Modelo predictivo para la gestión documental de los correos electrónicos de radicación para la Agencia Nacional de Defensa jurídica del Estado

Jorge Mario Carrasco Ortiz

Asesor

Rafael Gaitan Ospina

Universidad Nacional Abierta y a Distancia – UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería – ECBTI

Especialización en Ciencia de Datos y Analítica

Junio de 2024

Dedicatoria

A mis hijos Mateo y Catalina, quienes son el motor de mi vida; a mi esposa Katherine, quien me ha enseñado el verdadero significado del amor y cómo vivir plenamente; a mis padres y hermanos, quienes me dieron la vida, me vieron nacer y crecer, y contribuyeron a forjar la persona que soy hoy en día.

Agradecimientos

Especial reconocimiento a la Agencia Nacional de Defensa Jurídica del Estado (ANDJE) por el apoyo técnico y económico brindado durante esta etapa del proyecto de automatización de gestión documental de correos electrónicos de radicación, los insumos de datos y de información brindados para este proyecto han sido fundamentales para avanzar en el desarrollo y conclusión de este proyecto aplicado. Adicionalmente, agradecimiento al equipo docente de la Especialización en Ciencia de Datos y Analítica de la UNAD, en especial al profesor Rafael Gaitan Ospina por su guía.

Resumen

El proyecto se centra en desarrollar un modelo de aprendizaje computacional para automatizar la gestión documental de correos electrónicos en la Agencia Nacional de Defensa Jurídica del Estado (ANDJE). Se basa en un enfoque experimental dividido en las fases de preprocesamiento del texto, representación de texto y evaluación de métodos de clasificación. Utiliza técnicas avanzadas de procesamiento de lenguaje natural, como BoW, Word Embedding y Large Language Models, para transformar y comprender el contenido de los correos. El diseño se fundamenta en modelos de aprendizaje automático y algoritmos de clasificación supervisada. Esta metodología permitirá construir un modelo que optimice la categorización automática de correos, mejorando la eficiencia y precisión en la gestión documental de la Agencia.

Palabras clave: Word Embedding, LLM's, Clasificación de textos

Abstract

The project focuses on developing a computational learning model to automate email document management at the Agencia Nacional de Defensa Jurídica del Estado (ANDJE). It is based on an experimental approach divided into phases of text preprocessing, text representation, and evaluation of classification methods. It uses advanced natural language processing techniques such as Bag of Words (BoW), Word Embedding, and Large Language Models (LLMs) to transform and understand the email content. The design is based on machine learning models and supervised classification algorithms. This methodology will allow building a model that optimizes automatic email categorization, improving efficiency and accuracy in the Agency's document management.

Keywords: Word Embedding, LLMs, Text Classification

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Lista de Anexos

Glosario

Ciencia de Datos: Es un área de aplicación que combina múltiples disciplinas, en las que generalmente se incluyen la estadística y la ciencia de la computación. El insumo principal lo constituyen grandes volúmenes de información que contiene datos estructurados y no estructurados, el fin es extraer patrones, ofrecer visualizaciones y construir soluciones que automatizan procesos de tal manera que se agregue valor a la organización.

SGDEA: Sistema de Gestión de Documentos Electrónicos de Archivo

ORFEO: es el actual SGDEA en el que se almacenan los documentos propios de los procesos para la Entidad.

Recuperación de Información: Es la tarea en la que el computador presenta al usuario las unidades de información, generalmente documentos de texto, imágenes o vídeos, más relevantes para su búsqueda dentro de una colección de información que se requiera.

Grandes Modelos de Lenguaje (LLM): Los LLM por sus siglas en inglés (Large Language Models), son sistemas de Inteligencia Artificial avanzada que se centra en comprender y generar texto de manera similar a lo que haría un humano. Estos modelos utilizan técnicas de aprendizaje automático, procesamiento del lenguaje natural y grandes conjuntos de datos para ser entrenados.

Azure OpenAI: Plataforma de servicios en la nube que ofrece herramientas de inteligencia artificial desarrolladas por OpenAI en la nube de Azure.

Versión del modelo: Número que identifica una versión específica de un modelo de inteligencia artificial que se va a implementar.

MIRACL: (Multi-language Information Retrieval Accuracy at the CLAR division of ACM SIGIR) es un conjunto de pruebas comúnmente utilizado para evaluar el rendimiento de modelos de recuperación de información en varios idiomas. Este conjunto de pruebas proporciona una

medida del rendimiento de los modelos de *embedding* en tareas de recuperación de información en diferentes idiomas.

MTEB: (Multi-Task English Benchmark) es un conjunto de pruebas comúnmente utilizado para evaluar el rendimiento de modelos de *embedding* en tareas específicas del idioma inglés. Este conjunto de pruebas proporciona una medida del rendimiento del modelo en tareas como comprensión de texto, generación de texto y otras tareas del procesamiento del lenguaje natural en inglés

Radicado SGDEA: es un identificador único dentro del sistema ORFEO. Este número sirve para distinguir las comunicaciones oficiales internas que la Agencia emite o recibe. En los metadatos del radicado suele incluir los siguientes campos: asunto de la radicación, la cual describe brevemente el tema o motivo de la comunicación; remitente, indica la persona o entidad que envía la comunicación; radicador, identifica a la persona responsable de la radicación del documento en el SGDEA y finalmente fecha de radicación, registra la fecha en que se ingresó la comunicación en el sistema.

Introducción

La gestión documental eficiente es fundamental para el correcto funcionamiento de las entidades gubernamentales, especialmente en aquellas que manejan un alto volumen de información, como la Agencia Nacional de Defensa Jurídica del Estado Colombiano (ANDJE). Sin embargo, los métodos tradicionales de gestión documental, basados en procesos manuales, suelen ser lentos, propensos a errores y representar un obstáculo para la productividad y la toma de decisiones oportunas.

En respuesta a estos desafíos, el presente proyecto se propone desarrollar un modelo predictivo para la automatización de la gestión documental de correspondencia en la ANDJE. Este proyecto se fundamenta en la aplicación de tecnologías de vanguardia como el procesamiento del lenguaje natural (PLN) y el aprendizaje automático (ML) para transformar los datos de texto en representaciones numéricas que puedan ser procesadas por algoritmos de clasificación y categorización.

El proyecto se estructura en dos fases fundamentales: la primera fase se enfoca en la representación de textos mediante técnicas avanzadas como Word Embedding y Large Language Models (LLM), con el objetivo de transformar los datos de texto en representaciones numéricas para su posterior procesamiento por algoritmos de aprendizaje automático. Esta etapa incluye actividades como el preprocesamiento de datos, la aplicación de Word Embedding para capturar la semántica y contexto de las palabras, y la exploración de LLM como GPT-2 para comprender y generar texto coherente. Por otro lado, la segunda fase se centra en la comparación de métodos de clasificación y categorización de los correos electrónicos, con el fin de evaluar el rendimiento de diferentes enfoques en la tarea de gestión documental.

Este proyecto representa un avance significativo en la optimización de procesos internos de la ANDJE, contribuyendo a una gestión documental más eficiente, precisa y transparente. Al implementar un modelo predictivo basado en tecnologías de vanguardia como el procesamiento del lenguaje natural y el aprendizaje automático, se espera reducir drásticamente el tiempo dedicado a tareas manuales de clasificación y categorización de correspondencia. Esto no solo aumentará la eficiencia operativa, sino que también minimizará los errores inherentes a procesos manuales, garantizando la integridad y la precisión de la información gestionada.

Planteamiento del problema

El proyecto se enfoca en abordar la problemática de la gestión documental de los correos electrónicos de radicación para la Agencia Nacional de Defensa Jurídica del Estado (ANDJE). Esta entidad se enfrenta a desafíos significativos debido al manejo manual de dicha gestión, lo que ha desencadenado diversas problemáticas identificadas.

La problemática central identificada es la gestión manual de los correos electrónicos de radicación para la Agencia Nacional de Defensa Jurídica del Estado (ANDJE). Esta gestión manual ha generado una serie de efectos adversos que amenazan el proceso mismo de clasificación y manejo de correspondencia electrónica. Entre los efectos identificados se puedan destacar:

- **Contratación Excesiva de Personal:** La necesidad de contratar múltiples individuos para clasificar los correos electrónicos, evidencia una falta de sistema automatizado eficiente. Esta medida paliativa conlleva un gasto adicional de recursos financieros y humanos que podría ser optimizado mediante soluciones tecnológicas especializadas.
- **Riesgo de Pérdida de Información Crítica:** La gestión manual incrementa el riesgo inherente de pérdida de información importante debido a la falta de seguimiento adecuado, extravío o eliminación accidental de correos electrónicos relevantes. Esta situación compromete la integridad y seguridad de los datos fundamentales para la entidad.
- **Subjetividad en la Clasificación:** La falta de un sistema automatizado ha resultado en una clasificación subjetiva y etiquetado no estandarizado de los

correos electrónicos. Esta subjetividad podría llevar a imprecisiones y errores en la asignación de los documentos, afectando la integridad y coherencia en el manejo documental.

- **Consumo Significativo de Recursos:** El proceso manual demanda un consumo considerable de tiempo y recursos humanos, lo que impacta negativamente en la eficiencia operativa. Esta ineficiencia se traduce en tiempos de respuesta prolongados y una gestión menos ágil de los correos electrónicos de radicación.

Las causas identificadas se refieren a los factores fundamentales que contribuyen o generan el problema central o los efectos adversos dentro de un sistema o situación particular. En el contexto del proyecto de automatización de gestión documental de correos electrónicos de radicación para la Agencia Nacional de Defensa Jurídica del Estado (ANDJE), las causas identificadas son:

- Carencia de Capacitación y Recursos: La falta de capacitación y recursos adecuados para el personal encargado de la gestión documental conlleva a una ejecución subóptima del proceso. La ausencia de conocimientos especializados impide un manejo eficiente de los correos electrónicos recibidos.
- Falta de Herramientas Automatizadas: La carencia de sistemas o herramientas automatizadas específicas para la gestión de correos electrónicos de radicación representa un vacío tecnológico crítico. Esta ausencia limita la optimización del proceso, imponiendo una carga adicional de trabajo manual.
- Subutilización de Sistemas y Herramientas: La falta de utilización efectiva de sistemas de gestión documental diseñados para la ANDJE o de los buzones electrónicos disponibles

incide en la ineficiencia del proceso. La no utilización adecuada de estas herramientas agrava la complejidad de la gestión documental.

Partiendo de reconocer la particularidad de este conjunto de información, este proyecto aplicado de desarrollo tecnológico busca responder a las siguientes preguntas: ¿Cuáles son las tecnologías y técnicas de procesamiento de lenguaje natural más adecuadas para la identificación y clasificación de contenido relevante en los correos electrónicos de la ANDJE? ¿Cómo se puede entrenar y desplegar un modelo de aprendizaje automático que mejore continuamente la precisión en la clasificación y gestión de correos electrónicos, considerando la evolución de los patrones de comunicación y documentación de la agencia?

Justificación

Este proyecto de apoyo en la gestión documental para la Agencia Nacional de Defensa Jurídica del Estado (ANDJE) se justifica en virtud de una serie de desafíos identificados que afectan directamente la eficiencia, seguridad y operatividad de la entidad en su manejo de correos electrónicos de radicación. La problemática central revela un escenario donde la gestión manual actual ha generado ineficiencias evidentes. La necesidad de contratar múltiples personas para la clasificación de correos, el riesgo de pérdida de información crítica, la subjetividad en la clasificación y el uso ineficiente de recursos, son aspectos que resaltan la urgente necesidad de una solución automatizada.

La falta de capacitación especializada y recursos adecuados para el personal encargado de la gestión documental ha resultado en una ejecución subóptima del proceso, dada la ausencia de conocimientos específicos. La implementación de sistemas basados en IA y Machine Learning ofrece una solución prometedora a este desafío, dado que estos sistemas pueden aprender automáticamente patrones complejos en los datos de los correos, permitiendo una clasificación más precisa y ágil sin depender únicamente de la experiencia humana. Al aplicar algoritmos de aprendizaje automático, el sistema puede mejorar continuamente su capacidad de clasificación, reduciendo los errores, el tiempo del procedimiento y optimizando la gestión documental de manera eficiente y efectiva.

La carencia de sistemas o herramientas automatizadas diseñadas específicamente para gestionar correos electrónicos de radicación plantea una excesiva dependencia en procesos manuales. Esta falta de automatización no solo aumenta significativamente la probabilidad de errores y pérdida de información, sino que también restringe la capacidad de la entidad para manejar eficazmente el considerable volumen de correspondencia. Esta situación evidencia la

urgente necesidad de implementar un modelo de inteligencia artificial que permita la clasificación automática de la correspondencia, lo cual justifica la importancia y relevancia de llevar a cabo este proyecto aplicado. El modelo que se plantea en el desarrollo de este proyecto es crucial para asegurar la capacidad de la entidad para gestionar de manera efectiva y precisa el flujo constante de información

El desarrollo de un modelo de aprendizaje computacional a través de este proyecto permitirá abordar estas causas fundamentales y atender los efectos adversos observados. Al desarrollar un modelo que aplique tecnologías como el procesamiento de lenguaje natural y el aprendizaje automático, se busca no solo mitigar las ineficiencias actuales, sino también mejorar la eficacia y precisión en la gestión documental de la entidad, brindando una solución integral a la problemática identificada.

Una vez esté terminado este proyecto de aprendizaje computacional la Agencia podrá iniciar un proyecto de desarrollo donde se haga integración de este modelo con la parte productiva de la operación del sistema de gestión documental y de los procedimientos de la Agencia.

Objetivos

Objetivo general

Desarrollar un modelo de aprendizaje computacional que permita la clasificación eficiente y precisa de la gestión documental de las comunicaciones que son recibidas por notificación vía correos electrónicos en la ANDJE.

Objetivos específicos

- Implementar técnicas de procesamiento de lenguaje natural para el análisis del contenido de los radicados que se reciben por correo de la ANDJE y que posteriormente permitan la creación de algoritmos de clasificación personalizados.
- Aplicar un método de clasificación supervisado que permite la clasificación de correspondencia del correo electrónico en las diferentes categorías que requiere la ANDJE.

Marco conceptual

Este proyecto está dentro del ámbito de la gestión documental en la Agencia, área que se enfoca en actividades como la creación, captura, clasificación, almacenamiento y conservación de las diversas piezas documentales recibidas, en este caso, por esta vía electrónica. La gestión documental tiene como objetivo primordial optimizar la información manejada por la entidad, garantizando su disponibilidad, integridad y seguridad a través de políticas y procedimientos aplicados a estas actividades.

El propósito principal de esta labor es mejorar el proceso de radicación y distribución de las comunicaciones según sea la temática de la comunicación recibida, por ejemplo, en el caso de temas relacionados con la facturación de un proveedor de la agencia, se busca que estos documentos sean radicados de manera inmediata en el SGDEA (Es el acrónimo de Sistema de Gestión de Documentos Electrónicos de Archivo) de la Entidad y asignados a la bandeja de entrada del área correspondiente, en este caso el área financiera.

Dentro de las metodologías esenciales que se podrían aplicar para la automatización de la tarea radicación y reparto se encuentra la clasificación supervisada de datos, para utilizar los algoritmos de este enfoque se debe tener un conjunto de datos etiquetados previamente, es decir datos con respuesta previas que ya se conocen. El objetivo es hacer predicción sobre conjuntos de datos futuros o conjunto de datos en donde no se conoce el resultado evaluado (Raschka, 2015)

Un ejemplo de aplicación de este tipo de técnicas en el ámbito jurídico es la construcción de un modelo para predecir el sentido de la decisión de un caso (Desfavorable o Favorable), con base en una colección de casos con características similares, mediante modelos de razonamiento

basados en casos (CBR por sus siglas en inglés) o algoritmos de Machine Learning (Ashley, 2017) y en ocasiones, mediante la combinación de ambas estrategias.

Un sistema inteligente de categorización de documentos se perfila como una plataforma o conjunto de herramientas que, a través de técnicas de inteligencia artificial y procesamiento del lenguaje natural, automatiza la organización y clasificación de documentos o textos en categorías predefinidas. Dentro de la literatura se encuentran diferentes ámbitos donde estos sistemas son aplicados: En la categorización de artículos científico (Bayer et al., 1998), en casos más específicos para la clasificación en diferentes categorías de artículos biomédicos (Humphrey et al., 2009), detección de noticias falsas (fake news termino en inglés) (Ahmed et al., 2021), y finalmente otras investigaciones que tratan de mejorar los sistemas de clasificación de documentos combinando múltiples clasificadores (J. Lee et al., 2021; Zelaia et al., 2011).

Los conceptos presentados son esenciales en el entendimiento del problema del negocio y serán más ampliamente abordados en el marco teórico que limite este proyecto.

Marco teórico

El problema de clasificación supervisado de texto se puede entender como el conjunto de algoritmos utilizados para asignar una categoría a un texto, la particularidad de estas categorías es que ya fueron previamente definas. Esta tarea se ha convertido en un parte esencial de la investigación dentro del procesamiento de lenguaje natural (Hassan et al., 2022). El sistema que se va a implementar en este proyecto emplea este tipo de algoritmos y modelos de aprendizaje automático para analizar el contenido de los mensajes de correos electrónicos, identificar patrones y características clave, y asignarlos a categorías específicas sin intervención humana.

Dentro de un sistema de categorización de documentos se pueden distinguir tres grandes etapas o módulos de la arquitectura para completar la tarea de clasificación de documentos: la etapa de preprocesamiento, la etapa representación de los textos y la etapa de clasificación (Fatima et al., 2017). En la etapa de preprocesamiento, es la más común en el campo de procesamiento del lenguaje natural (NLP por sus siglas en ingles), son una seria de pasos para limpiar, normalizar y estructurar los textos antes de aplicar técnicas de análisis o modelado (Ahmed et al., 2021; Fatima et al., 2017).

Como segunda etapa del proceso, la representación de textos es crucial para la tarea de clasificación, esta representación implica convertir palabras y documentos en representaciones numéricas (es decir transformar el texto no estructurado en datos estructurados) para que los modelos de aprendizaje automático puedan procesarlos eficazmente (Cahyani & Patasik, 2021; Fatima et al., 2017). Entre las técnicas más destacadas se encuentra el uso de representaciones de bolsas de palabras (BoW por sus siglas en inglés), un enfoque común que cuenta con diversas variantes, entre las más usadas se encuentra el enfoque de "Frecuencia del Término - Frecuencia Inversa de los Documentos" (TF-IDF por sus siglas en inglés) (Baharudin et al., 2010; Cahyani

& Patasik, 2021; Fatima et al., 2017).

Otro enfoque utilizado es el de representación por medio de *Word embeddings*, como en los modelos Neural Network Language Model, word2vec, GloVe mode, FastText (S. Wang et al., 2020), tiene como objetivo capturar tanto el significado semántico como sintáctico de las palabras en un texto, estas clases de método asigna cada palabra o documento a un vector continuo, utilizando modelos como el skip-gram, reconocido mejora la comprensión y representación de los textos para tareas de clasificación en NLP (Singh et al., 2022; B. Wang et al., 2019) En estos últimos años, debido al crecimiento que ha tenido la información textual y gracias también en gran medida al crecimiento de los desarrollos de IA bidirectional LSTM network, BERT y OpenAI GPT (B. Wang et al., 2019; S. Wang et al., 2020), muchos investigadores han centrado esfuerzos en las técnicas de representación de los documentos basados en el enfoque conocido como Large Language Models (LLM por sus siglas en ingles), una gran ventaja de este tipo de modelos es que usando *transfer learning*, se pueden aprovechar modelos previamente entrenados en grandes conjuntos de datos para tareas específicas de clasificación para los cuales no han sido concebidos originalmente, lo que puede mejorar el rendimiento incluso con otros conjuntos de datos como es el caso de los documentos que se están estudiando.

Las representaciones de texto anteriormente descritas pueden ser de gran dimensión, dependiendo de la técnica de representación utilizada (Singh et al., 2022). La selección de características en clasificación de texto en procesamiento del lenguaje natural (NLP) implica identificar y elegir las variables más relevantes del texto para entrenar modelos de manera eficiente y precisa. Esto implica, en algunos casos, la reducción de la dimensionalidad del conjunto de datos al seleccionar las características más informativas (Singh et al., 2022), lo que

mejora la eficiencia computacional, evita el sobreajuste y aumenta la comprensión sobre qué aspectos del texto son más influyentes en la clasificación. Otro tipo de métodos de selección de características son los métodos que, por medio de enfoques estadísticos, como la frecuencia de términos, hasta métodos basados en modelos o índices como el índice Gini (Shang et al., 2007), buscan resaltar aquellas características que tienen un impacto significativo en la capacidad predictiva del modelo de clasificación.

Como etapa final de un sistema de categorización de documentos, se encuentra la fase de clasificación, como ya se ha mencionado con anterioridad, consiste en la identificación de si un fragmento en particular se encuadra en alguna de las categorías predefinidas (Antonie & Zaïane, 2002; Bayer et al., 1998). Esta labor es esencial en el procesamiento textual, ya que facilita la organización de documentos para su posterior búsqueda. Además, se destaca como una fase fundamental en los sistemas de procesamiento del lenguaje natural, aplicada en el análisis de contenidos (Fatima et al., 2017).

Mas formalmente hablando, esta categorización de documentos se conoce como clasificación supervisada, en este paradigma de aprendizaje automático un algoritmo aprende a asignar instancias de datos a un conjunto predefinido de clases o categorías, aprendiendo reglas de un conjunto de entrenamiento de ejemplos etiquetados (Kotsiantis et al., 2007). Estos ejemplos, denominados datos de entrenamiento, contienen tanto las características de los datos (atributos) como la clase a la que pertenecen.

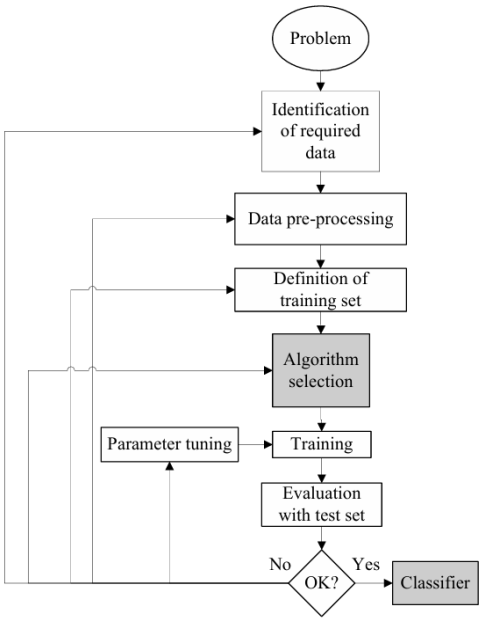
La Figura 1 muestra el proceso general seguido por los distintos algoritmos supervisados. Idealmente, este proceso comienza con las recomendaciones de un experto sobre las características o variables más informativas, basándose en el contexto y su experiencia en el tema (Kotsiantis et al., 2007). Posteriormente, la etapa de preprocesamiento de datos es crucial

para garantizar la calidad del modelo. En esta fase, se emplean técnicas para tratar valores faltantes y detectar anomalías. En el caso de los textos, se eliminan factores dependientes del idioma, como conjugaciones y palabras vacías (Baharudin et al., 2010). Además, se seleccionan subconjuntos representativos del conjunto original cuando se trabaja con grandes volúmenes de datos (Jindal et al., 2015).

Por último, se procede al entrenamiento del modelo de aprendizaje automático, seleccionando el algoritmo adecuado y ajustando sus parámetros para optimizar su desempeño en un conjunto. La evaluación del modelo con un conjunto de datos de prueba independiente es fundamental para validar su generalización y confiabilidad.

Figura 1

El proceso de obtención para el problema de aprendizaje de maquina supervisado.



Nota. Adaptada de Supervised machine learning: A review of classification techniques. (p162) por Kotsiantis, et al., 2007, Emerging artificial intelligence applications in computer engineering.

En el estado del arte de la clasificación de documentos y correos electrónicos, se destacan

varias técnicas avanzadas. Entre ellas se encuentra el uso de modelos de aprendizaje profundo, como las redes neuronales convolucionales (CNN) (Lan et al., 2008; Zelaia et al., 2011) y las redes neuronales recurrentes (RNN) (Raschka, 2015; R. Wang et al., 2019), máquina de vectores de soporte (SVM) (Fatima et al., 2017; T.-Y. Wang & Chiang, 2007), k-vecino más cercano (KNN) (Humphrey et al., 2009; Y.-H. Lee et al., 2021; R. Wang et al., 2019), clasificador de Bayes multinomial (MNB), regresión logística (LR) y bosques aleatorios (RF). En el desarrollo de aplicaciones de procesamiento del lenguaje natural, estos modelos permiten aprender representaciones complejas de los textos y capturar relaciones contextuales, lo que mejora la capacidad de clasificar documentos basados en su contenido.

En relación con el proyecto aplicado para la representación y clasificación de la correspondencia de la ANDJE, la Tabla 1 presenta los distintos algoritmos de clasificación que se probaron. Los detalles del experimento se pueden consultar en la sección "*Método de clasificación supervisado para la categorización de correspondencia ANDJE".* Para una mayor comprensión, se recomienda al lector consultar cada una de las referencias, ya que los modelos presentan diferencias significativas en su formulación.

Tabla 1

Relación de los métodos de clasificación supervisada usados en la experimentación

Etiqueta Clasificador	**Descripción de método de clasificación.**
AdaBoost Classifier	AdaBoost Combina múltiples clasificadores débiles para crear un clasificador final robusto. Cada clasificador débil se entrena en un subconjunto de datos ponderado y su error se utiliza para ajustar los pesos en la siguiente iteración.
HistGradientBoosting	Histogram-based Gradient Boosting Classifier combina árboles de decisión utilizando histogramas para mejorar el rendimiento del modelo, identificando regiones relevantes del espacio de características.

XGBoost Classifier	XGBoost el cual resulta ser una implementación eficiente y escalable de Gradient Boosting que utiliza árboles de decisión como clasificadores débiles. Optimiza el entrenamiento con técnicas como regularización y aprendizaje temprano para evitar sobreajuste y mejorar la generalización (Chen & Guestrin, 2016).
Linear SVM	SVM Clasifica datos utilizando un hiperplano que separa las clases en el espacio de características. Maximiza el margen entre las clases para aumentar la separación. (Baharudin et al., 2010).
SVM	SVM (Support Vector Machine) clasifica datos encontrando el hiperplano que mejor separa las clases en el espacio de características (T.-Y. Wang & Chiang, 2007).
Logistic Regresion	Modelo de regresión de la familia de modelos generalizados utilizado para problemas de clasificación binaria y multinomial (Ramirez-Loaiza et al., 2013).
Random Forest	Construye múltiples árboles de decisión y los combina para mejorar la precisión y evitar el sobreajuste. (Jindal et al., 2015)

Dado el enfoque experimental del proyecto presentado, se busca evaluar varios métodos en las tres etapas previamente mencionadas (preprocesamiento, representación y clasificación). El objetivo es identificar la secuencia de procesamiento óptima para el caso específico de la construcción de un modelo de aprendizaje computacional en la gestión documental de los correos electrónicos de radicación, especialmente orientado al caso de la Agencia Nacional de Defensa Jurídica del Estado (ANDJE).

Metodología

Este proyecto aplicado es un estudio exploratorio que busca probar diversas estrategias para categorizar las entradas de correo electrónico dirigidas a la Agencia Nacional de Defensa Jurídica. Estas estrategias incluyen diferentes representaciones de los textos y métodos de clasificación. En el diseño de la investigación se emplean datos cuantitativos derivados de los registros documentales y correos electrónicos que han sido previamente clasificados manualmente por el grupo de correspondencia. Además, este trabajo tiene un enfoque del tipo experimental, ya que busca recopilar pruebas sobre la validez de los métodos del estado del arte y la propuesta metodológica creada para abordar el problema planteado.

Para lograr estos objetivos, se han establecido dos grandes fases. A continuación, se presenta un breve resumen de cada fase del proyecto junto con las diferentes activades que se deben realizar:

Primera Fase (Representación de textos con Words Embedding y LLM)

Esta fase inicial, se enfocará en representar los textos de los correos electrónicos utilizando técnicas avanzadas como Word Embedding y Large Language Models (LLM por sus siglas en ingles). Esto implica transformar los datos de texto en representaciones numéricas para su posterior procesamiento por parte de algoritmos de aprendizaje automático. El uso de técnicas como Embedding permite capturar la semántica y contexto de las palabras, mientras que los LLM, como GPT-3, pueden comprender y generar texto coherente.

Actividades:

- **Preprocesamiento de Datos:** Limpieza y preparación de los correos electrónicos para eliminar ruido, normalizar texto y asegurar la coherencia de los datos.

- **Aplicación de Word Embedding:** Utilización de técnicas como Word2Vec, GloVe o FastText para convertir el texto en vectores numéricos, capturando significados semánticos y relaciones entre palabras.
- **Uso de Large Language Models (LLM):** Exploración y experimentación con modelos avanzados de lenguaje como GPT-3 para entender su capacidad de comprensión de texto y generación de representaciones contextualizadas.

Segunda Fase (Comparación de métodos de Clasificación/Categorización de entradas del correo)

Se compararán diferentes métodos de clasificación o categorización de entradas de correos electrónicos. Se analizarán y evaluarán algoritmos de aprendizaje automático para determinar cuál es más efectivo en la clasificación precisa y eficiente de los correos. Estos métodos pueden incluir desde algoritmos tradicionales hasta técnicas más modernas como redes neuronales o modelos de aprendizaje profundo.

Actividades:

- **Selección de Algoritmos:** Investigación y selección de algoritmos de clasificación adecuados para la tarea, considerando su eficiencia y precisión y su uso en tareas similares en diferentes investigaciones publicadas.
- **Entrenamiento y Evaluación:** Implementación de los algoritmos seleccionados utilizando los datos preparados en la primera fase. Se entrenarán y evaluarán estos modelos utilizando métricas relevantes como para la medición del desempeño de los algoritmos seleccionados.

- **Comparación y Selección:** Se compararán los resultados de los diferentes de clasificación para determinar cuál proporciona los mejores resultados en la categorización de los correos electrónicos.

Implementación de técnicas de procesamiento de lenguaje natural.

En este capítulo, se presenta la implementación de técnicas avanzadas de procesamiento de lenguaje natural (NLP, por sus siglas en inglés) para el análisis del contenido de la correspondencia recibidos por la Agencia Nacional de Defensa Jurídica del Estado (ANDJE). Uno de los objetivos de este proyecto aplicado es poder el preprocesamiento de texto, la representación de texto y la aplicación de modelos de aprendizaje automático para la clasificación de los textos. Además, se detalla el uso de técnicas como Word Embedding y Large Language Models (LLMs) para entender y categorizar el contenido de los correos de manera automatizada.

Este capítulo proporciona una visión detallada del proceso de implementación de estas técnicas en el conjunto de datos en que se experimentó, este apartado se divide en dos secciones uno que aborda los pasos que se siguieron en la depuración del conjunto de datos y posteriormente la descripción de metodología de la representación distribucional de los términos y la descripción de los diferentes *embeddings* utilizados.

Descripción de la limpieza y depuración del conjunto de datos

La ANDJE, en colaboración con su área de gestión documental y haciendo la consulta al Sistema de Gestión de Documentos Electrónicos de Archivo Orfeo, ha preparado una muestra de 19.499 radicados aleatorios para su análisis en este proyecto. Esta muestra, tras un proceso de depuración detallado en este documento, servirá como base para el estudio de la información contenida en los documentos de la entidad y la construcción de un clasificador de correspondencia.

En la Figura 2 se presenta la primera exploración del conjunto de datos original, en esta se realizó un análisis descriptivo de las variables incluidas, revelando varios detalles. En la columna

de Radicados, se encontraron un total de 19,999 registros. Respecto a la variable Asunto, se identificaron 19,999 registros únicos, siendo "C.E. ANGEL SAMUEL SEDA AND OTHERS V REPUBLIC O…." el valor más frecuente, con una frecuencia de 30 veces. La mayoría de los asuntos parecen ser únicos. En cuanto a Remitente, se registraron 19,997 entradas no nulas, siendo " REMITENTE ANÓNIMO 1" el remitente más frecuente, con 1,011 ocurrencias. Por otro lado, en la variable Radicador se identificaron 19,999 radiadores únicos, siendo "RADICADOR ANÓNIMO 1" el más frecuente, con 3,832 ocurrencias. En cuanto a la Fecha de Radicación, se observó que va desde el 1 de enero de 2022 hasta el 29 de diciembre de 2022, concentrándose la mayoría de las entradas entre el 31 de enero y el 9 de mayo de 2022. Finalmente, en la variable Tipo Documento se encontraron 79 tipos diferentes, siendo "No definido" el más común, con 10,660 ocurrencias, mientras que en Medio Recepción se identificaron 4 tipos diferentes, siendo "Mail (e-Mail)" el más frecuente, con 18,942 ocurrencias.

Figura 2

Descripción inicial del conjunto de datos de la iniciativa de clasificación de correspondencia

	Asunto	Remitente	Radicador	Fecha Radicacion	Tipo Documento	Medio Recepcion
count	19999	19997	19999	19999	19999	19999
unique	18946	3264	19	NaN	79	4
top	C.E. AND OTHERS V REPUBLIC O...	I		NaN	No definido	Mail (e-Mail)
freq	30	1011	3832	NaN	10660	18942
mean	NaN	NaN	NaN	2022-04-04 16:07:18.261913088	NaN	NaN
min	NaN	NaN	NaN	2022-01-01 00:07:00	NaN	NaN
25%	NaN	NaN	NaN	2022-01-31 23:39:30	NaN	NaN
50%	NaN	NaN	NaN	2022-04-09 03:33:00	NaN	NaN
75%	NaN	NaN	NaN	2022-05-09 23:01:30	NaN	NaN
max	NaN	NaN	NaN	2022-12-29 13:06:00	NaN	NaN

Como se puede observar en la Tabla 2, se realizó un conteo basado en la variable *Tipo de Documento*, nuestra variable de interés para predicción, dado que dependiendo de la identificación de este tipo de documento se puede categorizar la correspondencia y así enviar al

área encargada para su correspondiente tramite. La caracterización revela una distribución significativa entre los tipos de documentos radicados. "No definido" domina con un 53.303%, seguido por "Auto" con un 8.53%. Además, tipos como "Judiciales", "Comunicaciones" y "Sentencia" representan alrededor del 7-6% cada uno, mientras que otros tipos de documentos presentan porcentajes menores, desde un 5.5% hasta un 0.125%. Esta diversidad refleja la variedad en la naturaleza y frecuencia de los documentos manejados.

Tabla 2

Distribución de radicados Orfeo según tipo de documento

Tipo Documento	Radicados	%Radicados
No definido	10.660	53.3%
Auto	1.706	8.5%
Judiciales	1.442	7.2%
Comunicaciones	1.344	6.7%
Sentencia	1.240	6.2%
Solicitud de ConciliaciÃ³n Extrajudicial	1.111	5.6%
Acuerdo de gestiÃ³n	866	4.3%
Prejudiciales	302	1.5%
Acta de PosesiÃ³n	238	1.2%
Respuesta solicitud de InformaciÃ³n de Entidades	117	0.6%
Insumo para respuesta	116	0.6%
Escrito de los peticionarios	87	0.43%
Notificaciones	72	0.4%
Acuso de recibido de notas y observaciones	65	0.3%
Requerimientos CIDH	64	0.3%
Derecho de PeticiÃ³n	61	0.3%
Derecho de PeticiÃ³n - InterÃ©s General o Part...	57	0.3%
Orden de pago	48	0.2%
Carta de renuncia	46	0.2%
TrÃ¡mites CrIDH	25	0.1%

En cuanto a la distribución de los radicados, se destaca la categoría ‘No definido’, en la cual cabe aclarar, que la misma fue etiquetada por un humano y corresponde a correspondencia

que no cabe dentro de las categorías definidas, seguidos por una gama diversa de documentos como autos, documentos judiciales, comunicaciones y sentencias, cada uno contribuyendo con porcentajes significativos. Como se observa en la distribución de las categorías, las 20 categorías presentadas abarcan el 98.3% del total de documentos registrados en la muestra de este proyecto. Por esta razón, en la experimentación y en pasos posteriores se decide seleccionar estas categorías.

Encabezan la lista en su mayoría. Cabe aclarar que esta categoría fue establecida por un humano y corresponde a correspondencia que no se ajusta a las categorías definidas. Le siguen una gama diversa de documentos como autos, documentos judiciales, comunicaciones y sentencias, cada uno contribuyendo con porcentajes significativos. Se decidió incluir solo estas 20 categorías, ya que representan el 98.3% del total de documentos registrados en la muestra de este proyecto.

Figura 3

Distribución porcentual de los radicados, según medios de recepción de los documentos

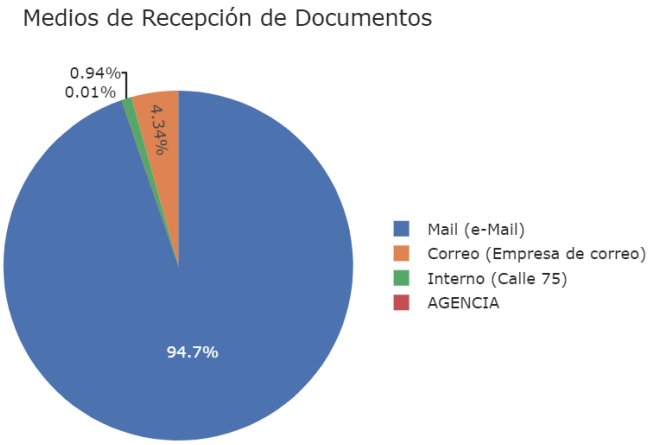


Tabla 3

Distribución de los radicados, según medios de recepción de los documentos

Medio Recepción	Número de Radicados
Mail (e-Mail)	18.942
Correo (Empresa de correo)	867
Interno (Calle 75)	188
AGENCIA	2

De acuerdo con

Tabla ***3*** y Figura 3, la mayor parte de los radicados, aproximadamente el 94.7%, se recibieron a través del medio de recepción denominado "Mail (e-Mail)". Este canal electrónico de comunicación fue utilizado de manera significativa, representando la mayoría abrumadora de los radicados procesados.

Por otro lado, aunque en menor proporción, otros medios de recepción también se utilizaron para recibir radicados. Por ejemplo, un total de 867 radicados por medio de empresa de correo (ver

Tabla ***3***), lo que representa aproximadamente el 4.3%, fueron recibidos a través de este otro servicio de correo. Además, se registraron 188 radicados que fueron procesados internamente, probablemente provenientes de entregas físicas en la ubicación específica en la Agencia, lo cual constituye alrededor del 0.9% del total.

En contraste, se observa que hubo una participación muy limitada de otro medio de recepción llamado "AGENCIA", con solo 2 radicados, lo que representa una proporción mínima

y marginal del total de radicados. Esto indica una utilización extremadamente baja de este medio para recibir documentación en comparación con las demás vías de recepción.

Tras la caracterización de los datos, se identificaron los siguientes filtros de información para la limpieza de los datos: eliminar los procesos donde el remitente sea desconocido, filtrar del conjunto de datos dejando solo radicados relacionados con las 20 categorías que representan el 98.3% de los documentos, y filtrar los radicados en los cuales la variable “medios de recepción” sean "Mail (e-Mail)" y "Correo (Empresa de correo)", para que pueda responder por el denuncio. Después de aplicar estos filtros, la base de datos pasó de tener 19,999 a 19,499 observaciones las que finalmente fueron aprobadas por la ANDJE para este proyecto, lo que representa una pérdida de solo el 3% de la información de la muestra. En esta etapa, también se corrigieron “caracteres raros”, los cuales se refieren a símbolos o signos que no forman parte del conjunto estándar de caracteres utilizados comúnmente en el español, identificando el *encoding* y se agregaron enlaces permanentes (los cuales son urls que permiten la descarga del anexo del documento que se tiene en cada uno de los radicados, tal como indico la Agencia al compartir este conjunto de datos para acceder a los documentos relacionados en la columna Urls anteriormente mencionadas se requiere procesarlos dentro de la infraestructura tecnológica de la Agencia) para acceder a algunos radicados.

Descripción de Metodología de representación usando *Embeddings*.

Teniendo como punto de partida la depuración de los diferentes radicados que componen el conjunto de muestra para este proyecto, el siguiente paso en nuestro análisis consiste en convertir las frases de los documentos en formatos que las computadoras puedan entender y

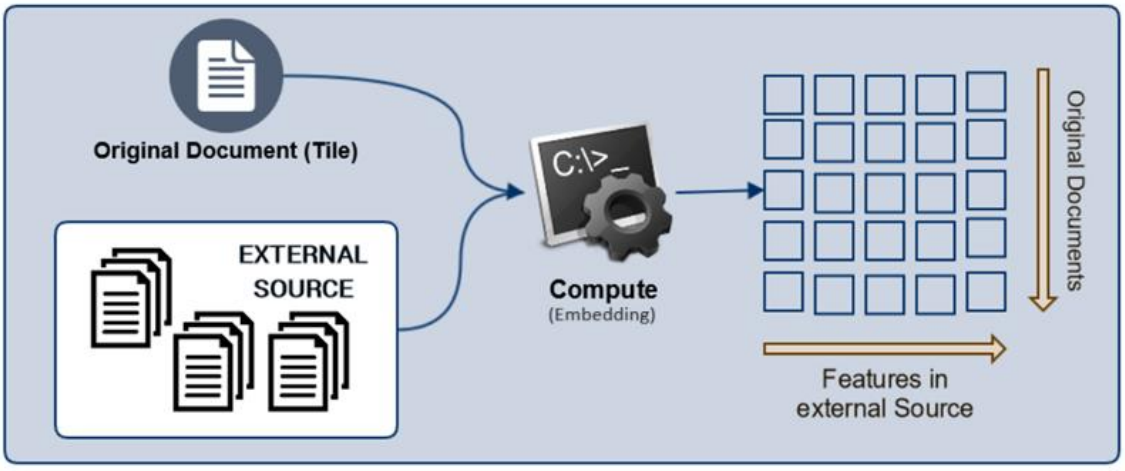
trabajar. Para esto, utilizaremos técnicas llamadas "*embeddings*", que transforman las palabras o frases en representaciones vectoriales, las cuales capturan los significados y relaciones entre las palabras, permitiendo que palabras con significados similares estén representadas por vectores similares.

Tal cómo presenta Figura 4, la idea principal de usar representaciones distribucionales de documentos o terminos llamadas embedding (Tambien son conocidos en la literatura como DTR, representaciones distribucionales de términos por sus siglas en inglés), la idea detrás de esto es construir un nuevo vector de representación numérica que pueda capturar el significado semántico de los documentos usando una fuente externa de información. Formalmente podemos describir el proceso de estos métodos tradicionales cómo (Carrasco Ortiz, 2017):

$$d_i^{dtr} = \sum_{t_j \in d_i} \alpha_{ij} \boldsymbol{w}_j$$

Figura 4

Estrategia de representación distribucional de los términos.



Donde α_{ij} es un número real que mide la contribución del término $t_j \in d_i$ dentro de la representación del documento d_i. El vector $\boldsymbol{w}_j$ es la representación vectorial del termino t_j obtenida a partir de la fuente externa de información (dicha vector se mantiene a lo largo de la colección de textos), estas representaciones de palabras buscan encontrar una fución tal que pueda convertir cada palabra en un vector, de forma que la distancia entre los vectores sea equivalente a la proximidad semántica de las palabras.

Existen diferentes métodos para poder representar $\boldsymbol{w}_j$, en el estado del arte exísten tres metodos de vanguardia para la generación de estas representaciones: Word2Vec, Glove y FastText (Perez, 2018).

Adicional a esto, cómo estrategia básica para encontrar el ponderador α_{ij} se puede usar cómo linea de base la metodologia descrita por Arora et al., 2017, que aumenta entre un 10% y un 30% el rendimiento en los problemas de similitud textos. En esta se pondera los $\boldsymbol{w}_j$ usando el primedio ponderado de los vectores de las palabras y luego usando análisis de componentes principales, haciendo la correción por la palabras más frecuentes de la colección de documentos analizadas (Arora et al., 2017).

Para la representación se busco fuentes actualmente disponibles en el idioma español y que tuvieran un número signiticativamente grande de terminos, en la siguiente tabla se describe cuales fueron las fuentes de información encontradas:

Tabla 4

Descripción de métodos de generación de *Embeddings*

Abreviatura	Corpus	Algoritmo de representación	N° de palabras	Autor
FastText_SUC	Spanish Unannotated Corpora	FastText	1.313.423	José Cañete (Cañete, 2019)
FastText_SBWC	Spanish Billion Word Corpus	FastText	855.380	Jorge Pérez (Perez, 2018)

Glove_SBWC	Spanish Billion Word Corpus	Glove	855.380	Jorge Pérez (Perez, 2018)
Word2Vec_SBWC	Spanish Billion Word Corpus	Word2Vec	1.000.653	Cristian Cardellino (Cardellino, 2016)
FastText_Wiki	Spanish Wikipedia	FastText	985.667	FastText team (Bojanowski et al., 2017)

Para determinar cuál de las diferentes fuentes de información descargadas es la más adecuada para la agencia, se llevaron a cabo diversas pruebas para evaluar el desempeño de los clasificadores. Estos experimentos, detallados en el Capítulo 2, "Método de clasificación supervisado para la categorización de correspondencia ANDJE," permitieron comparar la eficacia de cada fuente de datos. A través de estas pruebas, se identificó la fuente que mejor se ajusta a las necesidades específicas de la Agencia, garantizando una categorización precisa y eficiente de la correspondencia.

Métodos LLM's (Large Language Models)

En esta sección se describe en detalle el proceso de obtención de representaciones vectoriales (Word Embedding) de los documentos, utilizando transformers de la biblioteca Hugging Face. Se emplean tres modelos pre-entrenados de vanguardia: BERT, GPT-2 y XLNet, con el objetivo de analizar su potencial para la clasificación precisa y eficiente de la gestión documental de las comunicaciones recibidas por notificación vía correos electrónicos en la ANDJE (Agencia Nacional de Defensa Jurídica del Estado).

Modelos LLM's Pre-Entrenados:

- **BERT (Bidirectional Encoder Representations from Transformers):** Modelo de transformadores bidireccionales que ha demostrado un rendimiento superior en tareas como la clasificación de textos, la respuesta a preguntas y el resumen automático (Devlin

et al., 2018) Su arquitectura se basa en un enfoque de pre-entrenamiento enmascarado (Masked Language Model - MLM), que permite al modelo aprender a predecir palabras ocultas en un contexto dado, mejorando su capacidad para capturar la semántica y las relaciones contextuales del lenguaje

- **GPT-2 (Generative Pre-trained Transformer 2):** Modelo de transformadores pre-entrenados generativos desarrollado por OpenAI, que destaca por su capacidad para generar textos creativos y coherentes, como poemas, código, guiones y diálogos (Radford et al., 2018). Su arquitectura emplea un enfoque de auto regresión, donde el modelo aprende a predecir la siguiente palabra en una secuencia dada, permitiéndole generar textos con un alto grado de fluidez y coherencia. Además, GPT-2 se ha mostrado efectivo en tareas como la traducción automática y la creación de contenido.
- **XLNet (eXtreme Learning Machine Networks):** Modelo de transformadores auto atendidos permutados que supera las limitaciones de BERT al utilizar una estrategia de pre-entrenamiento permutado (Yang et al., 2019). Este enfoque permite al modelo capturar dependencias de largo alcance en el texto de manera más efectiva, mejorando su rendimiento en tareas que requieren una comprensión profunda del contexto, como la respuesta a preguntas complejas y el resumen de textos largos (Yang et al., 2019).

Procesamiento de los Textos:

Cada modelo LLM procesa los textos de la muestra aleatoria y genera representaciones vectoriales de alta dimensionalidad para cada documento. Estas representaciones, también conocidas como embeddings, capturan información semántica y contextual que será utilizada por el modelo de aprendizaje computacional para aprender patrones y realizar predicciones precisas

sobre la clasificación de los documentos. La forma del tensor que retorna cada modelo es (1, 512, 768), donde:

- **(1)** indica que solo se tiene un ejemplo de entrada para cada modelo.
- **(512)** representa la longitud máxima de la secuencia de entrada utilizada para el modelado (un máximo de 512 tokens).
- **(768)** corresponde a la dimensión de los embeddings generados por cada modelo (cada token en la secuencia de entrada convertido en un vector de 768 dimensiones).

Resultados relevantes representaciones de textos

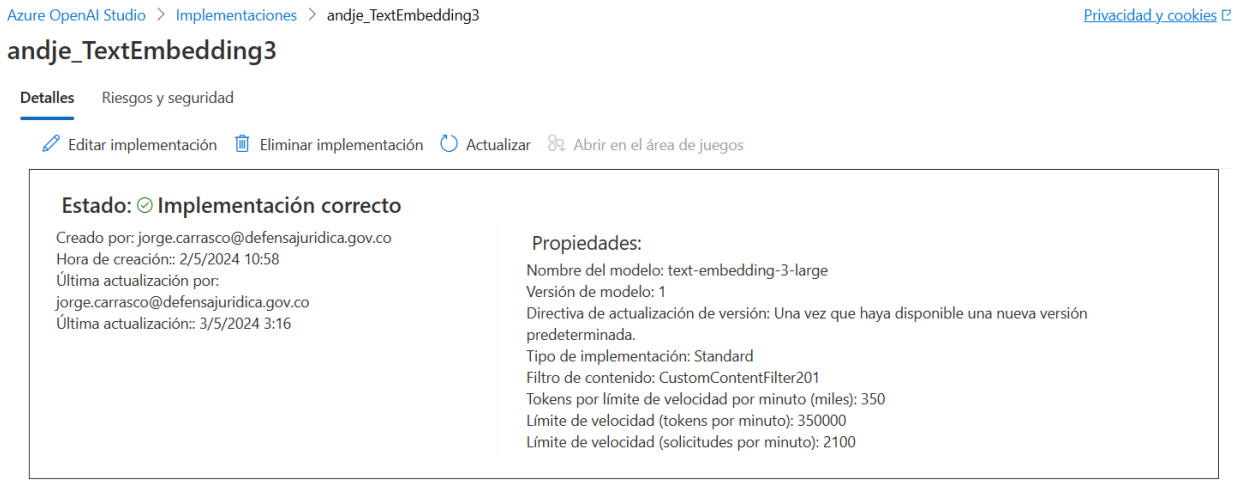
Los modelos LLM's proporcionan representaciones vectoriales de alta dimensionalidad que capturan información semántica y contextual, constituyendo un elemento fundamental para el aprendizaje computacional y la clasificación precisa de documentos. La selección del modelo LLM más adecuado dependerá de las características específicas de la tarea y la disponibilidad de datos.

La Figura 4 muestra la última estrategia implementada usando Azure OpenAI para la representación de textos a través de la representación vectorial de documentos. Esta estrategia utiliza el modelo "text-embedding-3-large", el cual se ha implementado exitosamente en la cuenta de Azure de la Agencia Nacional de Defensa Jurídica del Estado, de acuerdo con las capacidades de la plataforma. Este servicio fue creado por jorge.carrasco@defensajuridica.gov.co, y entre las propiedades que más se destacan de esta configuración se encuentra el límite de velocidad, con un máximo de 350,000 tokens por minuto y 2,100 solicitudes por minuto. Además, se ha configurado una directiva de actualización de versión para que una vez disponible una nueva versión, se implemente como predeterminada,

estas caracteristicas son adecuadas para el conjunto de datos de experimentación que se utiliza en este proyecto aplicado.

Figura 5.

Detalle de la implementación de servicio ***andje_TextEmbedding3***



Según OpenAI, 2024, la última actualización ha introducido el modelo 'text-embedding-3-large', que representa una nueva generación con mejoras significativas en tamaño y rendimiento. Este modelo crea *embeddings* de hasta 3.072 dimensiones, superando considerablemente a la anterior herramienta 'text-embedding-ada-002'.

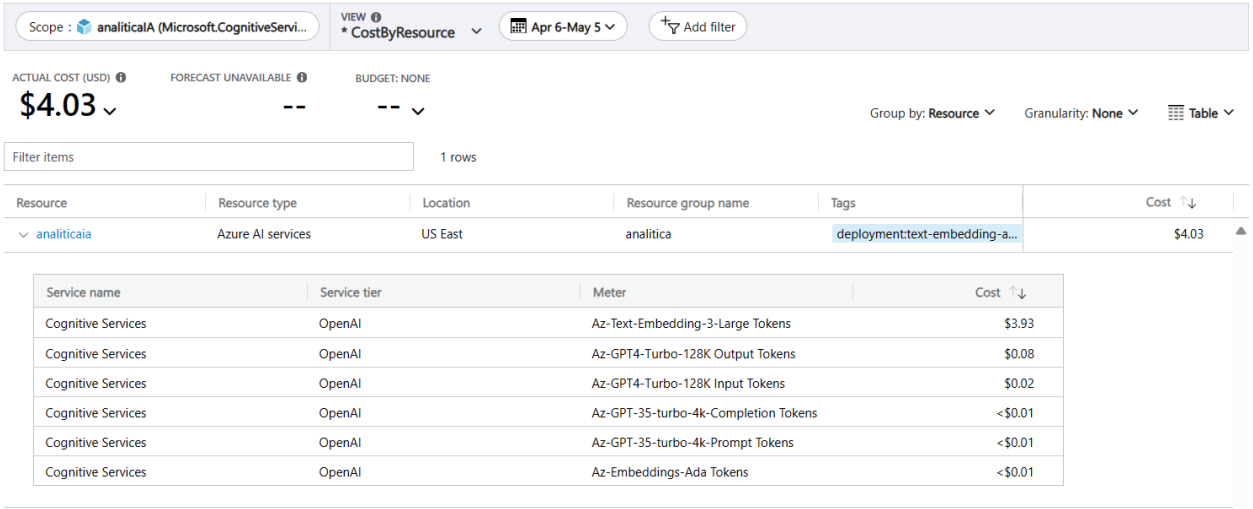
En pruebas como MIRACL (por sus siglas en ingles *Multilingual Retrieval Dataset Covering 18 Diverse Languages*. Los resultados de MIRACL se expresan como un puntaje promedio que indica la precisión del modelo en la recuperación de información en una variedad de idiomas. Un puntaje más alto en MIRACL indica un mejor rendimiento del modelo en la tarea de recuperación de información multi-lenguaje.), el puntaje promedio ha aumentado del 31.4% al 54.9%, mientras que en MTEB (por sus siglas en ingles *Multi-Task English Benchmark*, los resultados de MTEB se expresan como un puntaje promedio que indica la precisión del modelo

en estas tareas específicas del idioma inglés. Un puntaje más alto en MTEB indica un mejor rendimiento del modelo en estas tareas específicas del idioma inglés), el incremento fue del 61.0% al 64.6%. Esta mejora se debe a una representación más potente de los conceptos dentro del contenido, permitiendo a los modelos de aprendizaje automático comprender mejor las relaciones entre el contenido y realizar tareas como agrupación o recuperación (OpenAI, 2024).

A pesar de su mayor tamaño y rendimiento, text-embedding-3-large se ofrece a un precio competitivo de 0.00013 USD por cada 1,000 tokens, haciéndolo accesible para una amplia gama de aplicaciones y casos de uso.

Figura 6

Costo de implementación del modelo Az-Text-Embedding-3-Large



Después de utilizar los recursos para la representación de textos, Azure ofrece la opción de hacer un seguimiento del costo de los servicios según los diferentes recursos desplegados. En la Figura 6, se muestra el costo de las diferentes instancias utilizadas durante los últimos 30 días. Se observa que el servicio utilizado para la representación de textos mediante *embeddings* tuvo un costo de menos de 4 USD para un total de 19,378 documentos procesados y más de 66 millones de tokens.

Tal como se ha indicado en esta sección, la metodología empleada para el procesamiento de lenguaje natural en este proyecto consiste en generar representaciones vectoriales de alta dimensionalidad para cada uno de los documentos. Estas representaciones se obtienen mediante modelos preentrenados de alta calidad y herramientas que facilitan su integración y uso en el proyecto. La elección del método de embedding adecuado es crucial para el éxito de cualquier modelo de procesamiento del lenguaje natural, por ello, en la Tabla 5, se presenta la cantidad de documentos que fueron finalmente representados en cada embedding.

Tabla 5

Dimensión del espacio de características, después de aplicar cada método de embedding en los textos

Método de embedding	**FastText_SUC**	**FastText_SBWC**	**Glove_SBWC**	**Word2Vec_SBWC**	**FastText_Wiki**	**BERT**	**GPT-2**	**XLNet**	**Azure Open AI**
Número de filas	10.147	10.147	10.147	10.147	10.147	10.023	10.022	9.826	10.110
Número de columnas	300	300	300	300	300	768	768	768	3.072

En la tabla anterior (ver Tabla 5), se puede observar que los métodos FastText_SUC, FastText_SBWC, Glove_SBWC, Word2Vec_SBWC y FastText_Wiki, tienen un número constante de filas (10.147), indicando que pueden representar la misma cantidad de textos y no se pierde vocabulario. Por su parte, BERT, GPT-2, XLNet y Azure Open AI, Presentan ligeras variaciones en el número de textos que pueden representar. BERT tiene 10.023 filas, GPT-2 tiene 10.022, XLNet tiene 9.826 y Azure Open AI tiene 10.110 filas. Estas variaciones se deben a la diferente cobertura de vocabulario y a cómo cada modelo maneja las palabras no reconocidas o fuera de su vocabulario preentrenado.

Método de clasificación supervisado para la categorización de correspondencia ANDJE

En este proyecto aplicado se hicieron diversas pruebas para evaluar los métodos de *embedding* y los algoritmos de aprendizaje supervisado seleccionados para el problema en todos los aspectos de interés: la precisión del modelo de clasificación, tiempo de procesamiento y el desempeño del clasificador para las diferentes categorías. En este capítulo se explicará en detalle algunos experimentos para evaluar el desempeño y la precisión de la metodología propuesta para así hacer la selección del modelo que permita la categorización automática de la correspondencia de la ANDJE.

El desarrollo del segundo objetivo del proyecto está estructurado de la siguiente manera: en la primera parte, se proporciona una descripción general de las estadísticas comparativas del tiempo de ejecución y del rendimiento general de los diferentes métodos probados. En la segunda parte, se detallan las características de los diversos experimentos realizados con este conjunto de datos, junto con la comparación estadística entre las estrategias propuestas. Finalmente, en la tercera y última parte, se presentan varias estadísticas de rendimiento del mejor modelo obtenido para la tarea de clasificación, ofreciendo una descripción exhaustiva de su desempeño.

Diseño de experimentos y descripción de tiempos de ejecución

Después de realizar un análisis preliminar de los experimentos de clasificación de los datos utilizando este filtro, observamos que las categorías 'Acta de Posesión', 'Derecho de Petición', 'Notificaciones' y 'Escrito de los peticionarios' presentan un rendimiento muy deficiente en la clasificación. Por esta razón, hemos decidido excluir estas categorías del análisis, considerando que representan solo el 2.33% del total de elementos.

Además, como parte de las decisiones tomadas para mejorar el análisis, hemos optado por unificar las categorías “Prejudiciales” y “Solicitud de Conciliación Extrajudicial”, ya que ambas tratan exactamente la misma temática.

Con el objetivo de identificar el modelo óptimo para abordar el problema de clasificación de texto en esta iniciativa, se evaluaron los clasificadores resultantes de combinar las diferentes propuestas de representación textual (descritas en la sección "Descripción de Metodología de representación usando Embeddings") con los métodos de clasificación presentados en este trabajo, los cuales fueron mencionados en el "marco teórico". En este sentido, como se detalla en la Tabla 1, se probarán siete clasificadores (AdaBoost Classifier, HistGradientBoosting, XGBoost Classifier, Linear SVM, SVM, Logistic Regresión y Random Forest), en conjunto con nueve posibles representaciones de texto (FastText_SUC, FastText_SBWC, Glove_SBWC, Word2Vec_SBWC, FastText_Wiki, BERT, GPT-2, XLNet y Azure Open AI) presentadas en la sección “Implementación de técnicas de procesamiento de lenguaje natural.”, lo que resultará en un total de 63 combinaciones posibles para su evaluación.

Para cada una de las posibles combinaciones, el experimento consistió en evaluar el rendimiento del método utilizando diferentes tamaños de muestra mediante validación cruzada (cv = 5). Se generaron curvas de aprendizaje para visualizar cómo el desempeño del clasificador evoluciona a medida que aumenta el tamaño del conjunto de datos de entrenamiento. Estas curvas de aprendizaje fueron generadas utilizando el paquete Plotly, calculando las puntuaciones de entrenamiento y de validación cruzada en varios tamaños de conjunto de entrenamiento y

representándolas en un rango específico. De esta manera, las curvas de aprendizaje proporcionaron una herramienta para evaluar problemas de sesgo y varianza en el modelo.

En la Tabla 6 se presentan diferentes estadísticas relevantes sobre cada uno de los clasificadores puestos a prueba, en esta etapa se selecciona el tamaño de entrenamiento que mejor resultado se encontró para cada método de embedding para cada método de clasificación evaluado, es decir se construye las diferentes 63 curvas de aprendizaje y validación de las experimentos (Los parámetros utilizados en cada uno de los métodos de clasificación experimentaron fueron los parámetros por defecto de cada algoritmo), a manera de ejemplo, se presentan de los resultados obtenidos en el Anexo 1

Curvas de aprendizaje para modelo SVM y embedding extraído con Azure OpenAI.

Tabla 6

Estadísticas descriptivas de tiempo en segundos de ejecución de los algoritmos de clasificación y evaluación de desempeño

	AdaBoost Classifier (N=450)	**HistGradient Boosting (N=450)**	**Linear SVM (N=450)**	**Logistic Regresion (N=450)**	**Random Forest (N=450)**	**SVM (N=450)**	**XGBoost Classifier (N=450)**	**Total (N=3150)**	**p value**
train_scores									< 0.001
Mean (SD)	0.488 (0.224)	0.845 (0.281)	0.868 (0.094)	0.801 (0.108)	0.997 (0.002)	0.554 (0.207)	0.944 (0.037)	0.785 (0.244)	
Range	0.121 - 0.964	0.136 - 1.000	0.236 - 0.995	0.489 - 0.983	0.993 - 1.000	0.169 - 0.982	0.883 - 1.000	0.121 - 1.000	
test_scores									< 0.001
Mean (SD)	0.238 (0.053)	0.390 (0.158)	0.477 (0.182)	0.451 (0.168)	0.466 (0.167)	0.280 (0.122)	0.478 (0.173)	0.397 (0.178)	
Range	0.098 - 0.373	0.079 - 0.793	0.168 - 0.872	0.194 - 0.841	0.219 - 0.849	0.081 - 0.682	0.227 - 0.848	0.079 - 0.872	
fit_times									< 0.001
Mean (SD)	48.078 (69.517)	90.226 (106.614)	29.590 (47.911)	18.578 (27.792)	8.750 (6.972)	20.906 (47.070)	58.831 (103.786)	39.280 (72.791)	
Range	3.262 - 462.333	6.440 - 727.358	0.102 - 240.787	0.550 - 186.442	1.384 - 39.142	0.110 - 313.164	2.973 - 640.661	0.102 - 727.358	
score_times									< 0.001
Mean (SD)	0.276 (0.439)	1.047 (0.559)	0.008 (0.007)	0.007 (0.007)	0.543 (0.206)	9.125 (12.400)	0.027 (0.016)	1.576 (5.624)	
Range	0.068 - 1.999	0.249 - 2.888	0.003 - 0.053	0.002 - 0.049	0.062 - 1.156	0.246 - 66.564	0.009 - 0.098	0.002 - 66.564	

Al analizar los tiempos de ajuste (*fit_times*) de los diferentes clasificadores presentados en la Tabla 6

Estadísticas descriptivas de tiempo en segundos de ejecución de los algoritmos de clasificación y evaluación de desempeño, se observa una variabilidad significativa en esta variable entre los modelos analizados. Destaca que el clasificador Random Forest tiene el menor tiempo medio de ajuste, con un valor de 8.750 segundos (desviación estándar = 6.972 segundos), mientras que el Histogram-based Gradient Boosting Classifier presenta el mayor tiempo medio de ajuste, con 90.226 segundos (desviación estándar = 106.614 segundos).

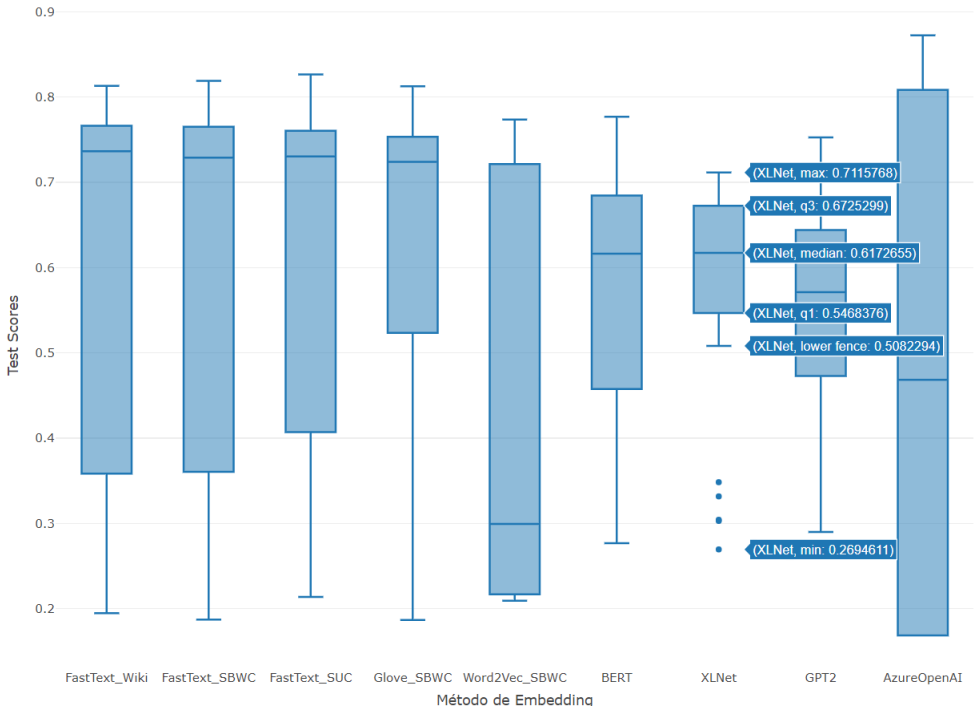
Por otro lado, al considerar los tiempos de puntuación (score_times), se observa que el clasificador SVM muestra una variabilidad excepcionalmente alta, con un tiempo medio de puntuación de 9.125 segundos (desviación estándar = 12.400 segundos) y un rango que va desde 0.246 hasta 66.564 segundos. Este amplio rango de tiempos de puntuación indica una gran variabilidad en el tiempo requerido para puntuar las muestras, lo que podría impactar en la eficiencia y la velocidad de respuesta del modelo en entornos de producción, especialmente en conjuntos de datos grandes o en tiempo real, donde la velocidad de predicción es crucial para la eficiencia del sistema. Estas diferencias en los tiempos de ajuste y puntuación entre los clasificadores resaltan la importancia de seleccionar un modelo que no solo tenga un buen rendimiento predictivo, sino también tiempos de ajuste y puntuación adecuados para el contexto de aplicación.

De acuerdo con los resultados obtenidos en la Tabla 6 y en lo presentado en la Figura 7, se puede observar que cada modelo de embedding tiene su propio rendimiento característico en términos de tendencia central y dispersión. En particular, el modelo FastText entrenado en

Spanish Unannotated Corpora (FastText_SUC) sobresale con una media de 0.628 y una mediana de 0.730, lo que indica una tendencia hacia valores más altos en comparación con otros modelos. Además, tiene una desviación de 0.199, lo que sugiere una dispersión moderada alrededor de la media. Este modelo también tiene un rango de 0.613, lo que indica una variabilidad considerable en los datos. Sin embargo, se puede observar que las representaciones FastTextWiki, FastText_SBWC, Glove_SBWC no están muy alejados del desempeño de clasificación presentado por el método FastText_SUC.

Figura 7

Gráfico Box-plot comparación de distribución del desempeño, según el método de embeddin utilizado.



Por otro lado, el modelo BERT muestra una media ligeramente inferior (0.572) pero una mediana más alta (0.616), lo que sugiere una distribución más balanceada de los datos. Además,

tiene la desviación estándar más baja (0.150) entre todos los modelos, lo que indica una menor dispersión alrededor de la media. Sin embargo, su rango (0.500) es considerable, lo que sugiere cierta variabilidad en los datos.

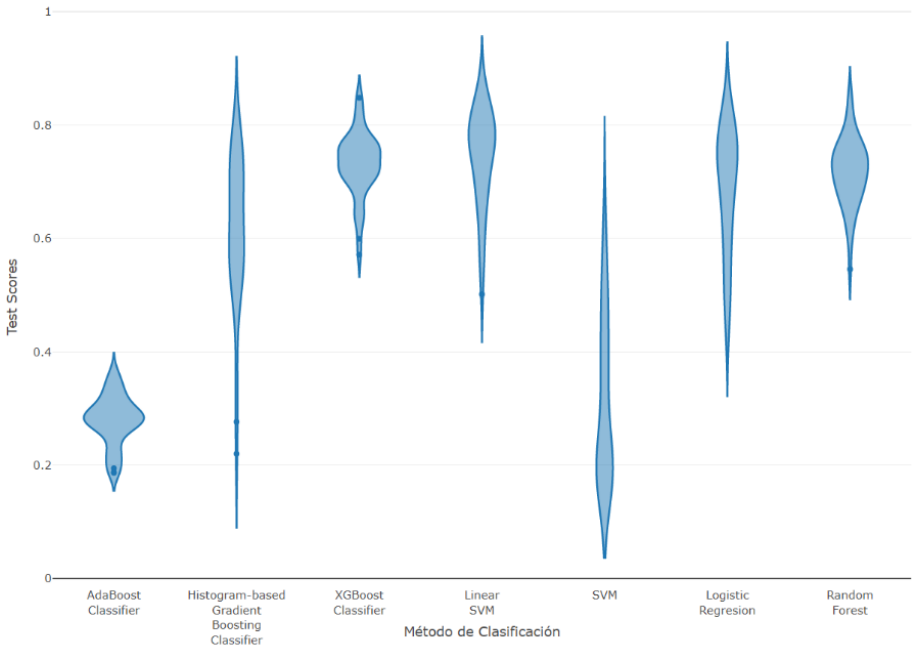
El modelo XLNet muestra una media de 0.574 y una mediana de 0.617 (ver Figura 7), lo que indica una tendencia similar a la de BERT hacia valores medios, sin embargo, tiene una desviación ligeramente más baja (0.122), lo que sugiere una dispersión más compacta alrededor de la media en comparación con otros modelos. Su rango (0.442) también es notablemente más estrecho, lo que indica una menor variabilidad en los datos en comparación con otros modelos.

En general, el modelo FastText entrenado en Spanish Unannotated Corpora (FastText_SUC) parece ofrecer los mejores resultados en términos de tendencia central y el modelo XLNet, como ya se ha mencionado es que presenta menor variabilidad entre los resultados, sin embargo, es importante considerar que cada modelo de *embedding* puede comportarse diferente dependiendo el método de clasificación que se esté utilizando.

En el Figura 8 se presenta una la comparación de la distribución entre los diferentes modelos de clasificación, basándonos en los resultados obtenidos, el modelo Linear SVM (Support Vector Machine) muestra el mejor desempeño en términos de media y mediana, con valores de 0.742 y 0.765 respectivamente. Además, se observa que tiene una de las dispersiones más baja en términos del desempeño en la clasificación. Lo que sugiere que este modelo es consistente y ofrece predicciones precisas y también estables.

Figura 8

Gráfico de Violin comparación de distribución del desempeño, según el método de clasificación utilizado



Entre las técnicas de ensamble se destacan, Random Forest y XGBoost Classifier también muestran un buen desempeño. Ambos modelos tienen valores similares de media y mediana, con desviaciones estándar relativamente bajas. Random Forest tiene una media de 0.717 y una mediana de 0.719, mientras que XGBoost Classifier tiene una media de 0.731 y una mediana de 0.733. Esto sugiere que ambos modelos son robustos y ofrecen resultados consistentes, gráficamente se pueden ver que se concentran en las medidas de tendencia central al tener una dispersión baja.

Por otro lado, el modelo AdaBoost Classifier muestra el peor desempeño entre todos los modelos. Tiene la media más baja (0.280) y la mediana más baja (0.285). Además, su desviación estándar es la más baja (0.044), lo que indica que el modelo AdaBoost Classifier puede no ser

adecuado para esta tarea de clasificación en particular dado que es consistentemente bajo su precisión en la tarea de clasificación.

Los resultados del ANOVA (ver Tabla 7) indican efectos principales significativos tanto para el Método de Embedding como para el Método de Clasificación. El valor p es menor que 0.001, lo que sugiere un efecto altamente significativo según el método de embedding en el puntaje de precisión del modelo. Esto significa que diferentes modelos de *embedding* conducen a puntajes promedio de desempeño en la tarea de clasificación estadísticamente diferentes. Del mismo modo, el valor p para el Método de Clasificación también es menor que 0.001, lo que indica existe un efecto altamente significativo del modelo de clasificación en los puntajes de prueba, es decir diferentes modelos de clasificación conducen a desempeño estadísticamente diferentes.

Tabla 7

Resultados del ANOVA de efectos fijos utilizando test_scores como criterio

Predictor	Suma de cuadrados	*Grados de libertas*	Mean Square	*Valor F*	*p-valor*
(Intercept)	0.51	1	0.51	153.05	.000
Method_Embedding	0.05	8	0.01	1.72	.093
Method_Clasifier	4.49	6	0.75	226.27	.000
Method_Embedding x Method_Clasifier	2.37	48	0.05	14.92	.000
Error	0.93	282	0.00		

Además, se observa un efecto de interacción significativo entre el Método de Embedding y el Método de Clasificación ($p < 0.001$ ***). Esto sugiere que el efecto de un factor (el método de embedding) en los puntajes de prueba depende del nivel del otro factor (el modelo de clasificación). En otras palabras, el rendimiento de diferentes modelos de embedding podría variar dependiendo del modelo de clasificación elegido. En resumen, los resultados del ANOVA

sugieren fuertemente que tanto la elección del modelo de embedding como la elección del modelo de clasificación impactan significativamente los puntajes de prueba. Además, existe una interacción significativa entre estos dos factores, lo que significa que la mejor combinación de modelos de embedding y clasificación puede diferir dependiendo del escenario específico.

Entrando más en detalle se puede sacar conclusiones entre la interacción de los métodos de embeddings con los métodos de clasificación (ver tabla Figura 9, Tabla 8 y Figura 10 que se presenta en los

Anexos), se presentan la distribución del desempeño en el conjunto de pruebas de varios clasificadores utilizando diferentes métodos de embedding, junto con los valores medios y las desviaciones estándar de los puntajes de prueba obtenidos. Adicionalmente, se discuten las diferencias significativas encontradas mediante pruebas estadísticas.

Tabla 8

Análisis descriptivo del desempeño de los clasificadores, según método de Embedding y método de clasificación supervisado

Method_Embedding		AdaBoost Classifier (N=45)	Histogram-based Gradient Boosting Classifier (N=45)	Linear SVM (N=45)	Logistic Regresion (N=45)	Random Forest (N=45)	SVM (N=75)	XGBoost Classifier (N=45)	Total (N=345)	p value
AzureOpenAI	**test_scores**									< 0.001
	Mean (SD)	0.318 (0.041)	0.589 (0.018)	0.848 (0.020)	0.808 (0.030)	0.815 (0.026)	0.169 (0.000)	0.825 (0.021)	0.488 (0.303)	
	Range	0.271 - 0.366	0.570 - 0.608	0.828 - 0.872	0.780 - 0.841	0.792 - 0.849	0.169 - 0.169	0.803 - 0.848	0.169 - 0.872	
BERT	**test_scores**									< 0.001
	Mean (SD)	0.306 (0.030)	0.593 (0.132)	0.734 (0.051)	0.563 (0.082)	0.660 (0.035)	0.469 (0.039)	0.678 (0.051)	0.572 (0.150)	
	Range	0.277 - 0.343	0.361 - 0.676	0.666 - 0.777	0.475 - 0.685	0.602 - 0.689	0.434 - 0.521	0.600 - 0.718	0.277 - 0.777	
FastText_SBWC	**test_scores**									< 0.001
	Mean (SD)	0.220 (0.030)	0.630 (0.229)	0.790 (0.025)	0.754 (0.025)	0.735 (0.025)	0.348 (0.030)	0.748 (0.019)	0.604 (0.228)	
	Range	0.187 - 0.267	0.220 - 0.751	0.767 - 0.819	0.727 - 0.789	0.709 - 0.766	0.312 - 0.378	0.722 - 0.767	0.187 - 0.819	
FastText_SUC	**test_scores**									< 0.001
	Mean (SD)	0.272 (0.033)	0.723 (0.014)	0.797 (0.026)	0.761 (0.024)	0.729 (0.023)	0.380 (0.040)	0.738 (0.024)	0.628 (0.199)	
	Range	0.214 - 0.294	0.707 - 0.743	0.762 - 0.827	0.738 - 0.799	0.702 - 0.762	0.321 - 0.423	0.711 - 0.761	0.214 - 0.827	
FastText_Wiki	**test_scores**									< 0.001
	Mean (SD)	0.248 (0.038)	0.675 (0.199)	0.786 (0.025)	0.756 (0.033)	0.736 (0.025)	0.348 (0.028)	0.749 (0.022)	0.614 (0.219)	
	Range	0.195 - 0.285	0.320 - 0.790	0.759 - 0.813	0.724 - 0.805	0.702 - 0.765	0.318 - 0.382	0.722 - 0.767	0.195 - 0.813	

Tabla 8 (Continuación)

Análisis descriptivo del desempeño de los clasificadores, según método de Embedding y método de clasificación supervisado

Method_Embedding		AdaBoost Classifier (N=45)	Histogram-based Gradient Boosting Classifier (N=45)	Linear SVM (N=45)	Logistic Regresion (N=45)	Random Forest (N=45)	SVM (N=75)	XGBoost Classifier (N=45)	Total (N=345)	p value
Glove_SBWC	**test_scores**									< 0.001
	Mean (SD)	0.258 (0.041)	0.560 (0.021)	0.777 (0.024)	0.761 (0.036)	0.729 (0.024)	0.507 (0.044)	0.747 (0.021)	0.620 (0.182)	
	Range	0.187 - 0.290	0.538 - 0.589	0.752 - 0.804	0.729 - 0.813	0.702 - 0.754	0.462 - 0.577	0.724 - 0.771	0.187 - 0.813	
GPT2	**test_scores**									< 0.001
	Mean (SD)	0.308 (0.015)	0.561 (0.037)	0.615 (0.084)	0.502 (0.061)	0.656 (0.070)	0.502 (0.052)	0.673 (0.070)	0.545 (0.129)	
	Range	0.290 - 0.323	0.508 - 0.594	0.502 - 0.725	0.427 - 0.587	0.546 - 0.734	0.451 - 0.572	0.571 - 0.753	0.290 - 0.753	
Word2Vec_SBWC	**test_scores**									< 0.001
	Mean (SD)	0.279 (0.023)	0.642 (0.205)	0.711 (0.030)	0.659 (0.023)	0.745 (0.023)	0.214 (0.003)	0.752 (0.025)	0.464 (0.250)	
	Range	0.261 - 0.318	0.276 - 0.753	0.685 - 0.759	0.635 - 0.689	0.710 - 0.768	0.209 - 0.217	0.718 - 0.774	0.209 - 0.774	
XLNet	**test_scores**									< 0.001
	Mean (SD)	0.311 (0.030)	0.535 (0.021)	0.623 (0.059)	0.617 (0.054)	0.652 (0.028)	0.608 (0.052)	0.674 (0.034)	0.574 (0.122)	
	Range	0.269 - 0.348	0.508 - 0.554	0.553 - 0.682	0.537 - 0.680	0.614 - 0.676	0.550 - 0.682	0.634 - 0.712	0.269 - 0.712	

Entre los clasificadores evaluados (ver tabla Figura 9 y Tabla 8), el Linear SVM destaca por su rendimiento superior, con una precisión media de 0.848 y una desviación estándar de 0.020. Su rango de puntajes de prueba, que va desde 0.828 hasta 0.872, indica una notable consistencia en su desempeño. Se puede observar que la línea del promedio de desempeño es muy superior a la de otros clasificadores. supera significativamente a la mayoría de los clasificadores. Por ejemplo, en comparación con el Random Forest, que tiene una precisión media de 0.815 y un rango de 0.792 a 0.849.

Al analizar la interacción entre los métodos de embedding y los clasificadores, se observan resultados significativos. El método de embedding "FastText_SBWC" combinado con el Linear SVM alcanza una precisión media de 0.790, una de las más altas entre todas las combinaciones. Similarmente, el método de embedding "FastText_SUC" combinado con el Linear SVM también muestra un rendimiento destacado, con una precisión media de 0.797.

En las diferentes salidas presentadas en la Figura 9 y Tabla 8, se observa que los mejores desempeños se obtienen utilizando la presentación “AzureOpenAI”, la cual muestra un rendimiento destacado en varios métodos de clasificación. Por ejemplo, al combinarse con el Linear SVM, el método "AzureOpenAI" alcanza una precisión media de 0.848, con un rango de puntajes de prueba que va desde 0.828 hasta 0.872. Este rendimiento es significativamente superior al de otros métodos de embedding en la misma configuración de clasificación. Además, el análisis de varianza y la prueba de Tukey HSD (ver Figura 10 en la sección de

Anexos) revelaron que, en comparación con otros métodos de embedding, como "Word2Vec_SBWC" y "FastText_SBWC", el método "AzureOpenAI" mostró diferencias significativas en el rendimiento cuando se combinó con el Linear SVM. Estos resultados resaltan la eficacia del método "AzureOpenAI" para la tarea de clasificación de correspondencia de la ANDJE.

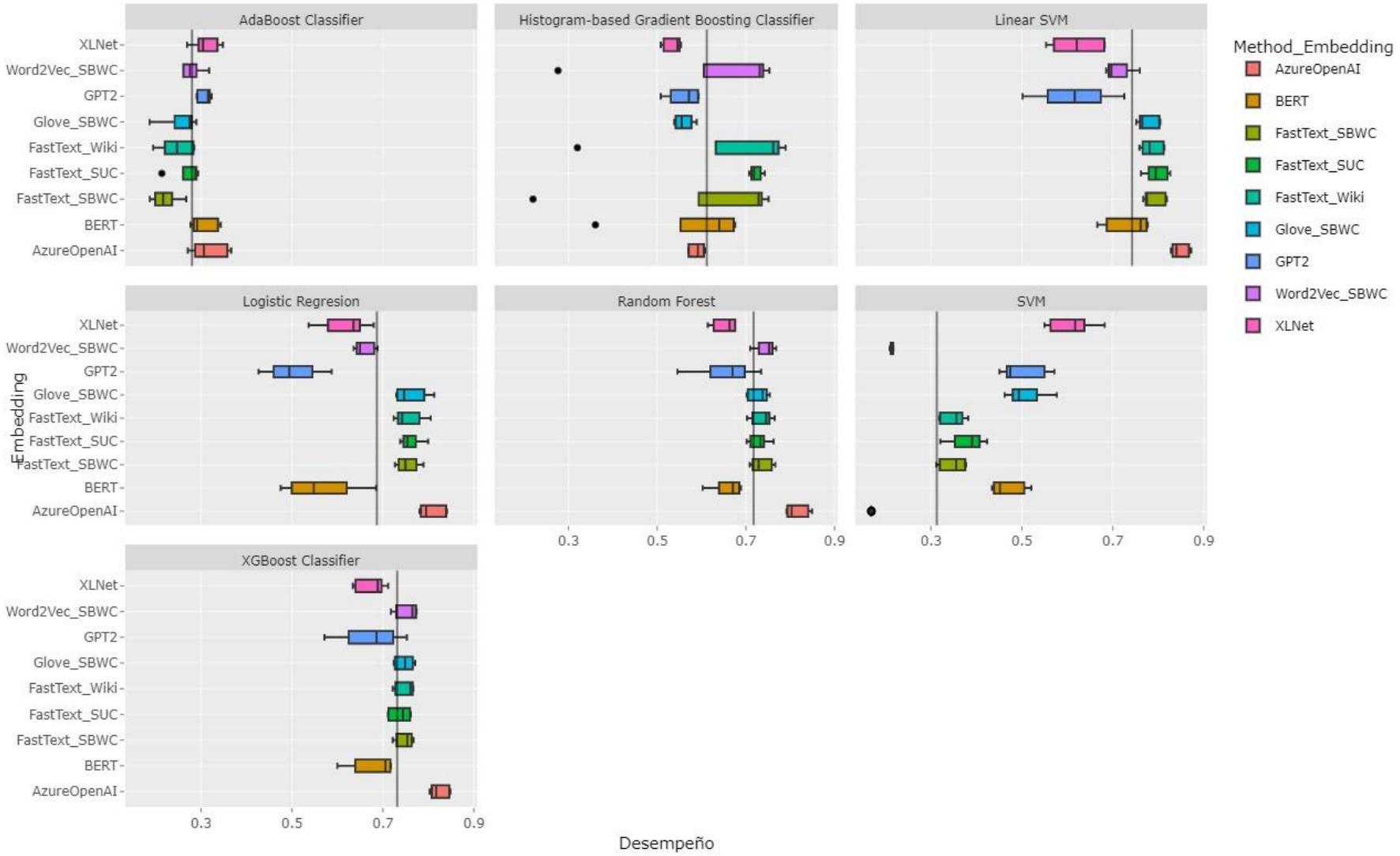
Al analizar los resultados del análisis de varianza (ANOVA) y la prueba de Tukey HSD (ver Figura 10 en la sección de

Anexos), se observa que existen diferencias significativas entre las combinaciones de métodos de embedding y clasificadores. Por ejemplo, se encontró una diferencia significativa en el rendimiento entre el Linear SVM y otros clasificadores, como el Random Forest y la Regresión Logística, para el método de embedding "AzureOpenAI". Además, se encontraron diferencias significativas entre los métodos de embedding para el mismo clasificador, como se observa en la diferencia de rendimiento entre "Word2Vec_SBWC" y "FastText_SBWC" cuando se combinan con el Linear SVM.

Basándonos en el análisis realizado, se concluye que el modelo más adecuado para la clasificación de la correspondencia de la Agencia es el método de embedding proporcionado por AzureOpenAI, combinado con el clasificador Linear SVM. El siguiente paso en el análisis implica experimentar con los diferentes parámetros disponibles para estos modelos en nuestro conjunto de datos. Además, se planea implementar una estrategia de ensamble generativo llamada Bagging, la cual tiene como objetivo mejorar la precisión y la estabilidad del modelo mediante la combinación de múltiples clasificadores entrenados con diferentes subconjuntos de datos. El Bagging busca reducir la varianza del modelo y evitar el sobreajuste al promediar los resultados de varios clasificadores.

Figura 9

Distribución del desempeño de la tarea de clasificación supervisada según método de embeddings y método de clasificación



Conclusiones

En este proyecto aplicado, se empleó una metodología basada en la clasificación supervisada de textos para categorizar las correspondencias de la Agencia Nacional de Defensa Jurídica (ANDJE). La metodología se fundamenta en la representación vectorial de los términos o textos que conforman el conjunto de datos disponible, utilizando embeddings como fuente de información externa y diversos modelos de última generación.

Los embeddings, o representaciones vectoriales, tienen como objetivo capturar el significado semántico y contextual de las palabras y frases. Al transformar los textos en representaciones numéricas, se facilita el procesamiento y análisis por parte de algoritmos de aprendizaje automático. En este caso, los embeddings permiten entrenar modelos de clasificación supervisada para categorizar las correspondencias de la ANDJE de manera eficiente y precisa.

Se realizó la evaluación experimental de 63 escenarios haciendo la combinación de varios modelos de clasificación supervisada y varias metodologías de representación vectorial de textos, haciendo la identificación de tres factores claves en la evaluación de dichos modelos: la precisión del modelo de clasificación, tiempo de procesamiento y el desempeño del clasificador para las diferentes categorías. Las principales conclusiones que se obtuvieron de los experimentos realizados son:

- Tal como se mencionó en el segundo capítulo, como estudio experimental en la muestra de conjuntos de datos, se concluyó que el modelo más adecuado para la clasificación de la correspondencia de la ANDJE es el método de embedding proporcionado por AzureOpenAI (text-embedding-3-large), combinado con el clasificador Linear SVM. Este modelo supera en un 6,4% al modelo FastText entrenado en Spanish Unannotated Corpora (FastText_SUC), en un 15,5% a BERT y en un 36,1% a XLNET. Cabe destacar

que estos métodos de vanguardia han demostrado ser altamente efectivos en diversas tareas relacionadas con la minería de textos, lo que respalda la solidez de la elección realizada.

- Se demostró que el uso de diferentes modelos de embedding mostró un rendimiento característico en términos de tendencia central y dispersión. El modelo FastText entrenado en Spanish Unannotated Corpora (FastText_SUC) ofreció los mejores resultados en términos de tendencia central. Sin embargo, el modelo BERT y XLNET mostró una distribución más balanceada de los datos y una menor dispersión alrededor de la media.
- En cuanto a los modelos de clasificación supervisado, el modelo Linear SVM muestra el mejor desempeño en términos de precisión y estabilidad, seguido de cerca por Random Forest y XGBoost Classifier. Por otro lado, AdaBoost Classifier muestra el peor desempeño y puede no ser la mejor opción para esta tarea específica de clasificación.
- Se observaron resultados significativos al analizar la interacción entre los métodos de embedding y los clasificadores. El método de embedding "AzureOpenAI" mostró un rendimiento destacado en varios métodos de clasificación, siendo especialmente efectivo cuando se combinó con el Linear SVM.
- El análisis de varianza (ANOVA) y la prueba de Tukey HSD (ver sección de anexos) revelaron diferencias significativas entre las combinaciones de métodos de embedding y clasificadores. Por ejemplo, se encontró una diferencia significativa en el rendimiento entre el Linear SVM y otros clasificadores, como el Random Forest y la Regresión Logística, para el método de embedding "AzureOpenAI".

Si bien el método propuesto demostró ser efectivo para categorizar y predecir la temática de los radicados, es importante destacar que se trata de una primera aproximación al problema y presenta algunas limitaciones. Una de las principales limitaciones es la falta de estandarización en las bases de datos utilizadas. Esto significa que las categorías no están bien definidas, hay inconsistencias en el formato de los datos y se utilizan caracteres especiales que no son compatibles con los modelos de aprendizaje automático. Estas deficiencias pueden afectar negativamente la precisión de las predicciones y dificultar el mantenimiento del sistema a largo plazo.

Para mejorar la precisión y generalización de los modelos, se requeriría contar con bases de datos más estandarizadas y de mayor tamaño. Además, se podrían explorar técnicas de aprendizaje automático más avanzadas, como el aprendizaje profundo, para mejorar aún más el rendimiento del sistema.

Otra limitación del estudio se encuentra relacionado con la interpretabilidad de los resultados, modelos de aprendizaje automático pueden ser difíciles de interpretar. Esto puede dificultar la comprensión de cómo los modelos llegaron a sus predicciones y puede dificultar la confianza en los resultados. Estas cuestiones de interés precisarían un tratamiento complementario a los textos y/o utilizar una metodología de clasificación diferente a la que fue propuesta en esta investigación.

Trabajo futuro

En el contexto de la clasificación de correspondencia de la ANDJE, se han identificado varios componentes del método propuesto que podrían optimizar los resultados obtenidos, por medio de nuevas exploraciones y la incorporación de información externa adicional, es posible mejorar la precisión y la efectividad del modelo, capturando de manera más precisa el

significado semántico y sintáctico de los textos. A continuación, se describen tres aspectos claves que se podrían mejorar:

1. La optimización de los hiperparámetros del mejor modelo resultante SVM puede conducir a mejoras significativas en el rendimiento. Se puede llevar a cabo una exploración exhaustiva de los hiperparámetros utilizando técnicas avanzadas como la búsqueda en cuadrícula (*Grid Search*) y la búsqueda aleatoria (*Random Search*). Además, la implementación de un *Bagging Classifier* podría combinar múltiples interacciones del modelo para aprovechar sus fortalezas individuales, mejorando así la precisión general del sistema.
2. Para capturar mejor el contexto semántico y sintáctico de los textos cortos, es esencial integrar fuentes de información externa. Esto podría incluso implicar el entrenamiento de embeddings propios con los textos del ámbito jurídico con los que cuenta la ANDJE, así como la incorporación de datos adicionales específicos del dominio legal. La inclusión de conocimiento especializado permitirá al modelo entender mejor las sutilezas y el vocabulario técnico propio de la correspondencia jurídico.
3. La reducción de dimensionalidad mediante Análisis de Componentes Principales (PCA, por sus siglas en inglés) o descomposición de valores singulares (SVD), puede ser una técnica efectiva para mejorar el tiempo de procesamiento y la eficiencia del modelo. Al reducir el número de dimensiones, estas técnicas facilitan la eliminación de ruido y redundancia en los datos, permitiendo que el modelo se enfoque en las características más relevantes. Esta estrategia no solo

podría optimizar el rendimiento del modelo, sino que también acelera la exploración de hiperparámetros y el entrenamiento del modelo.

Referencias Bibliográficas

Ahmed, A. A. A., Aljabouh, A., Donepudi, P. K., & Choi, M. S. (2021). Detecting fake news using machine learning: A systematic literature review. *arXiv preprint arXiv:2102.04458*.

Antonie, L., & Zaïane, O. (2002). *Text document categorization by term association*. 19–26. https://doi.org/10.1109/ICDM.2002.1183881

Arora, S., Liang, Y., & Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. *International conference on learning representations*.

Ashley, K. D. (2017). *Artificial intelligence and legal analytics: new tools for law practice in the digital age*. Cambridge University Press.

Baharudin, B., Lee, L. H., Khan, K., & Khan, A. (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. *Journal of Advances in Information Technology*, *1*. https://doi.org/10.4304/jait.1.1.4-20

Bayer, T., Kressel, U., Mogg-Schneider, H., & Renz, I. (1998). Categorizing paper documents: a generic system for domain and language independent text categorization. *Computer Vision and Image Understanding*, *70*(3), 299–306.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135–146.

Cahyani, D., & Patasik, I. (2021). Performance comparison of TF-IDF and Word2Vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics*, *10*, 2780–2788. https://doi.org/10.11591/eei.v10i5.3157

Cañete, J. (2019). *Spanish Word Embeddings [Data set]. Zenodo*. https://doi.org/10.5281/zenodo.3255001

Cardellino, C. (2016). *Spanish Billion Words Corpus and Embeddings*. https://crscardellino.github.io/SBWCE/

Carrasco Ortiz, J. M. (2017). *Agrupación de textos cortos para el análisis de temas latentes de investigación en un conjunto de datos de proyectos de investigación*.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fatima, S., Srinivasu, B., & others. (2017). Text Document categorization using support vector machine. *International Research Journal of Engineering and Technology (IRJET)*, *4*(2), 141–147.

Hassan, S. U., Ahamed, J., & Ahmad, K. (2022). Analytics of machine learning-based algorithms for text classification. *Sustainable operations and computers*, *3*, 238–248.

Humphrey, S. M., Névéol, A., Browne, A., Gobeil, J., Ruch, P., & Darmoni, S. J. (2009). Comparing a rule-based versus statistical system for automatic categorization of MEDLINE documents according to biomedical specialty. *Journal of the American Society for Information Science and Technology*, *60*(12), 2530–2539.

Jindal, R., Malhotra, R., & Jain, A. (2015). Techniques for text classification: Literature review and current trends. *webology*, *12*(2).

Kotsiantis, S. B., Zaharakis, I., Pintelas, P., & others. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3–24.

Lan, M., Tan, C. L., Su, J., & Lu, Y. (2008). Supervised and traditional term weighting methods for automatic text categorization. *IEEE transactions on pattern analysis and machine intelligence*, *31*(4), 721–735.

Lee, J., Kim, J., & Kang, P. (2021). Back-translated task adaptive pretraining: Improving accuracy and robustness on text classification. *arXiv preprint arXiv:2107.10474*.

Lee, Y.-H., Hu, P. J.-H., Tsao, W.-J., & Li, L. (2021). Use of a domain-specific ontology to support automated document categorization at the concept level: Method development and evaluation. *Expert Systems with Applications*, *174*, 114681. https://doi.org/https://doi.org/10.1016/j.eswa.2021.114681

OpenAI. (2024). *New embedding models and API updates*. https://openai.com/index/new-embedding-models-and-api-updates

Perez, J. (2018, mayo 15). *Spanish Word Embeddings*. GitHub. https://github.com/dccuchile/spanish-word-embeddings?tab=readme-ov-file#references

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., & others. (2018). *Improving language understanding by generative pre-training*.

Ramirez-Loaiza, M., Culotta, A., & Bilgic, M. (2013, mayo). *Anytime Active Learning*.

Raschka, S. (2015). *Python machine learning*. Packt publishing ltd.

Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., & Wang, Z. (2007). A novel feature selection algorithm for text categorization. *Expert Systems with Applications*, *33*(1), 1–5. https://doi.org/https://doi.org/10.1016/j.eswa.2006.04.001

Singh, K. N., Devi, S. D., Devi, H. M., & Mahanta, A. K. (2022). A novel approach for dimension reduction using word embedding: An enhanced text classification approach. *International Journal of Information Management Data Insights*, *2*(1), 100061.

Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C.-C. J. (2019). Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, *8*, e19. https://doi.org/10.1017/ATSIP.2019.12

Wang, R., Li, Z., Cao, J., Chen, T., & Wang, L. (2019). Convolutional Recurrent Neural Networks for Text Classification. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–6. https://doi.org/10.1109/IJCNN.2019.8852406

Wang, S., Zhou, W., & Jiang, C. (2020). A survey of word embeddings based on deep learning. *Computing*, *102*. https://doi.org/10.1007/s00607-019-00768-7

Wang, T.-Y., & Chiang, H.-M. (2007). Fuzzy support vector machine for multi-class text categorization. *Information Processing & Management*, *43*(4), 914–929. https://doi.org/10.1016/j.ipm.2006.09.011

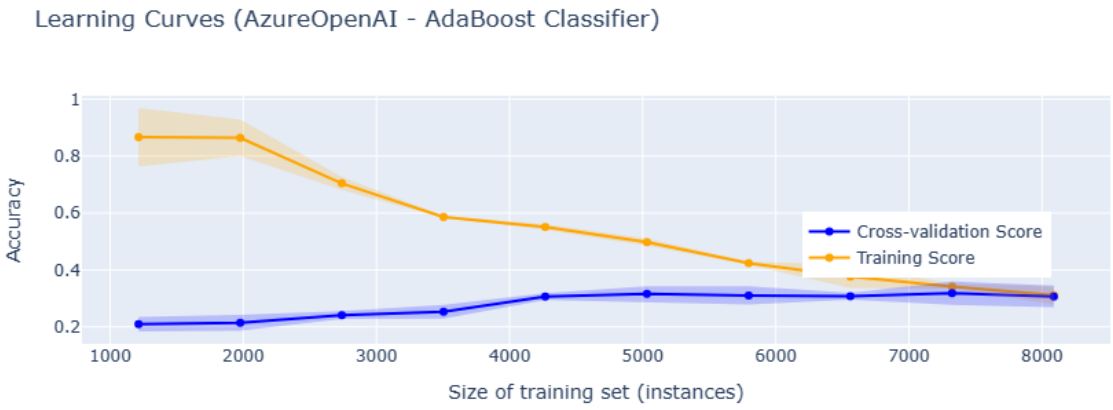
Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, *32*.

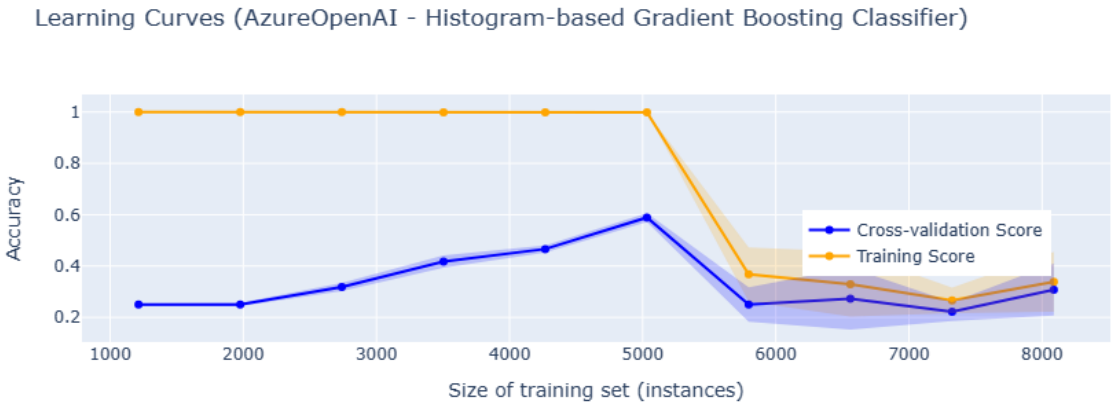
Zelaia, A., Alegria, I., Arregi, O., & Sierra, B. (2011). A multiclass/multilabel document categorization system: Combining multiple classifiers in a reduced dimension. *Appl. Soft Comput.*, *11*, 4981–4990. https://doi.org/10.1016/j.asoc.2011.06.002

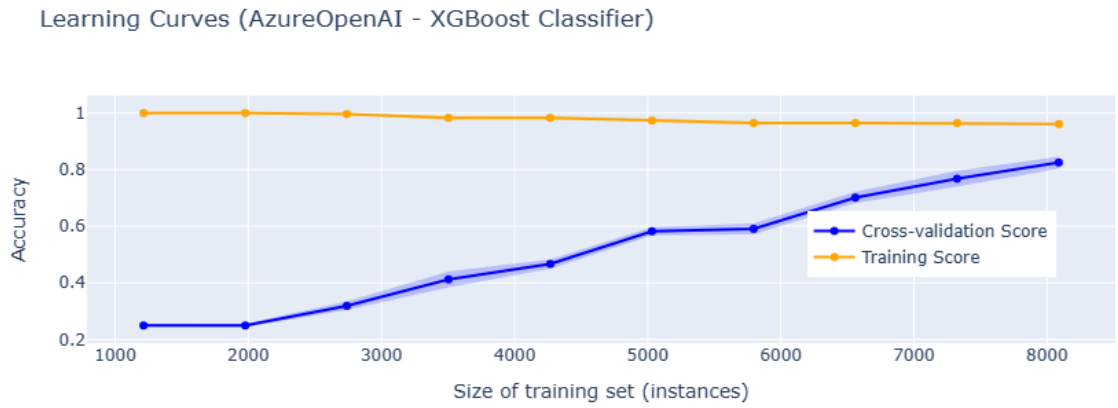
Anexos

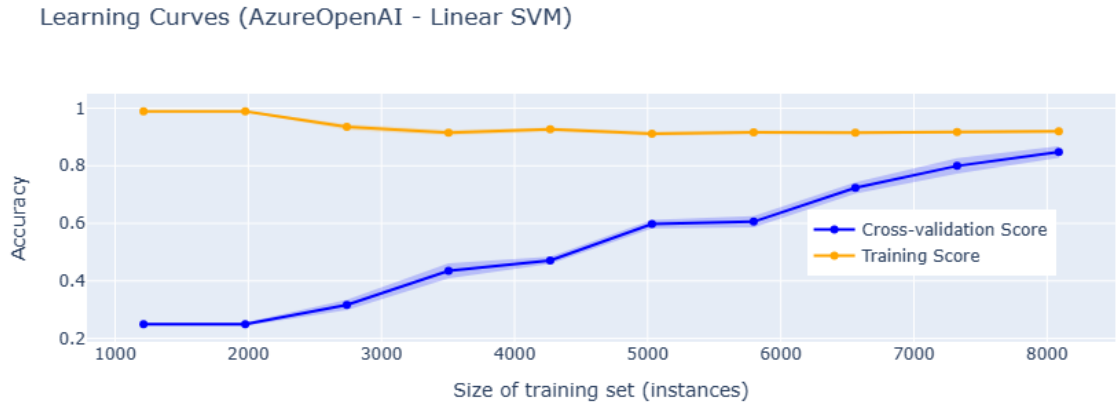
Anexo 1

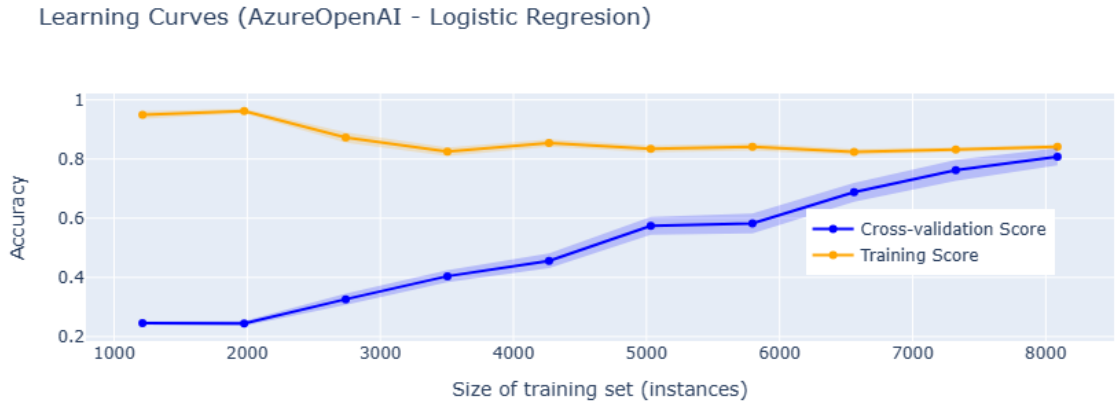
Curvas de aprendizaje para modelo SVM y embedding extraído con Azure OpenAI

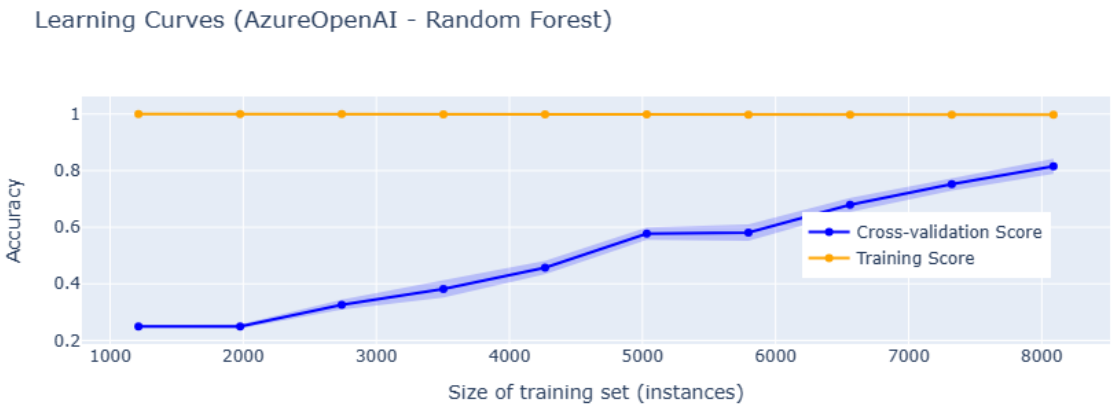





Learning Curves (AzureOpenAI - XGBoost Classifier)
Accuracy
1
0.8
0.6
0.4
0.2
1000
2000
3000
4000
5000
6000
7000
8000
Size of training set (instances)
Cross-validation Score
Training Score


Learning Curves (AzureOpenAI - Linear SVM)
Accuracy
1
0.8
0.6
0.4
0.2
1000
2000
3000
4000
5000
6000
7000
8000
Size of training set (instances)
Cross-validation Score
Training Score


Learning Curves (AzureOpenAI - Logistic Regresion)
Accuracy
1
0.8
0.6
0.4
0.2
1000
2000
3000
4000
5000
6000
7000
8000
Size of training set (instances)
Cross-validation Score
Training Score



Anexo 2

Salida análisis de varianza (ANOVA) y la prueba de Tukey HSD

Figura 10

Top 20 de principales diferencias encontradas por pruebas Tukey HSD

```
> aa = TukeyHSD(res.aov2)
> bb = data.frame(aa$`Method_Embedding:Method_Clasifier`)
> bb <- bb[order(bb$diff), ]
> head(bb, 10)
                                                  diff        lwr        upr        p.adj
AzureOpenAI:SVM-AzureOpenAI:Linear SVM           -0.6790307 -0.7983694 -0.5596919 5.139222e-13
AzureOpenAI:SVM-AzureOpenAI:Random Forest        -0.6464886 -0.7658274 -0.5271499 5.139222e-13
AzureOpenAI:SVM-AzureOpenAI:Logistic Regresion   -0.6389713 -0.7583100 -0.5196326 5.139222e-13
Word2Vec_SBWC:SVM-AzureOpenAI:Linear SVM         -0.6339181 -0.7532568 -0.5145793 5.139222e-13
AzureOpenAI:SVM-FastText_SUC:Linear SVM          -0.6284413 -0.7477801 -0.5091026 5.139222e-13
AzureOpenAI:SVM-FastText_SBWC:Linear SVM         -0.6214435 -0.7407822 -0.5021048 5.139222e-13
AzureOpenAI:SVM-FastText_Wiki:Linear SVM         -0.6176015 -0.7369402 -0.4982628 5.139222e-13
AzureOpenAI:SVM-Glove_SBWC:Linear SVM            -0.6083371 -0.7276758 -0.4889984 5.139222e-13
Word2Vec_SBWC:SVM-AzureOpenAI:Random Forest      -0.6013760 -0.7207148 -0.4820373 5.139222e-13
Word2Vec_SBWC:SVM-AzureOpenAI:Logistic Regresion -0.5938587 -0.7131974 -0.4745200 5.139222e-13
> tail(bb, 10)
                                                                 diff       lwr       upr        p.adj
FastText_Wiki:XGBoost Classifier-AzureOpenAI:SVM                0.5799550 0.4606162 0.6992937 5.139222e-13
Word2Vec_SBWC:XGBoost Classifier-AzureOpenAI:SVM                0.5837006 0.4643619 0.7030393 5.139222e-13
AzureOpenAI:Logistic Regresion-FastText_SBWC:AdaBoost Classifier 0.5876490 0.4366961 0.7386019 5.139222e-13
AzureOpenAI:Linear SVM-Glove_SBWC:AdaBoost Classifier           0.5901547 0.4392018 0.7411076 5.139222e-13
AzureOpenAI:Random Forest-FastText_SBWC:AdaBoost Classifier     0.5951663 0.4442134 0.7461192 5.139222e-13
AzureOpenAI:Linear SVM-FastText_Wiki:AdaBoost Classifier        0.6001098 0.4491569 0.7510627 5.139222e-13
AzureOpenAI:XGBoost Classifier-FastText_SBWC:AdaBoost Classifier 0.6044640 0.4535112 0.7554169 5.139222e-13
AzureOpenAI:XGBoost Classifier-Word2Vec_SBWC:SVM                0.6106737 0.4913350 0.7300125 5.139222e-13
AzureOpenAI:Linear SVM-FastText_SBWC:AdaBoost Classifier        0.6277083 0.4767555 0.7786612 5.139222e-13
AzureOpenAI:XGBoost Classifier-AzureOpenAI:SVM                  0.6557864 0.5364476 0.7751251 5.139222e-13
>
```

Anexo 3

visto bueno de la ANDJE para poder utilizar los datos en el proyecto aplicado

COLOMBIA POTENCIA DE LA VIDA

Defensa Jurídica del Estado

Al contestar por favor cite estos datos:
Radicado No. 20241030027591 - OAJ
Fecha: 27-03-2024 09:38

Bogotá D.C.,

Señor
JORGE MARIO CARRASCO
Bogotá D.C.
Correo electrónico: jmcarrascoo@unal.edu.co

Asunto: Respuesta a petición Nos. 20242400752172

Respetado señor:

Mediante escrito allegado el 5 de marzo de 2024 solicita le sean absueltos varios interrogantes referidos al mecanismo de extensión de jurisprudencia.

> En ejercicio del derecho de petición consagrado en el artículo 23 de la Constitución Política de Colombia y en la Ley 1755 de 2015, para efectos de solicitar información sobre radicación de la Agencia Nacional de Defensa jurídica para mi proyecto de grado titulado "Modelo predictivo para la gestión documental de los correos electrónicos de radicación para la Agencia Nacional de Defensa jurídica del Estado (ANDJE).".
>
> El objetivo de mi trabajo de grado es desarrollar un modelo de aprendizaje computacional que permita la clasificación eficiente y precisa de la gestión documental de las comunicaciones que son recibidas por notificación vía correos electrónicos en la ANDJE. Para lograr este objetivo requiero de la siguiente información en una muestra de radicados:

Variable	Descripción
Radicados	Número Único de Identificación en Sistema de Gestión documental en Orfeo
Asunto	Asunto en la radicación
Remitente	Persona que remite la comunicación
Radicador	Persona responsable en la radicación SGDEA
Fecha de Radicación	Fecha en que fue radicada la comunicación
Tipo de Documento	Tipología del documento a clasificar
Medio recepción	Medio en donde fue recibida la comunicación
Urls a incluir	Enlace permanente al documento de que trata la comunicación, si hay más de un anexo poner todos los anexos

En respuesta a su solicitud nos permitimos entregar la siguiente información:

Después de realizar la consulta en el Sistema Único de Gestión e Información de la Actividad Litigiosa del Estado – eKOGUI, en colaboración con el área de gestión documental de la ANDJE, se preparó una muestra de radicados junto con su clasificación, en donde se seleccionaron aleatoriamente 19.499 radicados, los cuales tiene la estructura requerida en su petición.

Dicha información se encuentra en el documento de Excel adjunto denominado anexo 20242400752172.

Es importante indicar que para acceder a los documentos relacionados en la columna Urls del archivo de Excel anteriormente mencionado se requiere procesarlos dentro de la infraestructura tecnológica de la Agencia.

Esto se realiza con e fin de evitar vulneraciones a la seguridad de la información contenida en estos radicados.

En los anteriores términos, damos respuesta a su solicitud, no sin antes comunicarle que estaremos prestos a brindar la información adicional que se requiera.

Cordialmente,

Firmado Electronicamente por:
JUAN CARLOS MENDEZ (JEFE-E)
No. Radicado: 20241030027591
Dependencia: OFICINA ASESORA JURIDICA - Jefe-E

Elaboró GRodríguez - Abogado OAJ
Anexo: lo enunciado