



CARRERA DE ESPECIALIZACIÓN EN INTELIGENCIA ARTIFICIAL

MEMORIA DEL TRABAJO FINAL

Procesamiento de contratos societarios

Autor:

Ing. Ezequiel Guinsburg

Director:

Dr. Luciano Del Corro (Microsoft Research)

Jurados:

Dr. Ing. María De Los Milagros Gutiérrez (UTN-FRSF)

Dr. Ing. Lucila Romero (UNL)

Ing. Juan Esteban Carrique (UNL)

*Este trabajo fue realizado en la Ciudad Autónoma de Buenos Aires,
entre marzo de 2022 y octubre de 2023.*

Resumen

En la presente memoria se describe la implementación de un sistema de inteligencia artificial para la validación de contratos societarios en español para la empresa MercadoLibre. Este desarrollo busca determinar, de manera automática, la veracidad de la documentación presentada por los nuevos usuarios que sean empresas jurídicas.

Para el trabajo se utilizaron las herramientas aprendidas en la carrera tales como modelos convolucionales, modelos secuenciales y herramientas para disponibilizar sistemas en servicios basados en la nube desarrollados en lenguajes de programación de alto nivel.

Índice general

Resumen	I
1. Introducción general	1
1.1. Validación de contratos societarios	1
1.2. Motivación	2
1.3. Estado del arte	3
1.3.1. Algoritmos y Modelos de NLP	4
Algoritmos de Reconocimiento y Extracción de Informa- ción (NER)	4
Modelos de Procesamiento del Lenguaje Natural Pre-entrenados	4
1.3.2. Tabla de algoritmos destacados	4
1.3.3. Preprocesamiento de las de imágenes	5
Procesamiento de imágenes mediante convoluciones	5
Preprocesamiento de imágenes escaneadas	5
Detección de regiones de interés por medio de algoritmos de Bounding Box	5
Integración de algoritmos NLP y convolucionales	5
1.3.4. Reconocimiento de texto en imágenes (OCR)	5
1.4. Alcance y objetivos	6
2. Introducción específica	7
2.1. Preprocesamiento de imágenes	7
2.1.1. Estandarización de formatos	7
Identificación del formato de origen	8
Conversión al formato intermedio	8
Compresión y optimización	8
Conversión al formato objetivo	8
2.1.2. Paginado de documentos	8
2.1.3. Rotación y corrección de ángulos en imágenes	8
2.2. Filtrado por <i>bounding box</i>	9
2.3. Algoritmos de reconocimiento óptico de caracteres	12
2.4. Name Entity Recognition	13
3. Diseño e implementación	15
3.1. Diagrama de flujo del sistema	15
3.2. Problemáticas presentadas	16
3.3. Consideraciones adoptadas	17
3.4. Implementación del sistema	17
3.4.1. Arquitectura de la solución	18
3.4.2. Módulo de conversión de imágenes	18
3.4.3. Módulo de procesamiento de imágenes	19
3.4.4. Módulo de conversión de imágenes a texto	19

3.4.5. Módulo de predicción NER	20
3.4.6. Módulo de clasificación binaria	21
4. Ensayos y resultados	23
4.1. Metodología de las pruebas	23
4.2. Resultados de modelos de procesamiento de imágenes	25
4.2.1. Consideraciones generales de los modelos	25
4.2.2. Consideraciones particulares de los modelos	25
4.2.3. Análisis de los resultados	28
4.3. Resultados de modelos de reconocimiento de entidades nombradas	29
4.4. Simulación del sistema completo	29
5. Conclusiones	31
5.1. Resultados obtenidos y cumplimiento de objetivos	31
5.2. Vínculo con la carrera	31
5.3. Oportunidades de mejora	32
Bibliografía	33

Índice de figuras

1.1. Captura de pantalla de la solicitud de documentación contractual para registro de nuevo usuario.	2
2.1. Ejemplo de documento contractual escaneado y luego alineado. . .	9
2.2. Ejemplo de extracción de texto útil mediante algoritmo de <i>bounding box</i>	11
2.3. Ejemplo de etiquetado de entidades en español.	14
3.1. Esquema de diagrama de flujo completo del sistema.	15
3.2. Arquitectura de la solución.	18
3.3. Arquitectura de capas del modelo VGG16.	19
3.4. Representación de la función <i>embed</i> en el <i>pipeline</i> de Spa.cy.	20
3.5. Representación de la función <i>encode</i> en el <i>pipeline</i> de Spa.cy.	21
3.6. Representación de la función <i>attend</i> en el <i>pipeline</i> de Spa.cy.	21
3.7. Representación de la función de predicción en el <i>pipeline</i> de Spa.cy.	21
3.8. Diagrama de flujo del clasificador binario.	22
4.1. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 1.	26
4.2. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 2.	26
4.3. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 6.	27
4.4. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 19.	28
4.5. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 19.	29

Índice de tablas

1.1. Algoritmos de imágenes y NLP	4
2.1. Algoritmos CNN más avanzados y sus características principales .	12

Capítulo 1

Introducción general

En este capítulo se presenta una breve introducción a la problemática que dio origen al trabajo, las motivaciones que lo precedieron, el estado del arte de la tecnología aplicada y, finalmente, el alcance propuesto.

1.1. Validación de contratos societarios

La validación de contratos societarios es una tarea compleja y de gran importancia en el ámbito legal y empresarial. Estos contratos contienen una variedad de formatos, cláusulas y términos específicos que deben ser revisados minuciosamente para garantizar su exactitud y cumplimiento legal. Sin embargo, el proceso de validación manual de contratos puede ser laborioso, propenso a errores humanos y consumir una cantidad significativa de tiempo y recursos.

Una problemática particular en la validación de contratos societarios en español es el idioma en sí. El español presenta una gran diversidad de términos y expresiones, y a menudo se utilizan variaciones regionales y terminología específica en diferentes países de habla hispana. Esto plantea desafíos adicionales al aplicar técnicas de inteligencia artificial y procesamiento de lenguaje natural para la validación automática de contratos.

La falta de modelos de inteligencia artificial y recursos específicos para la validación de contratos societarios en español también es un obstáculo significativo. La mayoría de los avances en este campo se han centrado en el inglés, lo que deja a un lado la necesidad de herramientas y modelos que sean efectivos y precisos en español. La adaptación y desarrollo de modelos de inteligencia artificial para procesar y validar contratos en español requiere esfuerzos adicionales para recopilar y etiquetar grandes cantidades de datos, así como para entrenar y afinar los modelos para el contexto y las particularidades legales del español.

Otro desafío es la interpretación y comprensión de las cláusulas y disposiciones contractuales en su contexto. Los contratos societarios a menudo contienen terminología legal compleja y estructuras sintácticas específicas que pueden ser difíciles de analizar y entender correctamente. Los modelos de inteligencia artificial deben ser capaces de capturar la intención y el significado preciso de las cláusulas y detectar posibles errores o inconsistencias, lo que implica un nivel avanzado de comprensión del lenguaje y conocimiento legal.

Además, la confidencialidad y seguridad de la información contenida en los contratos societarios son aspectos críticos a considerar al aplicar modelos de inteligencia artificial. La protección de datos sensibles y confidenciales es esencial,

especialmente en el ámbito legal, donde la privacidad y la confidencialidad son fundamentales. Se deben implementar medidas adecuadas para garantizar la seguridad de los contratos y la información relacionada durante el proceso de validación automatizada. En el caso puntual de este trabajo, la principal dificultad radicó en conseguir una cantidad razonable de contratos reales para llevar a cabo el entrenamiento de los modelos, justamente por la naturaleza confidencial de este tipo de información.

En conclusión, la validación de contratos societarios en español utilizando modelos de inteligencia artificial plantea desafíos específicos relacionados con la diversidad del idioma, la falta de recursos específicos, la interpretación contextual y la seguridad de los datos. En el trabajo realizado se utilizaron los recursos de última tecnología disponibles en búsqueda de los mejores resultados que se podían obtener en este campo.

1.2. Motivación

La motivación principal detrás de este trabajo es la necesidad de mejorar y agilizar el proceso de validación de contratos societarios en el contexto de los nuevos usuarios. Actualmente, el cliente lleva a cabo esta validación de manera manual, lo que requiere una cantidad significativa de tiempo y recursos humanos.

El cliente, la empresa MercadoLibre [1], recibe y registra una gran cantidad de contratos societarios de nuevos usuarios, que son personas jurídicas, en su plataforma diariamente. Estos contratos, junto con la información personal y los datos proporcionados por los usuarios en una planilla de datos, deben ser verificados y validados para garantizar la coherencia y precisión de la información.

La carga de documentación es requerida por la aplicación de la empresa como parte del proceso de registro de empresas jurídicas, como puede observarse en la figura 1.1.

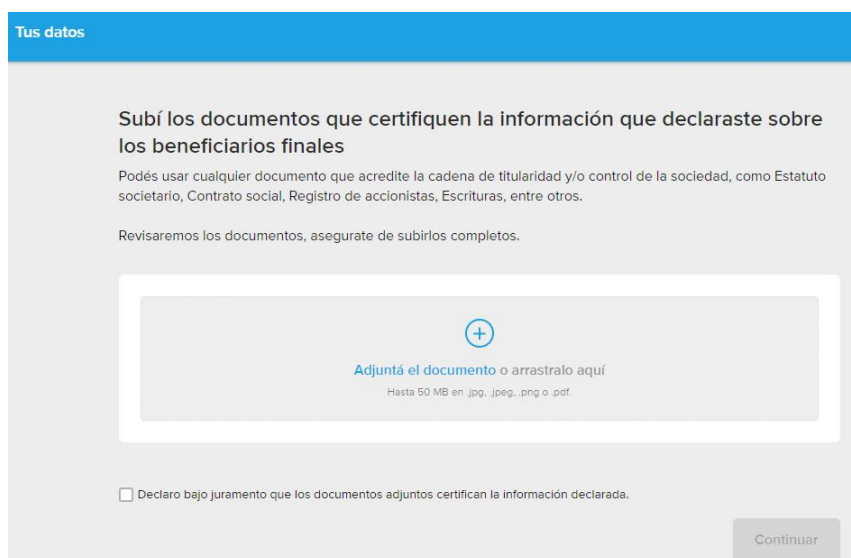
La imagen muestra una interfaz de usuario con un encabezado azul que dice "Tus datos". El contenido principal es un formulario para subir documentos. El título es "Subí los documentos que certifiquen la información que declaraste sobre los beneficiarios finales". Debajo, se explica: "Podés usar cualquier documento que acredite la cadena de titularidad y/o control de la sociedad, como Estatuto societario, Contrato social, Registro de accionistas, Escrituras, entre otros." Se añade la instrucción: "Revisaremos los documentos, asegurate de subirlos completos." En el centro hay un área de carga de documentos con un icono de "+" y el texto "Adjuntá el documento o arrástralo aquí" y "Hasta 50 MB en .jpg, .jpeg, .png o .pdf". Al final, hay un checkbox con el texto "Declaro bajo juramento que los documentos adjuntos certifican la información declarada." y un botón "Continuar".

FIGURA 1.1. Captura de pantalla de la solicitud de documentación contractual para registro de nuevo usuario.

La comparación manual de los contratos societarios con los datos cargados en la planilla es un proceso propenso a errores y laborioso. Dado que la cantidad de usuarios y contratos aumenta constantemente, la carga de trabajo se vuelve cada vez más intensa y puede resultar ineficiente a largo plazo.

La idea central de este trabajo es utilizar la automatización y la tecnología de procesamiento de lenguaje natural para comparar y validar automáticamente los contratos societarios con los datos proporcionados en la planilla de usuarios. Al hacerlo, se pretende optimizar el proceso de validación, reducir errores y agilizar el flujo de trabajo.

La automatización de este proceso de validación ofrece diversas ventajas. En primer lugar, la comparación automatizada permite detectar discrepancias y errores de manera más rápida y precisa que la revisión manual. Los algoritmos y modelos de inteligencia artificial pueden identificar inconsistencias en el contenido de los contratos y la planilla, así como realizar verificaciones de coherencia y validez de manera más eficiente.

Además, la automatización liberaría recursos humanos que actualmente se dedican a tareas repetitivas y de baja complejidad. Los equipos encargados de la validación podrían enfocarse en actividades más estratégicas y de mayor valor agregado, lo que aumentaría la eficiencia y productividad en general.

Por último, la implementación de una solución automatizada de validación de contratos societarios fortalecería la seguridad y confidencialidad de los datos. Al minimizar la intervención humana en el proceso, se reduciría el riesgo de errores y filtraciones de información sensible, lo que es especialmente relevante en el ámbito legal y de protección de datos.

En resumen, la motivación detrás de este trabajo radica en mejorar la eficiencia, precisión y seguridad en la validación de contratos societarios. La automatización del proceso mediante la comparación automatizada de los contratos y los datos de los usuarios en la planilla ofrece la oportunidad de optimizar la labor, liberar recursos humanos y mejorar la experiencia general del cliente al agilizar los procedimientos de registro y validación.

1.3. Estado del arte

En esta sección, se presentará el estado actual de la tecnología utilizada para la validación de documentos contractuales en español. Se explorarán los algoritmos, modelos y sistemas existentes que han sido desarrollados para abordar esta problemática, así como sus fortalezas y limitaciones.

Además de las técnicas de procesamiento de lenguaje natural (NLP, por sus siglas en inglés) [2], el procesamiento de imágenes [3] también desempeña un papel importante en la validación de contratos societarios. Los contratos a menudo se escanean y se convierten en imágenes digitales, lo que requiere algoritmos específicos para extraer y analizar la información contenida en ellas. En este sentido, los algoritmos convolucionales han demostrado ser eficaces en el procesamiento de imágenes escaneadas de contratos.

1.3.1. Algoritmos y Modelos de NLP

El procesamiento de lenguaje natural juega un papel fundamental en la validación de documentos contractuales en español. Existen diversos algoritmos y modelos de NLP que se han utilizado con éxito en este ámbito. Entre ellos, se destacan:

Algoritmos de Reconocimiento y Extracción de Información (NER)

Los algoritmos de reconocimiento y extracción de información permiten identificar entidades relevantes en los contratos, como nombres, fechas, cláusulas específicas, entre otros elementos. Algunos enfoques populares incluyen algoritmos basados en reglas, aprendizaje supervisado y técnicas de aprendizaje profundo como las redes neuronales recurrentes (RNN) y los Transformers [4].

Modelos de Procesamiento del Lenguaje Natural Pre-entrenados

Los modelos de lenguaje pre-entrenados, como BERT (Bidirectional Encoder Representations from Transformers) y GPT (Generative Pre-trained Transformer), han demostrado ser eficaces en tareas de NLP, incluyendo la validación de contratos. Estos modelos capturan la estructura y el significado del lenguaje, lo que facilita la detección de errores y la clasificación de cláusulas.

1.3.2. Tabla de algoritmos destacados

Los algoritmos más importantes de procesamiento de imágenes y NLP (Natural Language Processing) que se consideran del estado del arte y tenidos en cuenta en el desarrollo del presente trabajo, se muestran en la tabla 1.1:

TABLA 1.1. Algoritmos de imágenes y NLP

Algoritmo	Tipo	Año de desarrollo
AlexNet	Imagen	2.012
VGGNet	Imagen	2.014
ResNet	Imagen	2.015
DenseNet	Imagen	2.016
MobileNet	Imagen	2.017
EfficientNet	Imagen	2.019
Xception	Imagen	2.016
SqueezeNet	Imagen	2.016
NASNet	Imagen	2.017
LeNet	Imagen	1.998
Bidirectional LSTM-CRF	NLP	1.990
BERT (Transformers)	NLP	2.018
ELMO (Embeddings from Language Models)	NLP	2.018
GPT-3.5 (Generative Pre-trained Transformer)	NLP	2.022
RoBERTa	NLP	2.019
XLNet-RoBERTa	NLP	2.019
Flair	NLP	2.019
SpanNER	NLP	2.019

Cabe destacar que este listado representa solo una selección de algoritmos considerados actuales en el momento de redactar la presente memoria.

1.3.3. Preprocesamiento de las de imágenes

Los documentos originales cargados por el usuario nuevo suelen tener distintos formatos y problemáticas propias de las imágenes escaneadas. Por otro lado, la documentación contractual es muy habitual que cuente con sellos, firmas y otra información que es necesario filtrar para lograr mejores resultados en el traspaso a texto de dichas imágenes.

Procesamiento de imágenes mediante convoluciones

Estos algoritmos, también conocidos como redes neuronales convolucionales (CNN), son ampliamente utilizados en tareas de visión por computadora y reconocimiento de patrones. Estos algoritmos se basan en la idea de aplicar filtros convolucionales a la imagen para extraer características relevantes. En el contexto de los contratos, pueden utilizarse para tareas como la detección de cláusulas, la identificación de campos clave y la extracción de información relevante.

Preprocesamiento de imágenes escaneadas

Antes de aplicar algoritmos convolucionales, es necesario realizar un preprocesamiento de las imágenes escaneadas de los contratos. Esto implica la limpieza de ruido, la corrección de perspectiva, el ajuste de contraste y brillo, entre otros pasos. El preprocesamiento adecuado ayuda a mejorar la calidad de las imágenes y facilita la extracción precisa de la información mediante algoritmos convolucionales.

Detección de regiones de interés por medio de algoritmos de Bounding Box

La utilización de algoritmos de Bounding Box (caja delimitadora) es una técnica común para identificar y extraer la parte de texto de las imágenes escaneadas de contratos que contienen firmas, sellos y otros caracteres propios de ese tipo de documentos. Esta técnica se basa en el concepto de delimitar una región rectangular, dentro de la cual se encuentra el texto relevante.

Integración de algoritmos NLP y convolucionales

La combinación de algoritmos NLP y convolucionales puede potenciar la validación de contratos societarios, ya que se pueden aprovechar tanto la información textual como visual. Al integrar técnicas de procesamiento de lenguaje natural y procesamiento de imágenes, se pueden obtener resultados más precisos y completos en la validación de contratos.

1.3.4. Reconocimiento de texto en imágenes (OCR)

El reconocimiento óptico de caracteres (OCR)[5] es una técnica que permite convertir el texto contenido en las imágenes escaneadas en texto digital editable. Los algoritmos convolucionales se pueden utilizar en combinación con técnicas de OCR para extraer el texto de los contratos y realizar su validación mediante técnicas de NLP.

1.4. Alcance y objetivos

El presente proyecto se ha enfocado en el desarrollo de un sistema que permita la validación de documentos legales de nuevos usuarios de personas jurídicas. El sistema procesa los documentos de entrada y emite una salida binaria indicando el cumplimiento o no de las consignas propuestas, junto con la probabilidad asociada a dicho resultado mediante el uso de una capa de salida Softmax.

Existen dos objetivos principales que no se perdieron de vista para este trabajo:

- El desarrollo de una plataforma que sea de utilidad para el cliente, a partir de la cual pueda continuar con el desarrollo de un sistema completo.
- El aprendizaje del alumno en áreas como:
 - Planificación del proyecto.
 - Modelos de inteligencia artificial convolucionales.
 - Modelos de inteligencia artificial de procesamiento de lenguaje natural.
 - Disponibilización de sistemas de alto nivel.

Capítulo 2

Introducción específica

A lo largo de esta sección se exploran las metodologías, algoritmos y técnicas utilizados en cada etapa del preprocesamiento de imágenes, filtrado por *bounding box*, reconocimiento óptico de imágenes y reconocimiento de entidades nombradas. Se analizan las técnicas más recientes basadas en el aprendizaje automático y la inteligencia artificial. El objetivo final es establecer una base teórica para el desarrollo del sistema propuesto en esta memoria, que permitirá mejorar la comprensión de los capítulos subsiguientes donde se abordan estas mismas temáticas de manera específica.

2.1. Preprocesamiento de imágenes

Esta técnica desempeña un papel crucial en la etapa inicial del procesamiento de documentos, ya que tiene como objetivo mejorar la calidad y la legibilidad de las imágenes. Esto es especialmente relevante en el contexto de la aplicación del reconocimiento óptico de caracteres (OCR, por sus siglas en inglés), donde la calidad del escaneo influye directamente en la precisión y eficacia del reconocimiento de texto.

Al digitalizar documentos con diferentes tipos de dispositivos, como escáneres de escritorio o cámaras de teléfonos celulares, es común encontrarse con variaciones en los resultados, como por ejemplo diferencias en la resolución, la iluminación, el enfoque y otros aspectos técnicos. Además, al ser escaneadas por diferentes personas, surgen discrepancias en la forma en que se capturan las imágenes, lo que puede afectar la uniformidad y la legibilidad de los documentos.

El preprocesamiento busca abordar estos desafíos al aplicar técnicas y algoritmos que mejoren la calidad y la uniformidad de las imágenes escaneadas. Esto implica realizar acciones como la estandarización del formato, la segmentación por hojas, el ajuste de contraste y la corrección de ángulo entre otros procesos. Al contar con imágenes correctamente preprocesadas, se optimiza el rendimiento del OCR y se obtiene una mayor precisión en la extracción de texto y la posterior indexación y búsqueda de la información contenida en los documentos.

2.1.1. Estandarización de formatos

El objetivo principal de esta etapa es convertir los archivos de imagen capturados por los usuarios en diferentes formatos, como TIFF, BMP, PDF, entre otros, a un formato común y ampliamente compatible, como JPG.

A continuación, se presenta una descripción técnica de cómo se logra esta estandarización a nivel de procesamiento de imágenes:

Identificación del formato de origen

En primer lugar, se debe determinar el formato de imagen de origen del archivo escaneado cargado por el usuario. Esto se logra mediante la inspección de los metadatos del archivo o mediante el análisis de los bytes iniciales del archivo que pueden contener una firma distintiva del formato.

Conversión al formato intermedio

Una vez identificado el formato de origen, se lleva a cabo la conversión al formato intermedio. Por ejemplo, si el formato de origen es TIFF, se utiliza un algoritmo de procesamiento de imágenes para decodificar el archivo TIFF y obtener una representación de la imagen en un formato intermedio, como una matriz de píxeles.

Compresión y optimización

En esta etapa, se aplican técnicas de compresión y optimización a la imagen en el formato intermedio. Esto puede implicar el uso de algoritmos de compresión como JPEG, que reducen el tamaño del archivo sin perder una cantidad significativa de calidad visual. Estas técnicas permiten mantener la legibilidad de los documentos escaneados mientras se reduce el tamaño del archivo resultante.

Conversión al formato objetivo

Para finalizar, se realiza la conversión de la imagen del formato intermedio al formato objetivo, en este caso, JPG. Este proceso implica el uso de un algoritmo de compresión diseñado específicamente para el formato seleccionado. El algoritmo codifica la imagen en una secuencia de bytes que cumple con el estándar JPG. Como resultado, se obtiene un archivo de imagen en el formato deseado, listo para ser procesado y almacenado posteriormente.

2.1.2. Paginado de documentos

El contrato puede estar contenido en su totalidad en un sólo archivo en formato PDF. En este caso se agrega un paso previo a los descriptos en la subsección anterior que consiste en la separación de páginas en archivos individuales: cada página convertida a formato JPG se guarda como un archivo independiente. Se asigna un nombre de archivo único para cada página, asegurando la correspondencia con la página original del PDF. El paginado garantiza que cada página pueda ser procesada y analizada de forma individualizada durante el resto del flujo de trabajo.

2.1.3. Rotación y corrección de ángulos en imágenes

La alineación horizontal del texto es un factor crítico para lograr una buena eficiencia en un filtro de *bounding box*, o caja delimitadora. Dicho filtro consta de un rectángulo que encierra un objeto o región de interés en una imagen, en este caso, el texto.

Cuando el texto se encuentra perfectamente horizontal, es decir, alineado con el eje horizontal de la imagen, facilita la tarea de detección y extracción mediante *bounding boxes*. Esto se debe a que los algoritmos de procesamiento de imágenes y reconocimiento de patrones utilizados en el filtro están diseñados para trabajar con objetos que se encuentran en posiciones horizontales o verticales.

Si el texto presenta una inclinación o desalineación significativa, es más probable que se genere una caja delimitadora incorrecta o que se omita parte del texto durante el proceso de detección. Esto puede resultar en una pérdida de información o en una disminución en la precisión y eficiencia del filtro.

La herramienta de detección y reconocimiento de texto emplea algoritmos avanzados de procesamiento de imágenes y aprendizaje automático para identificar las regiones que contienen texto. Una vez que se han detectado estas regiones, se analizan en detalle para determinar el ángulo de inclinación o desalineación del texto.

En muchos casos, como en el presente trabajo, se utilizan algoritmos de OCR como parte de esta herramienta. Estos algoritmos permiten identificar los caracteres individuales en el texto y analizar su disposición espacial y orientación. Basándose en esta información, la herramienta puede calcular el ángulo de inclinación del texto.

Una vez que se ha determinado el ángulo de inclinación, se puede aplicar un proceso de corrección para enderezar el texto. Esto implica rotar la imagen o ajustar la perspectiva de manera que el texto quede perfectamente alineado horizontalmente.

En la figura 2.1 se ejemplifica con un documento escaneado que luego del procesamiento el texto se ubica de manera horizontal.

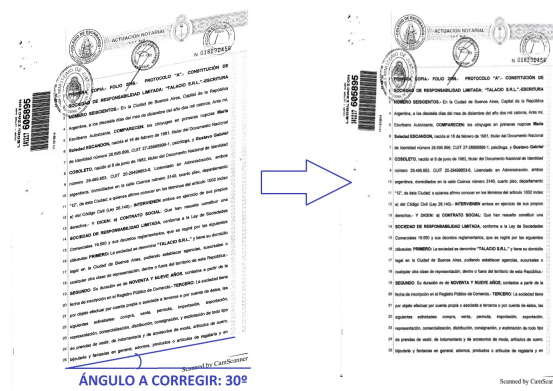


FIGURA 2.1. Ejemplo de documento contractual escaneado y luego alineado.

2.2. Filtrado por *bounding box*

Un algoritmo de redes neuronales convolucionales (CNN, por sus siglas en inglés) para *bounding box* se basa en la arquitectura y principios de funcionamiento de las CNN, que están diseñadas específicamente para el procesamiento de imágenes. En esta sección se presenta una explicación del funcionamiento de este tipo de algoritmo.

En primer lugar, la CNN realiza una etapa de extracción de características. Esto implica pasar la imagen de entrada a través de una serie de capas convolucionales y de *pooling*. Las capas convolucionales aplican filtros a pequeñas regiones de la imagen para extraer características visuales, mientras que las capas de *pooling* reducen la dimensionalidad de la información.

A medida que la imagen se procesa en las capas convolucionales, se generan múltiples mapas de características que representan diferentes aspectos visuales de la imagen. Cada mapa de características captura patrones específicos, como bordes, texturas o formas.

Después de la extracción de características, se utilizan capas totalmente conectadas para detectar objetos en la imagen. Estas capas reciben como entrada los mapas de características y aprenden a reconocer patrones que corresponden a objetos específicos. La red neuronal utiliza la información de los mapas de características para realizar la detección y localización de objetos en la imagen.

Una vez que se ha detectado un objeto en la imagen, se genera un *bounding box* alrededor de él. Esto se logra determinando las cuatro coordenadas de los vértices que definen la ubicación y el tamaño del objeto dentro de la imagen. El algoritmo utiliza los resultados de la detección de objetos y aplica cálculos para obtener dichas coordenadas.

Para mejorar la precisión, es común aplicar técnicas de ajuste y refinamiento. Se pueden aplicar técnicas de supresión de no máximos para eliminar detecciones redundantes y mantener solo las más relevantes. En el caso que nos compete se fijó que habría una sola caja delimitadora por página, con lo cual se suprime el problema de la redundancia.

El algoritmo de *bounding box* basado en CNN se entrena utilizando conjuntos de datos etiquetados que contienen imágenes de entrenamiento y las correspondientes detecciones. Durante este proceso, se ajustan los pesos y los parámetros de la CNN para optimizar la detección y precisión de las coordenadas. Esto se logra utilizando técnicas de optimización, como la retropropagación del error y la actualización de los pesos mediante algoritmos de descenso del gradiente.

Para mejorar el rendimiento en el entrenamiento del modelo suelen utilizarse dos técnicas que implican contar con pesos preentrenados:

- *Transfer learning*: la técnica de utilizar pesos de modelos de redes neuronales preentrenados se conoce como transferencia de aprendizaje o transferencia de conocimiento. Esta técnica se utiliza para mejorar la performance y acelerar el entrenamiento de nuevos modelos de redes neuronales. La transferencia de aprendizaje aprovecha el hecho de que los modelos de redes neuronales preentrenados han sido entrenados en conjuntos de datos masivos y representan conocimientos y características generales del dominio de datos en el que fueron entrenados. Estos modelos han aprendido a reconocer patrones y características relevantes en los datos, lo que los convierte en una excelente base para tareas similares.
- *Fine-tuning*: en esta estrategia, se congela parte del modelo preentrenado y solo se permiten actualizaciones en las capas finales o en las capas específicas relacionadas con la tarea específica. Esto permite que el modelo se adapte a los datos específicos del nuevo dominio mientras retiene los conocimientos generales aprendidos previamente.

Finalmente, una vez entrenado el modelo, se aplica a imágenes de entrada obteniendo las cuatro coordenadas que delimitan el *bounding box*, dejando dentro, idealmente, sólo el texto de interés y por fuera los sellos, firmas y los grafismos y simbología propia de la documentación contractual.

A continuación, en la figura 2.2 se ejemplifica con una imagen contractual el efecto logrado por el algoritmo de *bounding box* de manera ideal, recortando únicamente el texto de interés.

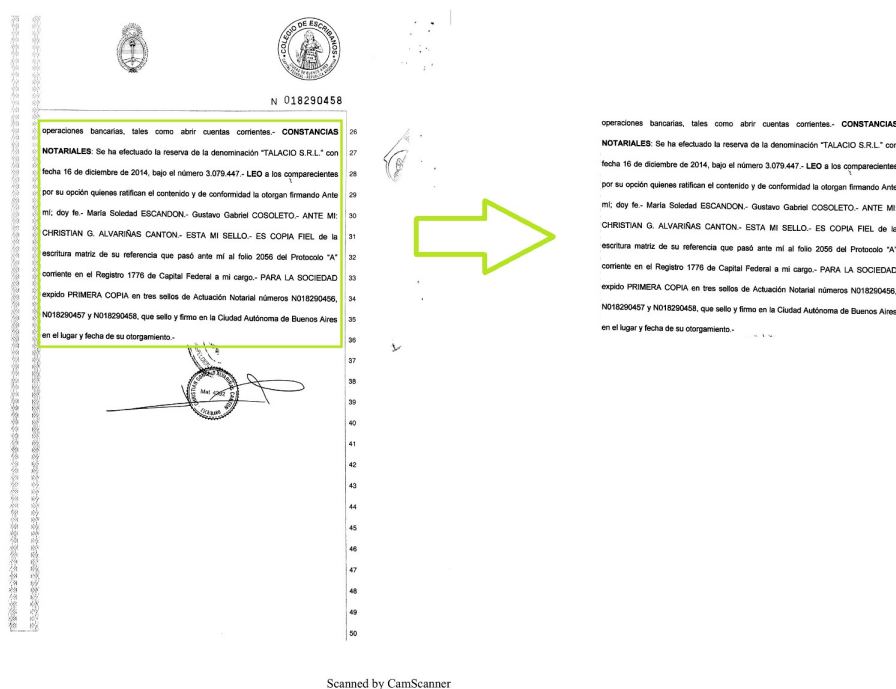


FIGURA 2.2. Ejemplo de extracción de texto útil mediante algoritmo de *bounding box*.

Los algoritmos de redes neuronales convolucionales han experimentado un desarrollo significativo a lo largo de los años. Desde la introducción de la propagación hacia atrás en la década de 1980 hasta el resurgimiento de las CNN en 2012, se han realizado avances en las capas convolucionales, la arquitectura de la red y las técnicas de transferencia de aprendizaje. Estos avances han llevado a una mejora en la precisión y la capacidad de representación de las CNN, y han permitido su aplicación en una amplia gama de tareas de visión por computadora, como clasificación de imágenes, detección de objetos y segmentación semántica.

La tabla 2.1 muestra los algoritmos que conforman el estado del arte con sus principales características:

TABLA 2.1. Algoritmos CNN más avanzados y sus características principales.

Algoritmo	Año de desarrollo	Características principales
AlexNet	2012	Arquitectura profunda con capas convolucionales y de <i>pooling</i> , uso de la función de activación ReLU.
VGGNet	2014	Estructura de red profunda con capas convolucionales de tamaño fijo, enfoque en la simplicidad y la profundidad.
GoogLeNet	2014	Utiliza módulos Inception que combinan diferentes tamaños de filtros convolucionales para capturar características.
ResNet	2015	Arquitectura profunda con conexiones residuales que permiten el entrenamiento de redes muy profundas.
DenseNet	2016	Conexiones densas entre capas que promueven el flujo de información y la reutilización de características.
Xception	2017	Uso de operaciones de convolución separable en lugar de convoluciones estándar para mejorar la eficiencia.
MobileNet	2017	Diseñado para aplicaciones en dispositivos móviles, utiliza capas convolucionales ligeras llamadas MobileNets.
EfficientNet	2019	Emplea un enfoque de escalado compuesto para optimizar el rendimiento de la red en términos de eficiencia y precisión.

2.3. Algoritmos de reconocimiento óptico de caracteres

El reconocimiento óptico de caracteres es una técnica que permite la extracción automática de texto contenido en imágenes o documentos escaneados. Esta tecnología es ampliamente utilizada en aplicaciones de procesamiento de documentos, digitalización de archivos y reconocimiento de texto en imágenes.

El OCR se basa en algoritmos y modelos entrenados para identificar y reconocer patrones de texto en imágenes. El objetivo principal de un algoritmo de OCR es convertir el texto contenido en una imagen en texto digital editable, lo que permite su procesamiento y análisis posterior.

La librería Tesseract [6] es un motor de OCR desarrollado por la comunidad de código abierto altamente preciso y eficiente que se actualiza y mejora continuamente. Utiliza un enfoque basado en aprendizaje automático para el reconocimiento de caracteres. Utiliza modelos de lenguaje y redes neuronales para identificar y clasificar los caracteres en una imagen. Este tipo de librerías pueden reconocer una amplia variedad de fuentes, tamaños y estilos de texto, incluyendo texto impreso, manuscrito y tipografías especializadas.

Una de las características destacadas de los motores OCR más avanzados es su capacidad para trabajar con múltiples idiomas, lo que los convierte en una herramienta adecuada para los fines de este trabajo. Además, ofrecen opciones de configuración y personalización para adaptarse a requisitos específicos de calidad y precisión en la extracción de texto.

Los motores de reconocimiento óptico de caracteres como el utilizado en este proyecto siguen una secuencia de procesamiento que se resume a continuación:

1. Preprocesamiento de la imagen: el motor de OCR comienza realizando un preprocesamiento de la imagen para mejorar la calidad y facilitar la extracción del texto.
2. Segmentación de caracteres: se realiza la segmentación de la imagen para identificar las regiones que contienen caracteres individuales.
3. Extracción de características: una vez que se han identificado las regiones de caracteres, el algoritmo extrae características de cada uno.
4. Reconocimiento de caracteres: el siguiente paso es el reconocimiento propiamente dicho, donde se asigna una etiqueta o clasificación a cada carácter.
5. Corrección y postprocesamiento: una vez que los caracteres han sido reconocidos, el motor OCR realiza un proceso de corrección y postprocesamiento para mejorar la precisión y coherencia del texto extraído.

Es importante destacar que los motores avanzados como Tesseract utilizan modelos de lenguaje y diccionarios para ayudar en el proceso de reconocimiento y corrección. Estos modelos contienen información sobre la estructura y las probabilidades de ocurrencia de las palabras en un idioma específico, lo que contribuye a la precisión y coherencia del texto reconocido.

2.4. Name Entity Recognition

Un algoritmo de Name Entity Recognition (NER) es una técnica utilizada en el procesamiento de lenguaje natural para identificar y clasificar entidades mencionadas en un texto, como nombres de personas, organizaciones, ubicaciones, fechas, cantidades, entre otros.

Los algoritmos de NER más avanzados utilizan modelos basados en algoritmos Transformers, como BERT (Bidirectional Encoder Representations from Transformers) o GPT (Generative Pre-trained Transformer), referidos en la tabla 1.1. Estos algoritmos han demostrado ser altamente efectivos en las tareas de procesamiento de lenguaje natural.

Los algoritmos de NER basados en Transformers aprovechan las capacidades de atención y transformación de la arquitectura Transformer para capturar relaciones contextuales y realizar una detección precisa de entidades en un texto. Estos modelos se entrenan en grandes conjuntos de datos anotados y pueden adaptarse a diferentes dominios e idiomas mediante el ajuste fino y la transferencia de conocimiento.

El funcionamiento de un algoritmo de NER basado en Transformers se puede resumir en los siguientes pasos:

- Preprocesamiento del texto: El texto de entrada se divide en unidades más pequeñas y se les asigna una representación numérica. Se aplican técnicas de normalización y tokenización.
- Codificación de palabras y posicionamiento: Las palabras y subpalabras se codifican utilizando técnicas de incrustación y se añade información de posicionamiento.
- Arquitectura Transformer: El modelo consta de capas de atención y transformación que capturan las dependencias y relaciones a largo plazo.
- Atención contextualizada: Se utiliza la atención contextualizada para analizar las dependencias entre las unidades de texto y capturar el contexto de las entidades.
- Etiquetado de entidades: Se aplica una capa de clasificación para asignar etiquetas a las unidades de texto y determinar las entidades específicas.

Para lograr una mejor performance sin la necesidad de contar con una gran cantidad de palabras etiquetadas se utilizan las técnicas de *transfer learning* y *fine-tuning*, explicadas en la sección 2.2

En la figura 4.5 se observa un ejemplo de etiquetado utilizando la aplicación Prodi.gy [7], del grupo Explosion.ia:

Orden FOM/2427/2012 LEGAL , de 29 de octubre TIME , por la
empresas navieras para percibir las correspondientes bonificaciones al
Autónomas de Canarias LOC y de las Illes Balears LOC .
El Real Decreto 1316/2001 LEGAL , de 30 de noviembre TIME
transporte aéreo y marítimo para los residentes en las Comunidades Au
Ciudades de Ceuta LOC y Melilla LOC , establece en su dispo:

FIGURA 2.3. Ejemplo de etiquetado de entidades en español.

En la figura del ejemplo anterior se distinguen tres tipos de entidades que se etiquetan con distintos colores: artículos legales (gris), localidades (naranja) y fechas (cyan).

Capítulo 3

Diseño e implementación

Este capítulo se centra en el diseño e implementación del trabajo, desglosando cada fase clave que condujo a la creación del sistema. Inicialmente, se explora el diagrama de flujo completo, una representación gráfica esencial que traza el camino operativo y revela la interconexión entre los componentes fundamentales. Luego, se analizan las problemáticas presentadas en el desarrollo del proyecto, las cuales se canalizan mediante la adopción de consideraciones de mitigación necesarias. Finalmente, se describe de manera modular cada una de las herramientas de implementación del sistema.

3.1. Diagrama de flujo del sistema

En esta sección, se detalla la totalidad del sistema utilizando un esquema de flujo. A través de la referencia a la figura 3.1, se observa cómo cada paso del flujo desencadena una serie de procesos interrelacionados.

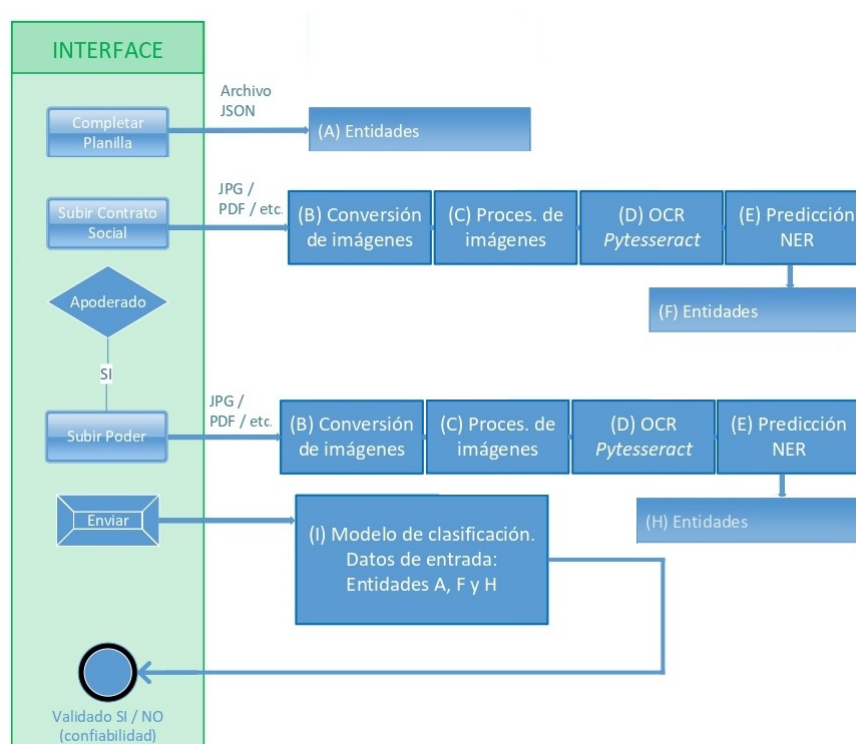


FIGURA 3.1. Esquema de diagrama de flujo completo del sistema.

El esquema sigue la secuencia que experimenta un nuevo usuario al ingresar los datos y los documentos al sistema para su aprobación. Inicialmente, debe completar una planilla y proporcionar la información básica de la persona jurídica que representa y del representante legal de la misma.

A continuación debe cargar en la aplicación el contrato social. Una vez tomada esa acción, el sistema sigue la secuencia que describe la figura 3.1: convierte las imágenes a un formato estandarizado, las recorta y luego las preprocesa de manera de optimizar el algoritmo de OCR. Una vez que se cuenta con las imágenes convertidas a texto se realiza la predicción NER, obteniendo como salida un archivo del tipo JSON que contiene las entidades comparables con las ingresadas por el usuario manualmente.

En el caso de ser la persona física que representa a la sociedad, un apoderado legal, se le requiere subir el poder que acredita dicha relación. El proceso de interpretación de las imágenes y conversión a entidades comparables es similar al descrito en el párrafo anterior para los contratos.

Finalmente, se someten las entidades extraídas junto con las ingresadas manualmente a una comparación mediante un modelo de clasificación. La salida binaria de este modelo señala la coherencia de los datos, validando así la incorporación del nuevo usuario al sistema.

3.2. Problemáticas presentadas

Durante el desarrollo de este trabajo, se identificaron diversas problemáticas que abarcan tanto aspectos técnicos como de gestión. Estas dificultades presentaron desafíos significativos que requirieron un enfoque estratégico y adaptativo para su resolución. Las problemáticas se listan a continuación:

1. Variedad en el formato y contenido de documentos: un obstáculo técnico importante surgió debido a la diversidad de formatos y contenidos en los documentos proporcionados por los usuarios para su procesamiento. La falta de uniformidad en los archivos añadió complejidad al diseño del sistema, ya que se requerían enfoques flexibles de procesamiento para garantizar la precisión y eficacia en la extracción de información.
2. Integración de sistemas de inteligencia artificial múltiples: la incorporación de sistemas de redes neuronales de distinta naturaleza, como son las convolucionales para el procesamiento de imágenes y las de procesamiento de lenguaje natural planteó un reto adicional. La interacción de múltiples procesos en un sistema completo exigió un diseño meticuloso para lograr una integración coherente y eficiente. La complejidad inherente a la combinación de estas tecnologías requirió un enfoque cuidadoso en la arquitectura y el flujo de trabajo.
3. Procesamiento de datos jurídicos en español: una dificultad técnica importante radicó en la necesidad de procesar datos en español de naturaleza jurídica, lo cual representa un nicho reducido en el que actualmente existen limitadas redes neuronales pre-entrenadas disponibles para su utilización. Esta escasez de recursos preexistentes demandó estrategias específicas para el entrenamiento y adaptación de modelos de procesamiento de lenguaje natural a esta área especializada.

4. Obtención de datos de entrenamiento de naturaleza confidencial: las dificultades de gestión más significativas, en términos de resultados, residen en las restricciones legales y contractuales para la obtención de datos societarios esenciales para el entrenamiento de los modelos. La colaboración con el cliente, MercadoLibre, implicó la necesidad de equilibrar los objetivos del proyecto con las limitaciones impuestas por los acuerdos contractuales y las políticas de privacidad.

3.3. Consideraciones adoptadas

Durante el desarrollo de este proyecto, se implementaron estrategias y enfoques específicos para abordar las problemáticas técnicas y de gestión previamente identificadas. Se listan a continuación las acciones implementadas para mitigar estas dificultades en el mismo orden en el que fueron descritas en la sección anterior:

1. Variedad en el formato y contenido de documentos: para abordar la variedad en el formato y contenido de los documentos cargados por el nuevo usuario, se diseñaron algoritmos de procesamiento flexibles capaces de adaptarse a diferentes estructuras y estilos de archivos. Como resultado se obtienen documentos con formato estandarizado que luego se preprocesan para lograr uniformidad previo a la conversión a texto mediante OCR.
2. Integración de sistemas de inteligencia artificial múltiples: dicha integración se logró mediante la creación de una arquitectura modular que facilitó la interacción fluida entre los diferentes componentes. Además, se utilizó tecnología comercial basada en la nube de manera de alojar todos los procesos en el mismo entorno.
3. Procesamiento de datos jurídicos en español: la limitada disponibilidad de modelos pre-entrenados para el procesamiento de datos jurídicos en español se abordó mediante la adaptación de modelos existentes a través de técnicas de transferencia de aprendizaje y fine-tuning. Se investigaron las bases de datos de modelos pre-entrenados que forman parte del estado del arte y se encontró, dentro de la plataforma SpaCy [8], matrices de *embeddings* y modelos NER con sus pesos pre-entrenados en español.
4. Obtención de datos de entrenamiento de naturaleza confidencial: como parte de los acuerdos con el cliente en los cambios de alcance durante el desarrollo del trabajo, se convino relegar la performance del sistema por no contar con información esencial para llevar el proyecto a nivel operativo.

3.4. Implementación del sistema

En la presente sección se detalla la implementación del sistema objeto de esta memoria. Inicialmente se presenta un diagrama de arquitectura que proporciona una visión general de donde están alojados y cómo están conectados los diferentes componentes. Luego, se profundiza en cada uno de los módulos que componen el programa, destacando sus funciones y roles dentro de la aplicación.

3.4.1. Arquitectura de la solución

Dentro de las múltiples posibilidades que ofrece la tecnología actual, se optó por implementar el sistema en la plataforma de Google Cloud, abarcando las funcionalidades de alojamiento, procesamiento y disponibilización global de la solución como se observa en la figura 3.2:

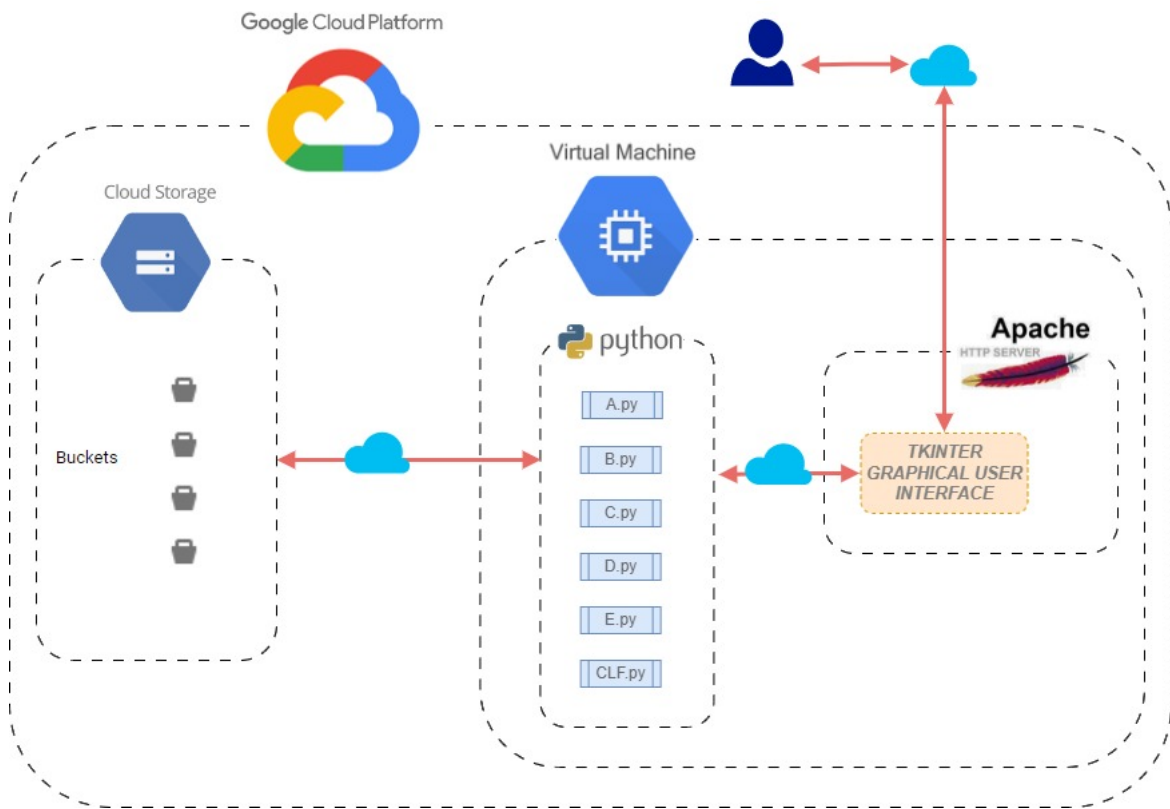


FIGURA 3.2. Arquitectura de la solución.

Dentro de la máquina virtual de la plataforma están alojados los módulos de Python y allí es donde se ejecuta el código. Los archivos de imagen, texto y resultados de los algoritmos en formato JSONL se guardan en repositorios propios de Google Cloud. El nuevo usuario interactúa por medio de la interfaz gráfica disponibilizada en un servidor Apache dentro de la misma máquina virtual.

3.4.2. Módulo de conversión de imágenes

Cuando el candidato a usuario carga un documento, ya sea un contrato societario o un poder legal, el proceso inicial se basa en homogeneizar el tipo de imagen para entregar a los módulos subsiguientes parámetros claros con los que continuar el procesamiento. El módulo "B", como aparece indicado en las figuras 3.2 y 3.1, está programado en Python y utiliza la librería Pillow [9] para realizar las conversiones. Cuenta con un clasificador para distinguir si se cargaron múltiples archivos o el documento está comprendido en uno solo. La interacción con la aplicación de almacenamiento Google Storage está integrada en el código de Python y se utiliza para guardar el resultado en un *bucket* específico.

3.4.3. Módulo de procesamiento de imágenes

La entrada al siguiente módulo (referenciado como "C ") entrega un archivo con formato JPG por cada hoja del documento. El objetivo del procesamiento es recortar el texto útil de los contratos descartando los bordes, firmas y formatos de fondo optimizando el posterior desempeño del algoritmo OCR. Para ello se utiliza un modelo de redes neuronales convolucionales de 16 capas VGG16 [10] con pesos de ImageNet, preentrenado con base de datos de millones de imágenes etiquetadas en miles de categorías. De todos los pesos se descartan los de la última capa, que se vuelven a entrenar con los sets de datos propios utilizando la técnica de *transfer learning*. En la figura 3.3 se observa el esquema de capas del modelo VGG16:

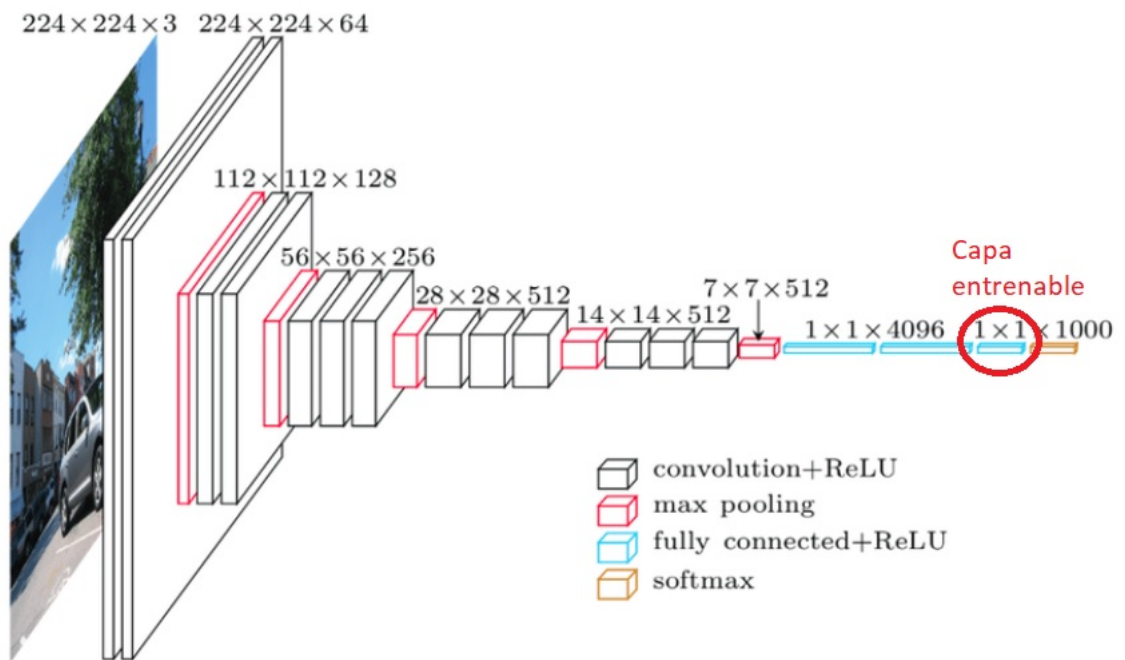


FIGURA 3.3. Arquitectura de capas del modelo VGG16.

Teniendo en cuenta la poca cantidad de datos se utilizó la técnica de *data augmentation*, en particular la rotación de las imágenes originales en 90, 180 y 270 grados de manera aleatoria. De esta manera, los archivos modificados se suman a los originantes duplicando el tamaño del dataset utilizado para el entrenamiento y posterior validación.

3.4.4. Módulo de conversión de imágenes a texto

Una vez obtenida la imagen con el texto puro de los contratos, el siguiente módulo, llamado con la letra "D" en las figuras 3.2 y 3.1, se encarga de convertir en caracteres alfanuméricos la información, guardando el resultado en un archivo del formato txt. Para ello se utiliza el paquete de Python Pytesseract, que proporciona una interface del motor de OCR Tesseract, desarrollado por Google y ampliamente utilizado para reconocer texto en imágenes y documentos escaneados. Los pasos que sigue el comando de conversión del paquete mencionado están detallados en la sección 2.3 de la presente memoria.

3.4.5. Módulo de predicción NER

Siguiendo con la secuencia programática del sistema, el módulo "E" es el encargado de extraer las entidades de los contratos, cuyos fundamentos se detallan en la sección 2.3. Las entidades que se esperan encontrar dentro de los documentos legales son las siguientes:

- Razón social.
- Integrantes de la sociedad.
- DNI/CUIT de los integrantes.
- Nacionalidad de los integrantes.
- Representante legal.
- Fecha de registro de la sociedad.
- Número de registro de la sociedad.

Para lograr la extracción óptima de entidades se utilizó el *pipeline* de la aplicación Spa.cy, cuya secuencia de procesamiento es la siguiente:

1. *Embed*:

El *pipeline* utiliza una matriz de embeddings de 128 dimensiones con pesos pre-entrenados en un *corpus* grande de texto en idioma español para llevar cada palabra o *token* a una representación numérica vectorial, como se observa en la figura 3.4. Su función es capturar relaciones semánticas y sintácticas entre palabras y permitir que el modelo aprenda de manera eficiente a partir de los datos de entrenamiento. Específicamente, el algoritmo utilizado toma en cuenta las palabras vecinas al *token* en cuestión para definir su representación vectorial.



FIGURA 3.4. Representación de la función *embed* en el *pipeline* de Spa.cy.

2. *Encode*:

La función de codificación reformula los vectores de *tokens* consecutivos en una matriz sensible al contexto de las palabras, como se observa en la figura 3.5. En el caso del *pipeline* de SpaCy utilizado, el *encode* se implementa mediante modelos de redes neuronales convolucionales (CNN). El algoritmo aplica convoluciones unidimensionales a la secuencia de *embeddings* de palabras. La ventana de convolución, en este caso, tiene un tamaño de 3, lo que significa que el algoritmo examinará tres *tokens* adyacentes a la vez. Esta operación se desliza a lo largo de la secuencia y realiza una multiplicación elemento por elemento entre los valores del *kernel* y los valores en la ventana actual. Esto permite detectar patrones locales en el texto.



FIGURA 3.5. Representación de la función *encode* en el *pipeline* de Spa.cy.

3. *Attend*:

El módulo de atención del modelo se utiliza para ponderar diferentes partes de la salida de la CNN según su importancia en la tarea específica. Como muestra la figura 3.6, la salida resultante de dicha funcionalidad se representa mediante un vector de características.



FIGURA 3.6. Representación de la función *attend* en el *pipeline* de Spa.cy.

La red neuronal del módulo de atención discrimina características locales mediante la asignación de pesos a los *tokens* en función de su relevancia en un contexto cercano. Esto permite que el modelo se enfoque en partes específicas de la entrada, resaltando características locales y discriminando información relevante en función del contexto.

4. Predicción de entidades nombradas (*predict*):

Identifica y clasifica las entidades nombradas en el texto asociando los *tokens* encontrados con cada una de las categorías, como se muestra en la figura 3.7. Para lograr la asociación mencionada, se utiliza un modelo de red neuronal de perceptrón multicapa.



FIGURA 3.7. Representación de la función de predicción en el *pipeline* de Spa.cy.

3.4.6. Módulo de clasificación binaria

Tal como se puede apreciar en la figura 3.1, la fase final del proceso tiene como tarea determinar la precisión de la información proporcionada por el nuevo usuario. Este análisis se lleva a cabo mediante la comparación de las entidades extraídas automáticamente y las ingresadas manualmente a través de un formulario. Por lo tanto, se dispone de tres archivos en formato JSONL correspondientes a los módulos identificados en la figura 3.1 como "A", "F" y "H". El último de estos archivos únicamente está presente cuando el usuario es el representante legal de la empresa que se busca habilitar en el sistema del cliente.

Los campos de entidades deben ser comparados en sentido amplio, ya que puede haber errores puntuales en el reconocimiento de algunos caracteres debido a reconocimiento OCR fallido. Por otro lado, los nombres y apellidos no siempre están en el mismo orden, pueden o no tener comas en el medio, etc. Por todo esto se decidió utilizar un algoritmo distancia de Levenshtein con peso para parametrizar la similitud entre palabras.

En la figura 3.8 se muestra el diagrama de flujo del módulo de clasificación binaria. Las flechas color verde representan comparaciones de similitud. Cada una de las ramas se la pondera con los parámetros "h" y el resultado final es cotejado con el umbral de aceptación.

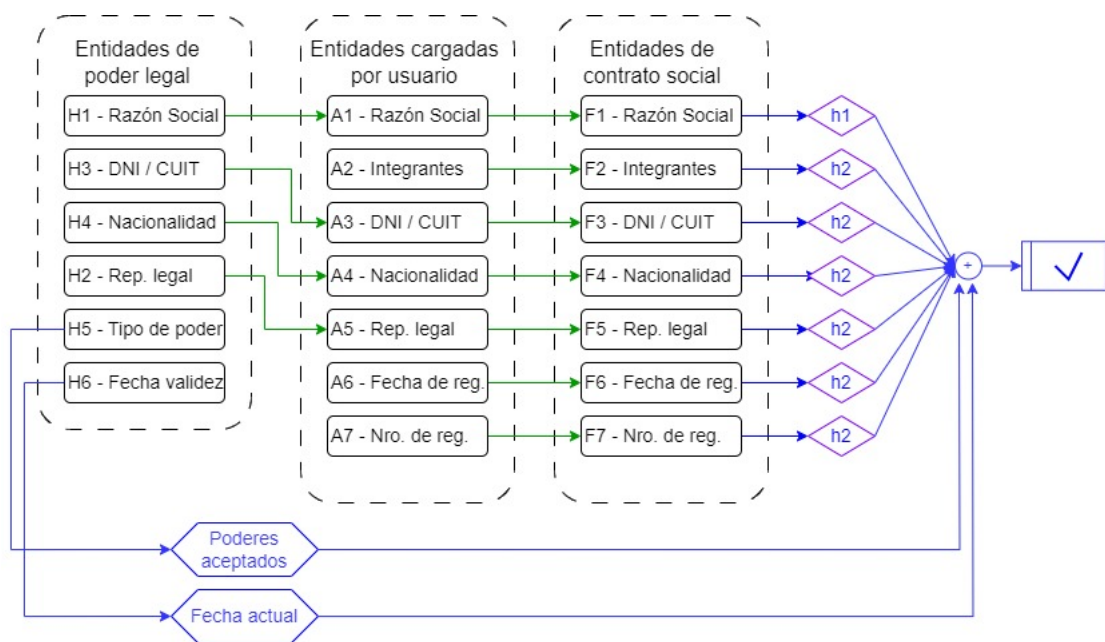


FIGURA 3.8. Diagrama de flujo del clasificador binario.

El algoritmo de distancia de Levenshtein se utiliza para medir la similitud entre dos cadenas de texto. El objetivo es determinar la cantidad mínima de operaciones de edición necesarias para transformar una cadena en otra. Estas operaciones son: inserción, eliminación y sustitución.

La variante del algoritmo con pesos por similitud de caracteres se diferencia al asignar pesos o costos variables a las operaciones de inserción, eliminación y sustitución de caracteres según su parecido. En lugar de utilizar una matriz de costos constante como en el algoritmo estándar, se crea una matriz de similitud en la que se establecen los parámetros entre todos los pares de caracteres posibles en las cadenas. Por ejemplo, "O" y "0" son considerados más similares que "X" e "I", se asignarán valores más bajos a las celdas correspondientes en la matriz para el primer caso. Luego, al calcular la distancia de Levenshtein, se refleja de manera más precisa la similitud entre caracteres en el cálculo de la distancia entre las cadenas. En el caso de textos escaneados y procesados con OCR se tienen en cuenta las características intrínsecas en esos procesos. En el presente proyecto se implementó el algoritmo con la librería de Python Weighted-levenshtein [11].

Capítulo 4

Ensayos y resultados

Este capítulo busca describir los ensayos, principalmente de forma teórica, que se desarrollan sobre los tipos de modelos utilizados en el trabajo. Lamentablemente, la cantidad de datos, contratos societarios y poderes legales que pudo suministrar el cliente y que fueron factibles conseguir dada la naturaleza confidencial de los mismos, fue muy escasa e insuficiente para entrenar modelos con métricas aceptables. En el caso del procesamiento de imágenes se pudo desarrollar los pasos metodológicos aquí descritos para la optimización debido a la menor cantidad de recursos que requiere el problema de *bounding boxes* y la posibilidad de implementar la técnica de *data augmentation*.

4.1. Metodología de las pruebas

La realización de pruebas para modelos de procesamiento de imágenes y NLP implica una serie de pasos comunes que incluyen la preparación de datos, el ajuste de modelos pre-entrenados, el entrenamiento, la validación y la evaluación en un conjunto de prueba. Los detalles específicos y las métricas se explican en esta sección. El proceso general sigue un enfoque sistemático para garantizar la calidad y el rendimiento de los modelos de IA. A continuación, se describe cómo se llevaron a cabo las pruebas de manera conjunta para ambos tipos de modelos:

1. División de datos: como se mencionó anteriormente, se dividió el dataset en tres partes: entrenamiento, validación y prueba. Este paso es común para ambos modelos. Debido a la escasez de datos también se hicieron pruebas resignando las pruebas para reforzar el entrenamiento.
2. Carga de pesos pre-entrenados: en ambas tareas se cargan los pesos pre-entrenados en el modelo base. Estos valores fueron obtenidos mediante el entrenamiento del modelo en una gran cantidad de datos no específicos y de objetivo general.
3. Personalización del modelo: en esta etapa se seleccionan las capas superiores del modelo a ser entrenadas con el dataset propio, adaptadas a la tarea específica que se está probando.
4. Entrenamiento y validación: se entrena el modelo personalizado utilizando el conjunto de entrenamiento y se valida su rendimiento en el conjunto de validación. En esta etapa se ajustan hiperparámetros, como la tasa de aprendizaje y el tamaño del lote, para optimizar el rendimiento del modelo.

5. Evaluación en el conjunto de prueba: una vez que el modelo ha sido entrenado y validado, se evalúa en el conjunto de prueba para obtener una evaluación imparcial de su rendimiento en situaciones reales donde el algoritmo desconoce por completo los datos de entrada. En esta etapa se registran métricas típicas de modelos de redes neuronales de modelos de clasificación que se explican a continuación:

- **Precisión:** mide la proporción de ejemplos positivos que fueron clasificados correctamente como positivos en relación con todos los ejemplos clasificados como positivos, ya sean verdaderos o falsos. Es una métrica útil cuando se busca minimizar los falsos positivos. La fórmula de precisión es:

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

- **Recall o sensibilidad:** mide la proporción de ejemplos positivos que fueron clasificados correctamente como positivos en relación con todos los ejemplos reales positivos en el conjunto de datos. Es una métrica importante cuando se busca minimizar los falsos negativos. La fórmula de recall es:

$$\text{Recall} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

- **Puntaje F1:** es una medida que combina tanto la precisión como el recall en una sola métrica. Es útil cuando se necesita un equilibrio entre la precisión y el recall. Se calcula como la media armónica de la precisión y el recall, y se expresa de la siguiente manera:

$$\text{F1 Score} = 2 \times \left(\frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}} \right)$$

El puntaje F1 tiende a ser más informativo que la precisión o el recall por separado, ya que considera tanto los falsos positivos como los falsos negativos. Es particularmente útil en situaciones en las que las clases no están equilibradas y cuando se necesita encontrar un compromiso entre la precisión y el recall.

Para la evaluación del modelo de procesamiento de imágenes, al ser un problema de regresión, se utiliza la técnica de error cuadrático medio (MSE por sus siglas en inglés), en la cual se elevan al cuadrado los errores para luego promediarse, como indica su fórmula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Donde:

- n es el número de ejemplos en el conjunto de datos.

- y_i es el valor real del ejemplo i .
 - \hat{y}_i es la predicción del modelo para el ejemplo i .
6. Análisis de resultados: Se analizaron los resultados de las pruebas para comprender las fortalezas y debilidades del modelo en ambas tareas. Esto incluyó la identificación de casos de error y la retroalimentación para mejorar el modelo.
 7. Mejora continua: Basándose en los resultados y el análisis de los mismos, se realizan cambios en los hiperparámetros y se vuelve a configurar para luego entrenar y probar el modelo hasta encontrar las métricas satisfactorias para el problema dadas sus características y finalidad.

4.2. Resultados de modelos de procesamiento de imágenes

En esta sección se detallan los resultados obtenidos por el modelo VGG16 de procesamiento de imágenes, con el objetivo de generar los *bounding boxes* que permiten recortar las imágenes dejando sólo el texto de interés en los contratos, como se describe en los anteriores. Inicialmente se describen los parámetros comunes para todos los entrenamientos y luego los hiperparámetros que se fueron modificando para lograr el mejor resultado.

4.2.1. Consideraciones generales de los modelos

Los pesos pre-entrenados utilizados son los provenientes de la librería Keras de Tensorflow [12], en base a la competición ImageNet [13], con más de 32.000 imágenes en tensores de dimensión 224x224x3 cada una. La configuración en común para todos los entrenamientos generados incluye:

- Optimizador Adam.
- Función de pérdida MSE.
- 10 % del total del dataset para *test*.

4.2.2. Consideraciones particulares de los modelos

A continuación, se lista una muestra de los entrenamientos en el orden en que fueron realizados. Se indican las configuraciones y los resultados de manera gráfica:

1. Modelo entrenado 1:
 - Estructura del modelo: se mantienen las capas originales con los pesos pre-entrenados y se agregan las siguientes entrenables:
 - Capa *flatten* de dimensiones de salida (1, 25.088).
 - Cuatro capas densas de dimensiones de salida (1, 128); (1, 64); (1, 32) y (1, 4).
 - Hiperparámetros:
 - Tasa de aprendizaje inicial: 1×10^{-4} .
 - Números de épocas: 25.

- Tamaño de lote: 32.
- Resultados obtenidos de acuerdo a la figura 4.1.

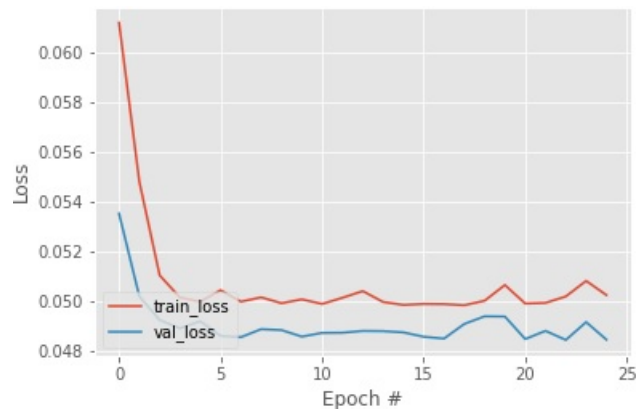


FIGURA 4.1. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 1.

2. Modelo entrenado 2:

- Estructura del modelo: se mantienen los pesos pre-entrenados de las primeras 13 capas y se entrenan las últimas tres convolucionales. Además, se agregan al final las mismas capas que en el modelo anterior:
 - Capa *flatten* de dimensiones de salida (1, 25.088).
 - Cuatro capas densas de dimensiones de salida (1, 128); (1, 64); (1, 32) y (1, 4).
- Hiperparámetros:
 - Tasa de aprendizaje: 1×10^{-4} .
 - Números de épocas: 25.
 - Tamaño de lote: 32.
- Resultados obtenidos de acuerdo a la figura 4.2.

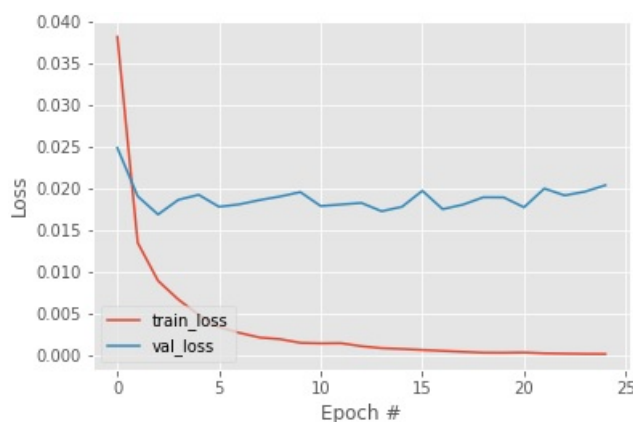


FIGURA 4.2. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 2.

3. Modelo entrenado 6:

- Estructura del modelo: se mantienen los pesos pre-entrenados de las primeras 13 capas y se entrenan las últimas tres convolucionales. Además, se agregan al final las mismas capas que en el modelo anterior sumando capa de *dropout*:
 - Capa *Flatten* de dimensiones de salida (1, 25.088).
 - Cuatro capas densas de dimensiones de salida (1, 128); (1, 64); (1, 32) y (1, 4).
 - Se agrega capa de *dropout* antes de la penúltima capa densa con parámetro de 0.2.
- Hiperparámetros:
 - Tasa de aprendizaje: 1×10^{-4} .
 - Números de épocas: 25.
 - Tamaño de lote: 6.
- Resultados obtenidos de acuerdo a la figura 4.3.

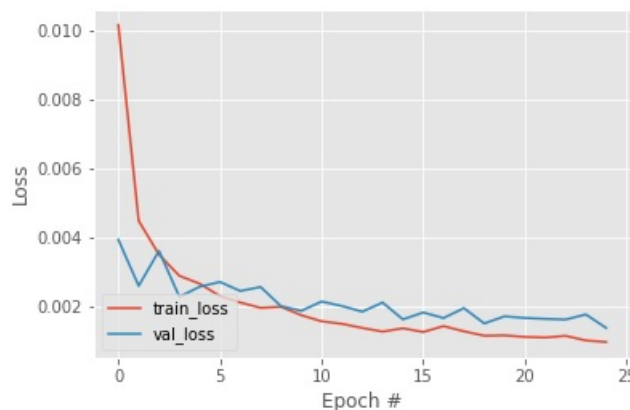


FIGURA 4.3. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 6.

4. Modelo entrenado 19:

- Estructura del modelo: se mantienen los pesos pre-entrenados de las primeras 13 capas y se entrenan las últimas tres convolucionales. Además se agregan al final las mismas capas que en el modelo anterior:
 - Capa *Flatten* de dimensiones de salida (1, 25.088).
 - Cuatro capas densas de dimensiones de salida (1, 128); (1, 64); (1, 32) y (1, 4).
 - Se agrega capa de *dropout* antes de la penúltima capa densa con parámetro de 0.2.
- Hiperparámetros:
 - Tasa de aprendizaje variable, inicial: 1×10^{-4} .
 - Números de épocas: 25.

- Tamaño de lote: 30.
- Resultados obtenidos de acuerdo a la figura 4.4.

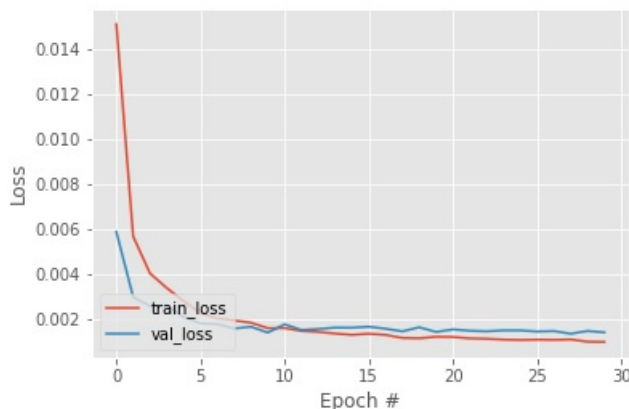


FIGURA 4.4. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 19.

4.2.3. Análisis de los resultados

En la subsección anterior se presentaron cuatro modelos parametrizados y sus resultados que se analizan mediante la comparación de las gráficas obtenidas. Las conclusiones son las siguientes:

- El modelo entrenado 1 resultó ser estable y converger a valores de pérdida cercanos a 4.9 %. Agregando mayor cantidad de capas entrenables en el modelo 2 se observa que se llega a mejores resultados con valores de pérdida cercanos al 2 %.
- El punto débil del modelo 2 es el sobre-entrenamiento que se manifiesta con la diferencia de performance entre los valores de entrenamiento y validación. Para ello se implementó una capa de *dropout* en el modelo 6, que funciona desactivando aleatoriamente un porcentaje de neuronas (unidades) durante el entrenamiento. Al incorporar esta capa, el modelo se ve forzado a aprender representaciones más robustas y generalizadas de los datos.
- El modelo 19 incorpora el concepto de tasa de aprendizaje variable, lo cual permite que el proceso converja más rápidamente sin penalizar la estabilidad del mismo, como se observa en la figura 4.4. Se utilizó para ello el parámetro de Keras ReduceLR0Plateau con una configuración que monitorea la pérdida en el conjunto de validación y reduce la tasa de aprendizaje en un 60 % si la pérdida no mejora durante 4 épocas consecutivas. El valor mínimo al que se puede reducir la tasa de aprendizaje es 0.

4.3. Resultados de modelos de reconocimiento de entidades nombradas

En esta sección, se presentan los resultados del entrenamiento del modelo de reconocimiento de entidades nombradas (NER) utilizando SpaCy. Se hizo un esfuerzo para entrenar el modelo, pero se enfrentó a limitaciones debido a la disponibilidad limitada de datos: únicamente 82 documentos que se utilizaron en proporciones de 80-20 para entrenamiento y validación respectivamente.

La figura 4.5 muestra la información que suministra SpaCy sobre el entrenamiento de los modelos de tokenización y NER con una tasa de aprendizaje variable de 0.001:

Training pipeline							
Components: ner							
Merging training and evaluation data for 1 components							
- [ner] Training: 82 Evaluation: 20 (20% split)							
Training: 67 Evaluation: 17							
Labels: ner (7)							
! Pipeline: ['tok2vec', 'ner']							
! Initial learn rate: 0.001							
E	#	LOSS TOK2VEC	LOSS NER	ENTS_F	ENTS_P	ENTS_R	SCORE
0	0	0.00	1448.63	0.23	0.12	1.62	0.00
2	200	15991.61	34462.76	53.61	73.86	42.07	0.54
5	400	2392.57	4632.48	56.16	74.73	44.98	0.56
8	600	22250.09	4087.05	49.22	78.17	35.92	0.49
11	800	998.98	2618.62	63.22	73.71	55.34	0.63
14	1000	1441.97	1690.21	62.12	74.89	53.07	0.62
17	1200	1339.32	1375.62	52.94	75.45	40.78	0.53
20	1400	889.96	1068.16	54.33	71.81	43.69	0.54
23	1600	3262.87	896.95	61.26	78.68	50.16	0.61
26	1800	3906.34	834.77	61.93	70.83	55.02	0.62
29	2000	856.94	678.83	60.69	81.87	48.22	0.61
32	2200	1087.92	680.12	57.20	74.87	46.28	0.57
35	2400	960.35	518.07	62.26	67.29	57.93	0.62

FIGURA 4.5. Métrica de regresión MSE en sets de entrenamiento y validación del modelo 19.

Las métricas referidas en la sección 4.1 son las siguientes:

- ENTS_F: F1 Score.
- ENTS_P: Precisión.
- ENTS_R: Recall.

Como se puede observar, a medida que avanzan las épocas de entrenamiento la pérdida no se estabiliza y las métricas no indican una mejora progresiva.

4.4. Simulación del sistema completo

El objetivo original de esta sección contemplaba la presentación de los resultados de la simulación del sistema completo involucrando todas las componentes desarrolladas, incluidos los modelos de procesamiento de imágenes y reconocimiento de entidades nombradas (NER). Sin embargo, tras realizar pruebas exhaustivas, se ha decidido no mostrar ningún resultado concreto. La razón principal detrás de esta decisión es la insuficiencia de datos de entrenamiento para los modelos. Aunque se realizaron esfuerzos de entrenamiento y ajuste, las falencias tuvieron un impacto significativo en el rendimiento general.

Capítulo 5

Conclusiones

5.1. Resultados obtenidos y cumplimiento de objetivos

Como se expresa en los capítulos anteriores de esta memoria, los resultados en términos de usabilidad presentan desafíos significativos debido a restricciones en el acceso a los datos necesarios para conformar el dataset. Sin embargo, a pesar de estas limitaciones, se logró alcanzar una performance aceptable en la implementación del modelo de *bounding boxes*. Además, es relevante subrayar que el sistema se disponibilizó de manera exitosa uniendo todos los elementos que lo componen. En definitiva, a pesar de los obstáculos encontrados, se ha logrado avanzar significativamente en el logro principal planteado en la sección 1.4.

5.2. Vínculo con la carrera

En cuanto al objetivo de aprendizaje, también planteado en la sección 1.4, se listan a continuación las materias de la especialización en inteligencia artificial cuyo contenido resultó imprescindible para la elaboración del trabajo:

- Gestión de proyectos: la planificación inicial resultó indispensable como guía del trabajo durante su desarrollo. A pesar de haber modificado sobre la marcha ciertos lineamientos, resultó provechoso plantear y justificar los cambios.
- Visión por computadora I y II: para el desarrollo del modelo de *bounding boxes* se utilizaron fundamentos de ambas materias, la primera de manera conceptual y la segunda de forma específica.
- Procesamiento del lenguaje natural: en el contenido de esta materia se encuentra el núcleo del trabajo: la interpretación del texto para validar los contratos societarios.
- Aprendizaje profundo: conocer el funcionamiento detallado de las redes neuronales fue muy importante para entender los procesos involucrados en el diseño y entrenamiento de los modelos.
- Aprendizaje de máquina II: esta materia brindó los conocimientos prácticos que fueron fundamentales para la disponibilización del sistema en la máquina virtual y *buckets* de Google Cloud.

5.3. Oportunidades de mejora

En esta sección se examinan las posibles áreas de desarrollo y perfeccionamiento identificadas durante el estudio del procesamiento de contratos societarios. Estas consideraciones proporcionan una visión esencial para futuras mejoras y refinamientos en el proyecto:

- Ampliación del dataset: dado que las restricciones de acceso a los datos han limitado en extremo la usabilidad del proyecto, la oportunidad de mejora fundamental es ampliar y diversificar el set de datos utilizado. Esto podría implicar colaboraciones con empresas o instituciones que estén dispuestas a compartir datos relevantes o incluso iniciar un proceso de solicitud formal al departamento de asuntos legales del cliente.
- Optimización del rendimiento del modelo de *bounding boxes*: A pesar de la performance aceptable del modelo VGG16 entrenado, siempre hay espacio para mejoras. Se pueden explorar técnicas de optimización en el aprendizaje automático como el ajuste de hiperparámetros o la incorporación de modelos más avanzados para lograr una mayor precisión.
- Exploración de técnicas avanzadas de procesamiento de lenguaje natural (NLP): La evolución en este área es tan rápida que, desde el inicio del trabajo hasta el día de hoy, hubo avances muy importantes, modelos mas complejos, pesos entrenados en base a gran cantidad de datos y aplicaciones que facilitan el uso de las herramientas. Como mayor ejemplo, el desarrollo y disponibilidad de ChatGPT modifica las expectativas del avance tecnológico a nivel global.
- Evaluación continua de resultados: Implementar un sistema de seguimiento y evaluación continua para medir la efectividad y el impacto del sistema a lo largo del tiempo y realizar ajustes en consecuencia.

Estas son algunas de las opciones de mejora que se pueden considerar. La elección de cuál implementar dependerá de los recursos disponibles, los objetivos a largo plazo y las necesidades específicas del cliente en la continuación del trabajo.

Bibliografía

- [1] *Página web oficial de la empresa MercadoLibre.* <http://mercadolibre.com.ar>.
- [2] Nitin Indurkha y Fred J. Damerau. *Handbook of Natural Language Processing*. Chapman y Hall, 2010.
- [3] Richard Szeliski. *Computer Vision Algorithms and Applications 2nd ed.* Springer, 2022.
- [4] *Página web de HuggingFace, el estado del arte en Transformers.* <https://huggingface.co/docs/transformers/index>.
- [5] Sujit Pal. *Named Entity Recognition A Practical Guide*. Springer, 2022.
- [6] *Repositorio de la librería Tesseract.* <https://github.com/tesseract-ocr/tesseract>.
- [7] *Página web de Prodi.gy.* <https://prodi.gy/>.
- [8] *Página web de spaCy.* <https://spacy.io/>.
- [9] *Página web de la librería de Python PILLOW.* <https://python-pillow.org/>.
- [10] Karen Simonyan; Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv, 2014.
- [11] *Página web de la librería de Python Weighted-Levenshtein.* <https://pypi.org/project/weighted-levenshtein/>.
- [12] *Página web de la librería de Tensorflow Keras.* <https://www.tensorflow.org/guide/keras/>.
- [13] *Página web oficial de ImageNet.* <https://www.image-net.org/>.