Proyecto LUISA – Subproyecto OCR
Reporte de resultados parciales

Ignacio Ramírez

26 de julio de 2021

1. Resumen

Presentamos un reporte de resultados parciales en lo que respecta a los esfuerzos para mejorar el desempeño del sistema de transcripción automática (OCR) utilizado en LUISA. Concretamente, se evalúan las mejoras obtenidas al utilizar el OCR Calamari, entrenado en un conjunto de datos extraído de LUISA, en comparación con los resultados disponibles hasta la fecha usando el sistema Tesseract 4.1. Las mejoras al momento son muy significativas, y se espera que sean aún mayores cuando se disponga del conjunto de datos completo.

2. Introducción

El objetivo del proyecto LUISA es el de transcribir los documentos del Archivo Berruti a partir de sus versiones digitales, obtenidas mediante escaneo.

En primera instancia se obtiene una transcripción automática usando un OCR. Esta transcripción es evaluada mediante una heurística y catalogada como buena o mala. Las transcripciones dadas como muy buenas son almacenadas directamente para su uso en las siguientes etapas. Las transcripciones malas son enviadas al sistema de transcripción colaborativo (el web de LUISA).

Al ser completadas, las transcripciones manuales son almacenadas junto con las automáticas.

El OCR entonces cumple dos funciones:

- transcribir documentos, en la medida de sus posibilidades
- evaluar la calidad de un documento

La gran ventaja de los OCR es su velocidad: es posible procesar en pocos días el Archivo Berruti completo. La desventaja es la calidad: los OCRs no son tan buenos como los humanos, en especial en documentos/imágenes muy deteriorados.

Por otra parte, las transcripciones generadas manualmente son relativamente pocas, pero de muy alta calidad.

El objetivo principal del esfuerzo actual en lo que refiere a OCR es el de juntar lo mejor de ambos mundos: la velocidad de los OCR y la calidad de LUISA.

La estrategia en cuestión es utilizar las transcripciones manuales como datos para entrenamiento de sistemas OCR. Teniendo esto último en cuenta, el sistema LUISA también cumple dos funciones:

- transcribir documentos
- generar datos de entrenamiento y validación de OCR

3. Metodología

En resumen, la idea es usar los datos de transcripción manual de LUISA para entrenar un sistema OCR para nuestro caso particular, y ver si eso mejora el desempeño del sistema *off-the-shelf* Tesseract utilizado actualmente.

3.1. Datos

Los datos de LUISA son de muy alta calidad en lo que refiere a las palabras transcritas, pero también contienen a menudo errores humanos y por sobre todo anotaciones que no deberían ser parte del texto: todo esto es de poco impacto para el sistema de extracción de información (objetivo ppal. de LUISA). Por ejemplo, no es un problema para este último la aparición de anotaciones al margen de la transcripción como "aquí hay una raya vertical". Sin embargo, ese tipo de anotaciones, o cambios en la puntuación, capitalización, etc., tienen el potencial de confundir severamente al OCR durante su entrenamiento.

Tanto con el OCR como con LUISA, los textos son transcritos de a una linea de texto a la vez. En ambos casos se utilizan las mismas lineas, que son recortadas mediante un algoritmo de preprocesamiento propio. Si bien este algoritmo no es perfecto, al ser las mismas imágenes en ambos casos, sabemos que eso no es un factor determinante en la diferencia de calidad entre los distintos modelos evaluados.

3.2. Curación

Por lo anterior, los datos de transcripción manual deben ser *curados* manualmente. Concretamente, en este proceso se hace lo siguiente:

- eliminar datos confusos o que no aportan (ej., lineas con ruido)
- corregir transcripción textual
- recortar las imágenes (lineas de texto) para eliminar basura en los bordes

El proceso de curación se realiza manualmente por voluntarios mediante una aplicación de escritorio y una aplicación web. Al momento de escribir este documento se disponía de unas 13000 lineas de texto curadas, de un total de aprox 48000 que falta curar.

3.3. Reescalado

Las imágenes del Archivo Berruti son *binarias*, es decir, los pixeles son o blanco o negro, sin escalas de grises. La resolución de escaneo es alta, entre 300 y 400 dpi (pixeles por pulgada lineal). Además de ese formato, para el entrenamiento probamos también con versiones reducidas en un 50 % (en ambas dimensiones) donde los pixeles reducidos son en 256 tonos de gris; este aumento en la precisión de los pixeles (de 2 niveles a 256) permite alivianar la pérdida de información generada al reducir la imagen, y permite acelerar los algoritmos, al ser imágenes más pequeñas.

Los datos en su formato original son identificados como *full1*, mientras que los datos reescalados los identificamos como *half8*.

3.4. Entrenamiento

Una vez curados, los datos son utilizados para el entrenamiento y la evaluación de OCR. Para esto se dividen, como es usual, en 3 partes: entrenamiento, validación y test.

Para el entrenamiento utilizamos el sistema de OCR Calamari, desarrollado por la Biblioteca Estatal de Berlín (SBB). Este es un OCR bastante típico y de una arquitectura muy similar a la de Tesseract: se basa en redes neuronales convolucionales combinadas con redes recurrentes tipo LSTM (Long-Short Term Memory) más un conjunto de heurísticas para preprocesar las lineas de texto.

El Tesseract, desafortunadamente, no pudo ser entrenado. Si bien tiene la capacidad de hacerse, la documentación es casi inexistente y confusa. El Calamari, por el contrario, funciona mejor. No menor es el hecho de que Calamari está basado en TensorFlow de Python, lo cual lo hace extremadamente eficiente en comparación con Tesseract, en particular en máquinas con GPUs potentes.

3.5. Evaluación

Una vez entrenado el Calamari, se aplica sobre el conjunto de test, lo cual genera un archivo de texto por cada linea que se le presenta; la extensión por defecto de estos archivos es `.pred.txt`. Luego se hace lo propio con el Tesseract; la extensión de los archivos es `.tess.txt`. Los textos transcritos manualmente son usados como referencia, y se almacenan en archivos con extensión `.gt.txt` (gt viene de ground truth). El resultado de la evaluación se expresa en CER (Character Error Rate). Hay varias medidas de CER. Tomamos como referencia la que usa Calamari. El resultado final sobre el conjunto de test es simplemente el CER promedio para todo el conjunto de test.

El conjunto de test consiste en un 10 % de los datos curados disponibles, tomados al azar. Al momento, son 1283 lineas de texto.

4. Resultados

Como se dijo anteriormente, los datos curados disponibles al momento constituyen menos de 1/3 del total de lineas de texto transcritas disponibles. Es de esperar que los resultados mejoren significativamente al culminarse el proceso de curado.

Los resultados abajo fueron obtenidos usando los datos en escala de grises y reducidos. Se repetirá lo anterior para los datos de 1 bit a resolución original más adelante.

La presentación es la que se obtiene con la utilidad `calamari-eval`, que calcula el CER dados los *ground truth* y las salidas de OCR correspondientes.

4.1. Tesseract

Se ejecutó el Tesseract 4.1 con los siguientes parámetros:

- `--dpi 150` resolución de 150 pixeles por pulgada lineal de las imágenes escaneadas
- `--lang spa` modelo preentrenado para lenguaje Español
- `--psm 13` entrada son lineas de texto pre-segmentadas

```
Evaluation result
=================

Got mean normalized label error rate of 38.74%
(24854 errs, 64154 total chars, 27203 sync errs)
GT       PRED     COUNT    PERCENT
{a}      {e}           223      0.82%
{ }      {}            183      0.67%
{e}      {o}           183      0.67%
{o}      {e}           160      0.59%
{o}      {a}            98      0.36%
{}       {.}            73      0.27%
{e}      {a}            70      0.26%
{.}      {,}            48      0.18%
{-}      {}             47      0.17%
{a}      {o}            40      0.15%
The remaining but hidden errors make up 95.86%
```

4.2. Calamari

Para el entrenamiento se utilizaron los siguientes parámetros:

```
calamari-train --dataset HDF5 \
--num_threads 8 \
```

```
--train_data_on_the_fly \
--files /luisa/ocr/data/luisa-curados/curados-train.h5 \
--validation /luisa/ocr/data/luisa-curados/curados-val.h5
```

La evaluación no requiere otro parámetro que no sea el modelo generado. El resultado fue el siguiente:

```
Evaluation result
=================

Got mean normalized label error rate of 19.04%
(12213 errs, 64154 total chars, 12978 sync errs)
GT       PRED     COUNT    PERCENT
{ }      {}           253      1.95%
{e}      {o}          213      1.64%
{o}      {e}          136      1.05%
{a}      {e}          121      0.93%
{"}      {''}         120      1.85%
{}       { }          120      0.92%
{e}      {a}           77      0.59%
{a}      {o}           68      0.52%
{.}      {}            56      0.43%
{o}      {a}           54      0.42%
The remaining but hidden errors make up 89.69%
```

4.3. Conclusiones

Mediante el entrenamiento de Calamari logramos reducir la tasa de error (según el CER) de 39 % a 19 %, habiendo explotado tan sólo el 30 % de las muestras disponibles desde LUISA. Es de esperar que se logren obtener resultados significativamente mejores cuando se pueda utilizar el 100 %.

Es muy importante notar que, tanto en Tesseract como en Calamari, buena parte de los errores de transcripción reportados se deben a diferencias en el espaciado (“ ” vs “”), doble comilla simple en lugar de una doble, u otras diferencias en la puntuación (“.” vs. “”). En particular, del error de 19 % obtenido con Calamari, un $1{,}95 + 1{,}85 + 0{,}92 + 0{,}43 = 5{,}12$ % corresponde a este tipo de diferencias, de escasa importancia. Dicho de otra forma, de ese 19 % sólo el 14 % son errores relevantes. En cambio, en Tesseract, estos errores constituyen el $0{,}67 + 0{,}27 + 0{,}18 + 0{,}17 = 1{,}29$ % del 39 % total.

Teniendo lo anterior en cuenta, la reducción en errores *relevantes* para su uso posterior es del 37.5 % al 13.9 %, es decir casi un *tercio*.