

Traductor inglés-español por reconocimiento de texto

Miriam Calderón Reyes, Daniela Jiménez Pano,
Norma Angélica Márquez Sulca, Luis Carlos Padierna García

Universidad de Guanajuato,
División de Ciencias e Ingenierías León,
México

{m.calderon.reyes, d.jimenezpano
na.marquezsulca, lc.padierna}@ugto.mx

Resumen. En la búsqueda continua de soluciones pragmáticas en el ámbito de la inteligencia artificial, presentamos un proyecto concreto que fusiona la visión computarizada y el procesamiento de lenguaje natural. Nuestro enfoque se centra en el desarrollo de un traductor automático que, a través de una Red Neuronal Convolutiva (CNN), analiza imágenes para extraer texto, y posteriormente utiliza una Red Transformer para realizar traducciones precisas. El uso de CNN permite identificar y aislar regiones de texto en imágenes, convirtiendo datos visuales en información legible. Este proceso sirve como punto de partida para la siguiente etapa, donde una Red Transformer se encarga de contextualizar y traducir el texto reconocido. Esta combinación técnica no solo simplifica la experiencia del usuario al eliminar la necesidad de ingreso manual de texto, sino que también demuestra aplicaciones prácticas en entornos donde la comunicación multilingüe basada en imágenes es esencial. Aunque la precisión actual del traductor sea del 60%, se están explorando posibles mejoras para maximizar su eficacia.

Palabras clave: CNN, red Transformer, padding, deep learning.

English-Spanish Text Recognition Translator

Abstract. In the ongoing quest for pragmatic solutions in the field of artificial intelligence, we present a concrete project that merges computer vision and natural language processing. Our approach focuses on developing an automatic translator that, utilizing a Convolutional Neural Network (CNN), analyzes images to extract text, and subsequently employs a Transformer Network to perform accurate translations. The use of CNN enables the identification and isolation of text regions in images, converting visual data into readable information. This process serves as a starting point for the subsequent stage, where a Transformer Network contextualizes and translates the recognized text. This technical combination not only simplifies user experience by eliminating the need for manual text input but also demonstrates practical applications in environments where multilingual image-based communication is essential. Although the current translator accuracy stands at 60%, we are exploring potential enhancements to maximize its effectiveness.

Keywords: CNN, red Transformer, padding, deep learning.

1. Introducción

La historia de las redes neuronales convolucionales (CNN) comienza en la década de 1980, esto sucede cuando Yann LeCun, nacido en Francia en 1960, desarrolló la arquitectura LeNet, dicha arquitectura fue pionera en el reconocimiento de patrones y visión por computadora, dado que era capaz de reconocer dígitos escritos a mano con gran precisión, como presentó en su artículo “Gradient-Based Learning Applied to Document Recognition” [1].

La arquitectura LeNet se diseñó para el reconocimiento de caracteres en imágenes, esto se logró realizando una serie de capas convolucionales y capas de pooling, continuando con capas completamente conectadas para alcanzar su objetivo. Como es habitual, la estructura LeNet se ha modificado al pasar de los años para optimizar y ampliar su funcionalidad como la segmentación de imágenes e incluso la traducción de texto a partir de imágenes, tal como se lleva a cabo en este proyecto.

En los años 90, se introdujeron capas de activación no lineales, tal como lo son la función ReLU o sigmoide, así como técnicas de regularización, para evitar el overfitting. En este último siglo, con el aumento de popularidad del deep learning, se incrementó la cantidad disponible de datasets y aparecieron arquitecturas como VGGNet o ResNet, las cuales son prueba de un avance notable en el reconocimiento de objetos.

Posteriormente, un artículo titulado *Attention is All You Need* introdujo una visión completamente nueva en el procesamiento de lenguaje natural, puesto que introdujo la arquitectura Transformer, esta es un modelo de red neuronal no recurrente independiente de capas recurrentes debido a que se basan en mecanismos de atención que permiten que cada palabra dentro de una secuencia interactúe con el resto de ellas dado que asignan pesos a cada palabra en función de su relevancia [2].

Hoy los Transformers son una base fundamental para algunos de los modelos de Inteligencia Artificial más conocidos y avanzados, como GPT y BERT.

2. Dataset

2.1 Identificación de texto a partir de imágenes

Para la parte de redes neuronales convolucionales (CNN) se utilizaron 2 conjuntos de datasets, EMNIST Y MNIST. Estos dos fueron obtenidos en Kaggle [3]. MNIST es un estándar para reconocer dígitos escritos a mano, que contiene 70,000 imágenes en total, cada una de 28x28 píxeles, representando números escritos a mano del 0 al 9. Por otra parte, el conjunto de datos EMNIST incluye más datos, siendo estas 814,000 imágenes de 28x28 píxeles para incluir las letras del alfabeto, representando tanto mayúsculas como minúsculas.

2.2 Traductor

El dataset utilizado para la traducción consta de dos conjuntos de datos distintos. El primero se obtuvo de Kaggle [4] y contiene un total de 118,964 palabras u oraciones en inglés y sus respectivas traducciones al español. Este dataset incluye una variedad de expresiones en inglés, algunas de las cuales se repiten para capturar las diferentes formas

de traducción al español. Además de las frases recopiladas, se agregaron traducciones básicas que abarcan listas de animales y colores.

Con el objetivo de ampliar la longitud de las traducciones y permitir que el Transformer no solo aprenda el equivalente de palabras, sino también la estructura textual en inglés se incorporó un segundo dataset [5]. Este conjunto de datos, creado por nuestro equipo para este proyecto, consta de 18,838 traducciones adicionales y se compone de la traducción verso a verso de más de 300 canciones, lo que proporciona una variedad de estructuras y estilos de texto en inglés.

3. Arquitectura

3.1 Identificación de texto a partir de imágenes

Se empleó la arquitectura de red neuronal convolucional (CNN) LeNet para crear un traductor eficiente que analiza imágenes y extrae texto para su traducción al inglés. Esta versión de LeNet [6] consta de 7 capas en total, incluyendo 2 de convolución, 2 de pooling y 2 totalmente conectadas, seguidas por una capa de salida.

Las capas de convolución (C1 y C3) detectan características locales como bordes y texturas en las imágenes de entrada, siendo esenciales para la extracción de texto. Las capas de pooling (S2 y S4) reducen la dimensionalidad de la salida de las capas convolucionales manteniendo las características más importantes.

Las capas totalmente conectadas proporcionan una representación más abstracta de la imagen, permitiendo el reconocimiento de patrones complejos, mientras que la capa de salida clasifica las imágenes en categorías específicas como letras o palabras, crucial para la tarea de traducción.

La adaptabilidad de LeNet a conjuntos de datos específicos como EMNIST y MNIST mejora la precisión del reconocimiento de letras y, por ende, la calidad de las traducciones generadas. La capacidad de LeNet para reconocer letras en imágenes [7], se alinea perfectamente con la tarea de traducción de texto.

Además, se extendió la arquitectura clásica de LeNet incorporando capas de Batch Normalization después de cada capa convolucional y completamente conectada. Se adaptó la red para permitir un número variable de clases de salida, aumentando así su flexibilidad y capacidad de generalización.

3.2 Traductor

Antes de utilizar el dataset para entrenar el modelo, se realizó un preprocesamiento. Este tratamiento incluyó la limpieza del texto, eliminando caracteres como acentos y signos de puntuación, además, se normalizó el texto convirtiéndolo completamente a minúsculas. Este preprocesamiento ayuda a garantizar que el modelo pueda aprender las relaciones entre las palabras y las estructuras de las oraciones, independientemente de las variaciones en la capitalización o los caracteres especiales. Posteriormente, se generó una bolsa de palabras con todas las palabras presentes en el dataset. A cada palabra se le asoció un número único y se incorporaron los términos clave *start*, *end* y *pad* para el padding. El vocabulario resultante fue de 15,209 palabras distintas.

Tabla 1. Configuración de la red transformer utilizada.

Parámetro	Valor	Parámetro	Valor
token_num	137802	head_num	4
embed_dim	32	hidden_num	128
encoder_num	2	dropout_rate	0.05
decoder_num	2	use_same_embed	False

Con el objetivo de representar cada oración en el dataset como un vector numérico, se generaron vectores compuestos por los números que representan cada palabra. Todos los vectores se ajustaron a la misma longitud, la cual corresponde a la dimensión del espacio vectorial más largo, en este caso, 43. Aquellos vectores de longitud inferior fueron rellenados con padding para alcanzar esta longitud estándar.

El modelo Transformer, importado de la librería Keras-Transformer [8], fue configurado en base a los parámetros descritos en el artículo *Attention is All You Need* [2], seleccionando valores cercanos, pero más bajos dada de la capacidad computacional del equipo utilizado y el tamaño del dataset. Estos valores se detallan en la tabla 1.

Se procedió a compilarlo utilizando el optimizador Adam y la función de pérdida `sparse_categorical_crossentropy`. Se optó por Adam debido a su capacidad para adaptar automáticamente las tasas de aprendizaje durante el entrenamiento, lo que es esencial para lidiar con la variabilidad en la longitud de las secuencias de entrada y salida en el proceso de traducción [9]. Por otro lado, la función de pérdida `sparse_categorical_crossentropy` se seleccionó debido a su capacidad para modelar la predicción de múltiples clases, palabras en el vocabulario de salida, de manera eficiente [10]. Es importante destacar que el preprocesamiento descrito en esta sección se basa en las recomendaciones de la investigación previa sobre el uso de transformers para tareas de traducción automática [11].

3.3 Metodología

La Figura 1 presenta el diagrama de la metodología del proyecto. Se trabaja en paralelo en dos aspectos: por un lado, el reconocimiento de texto utilizando una CNN, y por el otro, la traducción del texto mediante una red Transformer. Los parámetros seleccionados para cada proceso se especifican en la Figura 1.

La Figura 2 muestra la evaluación del modelo. En esta etapa, el texto reconocido en la imagen es generado como salida por la CNN y luego es procesado por el Transformer como entrada para la traducción.

4. Resultados

Para evaluar la precisión y asertividad del proyecto, utilizamos dos conjuntos de datos creados internamente. El primer conjunto cuenta con 667 imágenes frases y palabras en español, mientras que el segundo contiene 823 imágenes frases y palabras en inglés. Se evaluó la capacidad del traductor tanto para traducir del español al inglés como del inglés al español utilizando las imágenes reconocidas como entrada.

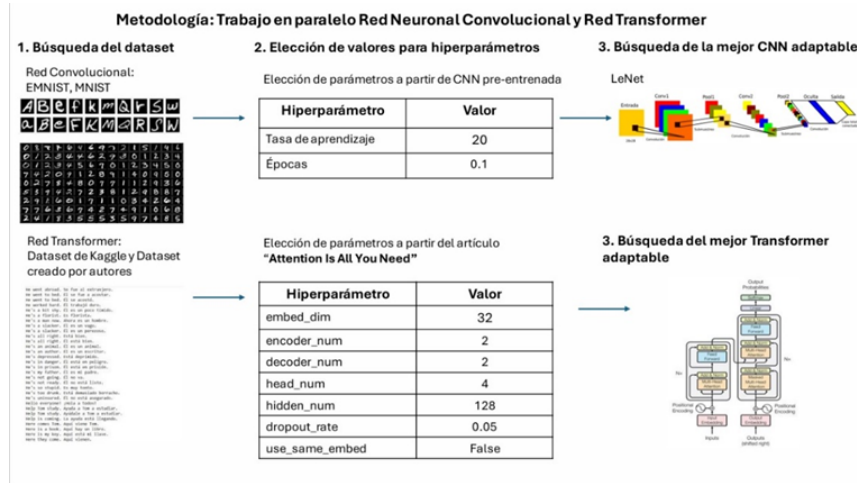


Fig. 1. Diagrama de pre-procesamiento y procesamiento de los inputs.



Fig. 2. Evaluación de los modelos, el producto de uno (CNN) siendo el input del otro (Red Transformer).

La evaluación de asertividad la realizó un humano especialista en inglés, donde se consideraron correctas aquellas traducciones que capturaban el significado contextual de las palabras y frases, siempre y cuando la entrada de imagen fuera correctamente reconocida y procesada por el sistema. Esto significa que el porcentaje de asertividad presentado se refiere específicamente a las traducciones realizadas con éxito en las entradas de imagen correctamente procesadas.

Durante la evaluación, se observó que las traducciones resultantes del programa requerían un proceso de postprocesamiento debido a posibles errores como caracteres

Tabla 2. Ejemplo de funcionamiento del traductor automático.

Imagen del texto	Texto detectado	Traducción	Revisión
	Always	Siempre	Correcto
	Barato	Cheap	Correcto
	Nacional	Vicious	Incorrecto
	Destroy	Lalalalala	Incorrecto
	Fecha	Changed i out	Incorrecto
	Healthy	Sano	Correcto
	Resfriado	Cold man	Correcto
	Red dress	Vestido rojo	Correcto

Tabla 3. Precisión de la traducción.

Modelo	Traducciones correctas	Porcentaje de asertividad
Inglés a español	526	63.9%
Español a inglés	376	56.4%

adicionales o repeticiones de frases. A pesar de estos detalles, se consideraron correctas aquellas traducciones que reflejaban con precisión el significado de las frases, incluso si presentaban errores en la estructura gramatical o las conjugaciones verbales.

En la Tabla 2 se presentan seis de las imágenes de entrada y las respectivas salidas del modelo que corresponden a las traducciones.

Notablemente, el modelo desarrollado no solo proporcionó traducciones palabra por palabra entre idiomas, sino que mostró un entendimiento rudimentario del contexto y significado de las frases. Por ejemplo, la palabra 'resfriado' fue traducida como "cold man" (véase Tabla 2), lo que refleja una interpretación del concepto más allá de una traducción literal.

Esto también se evidencia en la traducción de la frase en inglés "red dress" (véase Tabla 2), donde se observa que el traductor comprende la estructura de ambos idiomas. Por ejemplo, el orden del sustantivo y el adjetivo se invierte al pasar del inglés al español, lo que demuestra una capacidad para interpretar y adaptar la estructura lingüística de manera contextual.

La Tabla 3 muestra el porcentaje de asertividad obtenido con la ecuación 1:

$$Asertividad = \frac{Traducciones\ correctas}{Traducciones\ totales} \times 100. \tag{1}$$

Es importante destacar que estos resultados son producto de una evaluación interna. Para una evaluación más objetiva se requiere contar con especialistas externos que analicen el funcionamiento y rendimiento del traductor automático basado en reconocimiento de texto en imágenes.

5. Conclusiones

Durante este proyecto, se han mezclado armoniosamente la noción de visión por computadora con procesamiento de lenguaje natural para desarrollar un traductor, es decir, combina una red neuronal convolucional para la identificación de texto en imágenes con una arquitectura Transformer. Este modelo, logró una precisión del 56 – 64% entre la traducción bilateral de español e inglés, a pesar de que son resultados positivos, es evidente que aún hay limitaciones en la comprensión del significado, sobre todo contextual, como en el ejemplo mencionado anteriormente con “resfriado”.

Es importante destacar que el tamaño del dataset utilizado, si bien es adecuado y suficiente para los propósitos del proyecto, no es destacablemente grande. El modelo demostró poder aprender patrones significativos y generar traducciones coherentes para diversos casos.

Referencias

1. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324 (1998). DOI: 10.1109/5.726791.
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008 (2017)
3. Kaggle: <https://www.kaggle.com/datasets/crawford/emnist> (2024)
4. Kaggle: <https://www.kaggle.com/datasets/lonnieqin/englishspanish-translation-dataset?select=data.csv> (2024)
5. Github: Miricalderonr.: Traductor_por_reconocimiento_de_texto. Repositorio de GitHub, <https://github.com/miricalderonr/Traductor-por-reconocimiento-de-texto> (2024)
6. Github: Repositorio de GitHub, https://github.com/bmitt/emnist_ocr (2024)
7. Keras.io: https://keras.io/api/losses/probabilistic_losses/ (2024)
8. Kingma, D.P., Ba, J.: A Method for Stochastic Optimization. In: *ICLR* (2015)
9. Lin, T., Wang, Y., Liu, X., and Qiu, X.: A survey of transformers. *Artificial Intelligence Open Access*. vol. 8, no. 4 (2022). DOI: 10.1016/j.aiopen.2022.10.001.
10. Al-Jawfi, R.: Handwriting Arabic character recognition LeNet using neural network. *The International Arab Journal of Information Technology*, vol. 6, no. 3, pp. 304–309 (2009)
11. Github: Codificandobits: Traductor_con_redes_Transformer. Repositorio de GitHub, https://github.com/codificandobits/Traductor_con_redes_Transformer (2024)