- CASO PRÁCTICO -

AUTOMATIZACIÓN DEL PROCESAMIENTO DE LA INFORMACIÓN Y DE LOS DOCUMENTOS DE LOS CLIENTES PARA LA CONCESIÓN DE HIPOTECAS

Javier Martí Isasi
javiermartiisasi@gmail.com
https://linkedin.com/in/javier-marti-isasi
https://github.com/javier-marti-isasi

Índice

1. Presentación del proyecto

1.1. Definición del proyecto

El departamento de Procesos trabaja en el rediseño del proceso de negocio de Concesión de hipotecas y nos pregunta si podríamos identificar necesidades de IA y analizar la viabilidad de la implementación du una solución de IA para la automatización del procesamiento de la información y de los documentos que aportan los clientes en el proceso vía la oficina o los canales digitales. El procesamiento consiste en identificar los documentos y extraer una serie de información. El resultado del procesamiento se utiliza para validar y guardar de forma ordenada y consistente la documentación junto a los datos extraídos aportada en el gestor documental que servirá para otras fases del proceso (grabación de datos económicos y patrimoniales, scoring de riesgos, etc.).

Negocio indica que a futuro se rediseñaran otros procesos con necesidades idénticas y esperan que se reutilizara la solución implementada en este primer caso de uso.

El proceso se puede resumir de forma sencilla:

- Se abre una propuesta para todos los intervinientes (por ej. 2 personas piden una hipoteca)
- Cada interviniente declara sus ingresos, deudas y patrimonio (Declaración de bienes)
- A cada interviniente se le pide aportar unos documentos que acreditan lo que ha declarado en la Declaración de bienes (Nominas, Vida laboral, Contrato de trabajo, recibos de préstamos, escrituras, notas simples, etc.)

- Los documentos se mandan directamente al gestor documental para que se valide (reglas de negocio como vigencia, completitud, legibilidad, etc.) y se guarde en su hueco correspondiente (Nomina del Interviniente 1 de la empresa X de enero 2022). Si los documentos recibidos son incompletos, o no son los esperados o si no cumplen con las reglas de validación, se avisa a la oficina y/o al cliente para que aporte la documentación correcta

Una operación de concesión de hipotecas consiste en varios envíos (se repiten los puntos 3 y 4 porque el cliente puede aportar documentación varias veces, o bien por olvido o por documentos erróneos) y cada envío puede contener varios documentos y de cada documento se extraen varios campos.

1.2. Objetivos de negocio

El objetivo de negocio es de reducir los tiempos de respuesta a los clientes, mejorar la experiencia de los clientes, bajar los costes del servicio y disminuir los riesgos operativos.

Para llegar a estos objetivos, el equipo de IA tiene como objetivo desarrollar una solución que permite automatizar el proceso end-to-end sin revisión humana. Negocio insiste que se trata de un proceso de negocio donde el error por operación tiene que ser inferior al 2% (baseline humano) y que prefiere reducir la tasa de automatización para alcanzar este objetivo de precisión. Añade que la parte no automatizada derivara a un back-office para que se procese manualmente.

Los KPIs para el equipo de IA son los siguientes:

- Precisión (objetivo>=98%) (*)
- Automatización (objetivo>=60%) (**)
- SLA (<2 horas)

(*) 98% de precisión significa que 98% de los documentos se han procesado de forma correcta en todas las etapas (clasificación, extracción de datos y validación)

(**) 60% de automatización significa que 60% de los documentos se procesan de forma automática end-to-end sin requerir la intervención humana en ninguna de las etapas (clasificación, extracción de datos y validación)

1.3. Información sobre los documentos

- Mas de 20 tipos de documentos (por ej. Nómina, Vida laboral, Contrato de trabajo, recibos de préstamos, escrituras, notas simples, etc.)
- Los documentos no son plantillas. Ciertos tipos son semiestructurados (por ej. Nómina y Vida laboral) y otros no estructurados (por ej. Contrato de trabajo, escrituras)
- Pueden aparecer tipologías de documentos que no están en la lista identificadas de documentos a aportar (puede ocurrir por error del cliente o para ciertas excepciones no cubiertas por el proceso)
- Los documentos tienen un numero variable de páginas incluso dentro de una misma tipología
- En ciertos casos los documentos pueden venir mezclados y desordenados (en caso de que un cliente traiga toda la información en formato papel a la oficina y que se escanee toda la documentación a la vez)
- Hay un desbalanceo en la distribución (ciertos tipos de documentos representan hasta el 15% de la volumetría total y otros tipos menos del 1%)
- La información a extraer depende de los documentos, pero reducimos el alcance en la primera aproximación a los datos siguientes: DNI y nombre persona, CIF empresa, Periodo, Firma digital o manuscrita
- La calidad y el formato de los documentos aportados depende del canal (escáner oficina, fotos, documento digital original, etc.)

- No disponemos de anotaciones, es decir de un set de documentos con su tipología para la clasificación, ni de las coordenadas y de los valores de la información a extrae

Misión del grupo de IA

La misión del grupo de IA es identificar las necesidades de negocio, analizar los datos y revisar el estado de arte, diseñar una plataforma de IA de automatización de procesos y analizar la viabilidad del proyecto.

- Análisis del proceso e identificación de las necesidades de IA
- Análisis exploratorio de datos/documentos
- Anotación de datos: existe la posibilidad de contratar a un servicio de anotación, pero el equipo de IA quiere plantear aproximaciones para reducir el volumen de anotaciones manuales, como, por ejemplo:
    - Transfer Learning
    - Active Learning
    - Semi-supervised o Self-supervised - Etc.

- Diseño de la solución IA (la lista es orientativa y no exhaustiva, el candidato tiene la libertad de añadir o quitar unos elementos).
    - Flujo funcional de la solución: lista de componentes, flujo de ejecución, modelo
    - de datos.
    - Por ej. 1. Recepción envío => 2. Procesamiento documentos => 3. Extracción información (Layout+Texto) => 4. Clasificación => etc.
    - ➢ Definición Capacidades IA y Experimentación (el equipo de IA tiene que evaluar

- cómo abordar cada línea, utilizando capacidades de terceros o modelos in- house)
    - Mejora de imagen
    - OCR/ICR (layout del documento + texto)
    - Clasificación
    - Extracción de datos
    - Validación de documento: aplicar las reglas de negocio a partir de la tipología de los documentos de la información extraída
    - Modelo de decisión: desarrollar un modelo "on top" basado en el
    - conjunto de features generadas por el flujo completo (metadatos, inputs/outputs de modelos, etc.) que permite decidir si una tarea se puede automatizar o si se tiene que mandar a un humano para revisión. El objetivo de este modelo de decisión es maximizar la automatización (objetivo >= 60%) pero asegurando el nivel de precisión requerido (>= 98%)

2. Introducción

La comprensión de los documentos es una tarea esencial pero difícil, ya que requiere funciones complejas como la lectura del texto y una comprensión integral del documento.

En este caso práctico se pretende desarrollar una aproximación que permita automatizar el proceso *end-to-end* en la automatización de procesamiento, detección de errores, clasificación, extracción de datos, validación y reubicación de documentos aportados por los clientes.

3. Objetivos

A continuación, se enumeran los objetivos propuestos en el trabajo:

- Cumplir con el objetivo de negocio: reducir los tiempos de respuesta a los clientes, mejorar la experiencia de los clientes, bajar los costes del servicio y disminuir los riesgos operativos.
- Enmarcar la investigación dentro del campo del *Visual Document Understanding* (VDU) realizando un estudio sobre el campo.
- Desarrollar una solución que permite automatizar el proceso *end-to-end* sin revisión humana con una precisión objetivo>=98%, una automatización objetivo>=60% y un SLA <2 horas
- Identificar las necesidades de negocio
- Elegir un modelo de IA base y entender su funcionamiento.
- Elegir los distintos conjuntos de datos y prepararlos para el entrenamiento.
- Adaptar los modelos base elegidos.
- Evaluar los resultados obtenidos.
- Analizar la viabilidad del proyecto.
- Proponer futuras líneas de investigación.

4. Estado del arte

En este apartado se analiza brevemente el estado del arte de las últimas arquitecturas que pretenden solucionar el problema del *Visual Document Understanding (VDU).*

4.1. LayoutLM Models (2019)

LayoutLM (*Pre-training of Text and Layout for Document Image Understanding*) [15] utiliza *transformers* basados en técnicas OCR para etiquetar palabras o responder a determinadas preguntas a partir de la imagen de un documento.

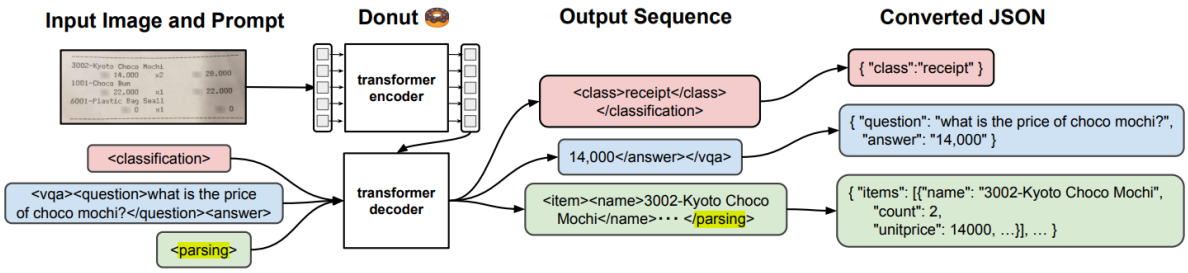
Este modelo, como la mayoría de las respuestas tradicionales al problema de la VDU, se basan en el análisis sintáctico de la salida OCR de esa imagen, junto con *visual encodings*. Sin embargo, el OCR es caro desde el punto de vista computacional (ya que suele requerir la instalación de un motor OCR como *Tesseract*) y la inclusión de otro modelo que debe ser entrenado y puesto a punto. Además, un modelo OCR inexacto conducirá a la propagación de errores en el modelo VDU.

4.2. Donut Model (2021)

OCR-free Document Understanding Transformer (DONUT) [4] fue publicado en noviembe de 2021, con la última revisión hasta la fecha, el 23 de agosto de 2022. Se trata de un modelo de transformadores codificadores-decodificadores sin uso de las técnicas OCR.

Codifica la imagen (dividida en parches utilizando un *transformer Swin*) en vectores de tokens que luego puede decodificar, o traducir, en una secuencia de salida utilizando el modelo de decodificador BART, preentrenado públicamente en conjuntos de datos multilingües. También se pueden

decodificar las indicaciones introducidas en el modelo en el momento de la inferencia en la misma arquitectura.



Pipeline de Donut. El codificador mapea una imagen de documento dada en *embeddings*. Con las incrustaciones codificadas, el descodificador genera una secuencia de tokens que pueden convertirse en un tipo de información objetivo de forma estructurada [4]

5. Métodos

5.1. METODOLOGÍA IMPLEMENTADA

El trabajo ha sido realizado mediante la implementación de la metodología CRISP-DM, desde la comprensión del negocio hasta la evaluación. Sus siglas corresponden a *Cross Industry Imagend Process for Data Mining* y es una metodología creada para dar forma a los proyectos de minería de datos (*data mining*).

Consta de 6 pasos para concebir un proyecto pudiendo tener iteraciones cíclicas según las necesidades de los desarrolladores. Estos pasos son la comprensión del negocio, la comprensión de los datos, la preparación de los datos, el modelado, la evaluación y el despliegue.

5.2. MODELO VDU EMPLEADO

Para resolver los objetivos propuestos se utilizará la arquitectura Donut. Se utilizará un modelo Donut para finetunear y resolver el problema de clasificación de los documentos y se empleará otro modelo en el que se realizará la inferencia directamente sobre él para la extracción de los datos dado un documento.

Por tanto, para el problema de clasificación de documentos, se realizará un entrenamiento para finetunear un modelo Donut preentrenado. Mientras que, para el problema de extracción de los datos, se aplicará la inferencia directamente sobre un modelo Donut preentrenado. Esto se concreta en el punto 5.3.

5.3. TÉCNICAS DE ENTRENAMIENTO EMPLEADAS

En la resolución del problema propuesto, se emplean distintas técnicas de entrenamiento que buscan aumentar el rendimiento de los modelos. Por un lado, el modelo Donut emplea técnicas de ***semi-supervised learning***. Este modelo ha sido preentrenado de forma no supervisada en el *dataset* IIT-CDIP, sin el uso de etiquetas. Este *dataset* cuenta con 11M de documentos en inglés escaneados.

Por otro lado, se emplean técnicas de ***transfer learning + fine tuning***. Se aplica *transfer learning* con el modelo Donut entrenado en el *dataset* de RVL-CDIP. Este *dataset* contiene 6 de las 10 clases que queremos clasificar. Partiremos de este modelo y lo finetunearemos en nuestro *dataset*.

Podría utilizarse técnicas de ***active learning***, reentrenando los modelos a partir de los documentos clasificados y los datos extraídos manualmente por *backoffice*.

6. Resultados

Este apartado contiene los resultados alcanzados tras la implantación de los métodos anteriormente citados.

6.1. WORKFLOF

Se diseña un *workflow* que permite automatizar el proceso *end-to-end* reduciendo la revisión humana.

NOTA: Ver el diagrama del workflow adjuntado en pdf: Workflow.pdf

El *workflow* propuesto se divide en las siguientes fases principales:

- Fase 0: Configuración del entorno de ejecución
- Fase 1: Aportación de los datos y
  la documentación por parte del cliente
- Fase 2: Validación formato, clasificación y reubicación de los documen
  tos
- Fase 3: Extracción, validación y almacenamiento de los datos y reubicación de los documentos
- Fase 4: Monitorización y reajuste de los modelos

En el diagrama del *workflow* y en el notebook 4 adjuntados se visualizan, detallan y programan las fases 0 a 3.

Respecto a la fase 4 (monitorización y reajuste de los modelos), se podrían utilizar las herramientas que permite *MLFlow* para reentrenar los modelos en función del criterio estimado. Los modelos podrían ir perdiendo precisión con el paso del tiempo. Es importante monitorizar el rendimiento de los modelos y reentrenar cuando sea necesario.

Respecto al modelo de clasificación, los documentos que han sido clasificados podrían servir para reentrenar al modelo, empleando técnicas de *active learning*. Esto es especialmente útil si se reentrena con los documentos clasificados por *backoffice*, que previamente habían tenido un grado de confianza bajo por el modelo. A su vez, habría que ir ajustando el índice de confianza del modelo para hacer un correcto equilibrio entre el porcentaje de los documentos automatizados y la precisión del modelo, en función de los KPIs definidos.

Por otro lado, un modelo preentrenado por nuestro sistema para la extracción de datos podría ajustarse con esta misma lógica.

6.2. NOTEBOOKS PRESENTADOS

Se entregan los siguientes notebooks:

- 1. Preparing dataset.ipynb
- 2. Fine-tune pretrained RVL-CDIP Donut for document classification.ipynb
- 3. DONUT for DocVQA evaluation.ipynb
- 4. Putting into production - document classification and data extraction.ipynb

En el notebook 1 se prepara el *dataset* que será utilizado para finetunear el modelo Donut en la tarea de clasificación de documentos.

El notebook 2 incluye el entrenamiento, monitorización y evaluación del modelo Donut para la clasificación de documentos. Se aplica *fine-tuning* sobre el modelo Donut preentrenado en el *dataset* RVL-CDIP. Se extrae la precisión del modelo y el índice de confianza que determinará si un documento será procesado por el modelo o derivado al *backoffice* para su clasificación manual.

En el notebook 3 se evalúa el modelo Donut oficial entrenado sobre el *dataset Document Visual Question Answering* (DocVQA) [11] sobre un set de datos definido. Este modelo permitirá la extracción de cualquier dato en cualquier

documento. Se obtiene la precisión del modelo y el índice de confianza que determinará si un documento será procesado por el modelo o derivado al *backoffice* para la extracción manual de los datos.

El notebook 4 consiste en una aproximación a la puesta en producción de todo el *workflow* propuesto. Incluye la introducción de los datos y la documentación por parte del cliente, la clasificación de los documentos, la extracción de los datos de los documentos, la validación de los datos, el almacenamiento de datos y reubicación de los documentos, el envío de los documentos al *backoffice* para su procesamiento manual y los avisos al cliente.

6.3. MODELOS UTILIZADOS

Para la tarea de clasificación de documentos, en una primera aproximación, se realiza *fine-tuning* sobre el modelo oficial base. Está facilitado por el equipo de Hugging Face: https://huggingface.co/nielsr/donut-base. En una segunda aproximación, se realiza fine-tuning sobre el modelo Donut preentrenado en el dataset RVL-CDIP, facilitado por el equipo de Hugging Face: https://huggingface.co/naver-clova-ix/donut-base-finetuned-rvlcdip. Este entrenamiento es el que aparece en el notebook 2.

Para la tarea de extracción de datos, se emplea el modelo oficial DocVQA Donut, facilitado por el equipo de Hugging Face: https://huggingface.co/naver-clova-ix/donut-base-finetuned-docvqa. Sobre este modelo se aplicará directamente la inferencia. Se podría finetunear el modelo base de Donut en un *dataset* con los datos deseados parseados, sin necesidad de que estén definidos en una bounding-box (*free-OCR*). Por ejemplo, se podría finetunear el modelo con el *dataset* Cord-v2: https://huggingface.co/datasets/naver-clova-ix/cord-v2

6.4. DATASET UTILIZADOS

Para la tarea de clasificación de documentos, se aplica *fine-tuning* sobre el modelo Donut oficial base y sobre el modelo Donut preentrenado en el dataset RVL-CDIP. Se ha pretendido ajustar el *dataset* a las necesidades de negocio del enunciado, recolectando imágenes de diversos *datasets* públicos.

Se emplean 10 clases en total. A continuación, se muestran las clases del *dataset* empleado, el *dataset* de las imágenes de origen y si se encuentra la clase o no en el dataset RVL-CDIP.

- Passport | Pardo dataset | No en RVL-CDIP
- ADVE | Tobacco-3482 dataset | Sí en RVL-CDIP
- Email | Tobacco-3482 dataset | Sí en RVL-CDIP
- Form | Tobacco-3482 dataset | Sí en RVL-CDIP
- Receipts | SROIE dataset | No en RVL-CDIP
- Memo | Tobacco-3482 dataset | Sí en RVL-CDIP
- News | Tobacco-3482 dataset | Sí en RVL-CDIP
- Note | Tobacco-3482 dataset | No en RVL-CDIP
- Report | Tobacco-3482 dataset | No en RVL-CDIP
- Resume | Tobacco-3482 dataset | Sí en RVL-CDIP

Cada clase tiene una media de 350 imágenes. La clase que menos muestras tiene es resume, con 112. La que más muestras tiene es memo, con 612. El 80% de cada clase se destina a train y el 20% a test. Excluimos de la partición train/test 60 muestras que utilizaremos para probar nuestro modelo.

Este *dataset* ha sido generado en el notebook 1 adjuntado y se ha hospedado en en:

https://huggingface.co/datasets/Mijavier/10_classes_custom_dataset_donut

En la documentación se adjuntan los *datasets* empleados en los notebooks 3 y 4 para la evaluación del modelo DocVQA y la puesta en producción end-to-end.

6.5. LIMITACIONES DE HARDWARE

Se realizan los entrenamientos en el entorno Google Colab. A pesar de utilizar una cuenta Pro que da acceso a la GPU Tesla P100-PCIE con 16Gb de RAM, todavía existen limitaciones en ejecuciones muy largas. Los entrenamientos alcanzan las 6 horas, por lo que se ha tenido que lidiar con el problema de las desconexiones.

Con mayor tiempo y mejores recursos, se podría ejecutar los modelos en otro entorno.

6.6. TECNOLOGÍAS EMPLEADAS

Se ha empleado el lenguaje de programación Python para todos los modelos, en la versión Python 3.7.

Respecto a las librerías, se han utilizado principalmente pytorch, transformers, NumPy, Pandas, json, mlflow, datablricks, huggingface_hub y tdqm.

6.7. MÉTRICAS DE EVALUACIÓN

Se han utilizado las métricas de la función de pédidas, accuracy, precision, recall, F1-score y support.

6.8. MODELOS DISEÑADOS

Como se ha definido anteriormente, para el problema de clasificación de documentos, se ha diseñado un modelo propio finetuneando el modelo Donut original y otro modelo finetunendo el modelo preentrendo RVL-CDIP. Mientras que, para la tarea de extracción de datos, se ha empleado directamente el modelo oficial DocVQA Donut.

A continuación, nos centraremos en los modelos entrenados para la clasificación de documentos. El primer modelo se nombra como DONUT_FT, en el que se aplica la técnica de *fine-tuning* sobre el modelo Donut base. El segundo modelo se nombra como DONUT_TL_FT, en el que se aplican las técnicas de *transfer learning* y *fine-tuning* sobre el modelo Donut base.

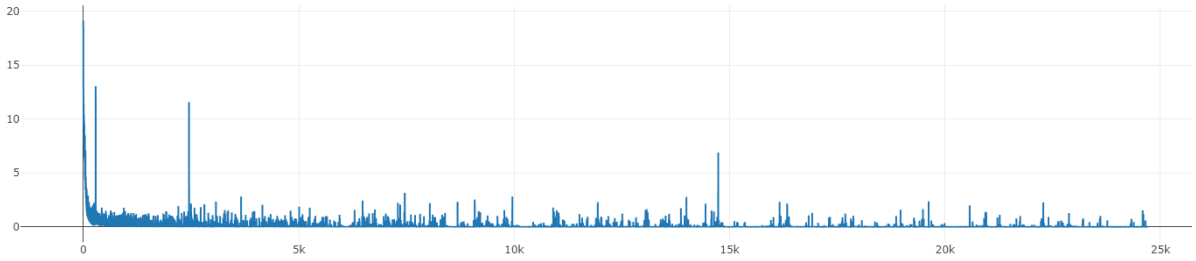
En el entrenamiento de ambos modelos se han empleado un un *batch size* de 1 y un *learning rate* de 1e-05. Se ha establecido un *batch size* de 1 para evitar posibles problemas de memoria al ejecutar los modelos en el entorno de Google Colab. Se entrenan ambos modelos durante 10 épocas durante unas 6 horas. Si bien, el modelo DONUT_FT no ha finalizado el entrenamiento de las 10 épocas, deteniéndose al final de la época 9, por problemas de desconexión de Colab después de más de 5 horas de entrenamiento.

6.9. RESULTADOS DE LOS MODELOS DE CLASIFICACIÓN

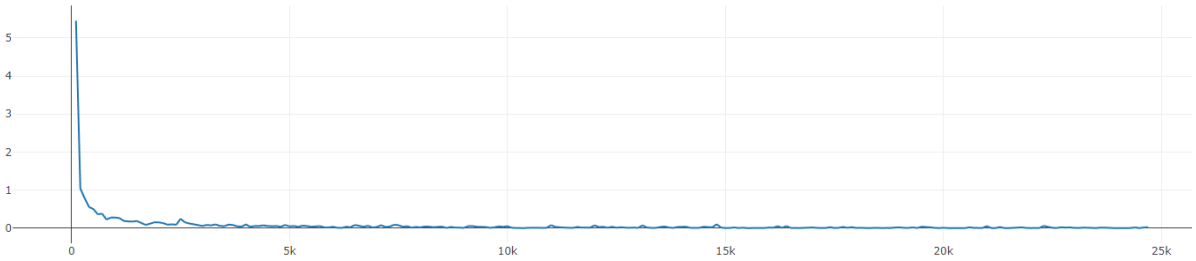
Los modelos entrenados han sido monitorizados en Databricks con MLFlow. Gracias a ello, podemos observar y comparar el rendimiento de los entrenamientos de forma ordenada y clara.

				Metrics			Tags		
Duration	User	Source	Models	loss	loss_100_ma	lo	MODEL	Note	TL model
	mijavi...	ipy...	-	4.253e-5	0.024		DONUT_FT	Model st...	-
6.0h	mijavi...	ipy...	pytorch	2.789e-5	8.927e-4		DONUT_TL_FT	-	naver...

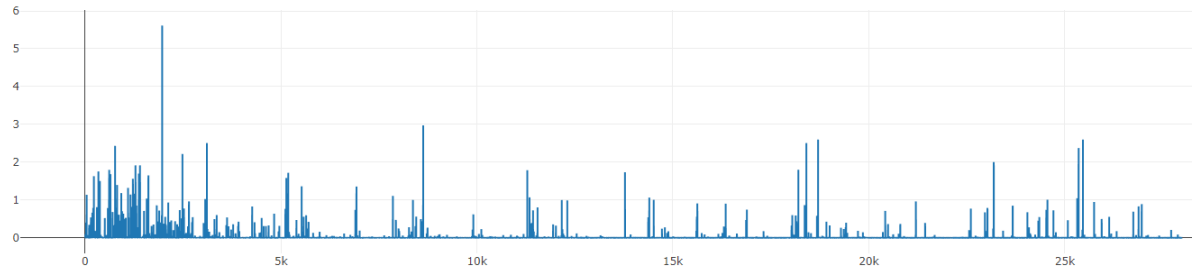
Resultados obtenidos de los modelos DONUT_FT y DONUT_TL_FT. Se representa la menor métrica *loss* y la menor métrica 100 *mean average loss*.



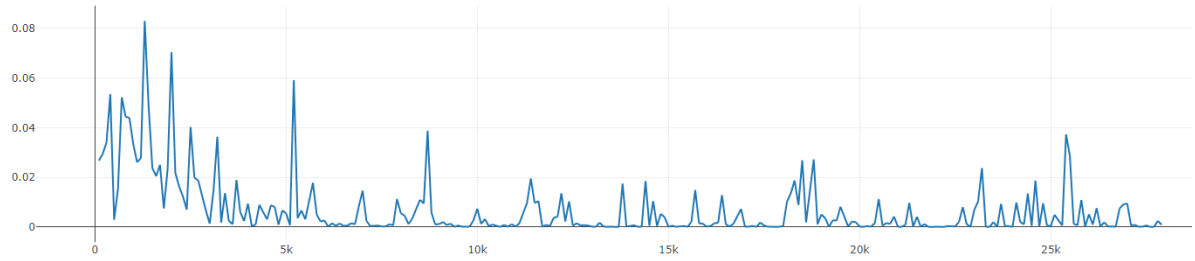
Loss por *step* del modelo DONUT_FT



100 *mean average loss* por *step* del modelo DONUT_FT



Loss por *step* del modelo DONUT_TL_FT



100 *mean average loss* por *step* del modelo DONUT_TL_FT

NOTA 1: En el notebook 2 se incluyen las métricas accuracy, precision, recall y F1-score obtenidas con el modelo DONUT_TL_FT.

NOTA 2: En el notebook 3 se incluyen las métricas evaluación del modelo Donut DocVQA.

En el modelo *DONUT_TL_FT* se ha alcanzado una precisión del 98.63% y una automatización del 62.43% para la tarea de clasificación de documentos. El modelo Donut DocVQA, ha obtenido una precisión del 100.00% y una automatización del 95.24% para la tarea de extracción de datos

7. Conclusiones

Muchos de los métodos actuales de comprensión visual de documentos confían la tarea de lectura de textos a motores OCR. Aunque estos enfoques basados en el OCR han mostrado un rendimiento prometedor, sufren de elevados costes computacionales por el uso del OCR y de la inflexibilidad en los idiomas.

Para resolver estos problemas, en este caso práctico, se emplea el modelo *OCR-free Document Understanding Transformer* (Donut) [4]. Es un modelo basado en *transformers* simple pero eficaz. Alcanza los mejores resultados en varias tareas de comprensión de documentos.

Se ha demostrado que el modelo Donut es capaz de alcanzar grandes resultados en varias tareas de VDU en términos de velocidad y precisión sin utilizar técnicas de OCR. Procesa adecuadamente documentos poco nítidos compitiendo, incluso, con la lectura de un humano. Muchos autores aseguran que se trata de un método más preciso que los métodos tradicionales de OCR.

En este trabajo se han alcanzado los objetivos propuestos. Se ha conseguido procesar con éxito los documentos testeados, clasificarlos, obtener información de estos, reubicarlos y validarlos con los datos aportados por el cliente. Se ha alcanzado una precisión del 98.63% y una automatización del 62.43% para la tarea de clasificación de documentos. Mientras que, para la tarea de extracción de datos, se ha obtenido una precisión del 100.00% y una automatización del 95.24%. Se cumplen, de esta forma, con los KPIs para el equipo de IA.

Por supuesto, esto resultados hay que saber interpretarlos, puesto que el *dataset* empleado en estos modelos difiera bastante de los documentos que negocio requiere en la vida real. Será necesario, por tanto, el acceso a este tipo de documentos para realizar entrenamientos más ajustados al problema. Sin embargo, los experimentos realizados son prometedores y, a falta de una experimentación en un *dataset* más acorde, hacen viable su aplicación práctica.

8. Líneas futuras

La comprensión de los documentos es una tarea vital con grandes avances en los últimos meses, incluso días. Como continuación de la investigación en el caso práctico, en primer lugar, consideraría adquirir un *dataset* más acorde a las necesidades del negocio y comprobar la eficacia de los modelos.

Por otro lado, consideraría el uso de técnicas de mejora de las imágenes proporcionadas. También se podría realizar un análisis de metadatos de los documentos que permita detectar errores en los mismos o ayudas a los modelos para la clasificación y de extracción de datos. Con un mayor número de experimentos, se podrían optimizar distinto parámetros de los modelos como el *batch size* y *learning rate*. Por otro lado, consideraría el modificar el modelo para obtener los datos de validación en entrenamiento.

Aunque se eligió el modelo Donut por encima de los modelos LayoutLM [15], se podrían estudiar, implementar y comprar resultados con el modelo Donut.

Respecto a la extracción de la firma digital o manuscrita, se podría emplear el uso del modelo YOLOv5 [12] haciendo *fine-tuing* sobre un *dataset* de firmas manuscritas, como el siguiente:

https://www.kaggle.com/datasets/victordibia/signverod.

YOLOv5 es un modelo con el que tengo bastante práctica y estoy seguro de que resolvería con éxito el problema. Tengo dos trabajos publicados en GitHub donde hago uso del modelo, con inferencias en imágenes y vídeos: YOLOv5m training using finetuning and data augmentation on a custom labeled dataset: https://github.com/javier-marti-isasi/YOLOv5m-training-using-finetuning-and-data-augmentation-on-a-custom-labeled-dataset y Finetuning YOLOv5m on custom dataset: https://github.com/javier-marti-isasi/Finetuning_YOLOv5m_on _custom_dataset

Respecto a la monitorización y reajuste del modelo, se podrían emplear técnicas de *active learning*, como las propuestas en el punto 5.1, *continuous training* o la activación de alertas tempranas. Estas técnicas podrían ajustar el modelo automáticamente en función de ciertas activaciones, como la

disponibilidad de nuevos datos o una caída en la precisión del modelo. Estos nuevos datos podrían venir de la clasificación manual de *backoffice*, especialmente de ayuda ya son los datos con los que el modelo tuvo un nivel de confianza bajo en la predicción. Todos estos flujos podrían generarse en *MLFlow*.

En la tarea de extracción de los datos de los documentos, se podría estudiar el modelo *Document Parsing* de Donut, implantarlo y comparar resultados con los obtenidos con el modelo Donut DocVQA. Se podría utilizar un modelo preentrenado y/o aplicar *fine-tuning*.

Este notebook es un tutorial para aplicar la inferencia sobre el modelo Donut para la tarea de document parsing: https://colab.research.google.com/github/ NielsRogge/Transformers-Tutorials/blob/master/Donut/CORD/Quick_inference_ with_DONUT_for_Document_Parsing.ipynb

Este notebook es un tutorial para finetunear el modelo Donut para la tarea de document parsing: https://colab.research.google.com/github/NielsRogge/ Transformers-Tutorials/blob/master/Donut/CORD/Fine_tune_Donut_on_a_ custom_dataset_(CORD)_with_PyTorch_Lightning.ipynb

9. Bibliografía

[1] https://towardsdatascience.com/ocr-free-document-understanding-with-donut-1acfbdf099be

[2] https://huggingface.co/docs/transformers/tasks/image_classification

[3] https://arxiv.org/abs/2111.15664

[4] https://github.com/clovaai/donut

[5] https://huggingface.co/spaces/nielsr/donut-rvlcdip

[6] https://github.com/NielsRogge/Transformers-Tutorials/tree/master/Donut

[7] https://github.com/NielsRogge/Transformers-Tutorials

[8] https://pytorch.org/docs/stable/data.html#data-loading-order-and-sampler

[9] https://wandb.ai/jack-morris/david-vs-goliath/reports/Does-Model-Size-Matter-A-Comparison-of-BERT-and-DistilBERT--VmlldzoxMDUxNzU

[10] https://www.toshibacenter.es/que-es-el-omr-icr-y-ocrcomo-funcionan/

[11] https://arxiv.org/abs/2007.00398

[12] https://medium.com/red-buffer/signature-detection-and-localization-using-yolov5-algorithm-7176ed19fc8b

[13] https://medium.com/the-point-collections/intro-to-mlflow-with-colab-part-1-2-beb80c960ad9

[14] https://medium.com/the-point-collections/intro-to-mlflow-with-colab-part-2-2-ae03ffd3930b

[15] https://arxiv.org/abs/1912.13318

[16] Adrian Rosebrock, PhD. OCR with OpenCV, Tesseract, and Python. Intro to OCR

[17] Adrian Rosebrock, PhD. OCR with OpenCV, Tesseract, and Python. OCR Practitioner Bundle