Segmentación de artículos periodísticos con transferencia de aprendizaje en redes neuronales convolucionales

Adolfo Simaz Bunzel[1,2], Agostina Filócomo[1,3], Ezequiel A. Pássaro[1]
[1]*Facultad de Ciencias Astronómicas y Geofísicas, UNLP, Paseo del Bosque S/N, La Plata, Argentina*
[2]*Instituto Argentino de Radioastronomía (CCT La Plata, CONICET; CICPBA; UNLP), C.C.5, (1894) Villa Elisa, Buenos Aires, Argentina*
[3]*Departamento de Investigación en Ciencias Exactas e Ingeniería, UNRN - Sede Atlántica, Don Bosco y Leloir S/N, Viedma, Argentina*

29 de mayo de 2024

1 INTRODUCCIÓN

La extracción de información de documentos basados en imágenes es una herramienta muy poderosa para la digitalización de los mismos. Para ello, se utilizan técnicas de *machine learning* específicas como lo es el uso de redes neuronales convolucionales. En el presente informe describimos los pasos seguidos para la extracción de información de diversos recortes de periódicos con un segmentador de instancias *Mask R-CNN*[1].

Para facilitar la reproducción de los resultados aquí mostrados todos los programas desarrollados tienen asociado un enlace a un repositorio con versionado de código. Se ha hecho especial hincapié en las licencias de código abierto de las herramientas utilizadas.

El informe se organiza de la siguiente manera: en la Sección 2 se presenta una descripción de los datos utilizados. En la Sección 3 se detallan los métodos aplicados sobre el conjunto de datos antes mencionado para su segmentación. Finalmente, la Sección 4 contiene un análisis de los resultados encontrados, junto con una breve discusión sobre los mismos.

2 DATOS

El primer paso de la tarea fue la producción de una muestra de datos para ser utilizado en el entrenamiento de las redes neuronales. Para ello, creamos un conjunto de datos en formato *Common Objects in Context* (COCO)[2] a partir de los 501 archivos en formato JSON producidos por los participantes durante la fase de etiquetado comunitario y recopilados por la organización de la competencia. Posteriormente, realizamos un filtrado de la muestra para garantizar la consistencia de la información, dando como resultado un total de 297 imágenes con sus respectivas anotaciones.

En el proceso notamos que, pese a haber entregado en tiempo y forma las 60 anotaciones que le fueron asignadas a este grupo, las mismas no se encontraban presentes en los datos recopilados. Por tal motivo, añadimos al conjunto de datos otras 200 imágenes con sus respectivas anotaciones (las que nos fueron asignadas originalmente, más 140), sumando un total de 497 imágenes.

Así, de esta etapa de procesamiento obtenemos un conjunto de datos a partir del cual entrenamos las redes neuronales, que describimos en la Sección 3. Todas las herramientas para reproducir el conjunto de datos mencionado se encuentran en el repositorio de entrenamiento: `https://github.com/epassaro/diaxi-training-tools`.

[1] `https://arxiv.org/abs/1703.06870`
[2] `https://cocodataset.org/`

3 MÉTODOS

El objetivo del trabajo consiste en la segmentación automática de artículos periodísticos para su posterior procesamiento con un software de reconocimiento óptico de caracteres (OCR, por sus siglas en inglés), `tesseract`[3]. Para ello, decidimos hacer *transfer learning* sobre un modelo de segmentación basado en el análisis de documentos, `PrimaLayout`[4]. En lo que a su entrenamiento respecta, utilizamos una librería de algoritmos de detección y segmentación, `detectron2`[5], desarrollada por *Facebook Research*. Cabe mencionar que el conjunto de datos utilizado en la fase de entrenamiento tuvo que ser modificado para incluir las máscaras de segmentación que no fueron provistas inicialmente, y luego separarlo en subconjuntos de *entrenamiento* y *validación*.

Para una explicación detallada del entrenamiento del modelo, referirse al repositorio de entrenamiento: `https://github.com/epassaro/diaxi-training-tools`.

De esta fase resulta un análisis asociado a la precisión en la detección de las diferentes secciones de una nota periodística sobre el conjunto de *validación*, basado en las regiones de la nota que las redes neuronales detectan, denominadas *bounding boxes*. Los resultados de esta fase se encuentran descriptos con mayor detalle en la Sección 4.

4 RESULTADOS

Debido al rol central que tiene el modelo para realizar la segmentación automática, dividimos la presentación de los resultados asociados al entrenamiento del modelo, y su posterior aplicación en el esquema provisto por la organización.

4.1 Fase de entrenamiento

Aquí presentamos un breve análisis asociado a la calidad de las predicciones hechas por el modelo previamente explicado, respecto a las *bounding boxes* que delimitan las diferentes secciones a obtener de una nota periodística. Para ello, utilizamos una métrica ampliamente utilizada en el campo de la segmentación en redes neuronales convolucionales, denominada intersección sobre la unión (IoU, por sus siglas en inglés), que cuantifica la similitud entre dos *bounding boxes*: la predicha por el modelo entrenado y la provista como valor verdadero. De esta manera, cuanto más se aproxime a 1 esta cantidad,

[3] `https://tesseract-ocr.github.io/`
[4] `https://layout-parser.readthedocs.io/en/latest/notes/modelzoo.html`
[5] `https://detectron2.readthedocs.io/en/latest/`

Figura 1. Segmentos detectados por el modelo luego de la inferencia sobre dos notas que no pertenecen al conjunto de datos de la Sección 2.

AP	AP50	AP75	AP-Título	AP-Volanta	AP-Copete	AP-Cuerpo	AP-Imagen	AP-Destacado	AP-Epígrafe
47.03	70.00	50.33	58.43	34.96	50.35	81.83	65.21	0.00	38.46

Tabla 1. Valores de la métrica correspondientes al modelo entrenado con 2 600 iteraciones. La definición de AP, AP50 y AP75 se puede encontrar en el siguiente enlace: `https://cocodataset.org/#detection-eval`.

mejor será el modelo utilizado. Cabe mencionar que, típicamente, un valor de IoU mayor a 0,5 es considerado una buena predicción.

A partir del modelo entrenado, evaluamos la métrica obtenida utilizando `COCOEvaluator`, provisto por `detectron2`, que mide la precisión promedio (AP, por sus siglas en inglés) de las antes mencionadas IoU. La ventaja de utilizar este evaluador consiste en que provee valores de la métrica separados por clase.

En lo que sigue, detallamos los resultados más importantes de esta evaluación:

- **Comparación en tamaños del conjunto de datos de entrenamiento**: en una primera instancia, utilizamos un subconjunto de 100 notas para el entrenamiento del modelo. Sin embargo, los resultados no fueron lo suficientemente precisos, obteniendo valores AP $< 0{,}5$ en los diferentes campos. Como era de esperarse, el aumento en número de notas mejoró la métrica, de forma que con la muestra total se obtuvieron promedios aún mejores. Consideramos importante mencionar que no fue posible aumentar el tamaño de la muestra con técnicas de *data augmentation* por limitaciones en la memoria de video de la unidad de procesamiento gráfico (GPU, por sus siglas en inglés) proporcionada por el servicio *Google Colab*.

- **Comparación de paramétros libres de la red neuronal**: debido al escaso tiempo para la ejecución del entrenamiento, mayoritariamente debido a las restricciones de tiempo de uso de la GPU proporcionada el servicio *Google Colab*, acotamos la búsqueda de hiperparámetros que optimicen el entrenamiento del modelo. Adoptamos un *learning rate* de 0,00025, como sugiere la documentación de `detectron2`, y un número máximo de 4 000 iteraciones. Una vez finalizado el entrenamiento, se inspeccionaron las curvas de las métricas obtenidas sobre el conjunto de *validación* y se conservó el modelo correspondiente a 2 600 iteraciones, por considerar que a partir de este punto las predicciones no mejoraron. Se obtuvieron métricas mayores al 50 % (AP $> 0{,}5$) en promedio para los campos asociados al título, el cuerpo y las imágenes, como se muestra en la Figura 1.

En resumen, el conjunto de datos utilizados para el entrenamiento del modelo elegido nos permite determinar con gran precisión los campos más importantes asociados a una nota periodística: Título, Cuerpo e Imágenes, como se puede ver en la Tabla 1. En la misma se puede notar el valor nulo para el campo Destacado; esto sucede debido a que el criterio utilizado por los participantes para etiquetar esta sección no fue uniforme.

4.2 Etapa de segmentación automática y reconocimiento de caracteres

La última parte del trabajo consiste en utilizar el modelo entrenado para segmentar las imágenes de acuerdo a la clase detectada, y enviar el resultado al OCR para la extracción del texto. Para ello, la consigna implica incorporar dicho modelo al código fuente provisto por la organización, dejando inmutables ciertas características del mismo, de manera que la ejecución continúe siendo similar a la original.

A su vez, se modularizó el código existente y se añadieron nuevas funciones y dependencias. El modelo final de segmentación implementado es el de 2 600 iteraciones sobre el conjunto de 497 imágenes, y tiene una suma de comprobación `MD5: fdbf58d42a41b4899762b544ea70c11d`.

Una explicación detallada sobre cómo utilizar el código y todas los cambios que se hicieron se encuentra en los archivos `README.md` y `CHANGELOG.md` del repositorio de entrega: `https://gitlab.com/epassaro15/desafio-iaxlaidentidad`.

Por último, se incluyeron en el repositorio de entrega de *GitLab* herramientas de integración continua que garantizan la fiabilidad y calidad del código. Todos los archivos que son resultado del proyecto se encuentran alojados en la siguiente página web: `http://xmm-newton.fcaglp.unlp.edu.ar`.

AGRADECIMIENTOS

Como grupo estamos muy agradecidos de poder colaborar con una organización tan loable como lo es “Abuelas de Plaza de Mayo”. Además, nos gustaría agradecer a quienes han sido organizadores de la competencia, ya que la misma nos permitió aprender sobre un campo del conocimiento sobre el que no teníamos experiencia previa, a la par de contribuir con la sociedad en la búsqueda de la verdad.