiempo de inicio del proceso
    starTime = time.time()
    # Definimos el Método de Extracción
    kwSingler = pke.unsupervised.SingleRank()
    # Invocamos al archivo de referencia
    kwSingler.load_document(
        input='/home/Data1/'+file,
        encoding='latin1',
        language='es',
        normalization='none'
        )
    # Seleccionamos los candidatos
    kwSingler.candidate_selection(pos=pos)
    #Calculamos los pesos
    kwSingler.candidate_weighting(window=10, pos=pos)
    # Obtenemos las Key words.
    keyWords = kwSingler.get_n_best(n=5)
   # FIN del tiempo del proceso
    endTime = time.time()
    #Cálculo del tiempo de proceso
    timess=endTime - starTime
    # Añadimos a la cadena el tiempo
    keyWords.append(('time',timess))
    #Creamos el archivo y almacenamos
    np.savetxt('/home/Modelos/M5/'+file, keyWords ,
         delimiter="\n", fmt=" %s")
    # Mensaje de Mapeo
    print("Archivo Generado "+file+"!")
```

Fuente: Elaboración Propia

Anexo 6.

Código Fuente Implementación TopicRank (BougouIN et al., 2013)

```
#  ****************************************************
#  *    Project     : KEYWORD EXTRACTION  [Thesis]
#  *    Model       : TopicRank (BougouIN et al., 2013)
#  *    Code Date   : 14-02-2021
#  ******************************************************
# Llamado a Librerías
IMPORT os
IMPORT pke
IMPORT sys
IMPORT time
IMPORT numpy as np
from string IMPORT punctuation
from spacy.lang.es.stop_words IMPORT STOP_WORDS #spacy
from nltk.corpus IMPORT stopwords #nltk
pos = {'NOUN', 'PROPN'}
#Lista de documentos a analizar
documentos = os.listdir('/home/Data1/')
# PreprocesamienTO  : Definimos las los caracteres de salto
stoplist = ['-lrb-', '-rrb-', '-lcb-', '-rcb-', '-lsb-', '-rsb-']
stoplist += stopwords.words('spanish')
stopW=list(STOP_WORDS)+list(punctuation)+list(stoplist)
FOR file IN documentos:
    # Iniciamos el tiempo de inicio del proceso
    starTime = time.time()
    # Definimos el Método de Extracción
    kwTR = pke.unsupervised.TopicRank()
    # Invocamos al archivo de referencia
    kwTR.load_document(
        input='/home/Data1/'+file,
        encoding='latin1',
        language='es',
        normalization='none')
    #Elegimos los candidatos
    kwTR.candidate_selection(pos=pos, stoplist=stopW)
    #Calculamos los pesos de las palabras
    kwTR.candidate_weighting(threshold=0.75, method='average')
    # Obtenemos las Key words.
    keyWords = kwTR.get_n_best(n=5)
    # FIN del tiempo del proceso
    endTime = time.time()
    #Cálculo del tiempo de proceso
    timess=endTime - starTime
    # Añadimos a la cadena el tiempo
    keyWords.append(('time',timess))
    #Creamos el archivo y almacenamos
    np.savetxt('/home/Modelos/M6/'+file,
         keyWords , delimiter="\n", fmt=" %s")
    # Mensaje de Mapeo
    print("Archivo Generado "+file+"!")
```

Fuente: Elaboración Propia

Anexo 7.

Código Fuente Implementación To*picalPageRank* (Sterckx et al., 2015)

```
#  ***************************************************
#  *    Project     : KEYWORD EXTRACTION  [Thesis]
#  *    Model       : TopicalPageRank   (Sterckx et al., 2015)
#  *    Code Date   : 17-03-2021
#  *****************************************************
# Llamado a Librerías
IMPORT os
IMPORT pke
IMPORT sys
IMPORT time
IMPORT numpy as np
from string IMPORT punctuation
from spacy.lang.es.stop_words IMPORT STOP_WORDS #spacy
from nltk.corpus IMPORT stopwords #nltk
#Lista de documentos a analizar
documentos = os.listdir('/home/Data1/')
# PreprocesamienTO  : Definimos las los caracteres de salto
stoplist = ['-lrb-', '-rrb-', '-lcb-', '-rcb-', '-lsb-', '-rsb-']
stoplist += stopwords.words('spanish')
stopW=list(STOP_WORDS)+list(punctuation)+list(stoplist)
#Definición de la gramatica admitida
grammar = "NP: {<ADJ>*<NOUN|PROPN>+}"
pos = {'NOUN', 'PROPN', 'ADJ'}
FOR file IN documentos:
    # Iniciamos el tiempo de inicio del proceso
    starTime = time.time()
    # Definimos el Método de Extracción
    kwTPR = pke.unsupervised.TopicalPageRank()
    # 2. load the content of the document.
    kwTPR.load_document(
        input='/home/Data1/'+file,encoding='latin1',
        language='es', normalization='none')
    # Seteamos la Gramatica
    kwTPR.candidate_selection(grammar=grammar)
    #Cargamos el archivo LDA
    kwTPR.candidate_weighting(window=20,pos=pos,
                              lda_model='/home/LDA-Model.gz')
    # Obtenemos las Keywords
    keyWords = kwTPR.get_n_best(n=5)
    # FIN del tiempo del proceso
    endTime = time.time()
    #Cálculo del tiempo de proceso
    timess=endTime - starTime
    # Añadimos a la cadena de resultado el tiempo de ejecución
    keyWords.append(('time',timess))
    #Creamos el archivo y almacenamos
    np.savetxt('/home/Modelos/M7/'+file, keyWords , delimiter="\n",
    # Mensaje de Mapeo                                    fmt=" %s")
    print("Archivo Generado "+file+"!")
```

Fuente: Elaboración Propia

Anexo 8.

Código Fuente Implementación *PositionRank (Florescu & Caragea, 2017)*

```
#  ***********************************************
#  *    Project     : KEYWORD EXTRACTION  [Thesis]
#  *    Model       : PositionRank  (Florescu & Caragea, 2017)
#  *    Code Date   : 25-04-2020
#  *************************************************
# Llamado a Librerías
IMPORT time
IMPORT sys
IMPORT os
IMPORT pke
IMPORT numpy as np
#Lista de documentos a analizar
documentos = os.listdir('Directorio/Data1/')
# Defenimos las partes válidas del texto
pos = {'NOUN', 'PROPN', 'ADJ'}
# define the grammar FOR selecting the keyphrase candidates
grammar = "NP: {<ADJ>*<NOUN|PROPN>+}"
# Declaramos la Iteración a los documentos
FOR file IN documentos:
    # Iniciamos el tiempo de inicio del proceso
    starTime = time.time()
    # Definimos el Método de Extracción
    kwPR = pke.unsupervised.PositionRank()
    # Cargamos el documenTO  iterado
    kwPR.load_document(
        input='Directorio/Data1/'+file,
        encoding='latin1',
        language='es',
        normalization='none'
        )
    kwPR.candidate_selection(grammar=grammar,
                             maximum_word_number=3)
    # Calculamos los pesos
    kwPR.candidate_weighting(window=10, pos=pos)
    # Obtenemos las palabras clave
    keyWords = kwPR.get_n_best(n=5)
    # FIN del tiempo del proceso
    endTime = time.time()
    #Cálculo del tiempo de proceso
    timess=endTime - starTime
    # Añadimos a la cadena de resultado el tiempo de ejecución
    keyWords.append(('time',timess))
    #Creamos el archivo y almacenamos
    np.savetxt('/d/models/M8/'+file,
      keyWords , delimiter="\n", fmt=" %s")
    # Mensaje de Mapeo
    print("Archivo Generado "+file+"!")
```

Fuente: Elaboración Propia

Anexo 9.

Código Fuente Implementación *MultipartRank (Boudin, 2018)*

```
#  *********************************************
#  *    Project     : KEYWORD EXTRACTION  [Thesis]
#  *    Model       : MultipartRank (Boudin, 2018)
#  *    Code Date   : 25-04-2020
#  *********************************************
IMPORT time
IMPORT sys
IMPORT os
IMPORT pke
IMPORT string
IMPORT numpy as np
from nltk.corpus IMPORT stopwords
from string IMPORT punctuation
from spacy.lang.es.stop_words IMPORT STOP_WORDS
#Lista de documentos a analizar
documentos = os.listdir('/home/Data1/')
# Definimos las reglas gramaticales
pos = {'NOUN', 'PROPN', 'ADJ'}
stoplist = list(string.punctuation)
stoplist += ['-lrb-', '-rrb-', '-lcb-', '-rcb-', '-lsb-', '-rsb-']
stoplist += stopwords.words('spanish')
stopwords=list(STOP_WORDS)+list(punctuation)+list(stoplist)
# Declaramos la Iteración a los documentos
FOR file IN documentos:
    # Iniciamos el tiempo de inicio del proceso
    starTime = time.time()
    # Definimos el Método de Extracción
    kwMtp = pke.unsupervised.MultipartiteRank()
    # Invocamos al archivo de referencia
    kwMtp.load_document(
        input='/home/Data1/'+file,
        encoding='latin1',
        language='es',
        normalization='none')
    # Definimos los candidatos y definimos las stopwords
    kwMtp.candidate_selection(pos=pos, stoplist=stopwords)
    kwMtp.candidate_weighting(alpha=1.1,threshold=0.74,
                              method='average')
    # 5. get the 10-highest scored candidates as keyphrases
    keyWords = kwMtp.get_n_best(n=10)
    # FIN del tiempo del proceso
    endTime = time.time()
    #Cálculo del tiempo de proceso
    timess=endTime - starTime
    # Añadimos a la cadena el tiempo
    keyWords.append(('time',timess))
    #Creamos el archivo y almacenamos
    np.savetxt('/home/Modelos/M9/'+file, keyWords ,
        delimiter="\n", fmt=" %s")
    print("Archivo Generado "+file+"!")
```

Fuente: Elaboración Propia