Actas Capitulares de la Habana

Edian Broche Castro
Roger Fuentes Rodríguez
Kevin Manzano Rodríguez
Massiel Paz Otaño
Jackson Claudio Vera Pineda

Índice

1. Introducción **1**
1.1. Objetivo del Proyecto 1
1.2. Contexto 1

2. Metodología **2**
2.1. Enfoque del Proyecto 2
2.2. Recursos Utilizados 3

3. Resultados **5**
3.1. Logros Principales 5
3.2. Desafíos Superados 8

4. Conclusiones **8**
4.1. Impacto del Proyecto 8
4.2. Recomendaciones para Futuras Implementaciones 8

1. Introducción

1.1. Objetivo del Proyecto

Se tiene como objetivo desarrollar una propuesta de transcripción y procesamiento de un conjunto de imágenes, para su posterior uso como dataset en futuros proyectos para identificar entidades nombradas y crear grafos de conocimiento.

1.2. Contexto

Se tiene una serie documental perteneciente al fondo Ayuntamiento de La Habana del Archivo Histórico de la Oficina del Historiador de la Ciudad de la Habana.

Esta serie documental se divide en dos grupos o subseries: los libros originales (1550 - 1898) y los libros trasuntados (1550 - 1809). Los primeros destacan por su riqueza de contenido y forma; los segundos por dejar constancia de la labor del Ayuntamiento para garantizar la perdurabilidad en el tiempo de este tipo documental, al ser copias realizadas en la segunda mitad del siglo XIX.

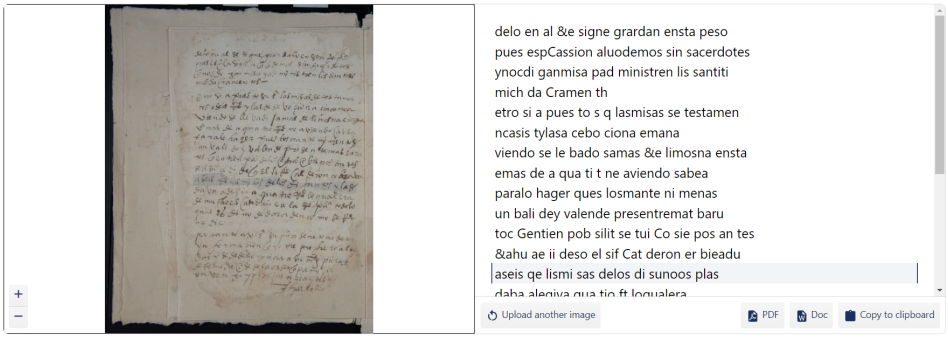
Las actas dejan la huella de una institución colonial y su devenir en el tiempo: el Ayuntamiento. Ellas recogen los planteamientos y discusiones de aquellos problemas que interesaron a los pobladores del lugar, ya fueran de índole económica, política o social; están reflejados los hechos más significativos de cada una de las épocas.

Se tiene como corpus el tomo 1 digitalizado, el resto está en proceso de digitalización.

2. Metodología

2.1. Enfoque del Proyecto

Existen múltiples herramientas para la transcripción de documentos históricos entre estas destacan Transkribus la cual fue recomendada por el cliente por su uso en anteriores trabajos, pero debido al deterioro de los documentos además del tipo específico de tipografía los resultados no eran satisfactorios. Debido al bajo **accuracy** provisto por esta herramienta creada por especialistas en el tema podemos asumir que el problema es bastante complejo.



Nos encontramos con dos grandes problemas: falta de datos para llevar a cabo el entrenamiento de algún modelo, pues muchos de los datasets encontrados no poseen el idioma requerido (Español) ni el tipo de letra (procesal-cortesana); y una tarea que solo un ojo humano especializado podría realizar (tipógrafo)

Debido a todo lo expuesto, el proyecto fue enfocado en una búsqueda exhaustiva de datos clasificados, un procesamiento minucioso de las imáge-

nes, el uso de varios OCRs, y un posprocesamiento con diccionarios del idioma y LLM.

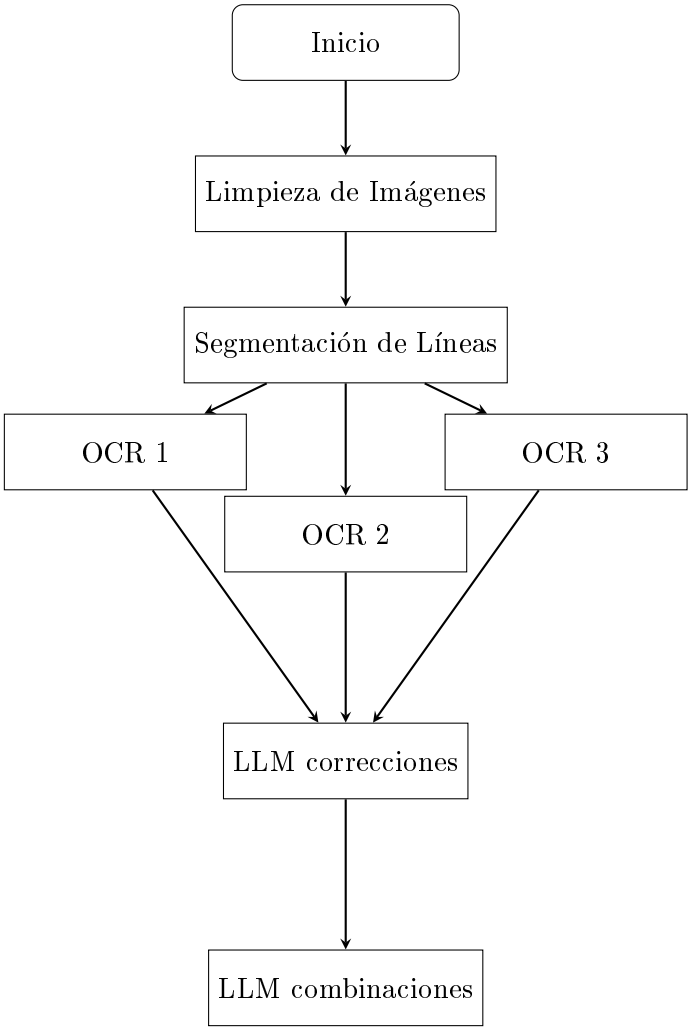


Figura 1: Diagrama de flujo con tres OCR que convergen en el LLM y un loop de retroalimentación.

2.2. Recursos Utilizados

Como lenguaje principal se utilizó Python, debido a que brinda un fácil acceso a bibliotecas de procesamiento de texto e imágenes. Entre las utilizadas destacan: **Kraken**, **PIL**, **TensorFlow**, **Numpy**, **matplotlib**, **scipy**, **spacy**, **symspellpy**, **transformers**, **cv2**, **sam2**.

El LLM utilizado fue **Gemini** por cuestiones económicas, y como diccionario de frequencias, spanish frequency dictionary (se consideró la creación

de un diccionario a partir de un libro escrito por Luis XV, pero se tendrían muy pocas palabras en comparación con el utilizado). El modelo de spacy empleado fue **es core news sm**.

Gemini emplea una arquitectura tipo transformer basada en autoatención, similar a los enfoques modernos de modelos de lenguaje (**LLM**). Además, se ha entrenado específicamente en un corpus mixto multilingüe que incluye textos en español. Aunque no es un modelo especializado en español antiguo, su mejor manejo del idioma y su facilidad de uso fueron factores decisivos.

En contraste, **GPT 2** no solo se centra mayoritariamente en inglés, sino que su despliegue eficiente requiere infraestructuras y optimizaciones adicionales. **T5** (incluso en variantes como **T5 base** o **T5 large**) tiende a ser más pesado en cuanto a recursos computacionales, haciendo que su uso para posprocesamiento resulte más costoso.

Por tanto, se optó por **Gemini** como un equilibrio entre rendimiento, soporte de español, punto clave en el proyecto al manejar un gran volumen de documentos y requerir un ciclo iterativo de corrección, donde la velocidad y la reducción de costos son fundamentales.

Entre los datos recopilados inicialmente se utilizó dataset de rodrigo. Este, es un dataset de Español antiguo, pero la tipografía no coincide, pues en este caso se utiliza letra gótica y nuestro (se necesita que sea procesalcortesana). Luego, se encontró todo un corpus de documentos en Español de años anteriores a 1900: CODEA y con una estructura que se puede aprovechar. La mayoría de los documentos presentan una triple representación: la imagen, una transcripción paleográfica, y una presentanción crítica. El dataset inicialmente tiene alrededor de 4000 documentos, pero luego de filtrar por el tipo de letra requerida se reduce a 546, recalcando la dificultad de los datos.

SimpleHTR se basa en una red neuronal recurrente (**LSTM**) que aplica **Connectionist Temporal Classification** (**CTC**) para alinear secuencias de caracteres sin necesidad de segmentarlas rígidamente. El modelo está entrenado en manuscritos latinos, cercanos en estructura al español antiguo, lo que lo hace relativamente apropiado para este tipo de documentos.

sinai sam rec v4 best emplea una arquitectura **CRNN** (red neuronal combinada de convoluciones y **LSTM**), común en **Handwritten Text Recognition** para capturar tanto características espaciales como temporales. Está optimizado para manuscritos históricos, lo que facilita el reconocimiento de trazos irregulares o antiguos.

McCATMuS nfd nofix V1 también aprovecha un enfoque neuronal profundo (recurrente y/o convolucional), especializado en la transcripción de textos históricos con variaciones tipográficas. En concreto, se ha ajustado para manejar grafías antiguas y caracteres poco comunes, buscando reducir la tasa de error en documentos premodernos.

AQUI FALTAN AGREGAR COSAS (resultado de cada OCR)

3. Resultados

3.1. Logros Principales

En la primera etapa del producto se logra una mejora considerable en la imagen, haciendo uso de un pipeline para limpiar las imágenes, llevándola a escala de grises, y aplicando técnicas de filtrado como filtro Gaussiano, para disminuir el ruido general de la imagen, ecualización de histograma para mejorar el contraste de la imagen y resaltar la letra en los documentos, y operaciones morfológicas de erosión y dilatación para mejorar el trazo en los documentos. Finalizando con una binarización de la biblioteca *Kraken*, con modelos basados en redes neuronales profundas y binarización adaptativa, y entrenados con documentos históricos. La figura 2 muestra el resultado de una foto pasando por las etapas claves del procesamiento.

Para el tratado de las manchas debido a la longevidad de los documentos(son amarillos) se implementó un conversor personalizado de la imagen de color a escala de grises, ponderando las componentes rgb de forma diferente a lo usual (que es el promedio);$r + g$ es el equivalente al amarillo por lo que se ponderó r y g en menor grado variando los parámetros para tratar de contrarrestar el ruido.

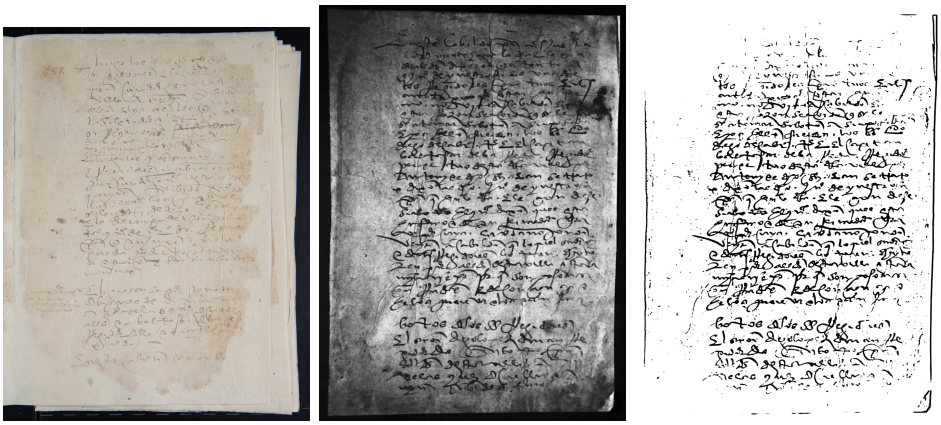


Figura 2: Original, en escala de grises con letra reconstruida, imagen binarizada

Por otra parte, se intentó el uso de algoritmos de detección de bordes como *sobel*, *Canny*, y *laplaciano* , el último no funcionó bien porque depende de la segunda derivada de los cambios pixel a pixel, y esto en documentos de texto, donde el documento es blanco y las letras son negras, sumado a que los cambios de papel tinta son muy abruptos, indefine la derivada y causó

malos resultados. En los otros dos casos el ajuste fue aceptable en algunos casos como se muestra en la figura 3, pero no así en la mayoría de los documentos.

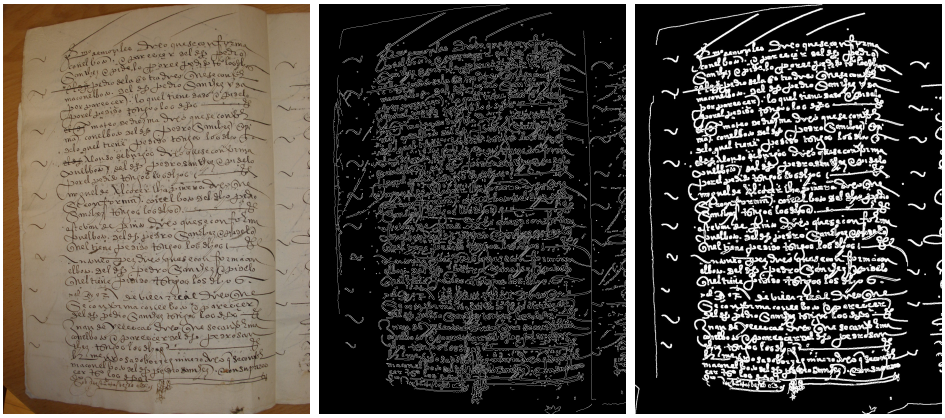


Figura 3: Original, detección de bordes, operaciones morfológicas de izquieda a derecha

Se intentó además el uso de filtros de mediana, para aminorar el ruido *sal y pimienta*, así como filtro bilateral, sin mejorías significativas en los resultados. Se probó la binarización de la imagen, definiendo un umbral de intensidad donde si $x > threshold$ se toma como letra y en caso contrario como fondo, pero en zonas de sombra, probando con diferentes valores para threshold no se detecta bien la escritura, también se intentó binarización adaptativa convencional pero se recuperaba mucho ruido y manchas en el documento.

Luego de binarizadas las imágenes son pasadas por dos procesos de segmentación, uno que brinda **Kraken** y una proyección de histogramas (se sugiere utilizar la primera pues otorga mejores resultados), y en caso de un proceso manual sería mejor utilizar **SAM2**, pero no logramos automatizar el proceso. Sin embargo, esta última herramienta, aunque poderosa, no ofrece resultados tan satisfactorios, pues en el momento de encerrar manualmente en *cajas* a cada línea del documento, es necesario que estas (las líneas) sean lo más rectas posibles; lo cual se hace complejo de encontrar dado que se trata de documentos manuscritos.
Luego de segmentada la imagen, esta es pasada a 3 OCRs distintos: **SimpleHTR**, **sinai-sam-rec-v4-best**, **McCATMuS-nfd-nofix-V1**, los cuales son modelos especializados en **handrwitten recognition**. Los dos últimos provienen de Kraken. Cada uno tuvo que tratarse de forma diferente dado que el primero necesita la imagen segmentada, y los otros solamente los **bounding box**.

En la última etapa, una vez procesadas la imágenes con las técnicas para mejorarlas y extraído el texto con los diferentes OCRs, se realizó un posprocesamiento para mejorar la calidad de las transcripciones. Inspirados en el DataSet de CODEA, se brindó una transcripción crítica del documento. Dada la naturaleza de los documentos y los errores inherentes al OCR, implementamos un pipeline para corregir errores ortográficos y ajustar la chorencia del texto, refinando la salida.

Se utilizó un modelo de lenguaje **Spacy** para segmentar el texto en tokens, para así identificar palabras y elementos no alfabéticos. Cada token fue corregido utilizando *SymSpell*, una herramienta basada en un diccionario de frecuencias, para separar correctamente palabras unidas y capitalizarlas. Tras las correcciones, se utilizó un modelo generativo, *Gemini* para realizar un refinamiento semántico y estilístico, corrigiendo así el formato del texto y la gramática, además de mantener el contexto histórico del español antiguio. También se consideraron las diferentes salidas y se combinaron en una sola.

Para unificar las tres salidas de OCR (SimpleHTR, sinai-sam-rec-v4-best y McCATMuS-nfd-nofix-V1) en una sola versión depurada, empleamos el mismo LLM (Gemini) con un enfoque de prompt colaborativo. Se le proporcionó como entrada las tres transcripciones generadas, por ejemplo en formato:

Texto A: (Salida OCR 1 luego pasada al LLM) Texto B: (Salida OCR 2 luego pasada al LLM) Texto C: (Salida OCR 3 luego pasada al LLM) A continuación, se formuló la instrucción al LLM para que:

Compare las tres versiones, Identifique coincidencias y divergencias, Corrija inconsistencias ortográficas o gramaticales, Genere una transcripción unificada que mantuviese la fidelidad al original histórico. Dado que el LLM ya contaba con un contexto de idioma español y cierta base en ortografía "antigua" (aunque no perfecta), pudo ponderar en cada fragmento cuál de las tres salidas era más coherente y acercarse a la mejor hipótesis conjunta. Este procedimiento automatiza la "votación" o ensamblaje de hipótesis sin necesidad de implementar manualmente reglas de combinación. El resultado final es un texto donde los errores u omisiones de un OCR se compensan con los aciertos de los otros, y se afina la coherencia mediante la capacidad generativa del modelo.

A continuación un ejemplo del flujo:

1. Essta es una prueva de ectraccion de teexto. Connoscida cosa sea a todos los queesta carta uieren como yo don Fferrando por la gracia de dios hey de Castiella

2. Esta es una prueba de extracción de texto. Conocida cosa sea a todos

los que esta carta vieren como yo don Fferrando por la gracia de Dios hay de Castilla

3. Esta es una prueba de extracción de texto. Conocida cosa sea a todos aquellos que esta carta vieran, como yo, don Fernando, por la gracia de Dios, rey de Castilla.

3.2. Desafíos Superados

En el momento de la segmentación jugó un papel fundamental la búsqueda de parámetros adecuados para que tuviera mejor rendimiento los procesos posteriores.

Entre los desafíos superados en el posprocesamiento nos encontramos con la conservación del español antiguo, pues no hallamos un diccionario de frecuencias de esa época; sin embargo, solventamos este problema con el LLM utilizado, la obtención del mismo fue compleja, pues se probaron con otros, por ejemplo: **GPT-2**, **"google/mt5-small"**, **flax-community/spanish-t5-small**, pero los resultados fueron pésimos en los tres casos, además de que no teníamos un API cómoda y los modelos eran bastantes pesados, por lo que optamos por **Gemini**.

4. Conclusiones

4.1. Impacto del Proyecto

El proyecto planea ser utilizado para la transcripción de documentos históricos archivados en la oficina del historiador, inicialmente se tiene un solo tomo, pero luego de terminar el proceso de digitalización de los siguientes tomos ya se podrá directamente pasar a transcripción para su posterior análisis. También puede utilizarse el resultado del mismo para convertirlo en producto reutilizable con el objetivo de crear alguna herramienta para encontrar entidades, y poder realizar búsqueda de interés, para encontrar información valiosa en estos documentos.

4.2. Recomendaciones para Futuras Implementaciones

Se recomienda indagar sobre la posibilidad de usar el algoritmo *DBSCAN* para la segmentación en este tipo de documentos, debido a la naturaleza curva e irregular de las líneas en documentos manuscritos con tipografía*procesal-cortesana*. Luego, seguir utilizando **SimpleHTR**, pero se recomienda tener un dataset para hacerle un **fine tunning** al igual que al modelo generativo que no es específicamente para esta tarea. Se pudiera volver a utilizar **GPT-2** pero entrenado en esta cuestión y debería de obtener buenos resultados.

Referencias

Referencias

[1] Vicente Marcet Rodriguez *El copus de documentos de Ávila del Hispanic Museum and Library (siglos XV y XVI). Descripción y análisis paleográfico y gráfico-fonológico.*

[2] A. Granet, E. Morin, H. Mouchère, S. Quiniou, y C. Viard-Gaudin, *Transfer Learning for Handwriting Recognition on Historical Documents*, ICPRAM, 2018.

[3] S. Torres Aguilar, *Handwritten Text Recognition for Historical Documents using Visual Language Models and GANs*, ArXiv, 2025.

[4] E. Granell, E. Chammas, L. Likforman-Sulem, C. D. Martínez-Hinarejos, C. Mokbel, y B. I. Cîrstea, *Transcription of Spanish Historical Handwritten Documents with Deep Neural Networks*, Journal of Imaging, vol. 4, no. 15, 2018.

[5] THI TUYET HAI NGUYEN, ADAM JATOWT, MICKAEL COUSTATY and ANTOINE DOUCET *Survey of Post-OCR Processing Approaches*, 2025.

[6] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, Haoshuang Wang *PP-OCR: A Practical Ultra Lightweight OCR System*

[7] Prof. Anuradha Thorat, Mayur Zagade, Shivani More, Manish Pasalkar, Anand Narute *Research Paper on Text Extraction using OCR*, IJARSCT, Volume 3, Issue 14, May 2023

[8] Ojas Kumar Barawal, and Dr Yojna Arora *Text Extraction from Image* , IJIREM, Volume 9, Issue 3, May 2022