



TIES Revista de Tecnología e Innovación en Educación Superior

**DE REDES NEURONALES RECURRENTE A MODELOS DE
LENGUAJE: LA EVOLUCIÓN DEL PLN EN LA GENERACIÓN DE
TEXTOS**

<https://doi.org/10.22201/dgtic.26832968e.2021.4.1>

David Emmanuel Maqueda Bojorquez

<https://www.ties.unam.mx/>

Fecha de recepción: julio 03, 2021 • Fecha de publicación: octubre de 2021

Octubre 2021 | número de revista 4 • ISSN 2683-2968

Acervos Digitales, Dirección General de Cómputo y de Tecnologías de Información y Comunicación, UNAM

Esta obra está bajo licencia de Creative Commons
Atribución-No Comercial 4.0 Internacional (CC BY-NC 4.0)

DE REDES NEURONALES RECURRENTE A MODELOS DE LENGUAJE: LA EVOLUCIÓN DEL PLN EN LA GENERACIÓN DE TEXTOS

Resumen

En el presente texto haremos un breve repaso por la evolución del procesamiento de lenguaje natural y la generación de texto artificial. Desde la década de 1990, vieron la luz las primeras ideas sobre generación artificial de textos. En una primera instancia, las técnicas existentes resultaron insuficientes para resolver tareas, tales como la propia generación de textos, la traducción automática o la búsqueda de información.

Sin embargo, al pasar del tiempo se fueron generando nuevas ideas, con base en la comprensión propia de la generación de textos desde la naturaleza humana, desde los primeros modelos de redes neuronales recurrentes hasta los modelos de lenguaje más actuales, desde la consecución de palabras más básicas hasta lograr la generación de obras literarias enteras de manera artificial. De esta forma visualizaremos en esencia cómo es que se dio esta evolución en el campo y cuál fue la lógica que siguió.

Palabras clave:

Procesamiento de Lenguaje Natural, Modelos de Lenguaje, Redes Neuronales Recurrentes, LSTM, Inteligencia Artificial

FROM RECURRING NEURAL NETWORKS TO LANGUAGE MODELS: THE EVOLUTION OF NLP FOR TEXT GENERATION

Abstract

In this text we will briefly review the evolution of natural language processing and the generation of artificial text. Since the 1990s, the first ideas about artificial text generation saw the light; In the first instance, the existing techniques were insufficient to solve tasks such as the generation of texts, automatic translation or the search of information.

However, as time went by, new ideas were generated based on the understanding of the generation of texts from human nature, from the first models of recurrent neural networks to the most current language models, from the achievement of words more basic until the generation of entire literary works artificially. In this way we will visualize in essence how this evolution occurred in the field and what was the logic that said evolution followed.

Keywords:

Natural Language Processing, Language Models, Recurrent Neural Networks, LSTM, Artificial Intelligence.

DE REDES NEURONALES RECURRENTE A MODELOS DE LENGUAJE: LA EVOLUCIÓN DEL PLN EN LA GENERACIÓN DE TEXTOS

Introducción

Procesamiento de Lenguaje Natural (PLN) es la parte de la Inteligencia Artificial (IA) y lingüística que interactúa con: el lenguaje humano, hablado o escrito y la computación [1]. Para llevar a cabo este proceso es distintivo el uso del lenguaje en general, se requiere tener estudios de fonética, fonología, morfología, semántica y sintaxis.

Además, se necesita acercarse al significado léxico, a la semántica composicional, para entender el concepto de cada palabra. Para interpretar las intenciones del lenguaje, se debe estudiar pragmática y diálogo. Finalmente, para deducir y asociar información dentro del lenguaje, se necesita comprensión del discurso. [2]

Este conjunto de conocimiento lingüístico se intenta brindar a un sistema, mediante un modelo matemático/computacional, que depende claramente del tipo de procedimiento a desarrollar. Mientras un sistema conversacional (e.g. Siri, Cortana o Alexa), con clara tendencia al reconocimiento y síntesis de voz, necesitaría una amplia cognición de fonética y fonología, ya que interpretaría y generaría voz. Otro tipo de generadores de texto necesitaría mayor noción de la sintaxis, la ortografía, el contexto léxico y la estructura de un lenguaje, idioma o dialecto, [3] siendo dependientes al igual que su fin, debido a las discre-

pancias entre el estilo de escritura que puede haber entre estilos de textos independientes de la lengua (e.g escritura poética, científica o coloquial).

En este punto cabe destacar la prueba de Turing, que él mismo llamó ‘*The Imitation Game*’ o el juego de imitación. [4] En esencia, esta prueba dice que es suficiente usar únicamente lenguaje para probar si una máquina se puede considerar pensante, todo esto logrando confundir a una persona mediante el origen de un cierto mensaje. Si el interlocutor considera que un mensaje, o respuesta a una pregunta, no es generado artificialmente, entonces el agente artificial puede considerarse inteligente. Es llamado el juego de imitación, ya que el agente artificial imita al ser humano. Para completar el experimento se contrastan las respuestas de un humano y un agente artificial con un tercero, el que desconoce la fuente de los mensajes que recibe.

Para muchos, la habilidad de procesar lenguaje como los humanos es un signo claro de máquinas inteligentes. Así que la tendencia en el campo ha ido en dirección a mejorar la forma en que se generan textos y lenguaje en general, viéndose mejorada la calidad de estos. Los avances recientes en sistemas conversacionales muestran a las personas de cierta forma. Comienzan a tratar a las máquinas como personas, incluyéndolas directamente en su vida cotidiana. [5]

Desarrollo

Un breve repaso sobre el funcionamiento de redes neuronales

Actualmente, la mayoría de los desarrollos en IA están fundamentados en redes neuronales, modelos matemáticos / computacionales que provienen de los estudios relacionados con el funcionamiento del cerebro. En 1943 McCulloch y Pitts aportaron una forma de describir las funciones cerebrales en términos abstractos, para mostrar que los elementos simples, conectados en una red neuronal, pueden tener un inmenso poder computacional. [6] Como lo indica su nombre, el concepto es crear una red de neuronas. Las neuronas por sí solas son unidades de cálculo, por las cuales fluye información (imágenes, textos, etcétera, codificados en vectores, los cuales son objetos matemáticos fácilmente manejables). Al final de esta tendremos una salida resultante, dependiente de los datos que fluyen por la red.

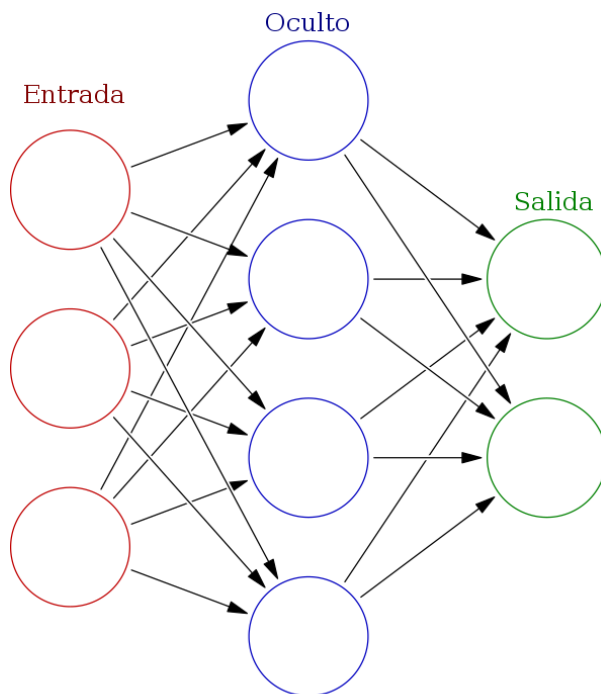


Figura 1. Estructura general de una red neuronal. Disponible en: https://commons.wikimedia.org/wiki/File:Colored_neural_network_es.svg. Consultado en junio 02, 2021.

En este punto, cabe destacar el concepto de capa neuronal. Como previamente se describió de manera breve, una neurona es una unidad de cálculo que también podría definirse como el elemento básico dentro de una red neuronal, mientras que una capa neuronal es en sí un conjunto de neuronas que tienen características homogéneas entre ellas.

Dentro de una red neuronal, la información fluye de dos formas y en ambas tiene utilidades distintas. Cuando alimentamos la red (*forward*), la información fluye de la generalidad a la particularidad (de la entrada de la red a la salida de la misma), es decir, la red neural, así como cada una de las neuronas, se va contextualizando matemáticamente con respecto a los datos que fluyen por cada una de las unidades de cálculo (neuronas). [7]

Cuando la información fluye a la salida, el siguiente tipo de flujo de información es el *backward*, que actúa sobre la salida y genera una retroalimentación a toda la red neuronal, relacionada con cómo mejorar la calidad de la salida, modificando los datos con que se calcula el valor de cada neurona. Esta modificación se hace en base a una resultante, entre la diferencia del valor esperado y lo que nos es ofrecido por la red neural. A esto se le denomina *error o pérdida* (hablando de aprendizaje supervisado). [8]

En la figura 1 se muestra la estructura general de una red neural. Se presentan gráficamente las neuronas de cada una de las capas de la red. Las neuronas de la capa de entrada son las que reciben la información directamente, de los datos con que se alimenta a la red. Esto significa que desde ahí inicia el *forward* de la red, mientras que a partir de la capa de salida se calcula un error, en base a la salida de la red, e inicia el *backward*. Cabe destacar que, típicamente, una capa oculta se define como toda aquella que se encuentre entre la capa de entrada y la de salida. Puede haber una o varias capas ocultas en una arquitectura neuronal. En el caso del diagrama, se visualiza entonces una capa oculta. Las neuronas, una a una, se conectan con fuerzas variables de conexión (o ponderaciones), comúnmente llamadas pesos.

A grandes rasgos, mediante el proceso antes descrito se entrena una red neuronal, es decir, la forma a través de la cual se va refinando un agente inteligente para generar mejores resultados. Cada vez que este proceso se repite con cada uno de los datos que están a disposición, se determina que se ha cumplido una *época* de entrenamiento. [9]

El concepto de memoria: Redes Neuronales Recurrentes

La idea de la generación de textos no es nueva en lo absoluto, si bien es cierto que los primeros esfuerzos se dieron en la década de los cuarenta con los n-gramas, el proyecto se remonta a los experimentos realizados por Claude Shannon, [10] en que la idea era: dada una secuencia de letras, ¿cuál es la siguiente letra más probable? Como tal, siendo un modelo probabilístico, los n-gramas permiten predicciones estadísticas del próximo elemento de una cierta secuencia de elementos, donde, debido a limitaciones computacionales y a la normalmente naturaleza abierta de los problemas (suele haber infinitos elementos posibles), se asume que cada elemento solo depende de los últimos n elementos de la secuencia. [11]

Alrededor de la década de los noventa, considerando las limitaciones de los n-gramas y, además, con los resultados prometedores de las redes neuronales tradicionales,

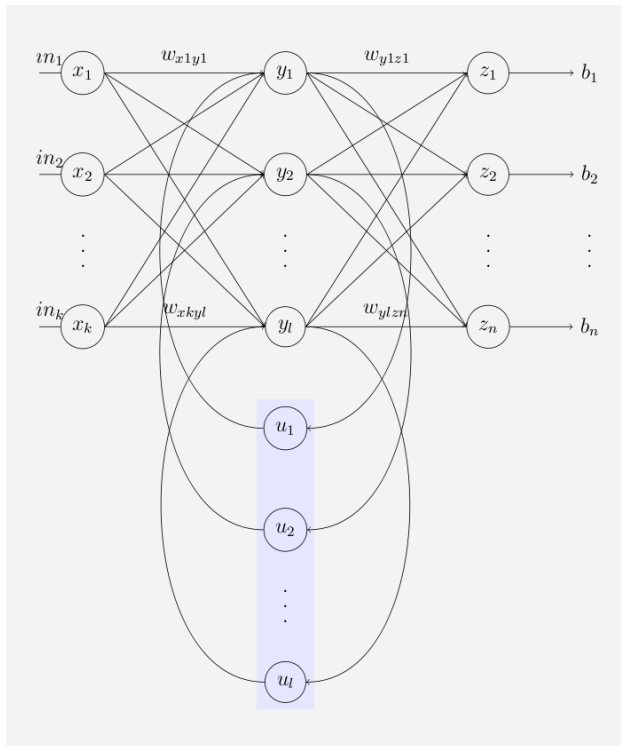


Figura 2. Estructura de la red neuronal recurrente simple de Elman. Disponible en: https://en.wikipedia.org/wiki/Recurrent_neural_network#/media/File:Elman_srnn.png Consultado en junio 02, 2021.

feed-forward, surgen las redes neuronales recurrentes o *Recurrent Neural Networks (RNN's)*. El principio básico de este tipo de redes se basa en la utilización de bucles dentro del flujo de información. Recordemos que en las redes neuronales convencionales solo existe el *backward* y el *forward*, para describir la naturaleza del flujo; sin embargo, como puede verse en la figura 2, en una *RNN* se incluyen bucles en el flujo, los cuales están representados por las líneas curvadas intermedias en el diagrama. Dichos bucles de información son interpretados como remanentes de información, que realimentan a la red junto con información nueva, que viene desde la entrada. Esto permite almacenar información dentro de la red.

En resumen, las *RNN* utilizan su razonamiento de experiencias anteriores para informar los próximos eventos. Un ejemplo común de *RNN* es la traducción automática. Por ejemplo, una red neuronal puede tomar una oración de entrada en español y traducirla a una oración en inglés. La red determina la probabilidad de cada palabra en la oración de salida, basándose en la palabra en sí y en la secuencia de salida anterior. [12]

Sin embargo, uno de los principales problemas de las *RNN* fue su limitado rango de información contextual, es decir, estos sistemas pueden considerar la información contextual de una situación previa, más no de la generalidad. En un contexto de traducción, esto fue problemático, debido a que en la mayoría de las situaciones la correcta traducción no depende exclusivamente de la palabra antecedente, sino también de un contexto mucho más generalizado en la oración o párrafo a traducir.

En conclusión, esto, aunado a problemas relacionados con el entrenamiento de estas redes, creó la situación idónea para desarrollar nuevas ideas. Así es como en 1997 surgen las redes *Long Short-Term Memory (LSTM)*. [13]

Introducción a los tipos de memoria: Long Short-Term Memory

El concepto de una *LSTM* sigue siendo una red recurrente, con los bucles de flujo de información, pero aplica transformaciones sofisticadas a las entradas. Las entradas de cada neurona se manipulan mediante maniobras complejas, lo que produce dos salidas, que se pueden considerar como la *memoria a largo plazo* y la *memoria a corto plazo*, dos bucles de flujo de información distintos.

Esta adición de un flujo de información a largo plazo amplía drásticamente el tamaño de la atención de la red.

Puede acceder a estados neurales anteriores y así obtener un mayor contexto que una RNN, sin embargo, la red LSTM tiene sus desventajas. Sigue siendo una red recurrente. Si bien la adición de un canal de memoria a largo plazo ayuda, existe un límite en la cantidad que puede contener. [14]

En la figura 3, se muestra la visualización de una celda de la red LSTM, compuesta por una compuerta de entrada, una puerta de salida y una compuerta de olvido. La celda recuerda valores en intervalos de tiempo arbitrarios y las tres puertas regulan el flujo de información dentro y fuera de la celda. De esta forma, se pueden procesar datos secuencialmente y mantener su estado oculto a lo largo del tiempo. [15]

La LSTM funcionó bien durante un tiempo. Pudo generar caracteres razonablemente bien en textos más cortos y no se dejó intimidar por muchos de los problemas que plagaron el desarrollo temprano del procesamiento del lenguaje natural, en particular, una mayor profundidad global y comprensión, no solo de palabras individuales, sino también de su significado colectivo.

Las LSTM son limitadas cuando se presentan problemas modernos más exigentes, como la traducción automática en varios idiomas o la generación de texto completamente indistinguible del texto escrito por humanos. Cada vez más, se utiliza una arquitectura más nueva para abordar tareas más desafiantes: *Transformers*.

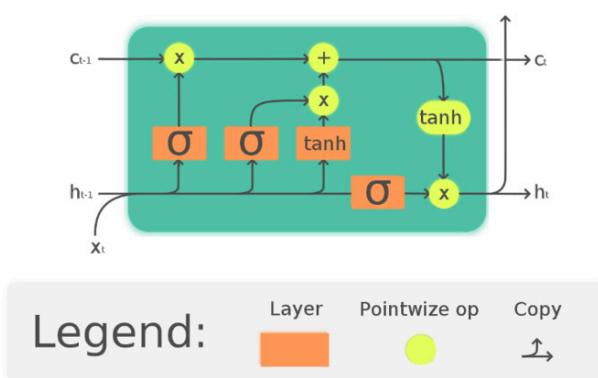


Figura. 3. Celda de LSTM para procesar secuencialmente la información. Su información se puede mantener oculta en el tiempo. Disponible en: https://en.wikipedia.org/wiki/Recurrent_neural_network#/media/File:Elman_s_rnn.png Consultado en junio 02, 2021.

Lo único que necesitas es atención: transformers

Los transformers o transformadores son un modelo introducido en 2017, [16] que se denomina arquitectura secuencia a secuencia. Secuencia a secuencia (o Seq2Seq) es una red neuronal que transforma una secuencia determinada de elementos (como la secuencia de palabras en una oración) en otra secuencia. Por ejemplo, palabras de un idioma a una secuencia de palabras diferentes en otro idioma.

Transformers hace uso de un concepto sumamente importante: la 'atención'. El mecanismo de atención mira una secuencia de entrada y decide en cada paso qué otras partes de la secuencia son importantes. Por ejemplo, al leer este texto, siempre se concentra en la palabra que lee, pero, al mismo tiempo, su mente aún guarda las palabras clave importantes del texto en la memoria, para proporcionar contexto.

Las redes recurrentes eran, hasta ahora, una de las mejores formas de capturar las dependencias oportunas en secuencias. Sin embargo, el equipo que presentó el artículo demostró que una arquitectura, con solo mecanismos de atención, puede mejorar los resultados en la tarea de traducción y otras tareas. Los transformers no requieren que los datos secuenciales se procesen en orden. Más bien, la operación de atención identifica el contexto para cualquier posición en la secuencia de entrada.

Al usar 'atención' se pondera la importancia de cada elemento de la secuencia de entrada, es decir, identifica el contexto que confiere significado a una palabra en la oración. Al dejar atrás el paradigma principal de las RNN, sobre los bucles de flujo de información y el concepto de memoria a corto y largo plazo, los transformers se han convertido rápidamente en el modelo de elección para los problemas de PLN, reemplazando los modelos de RNN más antiguos.

Modelos de Lenguaje, la implementación de Transformers

Un modelo de lenguaje se ha vuelto recientemente en un eje central del PLN, el cual analiza el patrón del lenguaje humano para la predicción de palabras. [17] Las aplicaciones basadas en PLN utilizan modelos de lenguaje para una variedad de tareas, como conversión de audio a texto, reconocimiento de voz, análisis de sentimientos, resumen, corrección ortográfica, etcétera.

Los modelos de lenguaje determinan la probabilidad de la siguiente palabra, analizando el texto en los datos. Los modelos se preparan para la predicción de palabras, aprendiendo las características de un idioma. Con este aprendizaje, el modelo se prepara para comprender frases y predecir las siguientes palabras en frases. La cantidad de datos de texto que se analizarán y las matemáticas aplicadas para el análisis, marcan la diferencia en el enfoque seguido para crear y entrenar un modelo de lenguaje.

Además de las ventajas que supone la implementación de los *Transformers* antes mencionados, también permite mucha más paralelización que los *RNN* convencionales. Esto, por lo tanto, reduce los tiempos de entrenamiento, [18] lo que ha permitido el desarrollo de modelos de lenguaje, como *BERT*, *GTP*, *GTP-2* y el recientemente introducido *GTP-3*, entrenados con conjuntos de datos de lenguaje general de gran tamaño, como *Wikipedia Corpus* (1.9 billones de palabras) [19] y *Common Crawl* (220 TB de información), [20] que pueden ser adaptados a tareas lingüísticas específicas y cuyos beneficios son mucho mayores de lo que era posible obtener antes de la llegada de *Transformers* en 2017.

Un modelo de lenguaje es, a grandes rasgos, una distribución de probabilidad de secuencias de palabras. ¿Por qué esto es necesario y a qué nos referimos?

Bien, supongamos que tenemos una serie de palabras: *mesa, sobre, plato, la, el, está*. Con estas palabras tendremos que generar una oración. Para un humano que tenga conocimiento del idioma español, resultará naturalmente sencillo generar la oración: “*El plato está sobre la mesa*”. Sin embargo, para una máquina la historia es distinta. Un modelo de lenguaje, usando la distribución de probabilidad, tendrá que definir cuál sucesión de palabras tiene la mejor probabilidad entre todas las posibilidades, a lo que un modelo con un entrenamiento correctamente realizado llegaría a una conclusión igual o similar a la de un humano.

Esto quiere decir que la sucesión de palabras “*El plato está sobre la mesa*”, tiene una mayor distribución de probabilidad que la sucesión “*La mesa está sobre el plato*” o “*Sobre el plato, está el mesa*”. Ahora bien, aquí podría surgir la pregunta: ¿cómo se logra esto?

Como dijimos antes, estos modelos, al ser entrenados con conjuntos de datos tan grandes, tienen gran contexto del manejo del lenguaje. Mientras más grandes sean estos conjuntos de datos, mayor será su conocimiento del lenguaje y el manejo de este (con el debido tiempo de entrenamiento). Esto significa que, al ya ser *refinado* el

modelo con una gran cantidad de datos y recordando el funcionamiento del método de ‘atención’ propuesto por *Transformers*, se abstrae la información contextual de cada palabra en una oración. Esto quiere decir que el modelo de lenguaje tendrá conocimiento de la importancia de cada elemento, de la secuencia de datos, con la que haya sido entrenado, [21] es decir, de cada una de las oraciones, de cada párrafo, de cada documento dentro de toda la información con la que haya sido entrenado el modelo de lenguaje. Es por ello por lo que entre mayor sea el conjunto de datos con el cual se entrena, mayores serán los fenómenos lingüísticos de los que el modelo tendrá contexto y, por ello, el mismo será más robusto.

Por este motivo, el modelo puede deducir cuál secuencia de palabras es la de mayor distribución de probabilidad. Resulta más probable que la secuencia “*El plato*” o “*La mesa*”, sean lingüísticamente correctas, que “*La plato*” o “*El mesa*”, dado que todos los eventos lingüísticos, de los cuales el modelo obtuvo contexto, eran del primer tipo, dado que la forma correcta del uso del lenguaje es ésta, mientras la otra es improbable, dado que no es la forma correcta de uso de lenguaje, por lo que los textos, de los que el modelo obtuvo contexto, no contenían estos tipos de secuencias.

En el caso de generación de textos, al introducir una secuencia inicial (*prompt*), ésta será la base que calcule una distribución de probabilidad, es decir, la predicción de la palabra consecuente estará directamente relacionada no solamente con la palabra antecedente (como habría sido en las *RNN* convencionales), sino también con toda la información contextual que brinden las demás palabras, que incluyan la secuencia inicial. Cuando esta nueva palabra sea generada, será considerada nuevamente para la generación de otra nueva palabra, así como del nuevo contexto que le haya dado a la secuencia completa y así sucesivamente.

Cada nueva palabra generada depende del contexto de la secuencia base y también, a su vez, del contexto que tenga el modelo del lenguaje, dado su entrenamiento. Volviendo al supuesto anterior, al tener la frase “*El plato*”, aunque puedan existir una infinidad de palabras consecuentes lingüísticamente correctas, los datos y a su vez los fenómenos lingüísticos con los cuales el modelo fue entrenado, decidirán cuál es la palabra consecuente con mayor probabilidad. Supongamos que dicha palabra es *está*, siendo la nueva secuencia base “*El plato está*”. En este punto la información contextual de la secuencia base se modificó: *el plato debe estar en algún lugar*. Así que se delimita la palabra a generar, con alguna que tenga sentido contex-

tual, más allá de una correcta sintaxis. El modelo incluye esa información en la generación de una nueva palabra y, dada la forma como se constituye el lenguaje (en este caso el español), analizando tanto el contexto global del entrenamiento (conocimiento general del lenguaje del modelo), como el de la secuencia base, se generará una palabra. Supondremos que las siguientes dos palabras generadas serán “en la”, siendo la nueva secuencia base “*El plato está en la*”. De nueva cuenta el contexto de la oración cambia, ya que *el plato debe estar en un objeto cuyo prefijo sea femenino*, delimitando así la generación de la nueva palabra. Pueden existir palabras compatibles, como “*nevera*”, “*alacena*”, etcétera. Sin embargo, aunque todas son lingüísticamente correctas, supongamos que el modelo decida que la palabra más probable es: “*mesa*”. Esto significa que es más común y, a su vez, más probable que la palabra más adecuada en el idioma español, o al menos en el conjunto de entrenamiento del modelo, sea esa y con esto terminamos con el ejercicio supuesto.

Conclusiones

Como puede notarse con estos modelos, el razonamiento y la generación de textos es similar a la forma como un humano generaría algún texto. Incluso, al estar escribiendo este texto, se está usando el mismo principio: se generó una idea, se inició una frase y con base en esa frase se construyó un texto contextualmente coherente y sintácticamente correcto, que represente la idea que se quiso expresar al momento de escribir. La diferencia radica en la generación inicial de la idea. Un humano, como cualquiera de nosotros, puede generarla intrínsecamente. Un modelo como el expuesto previamente, es dependiente de una secuencia de palabras inicial, que debe brindársele, aunque el texto de esa idea inicial y el rumbo de este es independiente de una idea explícitamente humana.

Mientras un modelo de lenguaje conocería la sintaxis del idioma, la coherencia contextual y la precisión léxica, con base en un conjunto de datos de entrenamiento y el uso del lenguaje, un humano lo conocería con base en años de interacción humana y un adiestramiento riguroso.

Recursos Adicionales

Demo con transformers: [En línea]. Disponible en <https://transformer.huggingface.co/> [Consultado en junio 02, 2021].

BIBLIOGRAFÍA

- [1] A. Gelbukh, "Procesamiento del lenguaje natural: estado de la investigación," en *Memoria del I Simposio Internacional sobre Organización del Conocimiento: Bibliotecología y Terminología*, C. Naumis Peña, 2009, pp. 337-357.
- [2] G. Guida y G. Mauri, "Evaluación de los sistemas de procesamiento del lenguaje natural: problemas y enfoques," en *Proceedings of the IEEE*, vol. 74, no.7, pp. 1026–1035, 1986.
- [3] T. Klüwer, "From chatbots to dialog systems," en *Conversational agents and natural language interaction: Techniques and Effective Practices*, 2011, pp.1-22.
- [4] A. M. Turing, "Computing Machinery and Intelligence," en *Mind*, 1950, pp. 433-460.
- [5] D. Jurafsky & J.H Martin, *Speech and language processing*. Pearson International Edition, 2009 ISBN 978-0-13-504196-3, Chapter 24.
- [6] W.S. McCulloch y W.H Pitts, "A logical Calculus of ideas immanent in nervous activity," en *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.
- [7] O. Varun Kumar, A. Ajith y S. Václav, "Metaheuristic design of feedforward neural networks: A review of two decades of research," en *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 97-116, 2017.
- [8] S. Hochreiter y J. Schmidhuber, "Long short-term memory," en *Neural Computation*, vol. 9 no.8, pp. 1735–1780, 1997.
- [9] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. New York: Macmillan College Publishing, 1998.
- [10] C.E. Shannon, "A Mathematical Theory of Communication," en *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.
- [11] C. D. Manning y H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press: ISBN 0-262-13360-1, 1999.
- [12] H. Hewamalage, C. Bergmeir y K. Bandara, "Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions," en *International Journal of Forecasting*, vol. 37, pp. 388–427, 2020.
- [13] O. Calin, *Deep Learning Architectures*, Cham, Switzerland: Springer Nature. ISBN 978-3-030-36720-6, p. 555.
- [14] K. Greff, R. K. Srivastava, J. Koutník, *et al.*, "LSTM: A Search Space Odyssey," en *IEEE Transactions on Neural Networks and Learning Systems*, vol.28, no.10, pp. 2222–2232, 2015.

- [15] S. Hochreiter y J. Schmidhuber “Long short-term memory,” en *Neural Computation*, vol. 9, no.8, pp. 1735–1780, 1997.
- [16] Vaswani, et al., “Attention is all you need,” en *NIPS’17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, 2017.
- [17] I. Solaiman, M. Brundage, J. Clark, et al., *Release Strategies and the Social Impacts of Language Models*, CoRR: 1908-09203, 2019.
- [18] J. F. Kolen, *A Field Guide to Dynamical Recurrent Networks*, 1st Edition, pp. 14-24, 2001.
- [19] The Wikipedia corpus [En línea]. Disponible en: <https://www.english-corpora.org/wiki/> [Consultado en junio 02, 2021].
- [20] Commoncrawl.org, *Rastreo común*, [En línea]. Disponible en: <https://commoncrawl.org/> [Consultado en junio 02, 2021].
- [21] Radford, et al., *Language Models are Unsupervised Multitask Learners*, OpenAI, 2019.

Cómo se cita:

D. E. Maqueda, “De redes neuronales recurrentes a modelos de lenguaje: la evolución del PLN en la generación de textos,” *TIES, Revista de Tecnología e Innovación en Educación Superior*, no. 4, octubre, 2021. [En línea]. Disponible en: <https://www.ties.unam.mx/> [Consultado en mes día, año].

Fecha de recepción: julio 03, 2021

Fecha de publicación: octubre de 2021