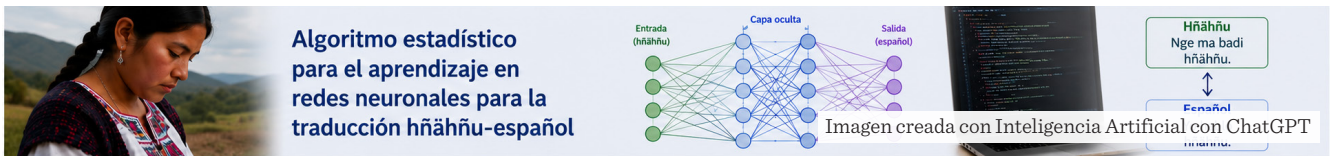


Algoritmo estadístico para el aprendizaje en redes neuronales para la traducción hñähñu–español

Statistical algorithm for learning in neural networks for the hñähñu–spanish translation

José Manuel Cruz-Olguín, Salvador Santos-Romero, Virgilio López-Morales, Manuel Alejandro Ojeda-Misses*

RESUMEN

La traducción automática de lenguas indígenas de bajo recurso enfrenta importantes desafíos derivados de la escasez de corpus digitales, la variación dialectal y la complejidad lingüística. El objetivo de este trabajo fue desarrollar y evaluar un traductor automático hñähñu–español basado en una red neuronal perceptrón multicapa entrenada mediante un algoritmo estadístico–probabilístico (EyP), fundamentado en el Teorema del Límite Central. Se generó una metodología que permitiera realizar el ajuste de pesos y sesgos a partir de probabilidades de error estimadas estadísticamente, sin emplear derivadas explícitas, diferenciándolo de los algoritmos clásicos de entrenamiento neuronal como Backpropagation y Levenberg–Marquardt. Se utilizó una estrategia de vectorización a nivel de caracter para representar palabras del hñähñu como vectores numéricos de longitud fija, lo que permitió capturar regularidades fonético–estructurales entre ambas lenguas. El desempeño del traductor se evaluó utilizando un corpus léxico de 145 palabras, métricas estadísticas objetivas a través del error cuadrático medio (ECM), validación cruzada k-fold y comparaciones directas con algoritmos de entrenamiento tradicionales. Los resultados mostraron que el algoritmo EyP alcanzó menores ECM y una convergencia más estable, especialmente en escenarios de datos limitados, aunque con un mayor número de épocas. En conjunto, los hallazgos confirman que el enfoque EyP propuesto constituye una alternativa robusta y viable para la traducción automática hñähñu–español, contribuyendo a la preservación de lenguas indígenas.

PALABRAS CLAVE: traducción automática, lengua indígena, hñähñu, español, redes neuronales.

ABSTRACT

Machine translation of low-resource indigenous languages faces significant challenges stemming from the scarcity of digital corpora, dialectal variation, and linguistic complexity. The aim of this work was to develop and evaluate a Hñähñu–Spanish machine translator based on a multilayer perceptron neural network trained using a statistical-probabilistic (S&P) algorithm grounded in the Central Limit Theorem. A methodology was generated that would allow the adjustment of the weights and biases based on statistically estimated error probabilities, without employing explicit derivatives, differentiating it from classical neural training algorithms such as Backpropagation and Levenberg–Marquardt. A character-level vectorization strategy was used to represent Hñähñu words as fixed-length numerical vectors, allowing for the capture of phonetic-structural regularities between the two languages. The translator's performance was evaluated using a 145-word lexical corpus, objective statistical metrics through the Mean Square Errors (MSE), k-fold cross-validation, and direct comparisons with traditional training algorithms. The results show that the S&P algorithm achieves lower MSE and more stable convergence, especially in data-constrained scenarios, albeit with a greater number of epochs. Overall, the findings confirm that the proposed statistical-probabilistic approach constitutes a robust and viable alternative for Hñähñu–Spanish machine translation, contributing to the preservation of indigenous language.

KEYWORDS: machine translation, indigenous language, Hñähñu, Spanish, neural networks.

***Correspondencia:** manuel_ojeda@uaeh.edu.mx**/Fecha de recepción:** 11 de julio de 2025**/Fecha de aceptación:** 21 de mayo de 2026**/Fecha de publicación:** 6 de julio de 2026.

Universidad Autónoma del Estado de Hidalgo, Instituto de Ciencias Básicas e Ingeniería, Área Académica de Computación y Electrónica, Ciudad del Conocimiento, carretera Pachuca – Tulancingo km. 4.5, s/n, col. Carboneras, Mineral de la Reforma, Hidalgo México, C. P. 42184.

https://doi.org/10.29059/cienciauat.v20i2.2042

INTRODUCCIÓN

La traducción automática consiste en transformar un texto desde una lengua de origen hacia otra de destino, generando como resultado un texto traducido. Este proceso se ha convertido en un recurso de gran utilidad para apoyar la labor de traductores humanos, impulsar investigaciones lingüísticas, facilitar la creación de diccionarios de manera automática, desarrollar analizadores de lenguas, mejorar sistemas de recuperación de información y fortalecer herramientas de traducción multilingüe (Santiago y col., 2024; González-Servín y col., 2025).

En México, algunos estados presentan un alto porcentaje de población hablante de lenguas indígenas como Oaxaca (27.3 %), Yucatán (26.1 %), Chiapas (23.4 %), Quintana Roo (14.1 %) y Guerrero (13.9 %). Cifras que evidencian la riqueza cultural y lingüística del país, ya que estas lenguas no son solo medios de comunicación, sino también expresiones de identidad y tradición.

El 86 % de los hablantes indígenas vive en su estado natal, el 13.7 % ha migrado dentro del país y solo el 0.3 % nació en el extranjero, reflejando un fuerte arraigo territorial. En cuanto a la migración reciente, el 96.4 % de los hombres y el 97.6 % de las mujeres hablantes no migraron en los últimos 5 años, lo que refuerza la conexión entre lengua, territorio e identidad (INEGI, 2024). Este contexto tiene implicaciones significativas para políticas lingüísticas, educativas y culturales. El reconocimiento legal de las lenguas indígenas como lenguas nacionales ha fomentado su inclusión en medios, educación y espacios públicos, fortaleciendo así la diversidad cultural, de acuerdo con el Instituto Nacional de lenguas indígenas (INALI, 2015). Las lenguas más habladas en México, son el náhuatl (23.6 %), maya (12.4 %), tzeltal (7.9 %) y zapoteco (7.2 %) (INEGI, 2024).

Aunque menos mencionada, la lengua otomí o hñähñu, como la denominan sus hablantes, es parte de la familia otomangue, subgrupo otopame. Según Lastra (2010), se concentra en los estados de Querétaro, Estado de México e Hidalgo. El INEGI (2015) reportó 307 928 hablantes, e Hidalgo concentró al 39.4 % (121 442 personas), particularmente en el Valle del Mezquital.

La elaboración de corpus (textos o documentos) paralelos, la tokenización (división de un texto en unidades menores) automática y el diseño de una metodología para la traducción hñähñu-español son estrategias que permiten contribuir a la preservación de las lenguas nativas como el otomí (De-la-Vega, 2017; Escorza-Sánchez y col., 2018). Uno de los retos centrales radica en obtener traducciones confiables del hñähñu al español, particularmente debido a la escasa disponibilidad de corpus. La mayoría de los textos se encuentran en formato físico o no están accesibles digitalmente (Hernández-Cruz y col., 2010; Hernández-Pérez y col., 2020). Asimismo, la estructura gramatical del hñähñu representa un desafío, pues la creación manual de reglas y su implementación pueden resultar complejas y demandar gran esfuerzo. A esto se suman problemas como la falta de estandarización ortográfica, la presencia de múltiples variantes dialectales y la carencia de herramientas lingüísticas esenciales como lematizadores (programas que intentan encontrar la raíz de las palabras), etiquetadores gramaticales y tokenizadores especializados (Hinton y Hale, 2001; HaCohen y col., 2020).

Se han desarrollado diversos esfuerzos para preservar esta lengua. Aguilar-Santiago y García-Zúñiga (2023) documentaron iniciativas desde el año 2000 orientadas a fortalecer las lenguas originarias realizadas por autores como Hinton y Hale (2001) y Dyson y col. (2007; 2016). Entre estas acciones destacan corpus digitales, diccionarios y materiales educativos. Gallardo-Arias (2012) presentó un libro con vocabulario básico otomí, desarrollado con apoyo de estudiantes en actividades de vinculación comunitaria, abarcando frutas, animales, números, colores y partes del cuerpo. Hernández-Cruz y col. (2010) y Vargas-García (2017) documentaron un diccionario hñähñu en el Valle del Mezquital, con recursos escritos y

orales. De-la-Vega (2017) publicó un libro que promueve la cultura e identidad otomí a través de temas cotidianos. Hernández-Pérez y col. (2020), ante la escasez de materiales pedagógicos en lenguas originarias para nivel básico, diseñaron un libro electrónico enfocado al rescate del hñähñu, respondiendo a las necesidades de niños indígenas y enfrentando el desplazamiento lingüístico. Finalmente, Escorza-Sánchez y col. (2018) desarrollaron una aplicación móvil para promover el aprendizaje del hñähñu en escuelas bilingües de Ixmiquilpan.

Por otra parte, la traducción automática ha evolucionado desde enfoques basados en reglas lingüísticas y modelos estadísticos hacia arquitecturas sustentadas en redes neuronales profundas. Las redes neuronales, junto con sus extensiones, unidad de memoria a largo y corto plazo (LSTM, por sus siglas en inglés: Long Short-Term Memory) y la unidad recurrente cerrada con compuertas (GRU, por sus siglas en inglés: Gated Recurrent Unit) (Remus y O'Connor, 2001; Ojeda y col., 2019; Mohebali y col., 2020; Khoussi y col., 2021; Linka y col., 2022; Chongchong-Li y col., 2024), permitieron avances significativos en la traducción secuencia a secuencia. Posteriormente, dichas arquitecturas fueron superadas por los modelos basados en mecanismos de atención, particularmente los transformadores (*transformers*), arquitecturas especializadas introducidas formalmente en el trabajo "Attention is all you need" (Vaswani y col., 2017). Estos modelos han demostrado un desempeño sobresaliente en traducción automática y otras tareas de procesamiento del lenguaje natural, dando lugar a modelos preentrenados como Representaciones de Codificador Bidireccional a partir de Transformadores (BERT, por sus siglas en inglés: Bidirectional Encoder Representations from Transformers), el Transformador Generativo Preentrenado (GPT, por sus siglas en inglés: Generative Pretrained Transformer) y el modelo preentrenado mediante un Transformador Autoregresivo Bidireccional (BART, por sus siglas en inglés: Bidirectional and Auto-Regressive Transformers), Red de Longitud Extrema (XLNet, por sus siglas en inglés: eXtra Length Network), y el convertidor de palabra en vector (Word2vec, por sus siglas en inglés: Word to Vector) (Kurniawan y Maharani, 2020; Nurdin y col., 2020; Çelik y Koç, 2021; Singh y col., 2022; Ramos-Aguilar y col., 2025).

A pesar del predominio de los transformadores, las redes neuronales tradicionales continúan siendo relevantes en escenarios con recursos computacionales limitados o conjuntos de datos reducidos, como es el caso de muchas lenguas indígenas. Su menor complejidad y su capacidad para modelar dependencias secuenciales, las convierten en alternativas viables cuando el entrenamiento de modelos de gran escala resulta inviable (Sennrich y Zhang, 2019).

En México, se han desarrollado propuestas recientes que emplean inteligencia artificial (IA) para la traducción y preservación de lenguas indígenas (Santiago-Benito y col., 2024; González-Servín y col., 2025), incluyendo la transferencia de patrones a modelos preentrenados y el uso de herramientas avanzadas para el análisis de audio. Estos trabajos destacan la efectividad de los enfoques neuronales actuales, pero también subrayan la necesidad de metodologías adaptadas a contextos de datos limitados e integradas con el conocimiento lingüístico de las comunidades hablantes.

El objetivo de este trabajo fue desarrollar y evaluar la capacidad de precisión y estabilidad de un modelo de traductor automático de hñähñu-español, basado en una red neuronal, usando un entrenamiento fundamentado en el análisis de datos estadísticos, así como comparar su desempeño de manera cuantitativa con algoritmos de entrenamiento Backpropagation y Levenberg-Marquardt, mediante el uso del error cuadrático medio; y contrastar de forma cualitativa con los sistemas de IA generalistas como ChatGPT y Grok.

MATERIALES Y MÉTODOS

Arquitectura de la red neuronal

Se utilizó una red perceptrón multicapa entre-

nada, usando un análisis de datos estadísticos. Matemáticamente, se tiene para cada neurona j en la capa l una entrada:

$$h_j^{(l)}(k) = \sum_i W_{ji}^{(l)}(k) y_i^{(l-1)}(k) + b_j^{(l)}(k) \quad (1)$$

Donde:
$W_{ji}^{(l)}$ = el peso que conecta la neurona i de la capa $(l\text{-}1)$ con la neurona j de la capa l.
$l.\, b_j^{(l)}(k)$ = el sesgo de la neurona j en la capa l.
$y_i^{(l-1)}(k)$ = la activación de la neurona i de la capa anterior.

Para la activación de la neurona en la capa anterior se utiliza la siguiente ecuación:

$$\hat{y}_j^l(k) = \Psi\left[h_j^l(k)\right] \quad (2)$$

Donde:
$\Psi[\cdot]$ = una función de activación; entre las más comunes están las de saturación, sigmoide y tangente hiperbólica, que poseen dimensiones compatibles. En este caso, se emplea la función tangente hiperbólica como activación (Baruch y col., 2017; Ojeda y col., 2019), para la arquitectura mostrada en la Figura 1.

Algoritmo de aprendizaje

El algoritmo de aprendizaje se propuso como un algoritmo estadístico-probabilístico (EyP), que se empleó en la red, considerando que el error, definido como la diferencia entre la entrada y salida, puede ser predicho por la red neuronal, esto es:

$$e_j^{(l)}(k) = y_j(k) - \hat{y}_j^{(L)}(k) \quad (3)$$

Donde:
$e_j^{(l)}(k)$ = error.
$y_j(k)$ = entrada.
$\hat{y}_j^{(L)}(k)$ = salida.

El modelo de aprendizaje se muestra en la Figura 2, donde se incluye el algoritmo EyP propuesto, basado en el análisis de datos muestrales obtenidos mediante muestreo aleatorio (Ojeda-Misses y col., 2024), con el fin de asegurar resultados válidos, estables y generalizables. Se fundamenta en el Teorema del Límite Central (TLC), que establece que, con muestras suficientemente grandes, la distribución de la media muestral se aproxima a una normal, sin importar la distribución original (Oakland y Oakland, 2024). Esto permitió estimar probabilidades de error sobre la media usando la

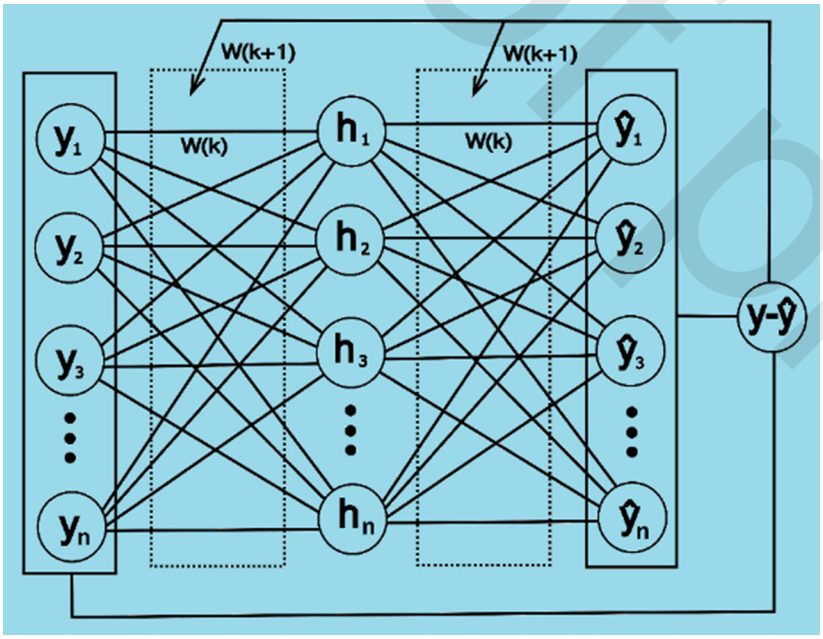


Figura 1. Arquitectura de un modelo de una red neuronal multicapa mediante corrección de error.
Figure 1. Model architecture of a multilayer neural network using error correction.

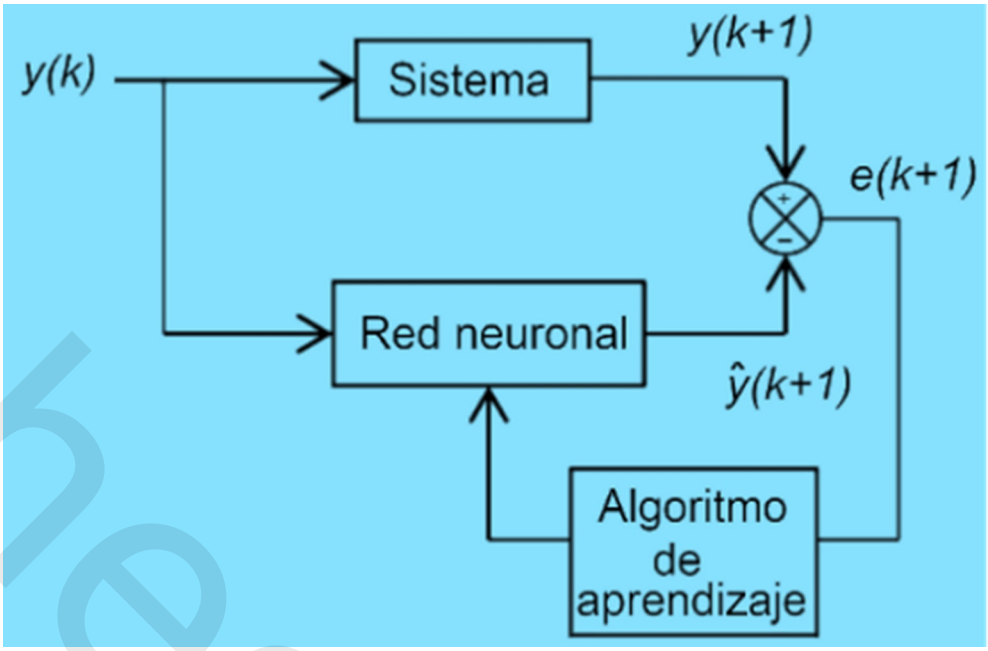


Figura 2. Modelo de aprendizaje usando el algoritmo propuesto.
Figure 2. Learning model using the proposed algorithm.

distribución normal, útil para intervalos de confianza y pruebas de hipótesis. El algoritmo propuesto aplicó dichos conceptos tomando muestras de la salida de la red neuronal y calculando su media y desviación estándar.

$$\bar{y}_j(k) = \frac{\hat{y}_1^{(L)}(k) + \hat{y}_2^{(L)}(k) + \cdots + \hat{y}_n^{(L)}(k)}{n} \quad (4)$$

Para las primeras n muestras, aplicando la ley de los grandes números, se consideró que los promedios muestrales convergen y, por lo tanto, convergen en probabilidad al valor esperado $\bar{y}_j^{(L)}(k)$ cuando $n \to \infty$. Debe tenerse en cuenta que, mientras mayor sea el número de muestras, más refinados y detallados serán los resultados estadísticos. Posteriormente, se obtuvo la desviación estándar utilizando los datos de la ecuación (4):

$$\sigma_j(k) = \sqrt{\frac{\sum_{j=1}^{n} \left(\hat{y}_j(k) - \bar{y}_j(k)\right)^2}{n}} \quad (5)$$

A continuación, se utilizó el TLC, el cual ayudó a determinar la forma de la distribución muestral de los datos obtenidos:

$$z_j(k) = \frac{y_j(k) - \bar{y}_j(k)}{\sigma_j(k) + \varepsilon} = \frac{e_j^{(L)}}{\sigma_j(k) + \varepsilon} \quad (6)$$

Donde:
$y_j(k)$ = la salida del sistema.
$\bar{y}_j(k)$ = la media estimada de los datos $\hat{y}_n^L(k)$ estimados en la salida de la red neuronal.
$e_j^{(L)}$ = el error dado por la diferencia $y_j(k) - \bar{y}_j(k)$.
$\sigma_j(k)$ = la desviación estándar.
ε = es una constante que permite que no haya divisibilidad entre cero cuando la desviación estándar decrezca y tienda a cero.

En el contexto del TLC, $z_j(k)$ es un valor estandarizado que indica a cuántas desviaciones estándar se encuentra una media muestral con respecto a la media poblacional (Ojeda-Misses y col., 2025). Esto permite utilizar la distribución normal estándar para calcular probabilidades de error, incluso cuando los datos provienen de una distribución no normal (Anderson y col., 2008; Ojeda-Misses y col., 2024).

El TLC permite estandarizar la media muestral, calcular $z_j(k)$ respecto a la media poblacional en errores estándar y estimar probabilidades de

error. Es fundamental para construir intervalos de confianza y realizar pruebas de hipótesis. Una vez estimada $z_j(k)$, la probabilidad es:

$$P_j^{(L)}(k) = \Phi(z_j(k)) \quad (7)$$

Donde:
$\Phi(k)$ = la función de distribución acumulada (FDA) de la distribución normal estándar.

En términos de cálculo, la probabilidad acumulada se expresa como:

$$\Phi(z_j(k)) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{\frac{-t^2}{2}} dt \quad (8)$$

Donde:
$\Phi(k)$ = es la función de distribución acumulada en términos de $z_j(k)$.
t = es la variable de integración.
$\frac{1}{\sqrt{2\pi}} e^{\frac{-t}{2}dt}$ = es la función de densidad de probabilidad.

La función probabilística expresa la probabilidad de que una variable adopte un valor particular, mientras que la función de distribución muestra la probabilidad acumulada hasta un cierto valor.

Ajuste de parámetros de la red neuronal
El ajuste de los parámetros de la red neuronal, específicamente los pesos sinápticos y los sesgos, se llevó a cabo mediante un proceso iterativo de aprendizaje en el que se emplearon funciones de activación no lineales, como la sigmoide y la tangente hiperbólica. Estas funciones permitieron introducir no linealidad en el modelo y regular la propagación de la señal a través de las capas, facilitando la minimización del error entre la salida estimada y la salida deseada, y contribuyendo así a la convergencia estable del algoritmo de entrenamiento.

Pesos sinápticos
Permiten determinar la importancia relativa de las conexiones neuronales, al reflejar el grado de influencia que una palabra o característica específica del idioma de origen ejerce sobre la selección de una palabra en el idioma de destino. De este modo, el análisis de los pesos proporciona información relevante sobre el proceso de aprendizaje y la contribución de cada entrada al resultado final.

$$\Delta W_j i^{(l)} = f\left(P_j^{(L)}(k)\right) = \frac{e^{P_j^{(L)}(k)} - e^{-P_j^{(L)}(k)}}{e^{P_j^{(L)}(k)} + e^{-P_j^{(L)}(k)}} \quad (9)$$

Donde:
$\Delta W_j i^{(l)}$= representa el ajuste del peso sináptico de la capa l, definida por una función tangente hiperbólica en términos de la probabilidad de error $P_j^{(L)}(k)$.

Se puede observar que $\Delta W_j i^{(l)}$ está en función de la probabilidad $P_j^{(L)}(k)$. Finalmente, los pesos sinápticos son actualizados de forma probabilística como:

$$W_{ji-n^{(l)}} \leftarrow W_j i^{(l)} \eta_1 \Delta W_j i^{(l)} \quad (10)$$

Donde:
$W_{ji_{-n}^{(l)}}$ = representa el peso sináptico actualizado de la última capa l.
$W_j i^{(l)}$ = es determinado como el peso sináptico anterior.
$\Delta W_j i^{(l)}$ = representa el ajuste del peso sináptico de la capa l.
n_1= es definida como la tasa de aprendizaje para el peso sináptico.

Los sesgos (biases)
Permiten ajustar el nivel de activación de las neuronas, actuando como términos independientes que desplazan la función de activación. Estos valores adicionales posibilitan que una neurona se active o inhiba independientemente de las entradas, lo que incrementa la flexibilidad del modelo y mejora su capacidad de representación.

$$\Delta b_j^{(l)} = f\left(P_j^{(L)}(k)\right) = \frac{L}{1 + e^{-\alpha P_j^{(L)}(k)}} \quad (11)$$

Donde:
$\Delta b_j^{(l)}$ = representa el ajuste del sesgo de la capa l, definida por una función sigmoide en términos de la probabilidad de error $P_j^{(L)}(k)$.

Se puede observar que, para el sesgo, se define como una función en términos de la probabilidad $P_j^{(L)}(k)$. Finalmente, los sesgos son actualizados de forma probabilística como:

$$b_{j_n}^{(l)} \leftarrow b_j^{(l)} + \eta_2 \Delta b_j^{(l)} \quad (12)$$

Donde:
$b_{j_n}^{(l)}$= representa el sesgo actualizado de la última capa *l*.
$b_j^{(l)}$ = es determinado como el sesgo anterior.
$\Delta b_j^{(l)}$= representa el ajuste del sesgo de la capa *l*.
n_2= es definida como la tasa de aprendizaje para el ajuste del sesgo.

Considerando como referencia que el algoritmo de BP usa la función de error $e_j^{(l)}(k)$, que actúa como un mecanismo de ajuste, cuyo fin es aplicar los ajustes correspondientes al vector de pesos sinápticos $W_j i^{(l)}(k)$ en las neuronas de cada capa. En este caso, no se emplea directamente el error numérico, sino la probabilidad de error $P_j^{(L)}(k)$, que permite medir qué tan probable es que la neurona *j* de la capa de salida haya contribuido en el error y así ajustar los pesos y sesgos. Además, se considera el índice de desempeño, definido como:

$$J(k) = \frac{1}{2} \sum^{j=1} e_j^{(l)} \quad (13)$$

Donde:
L = el número de neuronas en la capa de salida.
$e_j^{(l)}$ = definido como el error.

Vectorización de palabras

Se utilizó el contexto del procesamiento de lenguaje natural, para transformar palabras o secuencias de texto en vectores numéricos que preserven (Allgaier y col., 2024; De-la-Torre, 2025), en lo posible, características útiles del lenguaje original. Dado que las redes neuronales solo pueden operar sobre datos numéricos, el alfabeto para el hñähñu se definió como el siguiente conjunto ordenado:

alfabeto = {a, á,ä, ,b, c, d, e, é, f, g, h, i, í, j, k, l, m, n, ñ, o, ó, p, q, r, s, t, u, ú, v, w, x, y, z,!,(espacio)}.

Este conjunto de caracteres usado en la lengua hñähñu es denotado por *A*:

$$A = \{a_1, a_2, \ldots, a_n\} \quad (14)$$

Donde:
$a_1, a_2, \ldots, a_n$ representan los caracteres ordenados según su posición.

En este caso, el conjunto incluye no solo los caracteres estándar del alfabeto español, sino también las vocales acentuadas y la letra ñ. La posición de cada caracter en el conjunto se puede usar para asignarle un valor numérico real. Entonces, un carácter $c \epsilon A$ se define como:

$$f(c) = index(c) + 1/N \quad (15)$$

Donde:
index(c) = la posición del caracter.
N = el número total de caracteres en el alfabeto.

Es importante mencionar que, si el caracter no pertenece al alfabeto, se le asigna el valor de 0.

Dado que cada caracter tiene un valor numérico, una palabra se transforma en un vector de longitud fija. Se considera que una palabra está formada por una secuencia de caracteres.

El vector de la palabra se define como:

$$\vec{X} = (f(c_1), f(c_2), \ldots, f(c_m)), \quad (16)$$

Si $m < index$ del vector (posición numérica), se rellena con ceros hasta alcanzar la longitud deseada. Si $m > index$, se trunca a los primeros 20 elementos. Esta normalización asegura que todos los vectores tengan la misma dimensión, lo cual es un requisito para procesarlos con la red neuronal. En este caso, la longitud de las palabras es a lo más de 20 letras, por lo tanto, el vector se define como:

$$\vec{X} = (f(c_1), f(c_2), f(c_3), 0, 0, \ldots, 0) \in R^{20} \quad (17)$$

Finalmente, el corpus vectorizado se construye aplicando este proceso a cada palabra de la lista de palabras en hñähñu. Por ejemplo, dada una lista de tres palabras en hñähñu, cada palabra se transforma en un vector de longitud fija y el conjunto completo de vectores se presenta como una matriz definida como:

$$D = \begin{bmatrix} \overrightarrow{X_1} \\ \overrightarrow{X_2} \\ \overrightarrow{X_3} \end{bmatrix} \in R^{3\times 20} \quad (18)$$

La matriz D es el corpus de palabras de entrada, definido como un conjunto numérico que representa palabras en una forma interpretable por la red neuronal. Formalmente, $D \in R^{k\times m}$ actúa como el tensor de entrada, donde cada fila corresponde a una palabra vectorizada de longitud o dimensión fija (M) y todas ellas conforman el número total de palabras (o vocabulario total) en el corpus como:

$$D = \begin{bmatrix} \vec{X}_1 \\ \vec{X}_2 \\ \vec{X}_3 \\ \vdots \\ \vec{X}_n \end{bmatrix} \in \mathbb{R}^{K\times M} \quad (19)$$

En consecuencia, trabajar con palabras vectorizadas hace el aprendizaje más sencillo, eficiente y escalable, permitiendo que la red neuronal aprenda directamente de los datos sin depender de reglas lingüísticas rígidas. Esto resulta especialmente relevante en aplicaciones educativas, traducción automática y procesamiento de lenguas indígenas, donde la flexibilidad y capacidad de generalización del modelo son esenciales.

El algoritmo presentado fue puesto a prueba de manera automática al menos 50 veces usando un corpus de 145 palabras únicas, representativas del vocabulario cotidiano del hñähñu, y suficientes para evaluar la red sin necesidad de grandes volúmenes de datos. Cabe mencionar que, el algoritmo estadístico funciona con corpus pequeños porque aprovecha conceptos como el muestreo y el TLC, modelando la distribución de errores y patrones de las palabras. Permitiendo estimar parámetros fiables y garantizando convergencia aún con pocas palabras, maximizando la eficiencia del aprendizaje.

Cada caracter $c_i \in A$ y se convirtió en un número normalizado, por lo que, para la palabra *thani* se tiene: $t = 29$, $h = 10$, $a = 1$, $n = 15$, $i = 11$, *así:*

$$f(c_1) = \frac{29+1}{35} = 0.8571, f(c_2) = \frac{10+1}{35} = 0.3143$$

$$f(c_3) = \frac{1+1}{35} = 0.0571, f(c_4) = \frac{15+1}{35} = 0.4571$$

$$f(c_5) = \frac{11+1}{35} = 0.3429$$

De esta forma, la palabra “thani”, con 5 letras, se representó como un vector de dimensión $M = 20$ como:

$\vec{X}_{thani}$ = (0.857 1, 0.314 3, 0.057 1, 0.457 1, 0.342 9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

Evaluación del traductor mediante las métricas

El traductor automático español–hñähñu (Figura 3) se integró usando una red neuronal con arquitectura de tres capas (una de entrada, una oculta y una de salida) con 5 neuronas, respectivamente. Las pruebas de entrenamiento se realizaron utilizando 145 vocablos (como se mencionó previamente) que abarcaban pronombres, afirmaciones y negaciones, partes del cuerpo, órganos internos, números, prendas de vestir, animales, integrantes de la familia, frutas y verduras, colores y días de la semana, entre otras. Se desarrollaron pruebas de desempeño usando como función de activación la tangente hiperbólica y se evaluaron las métricas estadísticas objetivas del ECM, el ajuste de los pesos sinápticos y los sesgos.

Determinación del ECM

Se empleó como métrica principal para evaluar la precisión del entrenamiento, calculándose como el promedio de los cuadrados de la diferencia entre la palabra (o vector) predicha por el modelo y la traducción real esperada. En el análisis del ECM cada palabra de

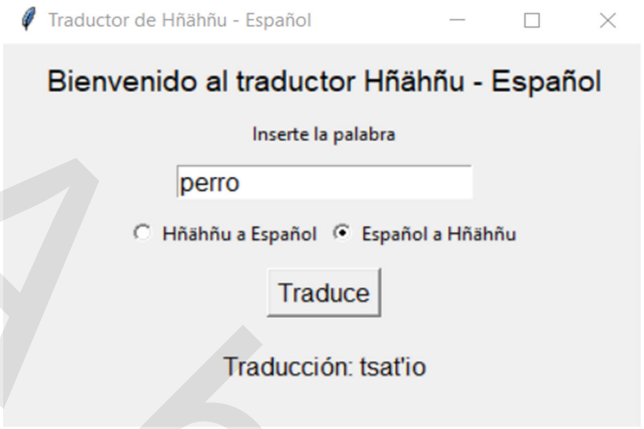


Figura 3. Ventana del traductor con palabra traducida de hñähñu a español (tsat'io-perro).
Figure 3. Translator window with word translated from Hñähñu to Spanish (tsat'io-dog).

entrada en hñähñu es representada en un espacio vectorial previamente definido, como se muestran en las Figuras 4a y 4b, y la red neuronal aprende a asociar dicho vector con su correspondiente representación en hñähñu. La validación se realiza comparando el vector de salida generado por la red con el vector objetivo de la palabra esperada, permitiendo evaluar de forma cuantitativa la precisión del proceso de traducción. El ECM se utilizó como métrica principal para medir la discrepancia entre la salida estimada y la salida deseada en el espacio vectorial, proporcionando un indicador numérico del desempeño del aprendizaje. Un valor menor de ECM indica una mayor eficiencia del modelo para aproximar correctamente el lenguaje de destino y reproducir las características del vocabulario aprendido. Se calculó en cada época con la siguiente fórmula:

$$ECM = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (20)$$

Donde:
y_i = es el valor esperado (vector de referencia de la palabra).
$\hat{y}_i$ = es el valor generado por la red neuronal.
n = es el número de elementos del vector.

Comparación entre modelos

Para la comparación del algoritmo EyP propuesto, se realizó el entrenamiento de la red neuronal utilizando el mismo conjunto de datos con los algoritmos BP y Levenberg-Marquardt (LM). El desempeño de cada método se evaluó mediante el ECM, permitiendo una comparación objetiva y directa de la eficiencia de aprendizaje entre los distintos enfoques.

Comparación contra IA

La comparación se realizó con 75 vocablos diferentes a los utilizados durante el entrenamiento de la red neuronal, durante 5 ocasiones, entre el traductor neuronal propuesto y sistemas de IA generalistas como ChatGPT y Grok (Chang y col., 2014; 2024; Al-Nazi y col., 2025; López, 2025). Se analizaron los datos promedios globales mediante una comparación, desde un enfoque cualitativo, considerando que los modelos de lenguaje de gran escala son sistemas cerrados, entrenados con corpus masivos y heterogéneos no accesibles públicamente, lo que impide una comparación directa en términos de estructura interna, parámetros y/o procesos de aprendizaje. En este contexto, la evaluación se limita deliberadamente a analizar el comportamiento de cada sistema como herramienta de traducción bajo condiciones de entrada controladas, lo cual garantiza reproducibilidad experimental y validez metodológica, especialmente en escenarios de lenguas de bajos recursos.

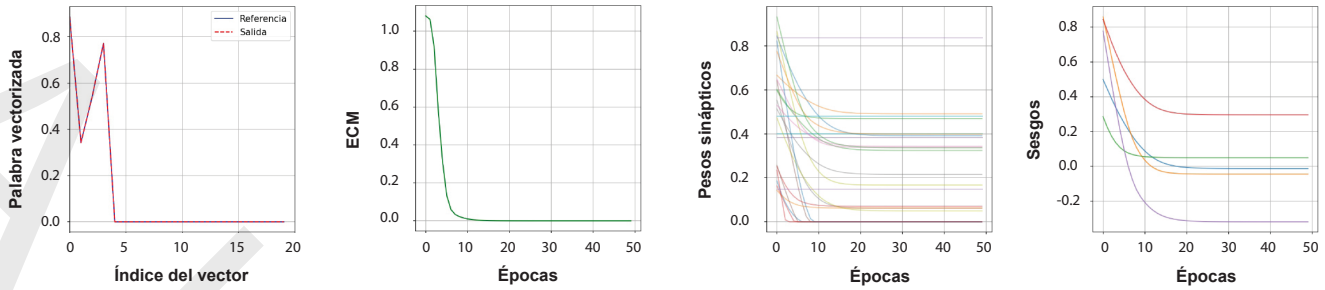
Capacidad de generalización

La red neuronal y el algoritmo de aprendizaje propuesto utilizaron 75 vocablos en hñähñu que no fueron utilizados durante la etapa de aprendizaje ni en la comparación contra IA, con el objetivo de analizar el desempeño del modelo ante palabras nuevas. Se realizó una validación cruzada k-fold con k = 10, para evaluar la eficiencia, la robustez y la capacidad de aprendizaje del método propuesto frente a variaciones en los datos de entrada no incluidos en el entrenamiento, mediante la determinación del ECM.

RESULTADOS Y DISCUSIÓN

En la comparación de la palabra vectorizada vs el índice del vector (Figuras 4a y 4b), es po-

a) xiñu-nariz



b) y'o-borrego

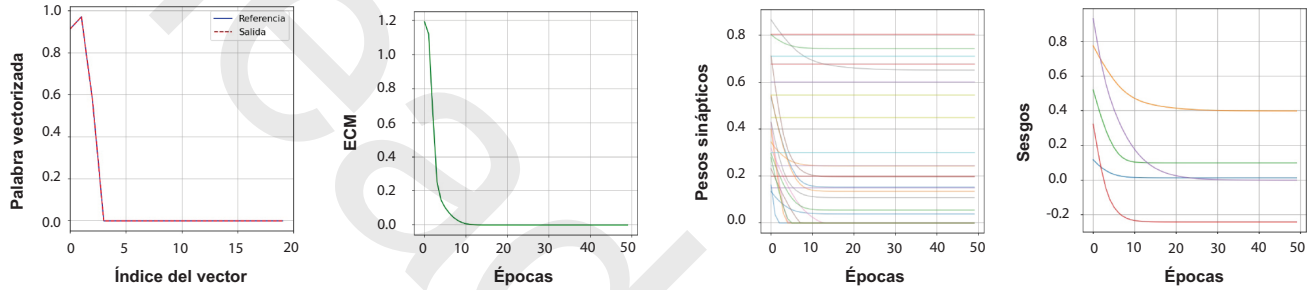


Figura 4. Aprendizaje de las palabras a) *xiñu-nariz* y b) *y'o-borrego* y sus respectivos (ECM), pesos sinápticos y sesgos.
Figure 4. Learning of the words a) *xiñu-nose* and b) *y'o-sheep* with their respective (MSE), synaptic weights, and biases.

sible observar la alta similitud en sus trayectorias de dos vocablos (*xiñu* y *y'o*), lo que indica la eficiencia del modelo. Los ECM representativos de ambos vocablos, reflejan una adecuada convergencia (tendencia a llegar a 0) del modelo, y con ello, una identificación correcta de la palabra traducida. De esta forma, la validación del traductor no se limitó a la comparación textual de las palabras traducidas, sino que se sustentó en el análisis formal del aprendizaje neuronal, donde la identificación correcta de la palabra vectorizada y la minimización del ECM confirman la efectividad del modelo (Suh y col., 2025).

El comportamiento de las convergencias de los pesos sinápticos y los sesgos de la red neuronal (Figuras 4a y 4b), indicó que la red aprendió de manera efectiva (Ying y col., 2024). Los 145 vocablos con que se entrenó a la red neuronal fueron integradas en una base de datos que permiten usar el traductor mediante una ventana como se muestra en la Figura 3. Este esquema resulta especialmente adecuado para lenguas indígenas de bajos recursos, ya que permite enriquecer la base de datos y analizar el desempeño del traductor aun cuando el corpus sea reducido.

Procesamiento de las palabras

La evaluación del entrenamiento del modelo de traducción automática español–hñähñu con el conjunto de 145 vocablos seleccionados, usando el algoritmo EyP mostró una a validación que arrojó un ECM promedio de 4.67×10^{-4} (Figura 5a), indicando alta correspondencia entre las traducciones estimadas y las de referencia en el espacio vectorial. Valores de error de

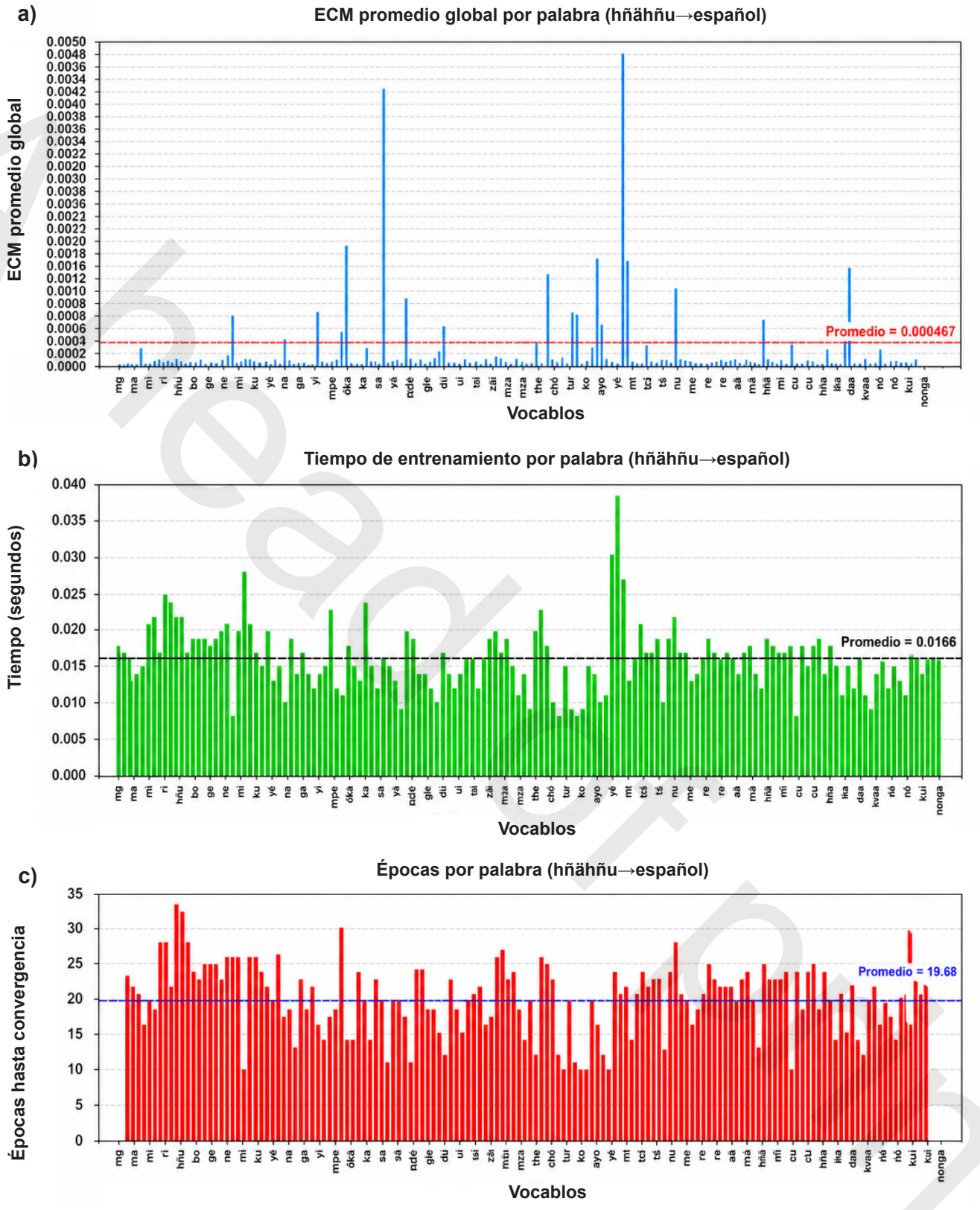


Figura 5. Errores cuadráticos medios (ECM) del promedio global (a), tiempo de entrenamiento (b) y épocas de convergencia (c), para las palabras entrenadas por la red neuronal con el algoritmo EyP para el traductor.

Figure 5. Mean squared errors (MSE) (a) average global, training time (b), and convergence epochs (c), for the words trained by the neural network using the S&P algorithm for the translator.

10^{-3} a 10^{-4} se consideran indicativos de convergencia adecuada y correcta identificación semántica, especialmente en corpus pequeños o lenguas de bajos recursos (Bengio y col., 2003; Koehn, 2010; Mikolov y col., 2013).

El tiempo promedio de procesamiento de 0.016 6 s por palabra (Figura 5b) evidencia un desempeño computacional eficiente. Tiempos menores a 0.1 s por unidad léxica son adecuados para entornos en tiempo real y herramientas de apoyo lingüístico (Koehn, 2010; Aharoni y col., 2019). Esto confirma la viabilidad del sistema para aplicaciones prácticas, incluso en contextos con recursos limitados, como entornos educativos o comunitarios, un aspecto clave para tecnologías lingüísticas en lenguas indígenas (Bird, 2020).

El número promedio de épocas para la convergencia fue de 19.68 (Figura 5c), lo que refleja la estabilidad y eficiencia del algoritmo EyP. Un bajo número de épocas indica procesos de aprendizaje bien condicionados, reduce el costo computacional y mitiga el riesgo de sobreajuste en conjuntos de datos pequeños (Joshi y col., 2020). Estos resultados brindan información sobre los tiempos de rendimiento del modelo para tareas de traducción automática en lenguas indígenas.

Comparación entre modelos

El entrenamiento del algoritmo propuesto EyP comparado con los algoritmos BP y LM, utilizando el mismo conjunto de vocablos (75) donde cada algoritmo vectoriza la palabra badu-pato, demostró que el algoritmo EyP produjo una representación inicial menor que BP y LM, las cuales muestran mayor dispersión en las etapas iniciales (Figura 6a). Aunque el EyP requirió un mayor número de épocas y tiempo de convergencia (Figura 5b y Tabla 1), mantuvo un comportamiento más estable en etapas avanzadas y fue menos sensible a las condiciones iniciales que BP y LM, los cuales, presentaron convergencias iniciales más rápidas, pero tendieron a estancarse en errores residuales mayores. Estos resultados son consistentes con reportes en la literatura sobre aprendizaje en redes neuronales con corpus pequeños o lenguas de bajos recursos. Por ejemplo, métodos como BP y LM suelen alcanzar rápidamente la minimización del error, pero su estabilidad depende fuertemente de la inicialización de (Bishop, 2006; Kayri, 2016). El algoritmo propuesto, al basarse en un enfoque estadístico–probabilístico, logra suavizar la trayectoria de aprendizaje, favoreciendo convergencia más estable y confiable, lo que es ventajoso en contextos donde la cantidad de datos es limitada (Choenni y col., 2023).

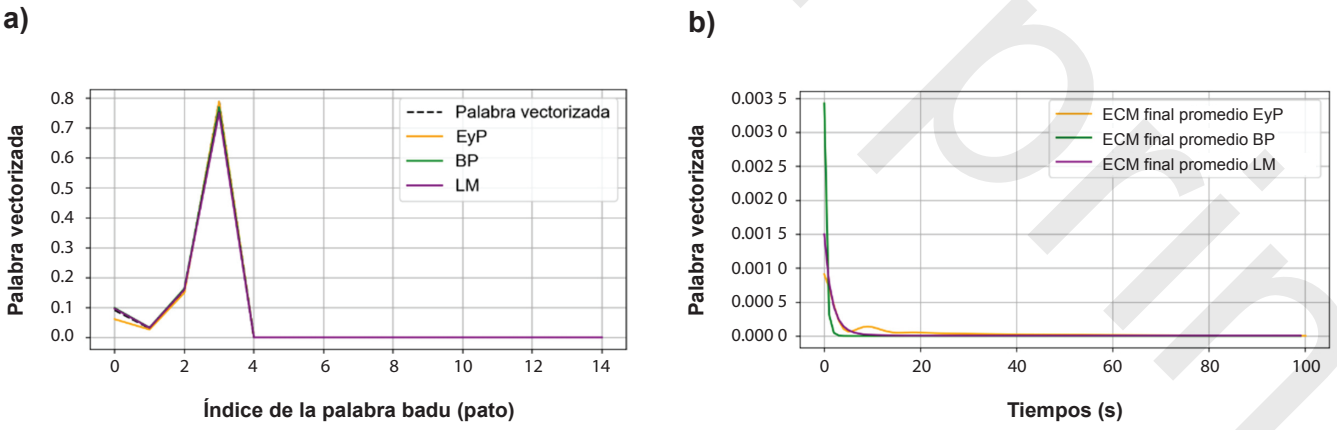


Figura 6. Comparación del aprendizaje (a) de la palabra *badu-pato* y (b) sus respectivos errores cuadráticos medios (ECM) iniciales.
Figure 6. Comparison of the learning (a) of the word badu-duck and (b) their respective average final mean square errors (MSE) initial.

Los resultados promedio globales de los tres algoritmos, al evaluar 5 vocablos, revelaron que el algoritmo EyP obtuvo el menor ECM global promedio de los 5 vocablos (1.2×10^{-5}) (Figura 7a), superando a BP y LM en términos de precisión global (ECM), lo que le requirió, en contraste, mayor tiempo (Figura 7b) y número de épocas (Figura 7c), donde una época en una red neuronal es una pasada completa de todo el conjunto de datos de entrenamiento a través del modelo. En conjunto, los resultados evidenciaron cuantitativamente que el algoritmo EyP superó a BP y LM en precisión, pero requirió mayor tiempo de procesamiento y presentó menor estabilidad en su adaptación a corpus pequeños.

Eficiencia para traducir nuevos vocablos

La validación cruzada k-fold con k = 10 para la evaluación de la capacidad de generalización

Tabla 1. Errores cuadráticos medios (ECM) para los algoritmos de aprendizaje basado en estadística y probabilidad (EyP), Backpropagation (BP) y Levenberg-Marquardt (LM) para la palabra badu (pato) en hñähñu.
Table 1. Mean squared errors (MSE) for the learning algorithms based on statistics and probability (Sand P), Backpropagation (BP), and Levenberg-Marquardt (LM) for the word badu (duck) in hñähñu.

Épocas	ECM		
	(EyP)	ByP	LM
0	0.000 914	0.003 430	0.001 503
2	0.000 456	0.000 052	0.000 435
3	0.000 236	0.000 014	0.000 248
5	0.000 069	0.000 006	0.000 093
8	0.000 136	0.000 005	0.000 033
10	0.000 137	0.000 005	0.000 022
20	0.000 058	0.000 005	0.000 014
40	0.000 031	0.000 005	0.000 014
60	0.000 020	0.000 005	0.000 014
80	0.000 014	0.000 005	0.000 014
100	0.000 009	0	0

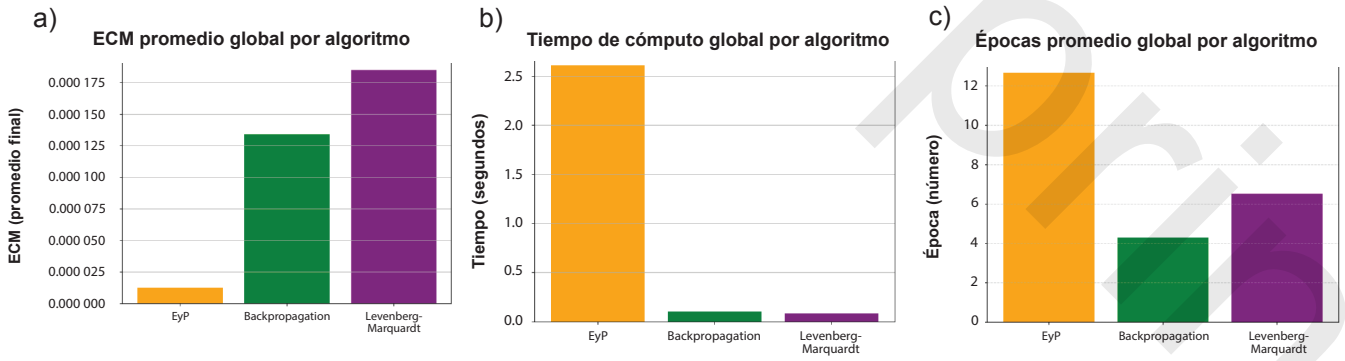


Figura 7. Resultados promedio globales de la evaluación de 5 vocablos (badu-pato, boxi-gallo, dämoni-guajolota, mey'o-chivo y tan'i- guajolote), error cuadrático promedio global (ECM) (a), tiempo de cómputo (b) número de épocas (c) para los algoritmos EyP, BP y LM.
Figure 7. Avarage overall results of the evaluation of the 5 words (badu-duck, boxi-rooster, dämoni-turkey hen, mey'o-goat y tan'i- turkey tom), Mean Squared Error (MSE) (a), computation time (b), and number epochs (c) for the EyP, BP, and LM algorithms.

de la red neuronal con el algoritmo propuesto, utilizando 75 vocablos diferentes a los utilizados durante el entrenamiento de la red neuronal hñähñu–español no incluidos en el entrenamiento original, reveló que el ECM promedio global obtenido fue de 0.076 55 (Figura 8), lo que indica un desempeño satisfactorio en la traducción de palabras no vistas. Si bien, se observaron variaciones entre folds (con valores mínimos de 0.036 y máximos de 0.216), el comportamiento general demostró que el modelo no memoriza el corpus, sino que aprende regularidades fonético–estructurales entre ambas lenguas, usando nuevas palabras como dehe-agua que logró identificar la palabra vectorizada (Figura 9a) en aproximadamente 10 épocas (Figura 9b), mientras que, el comportamiento del ECM se puede observar que converge a cero.

Comparación con otras herramientas de IA

La ventana del traductor modificada permite incorporar nuevos datos lingüísticos de manera dinámica al sistema (Figura 10). Su diseño

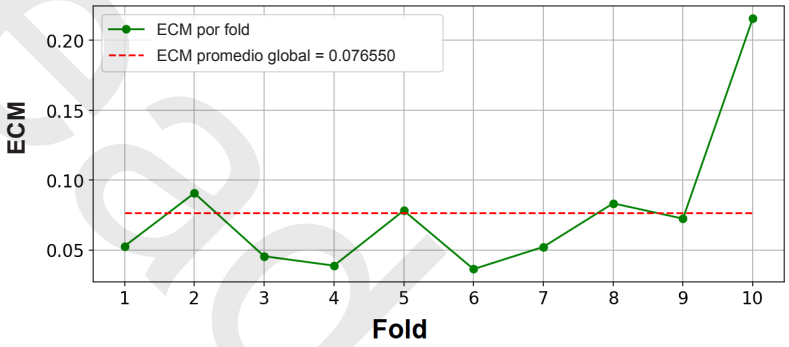


Figura 8. Errores cuadráticos medios (ECM) para los diez folds de 75 vocablos y su promedio ECM promedio global.
Figure 8. Mean squared errors (MSE) for the ten folds of the 75 words and their average MSE average global.

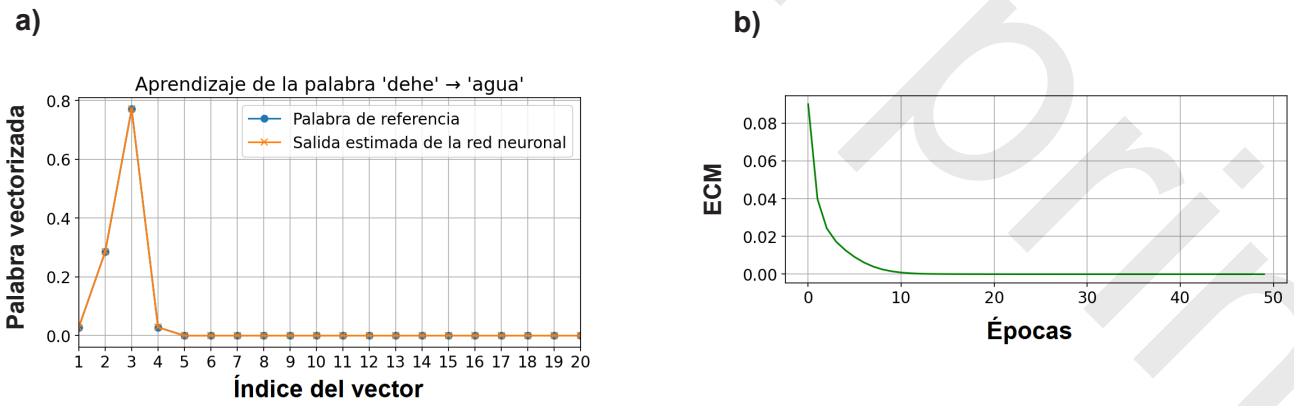


Figura 9. Identificación de la palabra vectorizada (a) dehe-agua agregada al traductor (b) y su error cuadrático medio (ECM).
Figure 9. Identification of the vectorized word (a) dehe-water added to the translator and (b) its mean squared error (MSE).

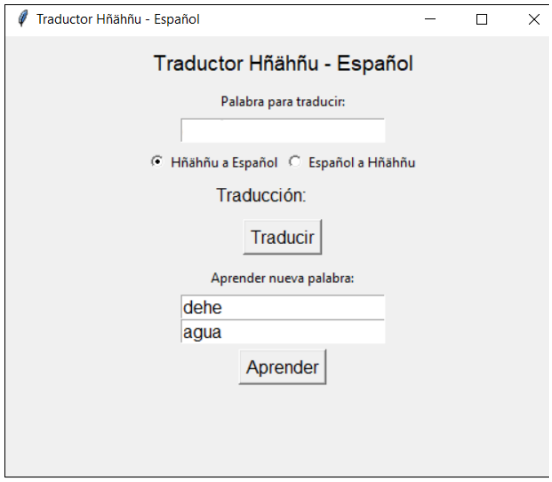


Figura 10. Ventana del traductor modificado con el fin de agregar corpus nuevo para el modelo de la red neuronal mediante el algoritmo EyP.
Figure 10. Modified translator interface designed to incorporate new corpus data into the neural network model through the EyP algorithm.

facilita la integración de un corpus adicional, enriqueciendo el proceso de aprendizaje del modelo. Esto se logra mediante el algoritmo EyP, optimizando la actualización y adaptación de la red neuronal.

En la comparación de la consistencia en la traducción entre el algoritmo EyP propuesto, Grok y ChatGPT, ante múltiples ejecuciones repetidas de una misma entrada, se observó que el algoritmo EyP mantiene una respuesta altamente estable, con variaciones mínimas y valores cercanos a la consistencia máxima, lo que refleja su comportamiento determinista y la estabilidad introducida por el esquema de aprendizaje estadístico, como puede observarse en la Figura 11a. En contraste, Grok y ChatGPT presentaron fluctuaciones más evidentes entre ejecuciones, asociadas a la naturaleza probabilística de los modelos generativos de lenguaje. Estas variaciones pueden afectar la precisión terminológica en aplicaciones donde se requiere alta repetibilidad y coherencia léxica, especialmente en contextos técnicos o educativos (Franganillo, 2023; Yi y col., 2024; Herrera-Poyatos y col., 2025).

En lo que respecta al nivel de reproducibilidad comparativo entre los tres enfoques evaluados, el algoritmo EyP alcanzó el mayor nivel de reproducibilidad debido a que su ejecución se basa en un proceso completamente definido y controlado localmente, donde los parámetros del modelo y el proceso de entrenamiento permanecen constantes, como se observa en la Figura 11b. Por el contrario, Grok y ChatGPT mostraron niveles de reproducibilidad inferiores, lo cual puede atribuirse a la dependencia de infraestructuras externas, ajustes dinámicos del modelo y mecanismos internos no completamente accesibles al usuario. Esta diferencia resalta la ventaja de los sistemas deterministas en escenarios donde la replicación exacta de resultados es un requisito fundamental (Antunes y col., 2024; Semmelrock y col., 2025).

Finalmente, el tiempo de procesamiento relativo entre el EyP, Grok y ChatGPT mostró que evidenció que el EyP presentó el menor tiempo de procesamiento, con un promedio aproximado de 0.016 6 s por palabra, demostrando una elevada eficiencia computacional (Figura 11c). En contraste, Grok y ChatGPT requirieron mayores tiempos de respuesta debido al procesamiento distribuido y a la comunicación con servidores remotos, lo que incrementó la latencia y el consumo de recur-

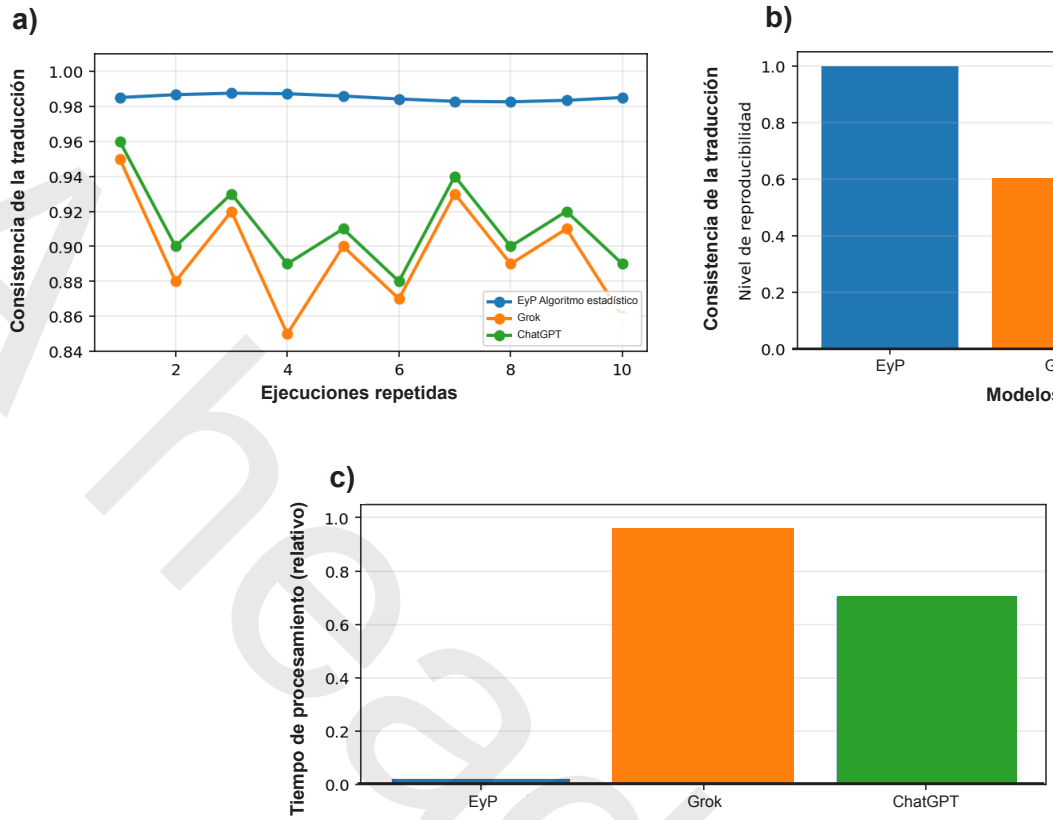


Figura 11. Comparativa entre el traductor EyP (algoritmo estadístico) y las herramientas de IA (Grok y ChatGPT), ejecuciones repetidas (a), nivel de reproductibilidad (b) y tiempos de procesamiento (c).
Figure 11. Comparison between the translator (statistical algorithm) and AI tools (ChatGPT and Grok), repeated runs (a), level of reproducibility (b), and processing times (c).

sos. Este comportamiento sugiere que el algoritmo propuesto resulta especialmente adecuado para aplicaciones en tiempo real, entornos educativos y sistemas con restricciones computacionales, donde la rapidez de respuesta y la estabilidad son factores críticos.

Los resultados de la comparación de la eficiencia del traductor propuesto contra los sistemas de IA generalistas ChatGPT y Grok sugieren que, debido a que estos no cuentan con entrenamiento específico en hñähñu, presentan errores en vocabulario indígena poco documentado. También carecen de información que ofrezca métricas cuantitativas de error y validación estadística, además de no permitir aprendizaje incremental controlado (Chang y col., 2014; 2024). En contraste, el sistema propuesto generó traducciones consistentes dentro de su corpus, identificó palabras fuera de vocabulario y ofreció métricas estadísticas verificables (López, 2025). Esto indica que, aunque las IA generalistas son adecuadas para lenguas de alto recurso, aún no son adecuadamente competitivas en lenguas indígenas de bajo recurso, donde los modelos especializados presentan ventajas claras (Al-Nazi y col., 2025).

La comparación entre el traductor neuronal propuesto y sistemas de IA generalistas como ChatGPT y Grok se realizó exclusivamente a nivel funcional y cualitativo, considerando el desempeño observable como herramienta de traducción. Mientras que los modelos generalistas están diseñados para maximizar la versatilidad lingüística y la generación de texto en múltiples dominios, el algoritmo propuesto se enfoca en la traducción léxica precisa me-

diante representaciones vectoriales explícitas y un proceso de aprendizaje estadístico–probabilístico controlado.

La Tabla 2 compara cualitativamente el traductor propuesto con sistemas de IA generalistas, enfocándose en la traducción léxica de una lengua indígena de bajos recursos, no en inteligencia general. El traductor propuesto es determinista, garantizando que una misma palabra de entrada siempre genere la misma traducción, lo cual es crucial para documentación lingüística, preservación cultural y educación. En contraste, los modelos generalistas producen salidas probabilísticas que pueden variar incluso ante entradas idénticas. Además, el algoritmo propuesto es altamente reproducible, pues puede ejecutarse con pesos sinápticos, parámetros y corpus conocidos. Los sistemas generalistas, al operar como cajas negras con actualizaciones constantes, limitan la reproducibilidad científica y el control sobre sus resultados.

Finalmente, los resultados muestran que, aunque los sistemas de IA generalistas como ChatGPT y Grok poseen una gran capacidad expresiva y cobertura lingüística amplia, no están optimizados para la traducción léxica precisa ni para la preservación de lenguas indígenas de bajos recursos (Iyer y col., 2024). El traductor propuesto, al estar basado en palabras vectorizadas y un algoritmo estadístico–probabilístico diseñado específicamente para el hñähñu, ofrece ventajas claras en términos de consistencia, reproducibilidad, eficiencia y adaptación cultural (Wang y Wang, 2025). En este sentido, ambos enfoques deben considerarse complementarios: mientras que los modelos generalistas son adecuados para tareas abiertas y de lenguaje natural amplio, el traductor propuesto resulta más apropiado para aplicaciones lingüísticas especializadas, controladas y orientadas a la preservación cultural (Ataman, 2025).

La utilización de principios estadísticos, como la normalización del error mediante funciones de distribución acumulada, introdujo un enfoque alternativo al aprendizaje tradicional en redes neuronales. Dicho mecanismo permitió ajustar los pesos sin recurrir al cálculo de derivadas, lo cual representó una ventaja significativa frente a métodos clásicos como el BP y LM, que dependen fuertemente del gradiente (Tabla 1 y Figura 7). En este sentido, el algoritmo EyP mostró mayor estabilidad numérica y menor sensibilidad a problemas como mínimos locales o gradientes nulos. Además, al evitar operaciones derivativas

Tabla 2. Comparación cualitativa entre el traductor propuesto y sistemas de IA generalistas.
Table 2. Qualitative comparison between the proposed translator and generalist AI systems.

Criterio	Traductor propuesto	ChatGPT/Grok
Tipo de modelo	Red neuronal determinista con corrección estadística	Modelo generativo probabilístico de gran escala
Tipo de salida	Determinista (misma entrada → misma salida)	Estocástica (salida variable ante misma entrada)
Consistencia	Alta	Variable
Reproducibilidad	Alta (ejecución local, pesos fijos)	Limitada (modelo cerrado, versiones dinámicas)
Precisión léxica	Alta (comparación vector–vector)	Variable (prioriza fluidez semántica)
Adaptación cultural	Alta (corpus hñähñu específico)	Generalista
Eficiencia computacional	Alta (0.016 6 s por palabra)	Dependiente de infraestructura remota
Lenguas de bajos recursos	Limitado	Limitado

complejas, es posible reducir la carga computacional en ciertos escenarios, favoreciendo su implementación en contextos con recursos limitados o datos ruidosos. No obstante, aunque estas ventajas le otorgan competitividad, es importante considerar que los métodos basados en gradiente como Backpropagation y Levenberg-Marquardt siguen siendo altamente eficientes en problemas bien condicionados, como se muestra en la Tabla 1, por lo que la elección del enfoque depende del tipo de aplicación y las características del conjunto de datos.

CONCLUSIONES

El algoritmo EyP demostró tener un enfoque efectivo para la traducción automática hñähñu–español, mostrando convergencia estable, precisa y robusta frente a variaciones morfológicas (estructurales) y fonológicas propias del hñähñu. Estas características lo hacen especialmente adecuado para lenguas indígenas como el hñähñu con alta complejidad lingüística y escasez de datos. Los resultados evidenciaron que EyP no solo aprendió correspondencias léxicas directas, sino que capturó patrones estructurales a nivel de caracter, favoreciendo la representación de relaciones semánticas entre hñähñu y español. Su estructura adaptativa permitió el aprendizaje incremental, incorporando nuevas palabras sin reiniciar el entrenamiento completo, lo que refuerza su aplicabilidad en contextos educativos bilingües y proyectos de preservación lingüística. El sistema basado en EyP mostró ventajas contra las IA generalistas ChatGPT y Grok, al ofrecer traducciones consistentes, métricas verificables y posibilidad de ampliación, y al no ser estos, actualmente, modelos especializados para lenguas de bajo recurso. Entre los desafíos críticos que persisten, destacan la falta de estandarización de la escritura hñähñu, variación dialectal y tonal, escasez de corpus digitales y limitada disponibilidad de herramientas para generar texto a partir de registros orales. Por ello, es necesario complementar los modelos con datos sintéticos, ampliar corpus léxicos y morfológicos y desarrollar módulos de reconocimiento de voz. Como perspectiva futura, se propone extender EyP a otros pares de lenguas indígenas, integrar módulos de audio para enseñanza de pronunciación y generar traductores estadístico-neurales culturalmente adaptados. Actualmente, EyP representa un avance significativo al combinar estadística, IA y lingüística aplicada, proporcionando una herramienta científicamente sólida y socialmente relevante para la preservación y revitalización de lenguas originarias.

DECLARACIÓN DE CONFLICTO DE INTERESES

Los autores declararon no tener conflictos de intereses de ningún tipo.

REFERENCIAS

Aguilar-Santiago, C. A. y García-Zúñiga, H. A. (2023). Tecnologías del lenguaje aplicadas al procesamiento de lenguas indígenas en México: una visión general. *Lingüística y Literatura*, 44(84), 79-102. https://doi.org/10.17533udea.lyl.n84a04

Aharoni, R., Johnson, M., & Firat, O. (2019). Massively multilingual neural machine translation. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computatioal Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 3874-3884).

Allgaier, J. & Pryss, R. (2024). Cross-Validation Visualized: A Narrative Guide to Advanced Methods. *Machine Learning and Knowledge Extraction*, 6(2), 1378-1388. https://doi.org/10.3390/make6020065

Al-Nazi, Z., Hossain, M. R., & Al-Mamun, F. (2025). *Evaluation of open and closed-source LLMs for low-resource language with zero-shot, few-shot, and chain-of-thought prompting. Natural Language Processing Journal*, 10, 100124. https://doi.org/10.1016/j.nlp.2024.100124

Anderson, D. R., Sweeney, D. J., Williams, T. A., Camn, J. D., & Martin, K. (2008). Estadística para administración y economía (Tenth edition). Cengage Learning. https://books.google.com.

mx/books/about/Estad%C3%ADstica_Para_Administraci%C3%B3n_Y_Econ.html?id=8SfLwAEACAAJ

Antunes, B. A. & Hill, D. R. C. (2024). Reproducibility, replicability and repeatability: A survey of reproducible research with a focus on high performance computing. *Computer Science Review*, 53, 100655. https://doi.org/10.1016/j.cosrev.2024.100655

Ataman, D. (2025). Machine translation in the era of large language models. *Information,* 16(9), 723. https://doi.org/10.3390/info16090723

Baruch, I. S., Quintana, V. A., & Reynaud, E. P. (2017). Complex-valued neural network topology and learning applied for identification and control of nonlinear systems. *Neurocomputing*, 233, 104-115. https://doi.org/10.1016/j.necom.2016.09.109

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3, 1137-1155.

Bird, S. (2020). Decolonising speech and language technology. In Proceedings of the 28th international conference on computational linguistics (pp. 3504-3519).

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer. https://doi.org/10.1117/1.2819119

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K.,Chern, K., Yi, X., Wang, C., Ye, W., Zhang, Y., Chang, Y., Yang, Q, & Xie, X. (2024). A survey on evaluation of large language models: Metrics, applications, and methodologies. ACM Computing Surveys. https://doi.org/10.1145/3641289

Çelik, Ö. & Koç, B. C. (2021). TF IDF, Word2vec ve fasttext vektör model yöntemleri ile türkçe haber metinlerinin sınıflandırılması. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, 23(67), 121-127. https://doi.org/10.21205/deufmd.2021236710

Choenni, R., Garrette, D., & Shutova, E. (2023). Cross-lingual transfer with language-specific subnetworks for low-resource dependency parsing. *Computational Linguistics*, 49(3), 613-641. https://doi.org/10.1162/coli_a_00482

Chongchong-Li, Y., Liu, Y., & Ma, Z. M. (2024). Neural networks taking probability distributions as input: A framework for analyzing exchangeable networks. *Neurocomputing,* 584, 127572. https://doi.org/10.1016/j.neucom.2024.127572

De-la-Vega, L. M. (2017) Aprendiendo otomí (Hñähñu). Comisión Nacional para el Desarrollo de los Pueblos Indígenas. [En línea]. Disponible en: https://www.gob.mx/cms/uploads/attachment/file/302157/aprendiendo-otomi-temoaya-estado-de-mexico-web.pdf. Fecha de consulta: 15 de abril de 2025.

De-la-Torre, J. (2025). Transformadores: Fundamentos teóricos y aplicaciones. arXiv. https://arxiv.org/abs/2302.09327

Dyson, L. E., Grant, S., & Hendriks, M. (2016). Indigenous people and mobile technologies. Routledge.

Dyson, L. E., Hendriks, M., & Grant, S. (2007). Information technology and indigenous people. In L. E. Dyson, M. Hendriks, & S. Grant (Eds.), *Information Technology and Indigenous People* (pp. 1-12). IGI Global. https://doi.org/10.4018/978-1-59904-298-5

Escorza-Sánchez, Y. M., Martínez-Martín, G., Saldaña-Tapia, Y. y Maldonado-Catalán, O. (2018). Aplicación móvil para reforzar el aprendizaje de la lengua Hñähñu. *Revista de Tecnología y Educación*, 2(6), 23-31. https://www.ecorfan.org/republicofperu/research_journals/Revista_de_Tecnologia_y_Educacion/vol2num6/Revista_de_Tecnolog%C3%ADa_y_Educaci%C3%B3n_V2_N6_4.pdf

Franganillo, J. (2023). Los grandes modelos de lenguaje: una oportunidad para la profesión bibliotecaria. Anuario ThinkEPI, 17. https://doi.org/10.3145/thinkepi.2023.e17a28

Gallardo-Arias, P. (2012). Ritual, palabra y cosmos otomí: yo soy costumbre, yo soy de antigua. UNAM. [En línea]. Disponible en: https://historicas.unam.mx/publicaciones/publicadigital/libros/ritualpalabra/RPC004.pdf. Fecha de consulta: 25 de abril de 2025.

González-Servín, C., Sidorov, G., Maldonado-Sifuentes, C. E., & Núñez-Prado, C. J. (2025). Transformer-based approaches for purépecha translation: Advancing indigenous language preservation. *International Journal of Combinatorial Optimization Problems and Informatics*, 16(1), 64-74. https://doi.org/10.61467/

2007.1558.2025.v16i1.595

HaCohen, Y., Miller, D., & Yigal, Y. (2020). The influence of preprocessing on text classification using a bag of words representation. *Plos One*, 15(5), e0232525. https://doi.org/10.1371/journal.pone.0232525

Hernández-Cruz, L., Victoria-Torquemda, M. y Sinclair-Crawford, D. (2010). Diccionario del Hñähñu (otomí) del Valle del Mezquital, estado de Hidalgo. Instituto Lingüístico de verano, A.C. [En línea]. Disponible en: http://docencia.uaeh.edu.mx/estudios-pertinencia/docs/hidalgo-municipios/Valle-Del-Mezquital-Diccionario-Hnahnu.pdf. Fecha de consulta: 15 de abril de 2025.

Hernández-Pérez, Y., Guzmán-Villa, M. G. y Simón-Ñonthe, E. (2020). Un libro electrónico en lengua hñähñu para promover el rescate y la ingesta de alimentos saludables. *Revista Lengua y Cultura*, 1(2), 77-84. https://doi.org/10.29057/lc.v1i2.5432

Herrera-Poyatos, D., Peláez-González, C., Zuheros, C., Herrera-Poyatos, A., Tejedor, V., Herrera, F., & Montes, R. (2025). An overview of model uncertainty and variability in LLM-based sentiment analysis: challenges, mitigation strategies, and the role of explainability. *Frontiers in Artificial Intelligence*, 8, 1609097. https://doi.org/10.3389/frai.2025.1609097

Hinton, L. & Hale, K. (2001). The Green Book of Language Revitalization in Practice. Academic Press. https://www.journals.uchicago.edu/doi/10.1086/424555

INEGI, Instituto Nacional de Estadística y Geografía (2024). Estadísticas a propósito del día Internacional de los Pueblos Indígenas. [En línea]. Disponible en: https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2024/EAP_PueblosInd24.pdf. Fecha de consulta: 15 de abril de 2025.

INEGI, Instituto Nacional de Estadística y Geografía (2015). Encuesta Intercensal 2015. Lenguas indígenas nacionales y su distribución geográfica. Instituto Nacional de Estadística y Geografía. [En línea]. Disponible en: https://www.inegi.org.mx/programas/intercensal/2015/ Fecha de consulta: 15 de abril de 2025.

INALI, Instituto Nacional de Lenguas Indígenas (2015). Estadísticas de la lengua otomí. INALI. [En línea]. Disponible en: https://site.inali.gob.mx/Micrositios/normas/estadisticas_otomi.html. Fecha de consulta: 1 de marzo de 2025

Iyer, V., Malik, B., Zhu, W., Stepachev, P., Chen, P., Haddow, B., & Birch, A. (2024). Exploring very low resource translation with LLMs, The University of Edinburgh's submission to Americas, NLP 2024 translation task. *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas*, 209-220. https://doi.org/10.18653/v1/2024.americasnlp-1.25

Joshi, P., Santy, S., Budhiraja, A., Bali, K., & Choudhury, M. (2020). The state and fate of linguistic diversity and inclusion in the NLP world. ACL.

Kayri, M. (2016). Predictive abilities of Bayesian regularization and Levenberg–Marquardt algorithms in artificial neural networks: A comparative empirical study on social data. *Mathematics and Computational Applications*, 21(2), 20. https://doi.org/10.3390/mca21020020

Khoussi, S., Heckert, A., Battou, A., & Bensalem, S. (2021). A neural networks-based methodology for fitting data to probability distributions. *En 2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA)* (pp. 1-7). IEEE. https://doi.org/10.1109/AICCSA53542.2021.9686821

Koehn, P. (2010). Statistical Machine Translation. Cambridge University Press.

Kurniawan, F. W. & Maharani, W. (2020). Indonesian Twitter Sentiment Analysis Using Word2Vec. 2020 International Conference on Data Science and Its Applications (ICoDSA). IEEE. https://doi.org/10.1109/ICoDSA50139.2020.9212906

Lastra, Y. (2010). Los otomíes, su lengua y su historia. Publicaciones del Instituto de Investigaciones Antropológicas, UNAM. https://editorialiia.unam.mx/omp/index.php/publicaciones/catalog/book/otomies_lengua_historia

Linka, K., Schäfer, A., Meng, X., Zou, Z., Karniadakis, G. E., & Kuhl, E. (2022). Bayesian physicsinformed neural networks for real world nonlinear dynamical systems. *Computer Methods*

in Applied Mechanics and Engineering, 402, 115346. https://doi.org/10.1016/j.cma.2022.115346

López, D. (2025). Dificultades asociadas en el uso de CHATGPT desde la perspectiva del estudiante. *CIENCIA UNEMI*, 18(47), 87-95. https://doi.org/10.29076/issn.2528-7737vol18iss47.2025pp87-95p

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. ICLR.

Mohebali, B., Tahmassebi, A., Meyer-Baese, A., & Gandomi, A. H. (2020). Probabilistic neural networks: A brief overview of theory, implementation, and application. In P. Samui, D. T. Bui, S. Chakraborty, & R. C. Deo (Eds.), *Handbook of Probabilistic Models* (pp. 347-367). Butterworth-Heinemann. https://doi.org/10.1016/B978-0-12-816514-0.00014-X

Nurdin, A., Seno-aji, B., Bustamin, A., & Abidin, Z. (2020). Perbandingan kinerja word embedding Word2Vec, GloVe, dan FastText pada klasifikasi teks. *Jurnal Tekno Kompak,* 14, 74. https://doi.org/10.33365/jtk.v14i2.732

Oakland, J. & Oakland, R. (2024). Statistical process control and data analytics (8th ed.). Routledge. https://doi.org/10.4324/9781003439080

Ojeda, M. A., Baruch, I. S., & Soria, A. L. (2019). A real-time identification for hand-based movements using Recurrent Complex-Valued Neural Networks. *2019 IEEE 4th Colombian Conference on Automatic Control (CCAC),* 1-6. https://doi.org/10.1109/CCAC.2019.8920864

Ojeda-Misses, M. A., Martines-Arano, H., López-Morales, V., Franco-Árcega, A., & Márquez-Grajales, M. (2024). Self-tuned closed-loop controller based on statistical data using a servomechanism. *2024 XXVI Robotics Mexican Congress (COMRob),* 27-32. https://doi.org/10.1109/COMRob64055.2024.10777440

Ojeda-Misses, M. A., Martines-Arano, H., Sampedro-Mendoza, A., Franco-Árcega. A. y López-Morales, V. (2025). Diseño de un controlador mediante datos estadísticos en lazo cerrado para un servomecanismo mediante una técnica de autosintonización. *RIIIT Revista Internacional de Investigación e Innovación Tecnológica,* 12(72). https://riiit.com.mx/apps/site/files_v2450/controlador_hgo._2_riiit_ene-feb_2025.pdf

Ramos-Aguilar, E., Olvera-López, J. A., & Olmos-Pineda, I. (2025). WavLM-Based Automatic Pronunciation Assessment for Yuhmu Speech: A Low-Resource Language. *Computación y Sistemas,* 29(3), 1257-1270. https://doi.org/10.13053/CyS-29-3-5913

Remus, W. & O'Connor, M. (2001). Neural networks for Time-Series Forecasting. In: Armstrong, J. S. (Eds.), Principles of Forecasting. *International Series in Operations Research & Management Science,* 30. Springer. https://doi.org/10.1007/978-0-306-47630-3_12

Santiago-Benito, H., Córdova-Esparza, D. M., Castro-Sánchez, N. A., García-Ramirez, T., Romero-González, J. A., & Terven, J. (2024). *Automatic Translation between Mixtec to Spanish Languages Using Neural Networks. Applied Sciences,* 14(7), 2958. https://doi.org/10.3390/app14072958

Semmelrock, H., Ross-Hellauer, T., Kopeinik, S., Theiler, D., Haberl, A., Thalmann, S., & Kowald, D. (2025). Reproducibility in machine-learning-based research: Overview, barriers and drivers. *IEEE ACCESS* 46(2), 211860-211868. https://doi.org/10.1109/ACCESS.2020.3039833

Sennrich & Zhang (2019). Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* pages 211-221, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1021

Singh, S., Kumar, K., & Kumar, B. (2022). Sentiment Analysis of twitter data using TF-IDF and machine learning techniques. 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON). 10.1109/COM-IT-CON54601.2022.9850477.

Suh, N. & Cheng, G. (2025). A survey on statistical theory of deep learning: Approximation, training dynamics, and generative models. *Annual Review of Statistics and Its Application*, 12, 177-207. https://doi.org/10.1146/annurev-statistics-040522-013920

Vargas-García, I. (2017). Experiencias de un proyecto de revitalización lingüística del hñähñu

(otomí) del Valle del Mezquital, Hidalgo. *Zeitschrift für romanische Philologie*, 133(4), 1064-1090. https://doi.org/10.1515/zrp-2017-0055

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. arXiv. https://doi.org/10.48550/arXiv.1706.03762

Wang, H. & Wang, X. (2025). Enhancing translation accuracy and learning outcomes for low resource languages: AI fine tuning and learner adoption factors. *System*, 134, 103807. https://doi.org/10.1016/j.system.2025.103807

Yi, Q., Chen, X., Zhang, C., Zhou, Z., Zhu, L., & Kong, X. (2024). Diffusion models in text generation: a survey. *PeerJ Computer Science*, 10, e1905. https://doi.org/10.7717/peerj-cs.1905

Ying, H., Song, M., Tang, Y., Xiao, S., & Xiao, Z. (2024). Enhancing deep neural network training efficiency and performance through linear prediction. Scientific Reports, 14, Article 15197. https://doi.org/10.1038/s41598-024-65691-0