

Pontificia Universidad Católica del Ecuador

Facultad de Ingeniería

Escuela de Sistemas



TEMA:

ESTUDIO COMPARATIVO DEL TRADUCTOR DE IDIOMAS GOOGLE TRANSLATE Y
CHATGPT VS LA UTILIZACIÓN DE ALGORITMOS DE MACHINE TRANSLATION EN
PYTHON

AUTOR:

Mathias Giuseppe Lettere Dragosavljevich

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS Y
COMPUTACIÓN

QUITO DM, Diciembre – 2023

DEDICATORIA

Dedicado a mi madre Milica, padre, Angelo y mis abuelos, Laura y Matja, Elfriede y Gerhard.

AGRADECIMIENTO

Agradezco a Milica Dragosavljevich y Angelo Lettere por el apoyo incondicional y su experiencia en los campos de Lingüística y Traducción que fueron esenciales para lograr completar este trabajo. Asimismo, agradezco a mi tutor Ing. Charles Escobar igualmente por su apoyo y guía durante el proceso y a los ingenieros Oswaldo Espinoza y Nelson Salgado por ayudar en concretizar y definir este trabajo. Extiendo mi gratitud también a los estudiantes y traductores oficiales que tomaron parte en las entrevistas y evaluación de las traducciones.

RESUMEN

Actualmente, se han incrementado la cantidad de modelos de Inteligencia Artificial disponibles en el mercado para una gama de temas. En este estudio se enfocó en la capacidad de traducción que tienen tanto las IA para el Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés: *Natural Language Processing*); comparando modelos de Traducción automática Neuronal (NMT, o Neural Machine Translation, por sus siglas en inglés) y Grandes Modelos de Lenguaje o LLM (Large Language Models). Se siguió un tutorial para crear un modelo de NMT con PyTorch , en el cual se encontró que este tipo de modelos requieren de grandes cantidades de datos, infraestructura y tiempo para lograr resultados como los que se observa con Google Translate o Chat GPT. Igualmente, Chat GPT demostró una gran capacidad de generar traducciones fieles al texto original, lo que resultó sorprendente tanto para traductores oficiales como estudiantes de traducción, recibiendo las mejores calificaciones tanto en fidelidad como en una revisión de calidad de la traducción. Al entrevistar algunos traductores oficiales y estudiantes, se encontró que principalmente, los modelos de IA dedicada a la traducción son utilizadas de manera referencial, para revisar un término o confirmar si existe una traducción para terminología específica, además de que los LLM no son usados ni conocidos ampliamente en este contexto.

ÍNDICE

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS	IV
ÍNDICE DE FIGURAS	IV
CAPÍTULO I: INTRODUCCIÓN	8
1. MARCO DE REFERENCIA	8
1.1. TEMA	8
1.2. JUSTIFICACIÓN	8
1.3. PLANTEAMIENTO DEL PROBLEMA.....	9
1.4. OBJETIVOS	10
1.4.1. GENERAL:.....	10
1.4.2. ESPECÍFICOS:.....	10
1.5. ALCANCE	11
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA.....	13
2. Marco Teórico.....	13
2.1. Generalidades	13
2.1.1. Traducción	13
2.1.2. Semántica.....	14
2.1.3. Sintaxis	15

2.1.4.	Pragmática.....	15
2.1.5.	Procesamiento de Lenguaje Natural	16
2.1.6.	Traducción de Máquina.....	17
2.1.7.	Large Language Models (LLM)	18
2.1.8.	Neural Machine Translation (NMT)	19
2.2.	Arquitecturas de Redes Neuronales.....	19
2.2.1.	Redes Neuronales Estándar	19
2.2.2.	Redes Neuronales Recurrentes (RNN).....	21
2.2.3.	Backpropagation Through Time (BPTT).....	22
2.2.4.	Descenso por Gradiente	23
2.2.5.	Arquitecturas de RNN	24
2.2.6.	Redes Adversarias Generativas.....	28
2.2.7.	Redes Neuronales con Transformadores.....	29
2.3.	GPT-3	33
2.4.	Machine Translation en Google y con Python	34
2.4.1.	Google Translate (GNMT).....	34
2.4.2.	PyTorch torch.nn.....	35
2.5.	Métricas de Evaluación	36
2.5.1.	Cross-Entropy Loss	36
2.5.2.	Métrica BLEU.....	37

CAPÍTULO III: METODOLOGÍA	39
3. Metodología de desarrollo del plan de trabajo de titulación.....	39
3.1. Investigación Cualitativa.....	39
3.2. Enfoque Metodológico.....	43
3.2.1. Metodología de Ciencia de Datos	43
CAPÍTULO IV:	44
4.1. Comparativa de Metodologías de IA para NLP en línea	44
4.1.1. Tabla comparativa de traducciones.....	46
4.2. Diseño de la RNN para realizar NMT con PyTorch.....	53
4.2.1. Entendimiento de los Datos	53
4.2.2. Preparación de los Datos	55
4.2.3. Modelado	56
4.2.4. Evaluación del modelo	66
4.3. Comparativas de la calidad de traducción de las IA	70
4.4. Análisis del impacto que tienen los softwares de traducción IA sobre la profesión de traducción y sus implicaciones	75
4.4.2. Aceptación social	78
CONCLUSIONES Y RECOMENDACIONES	82
Conclusiones.....	82
Recomendaciones.....	85

BIBLIOGRFÍA.....	87
ANEXOS	90
6.1. Anexo A:.....	90
6.2. Anexo B:	90
6.3. Anexo C:	90
6.4. Anexo D:	90
6.5. Anexo E:	90
6.6. Anexo F:.....	90

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

ÍNDICE DE FIGURAS

Ilustración 1	Arquitectura de una Red Neuronal estándar	https://images.app.goo.gl/tZjGkXgk9ja9bJRC721
Ilustración 2	Arquitectura de una RNN estándar	https://images.app.goo.gl/U1G1vqMxh6cv25F1A22
Ilustración 3	Arquitectura de una célula de memoria para LSTM	http://bit.ly/3LNnmGJ25
Ilustración 4	Célula de GRU con las fórmulas de las compuertas y su forma	https://images.app.goo.gl/hcv4HHYhw8FCdjtW626

Ilustración 5 Estructura general de una BRNN a través de tres pasos de tiempo
<http://bit.ly/3PFOmt727>

Ilustración 6 Diagrama del proceso de entrenamiento de una GAN
<https://bit.ly/3LRmSzv29>

Ilustración 7 Modelo de la arquitectura del Transformador <https://bit.ly/3LR0B4N30>

Ilustración 8 Arquitectura estándar de Codificador-Decodificador
<https://bit.ly/46i98WN32>

Ilustración 9 Modelo de la arquitectura de GNMT <https://bit.ly/45mNX4D35>

Ilustración 10 Ecuación de la pérdida de entropía cruzada explicada en el tutorial
<https://bit.ly/3ZQDayk37>

Ilustración 11 Información acerca del archivo deu.txt54

Ilustración 12 Primeras 16 líneas del archivo deu.txt55

Ilustración 13 Arquitectura del modelo de MT desarrollado56

Ilustración 14 Librerías usadas para la generación del modelo57

Ilustración 15 Fragmento del código de la clase Lang58

Ilustración 16 Funciones para convertir las oraciones a vectores One-Hot por medio de tensores59

Ilustración 17 Separación de los vectores en baches para el entrenamiento y prueba60

Ilustración 18 Fragmento de código de la clase del Codificador61

Ilustración 19 Fragmento de código de la clase de Decodificador61

Ilustración 20 Fragmento del código de entrenamiento por baches62

Ilustración 21 Fragmento de código de la función `test_batches` para la evaluación de los datos de prueba⁶³

Ilustración 22 Función para probar el modelo con una o varias oraciones fuera del dataset de entrenamiento y prueba⁶⁴

Ilustración 23 Función de evaluación randómica para observar la pérdida y BLEU a través de las distintas épocas con oraciones de los datos de prueba⁶⁴

Ilustración 24 Fragmento del código de la función máster que coordina al resto de funciones⁶⁵

Ilustración 25 Gráfico de Pérdida vs Tiempo de ejecución del modelo (en horas)⁶⁷

Ilustración 26 Ejemplo del output del modelo con las métricas de pérdida en entrenamiento y prueba y la calificación BLEU de algunas oraciones de prueba⁶⁷

Ilustración 27 Ejemplos de traducciones de oraciones de prueba con su calificación BLEU⁶⁸

Ilustración 28 Ejemplo de traducción con el modelo entrenado⁶⁹

Ilustración 29 Primera pregunta del apartado de evaluación de la traducción presentada, esta fue hecha con el modelo con PyTorch⁷⁶

Ilustración 30 Quinta pregunta del apartado de evaluación de las traducciones, esta fue hecha con Chat GPT⁷⁶

Ilustración 31 Octava pregunta del apartado de evaluación, traducción generada con Chat GPT⁷⁷

Ilustración 32 Sexta pregunta del apartado de evaluación, traducción hecha por Google Translate⁷⁷

Ilustración 33 Pregunta 5 del tercer apartado⁷⁸

Ilustración 34 Pregunta 7 del tercer apartado79

Ilustración 35 Tercera pregunta del tercer apartado80

ÍNDICE DE TABLAS

Tabla 1 Comparativa entre las aplicaciones de MT44

Tabla 2 Traducciones de frases propuestas con Google Translate (GNMT)47

Tabla 3 Traducciones de frases propuestas con Chat GPT (GPT-3.5)49

Tabla 4 Traducciones de frases propuestas con el modelo credo con PyTorch (NMT)51

Tabla 5 Comparativa detallada de la calidad de las traducciones generadas por las diferentes aplicaciones traductoras calificada por experto #170

Tabla 6 Comparativa detallada de la calidad de las traducciones generadas por las diferentes aplicaciones traductoras calificada por experto #272

Tabla 7 Promedio de las calificaciones dadas en las tablas 5 y 674

CAPÍTULO I: INTRODUCCIÓN

1. MARCO DE REFERENCIA

1.1. TEMA

Estudio comparativo del traductor de idiomas Google Translate y ChatGPT vs la utilización de algoritmos de Machine Translation (MT) en Python.

1.2. JUSTIFICACIÓN

Actualmente, con el incremento de herramientas de Inteligencia Artificial para la traducción (Traducción de Máquina o Machine Translation en inglés) y la popularidad de su uso, es importante conocer los límites y la viabilidad de dichas herramientas tanto para el uso cotidiano como profesionalmente. Existen ya herramientas que toman ventaja del Deep Learning, como Linguee que utiliza DeepL o ChatGPT con GPT3. Google igualmente invierte continuamente en investigaciones para mejorar sus algoritmos de Traducción de Máquina (MT).

En este escenario, es importante como comunidad universitaria la exploración de las capacidades que tienen estas herramientas actuales, evaluando los retos e implicaciones que tienen las mejoras de IA para la TM. Conociendo esto, se puede concientizar, recomendar y tener una idea clara de los usos que se deberían dar a estas herramientas. Para que las mismas sean utilizadas en beneficio de la comunidad, más no entorpecer las actividades de estudiantes o profesionales.

1.3. PLANTEAMIENTO DEL PROBLEMA

La traducción automática y la inteligencia artificial han experimentado avances significativos en los últimos años, lo que ha llevado a un aumento en la disponibilidad y el uso de traductores en línea y sistemas de Traducción Neuronal Automática (NMT, por sus siglas en inglés: *Neural Machine Translation*). Sin embargo, no se cuenta con una comparación de los traductores disponibles en línea con una red para NMT en Python, o de alguna evaluación que determine la precisión de sus capacidades de traducción del alemán al inglés. Esto aún a pesar de que la calidad de la traducción es un factor crítico en la comunicación interlingüística.

Tampoco se han explorado las diferencias en las metodologías utilizadas por las herramientas de IA para el Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés: *Natural Language Processing*) en línea y no se conoce cuales enfoques pueden producir traducciones más fieles al texto original y cómo se pueden mejorar.

Además, no se conocen las ventajas y desventajas de utilizar IA dedicada a la traducción tanto para usuarios expertos como para usuarios regulares, y tampoco se ha identificado sobre el papel de la tecnología en la industria de la traducción y su impacto en diferentes tipos de usuarios.

De esta discusión se puede identificar los siguientes problemas, el principal siendo, que no existe una comparación entre estas herramientas más allá de listas de top 10 traductores en línea que no hacen una investigación exhaustiva para compararlas.

Muchos usuarios usan ciegamente Google Translate sin considerar si de hecho es la mejor herramienta para dicho caso.

Otros problemas secundarios siendo:

- Ausencia de comparación entre las metodologías que utilizan IA para PLN en línea para evaluar cuál produce una traducción más fiel al texto original.
- Carencia en el estado del arte acerca de la efectividad del lenguaje Python para generar RNN para realizar PLN.
- Carencia de evaluación de las ventajas y desventajas de utilizar IA dedicada a la traducción como una herramienta para un usuario experto y uno casual.
- Inexistencia de evaluación del impacto que tienen estos tipos de software sobre la profesión de traducción y sus implicaciones.

1.4. OBJETIVOS

1.4.1. GENERAL:

Comparar los traductores disponibles en línea con una red para Neural Machine Translation en Python, determinando la precisión de sus capacidades de traducción del inglés al alemán.

1.4.2. ESPECÍFICOS:

- 1.4.2.1.** Comparar las metodologías que utilizan AI para NLP en línea, evaluando cuál produce una traducción más fiel al texto original.
- 1.4.2.2.** Crear una RNN (Recurrent Neural Network) para realizar NMT con PyTorch en un cuaderno de Jupyter.

1.4.2.3. Evaluar las ventajas y desventajas de utilizar IA dedicada a la traducción como una herramienta para un usuario experto y uno casual.

1.4.2.4. Evaluar el impacto que tienen estos tipos de software sobre la profesión de traducción y sus implicaciones.

1.5. ALCANCE

El presente estudio se limitará a analizar la traducción del inglés al alemán. Se creará unas RNN (o Redes Neuronales Recurrentes en español) para realizar MT, evaluando su desempeño y luego ingresando algunos fragmentos de textos con traducciones oficiales. El estudio se realizará con las siguientes herramientas:

- Google Translate
- ChatGPT
- RNN desarrollada con PyTorch para relazar NMT

Principalmente, se busca encontrar si el fragmento es una fiel traducción del original y se lo evaluará por mi persona y un traductor, comparando la traducción oficial del texto con lo que generen estas herramientas. Se utilizará las siguientes páginas para extraer los fragmentos:

Artículo sobre energía sostenible del Ministerio de Asuntos Exteriores de Alemania:

<https://www.auswaertiges-amt.de/en/aussenpolitik/-/2621982>

Unos fragmentos del poema *Note Left at a U.S. Camptown Brothel for My Missing*

Imo: <https://www.lyrikline.org/de/uebersetzerinnen/details/2760/irina-bondas>

Fragmentos del cuento para niños *Jack and the Beanstalk* (Jack y las habichuelas mágicas): <https://www.thefablecottage.com/fables/jack-und-die-bohnenranke>

Específicamente, para la creación de las RNN se tomará como ejemplo un tutorial de PyTorch (ver Anexos) para hacer NMT. En el cual la red será entrenada para poder traducir del inglés al alemán.

Se realizará pruebas haciendo *fine tuning* a diversos parámetros de la red y se entrenará con el dataset deu.txt (ver Anexos).

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2. Marco Teórico

2.1. Generalidades

2.1.1. Traducción

La traducción, según la RAE, se define como: la acción y efecto de traducir, la obra del traductor, o la interpretación que se da a un texto. Proveniente del latín *traductio*, que significa trasladarse, ir de un punto a otro. En cuanto a la profesión como tal, se lo define como: “(...) el proceso de traducción realizado por traductores profesionales formados y entrenados que se encargan de transmitir mensajes de un idioma a otro con precisión, calidad y garantía de satisfacción del cliente. Los traductores profesionales son expertos en lenguas que comprenden el idioma original con todos sus matices y los trasladan al idioma final.” (*Lionbridge, 2020*)

Existen varios tipos de traducción, entre ellos:

Directa: del idioma extranjero al del traductor.

Inversa: del idioma del traductor a un idioma extranjero.

Libre o literaria: “(...) siguiendo el sentido del texto, se aparta del original en la elección de la expresión.”

Simultánea: lo que se hace en la interpretación, hecho oralmente al mismo tiempo que se pronuncie un discurso o conferencia.

(*RAE, 2023*)

Para lograr automatizar un proceso, es necesario conocer de que se trata. Como tal, la traducción es una profesión milenaria que ha existido, documentadamente, desde los Mesopotámicos; pero recientemente, gracias a las Inteligencias Artificiales (IA o AI por sus siglas en inglés) se ha logrado hasta cierto punto automatizar este proceso. El pasar un texto de un idioma a otro no es ninguna tarea simple y, según la materia, se tiene diferentes grados de libertad en cuanto a la interpretación que se pueda dar al texto original. Si se compara, por ejemplo, una traducción técnica con una literaria. El reto principal yace, en todo caso, en transmitir el mensaje expresado en el idioma original de una manera equivalente en el idioma objetivo, perdiendo lo menor del significado como posible. Esto significa, principalmente, que se debe lograr que la AI entienda, o realice procesos equivalentes que simulan entender la semántica de ambos idiomas.

2.1.2. Semántica

La semántica es: “Una de las principales ramas de la Lingüística, dedicada al estudio del significado del lenguaje. (...) La semántica lógica o pura es el estudio del significado de las expresiones siguiendo sistemas lógicos de análisis -o cálculos- y es por tanto más afín a la lógica formal o a las matemáticas que a la lingüística.” (*Crystal*, 2000)

Cabe recalcar que, como se explicará al explorar el concepto de MT, el algoritmo como tal no puede “comprender” la semántica ni el contexto de las oraciones que se utilicen para el entrenamiento, pruebas o inputs externos. A pesar de esto, y justo por

ello, es importante conocer que no pueden hacer los algoritmos de MT y cómo para aclarar el proceso que siguen cada uno de estos para llevar a cabo sus tareas, lo que se explicará más adelante.

2.1.3. Sintaxis

La sintaxis es el “(...) estudio de las interrelaciones entre elementos de la estructura oracional y de las reglas que rigen la ordenación de las oraciones en secuencias.” (Crystal, 2000) En cuanto a la sintaxis, ella es clave para el funcionamiento de los algoritmos de MT. Al evaluar las expresiones y palabras más probables que vengan a continuación, seguirá una cierta lógica que es presente en las oraciones (sintaxis) que fueron empleadas para el entrenamiento, pero más como una serie de características que pueden o no ser tomadas en cuenta por los mecanismos de atención de un algoritmo de MT. Principalmente, la AI encuentra patrones entre palabras y oraciones durante su entrenamiento, por lo que se puede decir que tiene un sentido rudimentario de sintaxis.

2.1.4. Pragmática

“Término tradicional para etiquetar una de las tres divisiones principales de la semiótica (junto a la semántica y sintaxis). En la Lingüística moderna, se ha venido aplicando al estudio del lenguaje desde un punto de vista de los usuarios, especialmente de sus elecciones, de las restricciones que encuentran en el uso del lenguaje en las interacciones sociales y del efecto que el uso del lenguaje tiene en los otros participantes de un acto comunicativo.” (Crystal, 2000)

En corto, este es el *contexto* anteriormente referido, por lo que los algoritmos de MT no tienen noción de este concepto. Pero otros algoritmos en el campo de NLP si lo tienen, y es clave para su funcionamiento. Chat GPT, por ejemplo, logra un cierto nivel de incorporación de pragmática, pero este es limitado principalmente a patrones en los textos con los cuales fue entrenado y las asociaciones estadísticas entre palabras y frases.

2.1.5. Procesamiento de Lenguaje Natural

Conocido también como NLP (Natural Language Processing), es un conjunto de técnicas computacionales para el análisis automático y la representación de lenguajes humanos, motivado por la teoría. Sin embargo, el análisis automático de texto, al nivel de los seres humanos, requiere una comprensión mucho más profunda del lenguaje natural por parte de los computadores; lo cual no se ha logrado en la actualidad. Hay muchos ejemplos de NLP, como: la recuperación de información en línea, la agregación y la pregunta-respuesta, basados principalmente en algoritmos que dependen de la representación textual de las páginas web (como de NLP). Estos algoritmos son muy buenos para recuperar textos, dividirlos en partes, verificar la ortografía y realizar análisis a nivel de palabras, pero no son exitosos para el análisis a nivel de oraciones y párrafos. (*Chowdhary, 2020*)

NLP requiere de capacidades simbólicas de alto nivel, lo cual incluye lo siguiente:

- Acceso y adquisición de características léxicas, semánticas y episódicas.

- Creación y propagación de ligaduras dinámicas.
- Manipulación de estructuras constituyentes recursivas.
- Coordinación de varios modelos de procesamiento y aprendizaje.
- Identificación de estructuras básicas de lenguaje (ej...: objetos y acciones)
- Representación de conceptos abstractos.

(2020)

NLP es el campo de Inteligencia Artificial con el cual trata esta investigación, y se trabajó particularmente con algoritmos de MT, una herramienta clave de NLP. Este es un campo relativamente moderno y que ha visto un auge recientemente con las estructuras basadas en atención como LSTM y GRU que permiten obtener traducciones cada vez más acertadas.

2.1.6. Traducción de Máquina

La Traducción de Máquina (Machine Translation) es el proceso de utilizar la inteligencia artificial para traducir automáticamente texto de un idioma a otro sin intervención humana. MT moderna va más allá de una simple traducción palabra por palabra para comunicar el significado completo del texto original en el idioma de destino. Analiza todos los elementos del texto y reconoce cómo las palabras se influyen mutuamente. (AWS, 2023)

Sus principales beneficios yacen en:

- Velocidad y volumen: puede traducir una gran cantidad de texto casi instantáneamente, o hacerlo en tiempo real, facilitando el trabajo de interpretación.
- Amplia gama de lenguajes: dispone de cientos de idiomas a los cuales se puede traducir directa e inversamente.
- Costo-efectividad: incrementa la productividad y en conjunto con la velocidad y volumen, reduce el tiempo al mercado del producto (la traducción).

(2023)

Según la Asociación Americana de Traductores (ATA), MT es el uso de software automatizado que traduce texto sin el involucramiento humano. Y Adaptive MT (MT Adaptivo) es una tecnología que aprende y ajusta en tiempo real con retroalimentación humana. Aclara también que MT se basa en probabilidad, no en el significado, por lo que no entiende el significado o contexto de lo que traduce, sólo predice la traducción más probable.

2.1.7. Large Language Models (LLM)

Modelos de Lenguaje Masivos (o LLM por sus siglas en inglés), según Nvidia (2023), son algoritmos de Deep Learning que pueden reconocer, resumir, traducir, predecir y generar contenido usando datasets grandes. Representan principalmente una clase de arquitecturas de Deep Learning llamadas redes de transformadores (transformer networks). Un modelo de transformadores es una red neuronal que aprende contexto y significado al rastrear relaciones en datos secuenciales, como las palabras en las anteriores oraciones. (*What Are Large Language Models? | NVIDIA Glossary, 2023*)

2.1.8. Neural Machine Translation (NMT)

Traducción de Máquina por Neuronas (NMT) es una aproximación a MT propuesta recientemente. Diferente a MT estadística, que se basa en la probabilidad de que siga una cierta palabra, NMT trata de construir una sola red neuronal que puede ser mejorada conjuntamente para maximizar el rendimiento de la traducción. Los modelos propuestos actualmente para NMT pertenecen a la familia de codificadores-decodificadores y codifican una oración fuente a un vector de largo fijo del cual un decodificador genera una traducción. (*Bahdanau, 2014*)

2.2. Arquitecturas de Redes Neuronales

2.2.1. Redes Neuronales Estándar

Modelos de Redes Neuronales (NN, por sus siglas en inglés: *Neural Networks*) son algoritmos para tareas cognitivas, como aprendizaje y optimización, el cual es basado grosso modo en estudios hacia la naturaleza del cerebro. En términos matemáticos, una NN es definida como un grafo dirigido con las siguientes propiedades:

- Una variable de estado n_i asociada con cada nodo i .
- Un peso w_{ik} asociado con cada vínculo entre dos nodos (i y k respectivamente).
- Un sesgo v_i Real asociado con cada nodo i .
- Una función de transferencia $f_i[n_k, w_{ik}, \vartheta_i, (k \neq i)]$

(*Müller, 1995*)

Esta función de transferencia está definida de tal manera que, para cada nodo i , se determina el estado de un nodo en función de su sesgo, de los pesos provenientes de

sus vínculos y de los estados de los nodos conectados a él por estos vínculos. Específicamente, cada nodo se llama neurona y los vínculos son sinapsis, con el sesgo conocido como umbral de activación. (1995) Este umbral es activado gracias a dos variables adicionales, la función de activación y la tasa de aprendizaje, las cuales ajustan los pesos luego de pasar cada capa.

En cuanto a su arquitectura, la red más simple consta de una capa de entrada, las capas ocultas y la capa de salida, donde se puede ajustar la cantidad de neuronas existentes en cada una. En este algoritmo, ingresan las entradas como valores de entrenamiento y al recibir la salida, esta se compara con los valores de prueba que se dividieron de los datos originales para evaluar la precisión del modelo. Típicamente se divide los datos por medio de la Ley de Pareto (80% de entrenamiento, 20% de prueba), pero igual como el número de neuronas por capa, número de capas, tasa de aprendizaje, y otros, estos valores se ajustan para determinar el modelo que mejor se ajuste al objetivo del entrenamiento. Generalmente tratando evitar que el modelo se sobreajuste a los datos.

Esta es la base teórica para todas las redes que se describirán a continuación. A partir de este concepto se han introducido varias mejoras y adaptaciones a este modelo para acoplar una serie variada de problemas. Esta es igualmente la base para NMT el mismo modelo que utiliza Google para Google Translate.

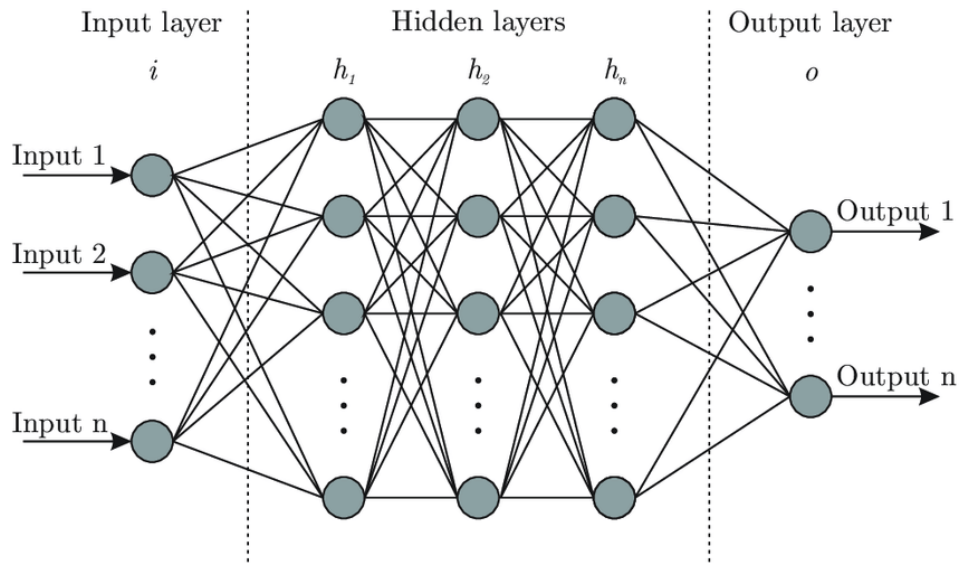


Ilustración 1 Arquitectura de una Red Neuronal estándar <https://images.app.goo.gl/tZjGkXgk9ja9bJRC7>

2.2.2. Redes Neuronales Recurrentes (RNN)

Según Grossberg (2013), una RNN (Red Neuronal Recurrente, o Recurrent Neural Network en inglés) es cualquier red cuyas neuronas mandan señales de feedback unas a otras. Este concepto incluye una gran gama de posibilidades. IBM (n.d.) lo define como un tipo de red neuronal artificial que utiliza datos secuenciales o datos de series de tiempo, por lo cual son utilizados normalmente para tratar con problemas de tipo ordinal o temporal; tal como traducción, NLP, reconocimiento de voz y captionamiento de imágenes. Algunos ejemplos populares siendo: Siri, búsqueda por voz y Google Translate.

Tal como redes *feed-forward* y redes neuronales convolucionales (CNN), utilizan datos de entrenamiento para aprender, pero su factor distintivo es su capacidad de “memoria”, por lo que toman información de entradas previas para influenciar las entradas y salidas actuales. Debido a esto, la salida de las RNN depende de los

elementos previos en la secuencia. Vale destacar también que los parámetros de la red se comparten a través de capas, y no son variables como en redes neuronales tradicionales. Pero los pesos si son ajustados a través del proceso de *backpropagation* y descenso por gradiente para facilitar el refuerzo del aprendizaje. Este tipo de redes utiliza para *backpropagation* un algoritmo específico para ello, conocido como *backpropagation through time* para determinar las gradientes, el cual es específico para datos secuenciales. (*What Are Recurrent Neural Networks?* | IBM, n.d.)

Desafortunadamente, este tipo de redes sufren de un famoso problema, el de la desaparición del gradiente (*vanishing gradient problem*). Esto lo hace menos adaptivo en capturar dependencias a largo plazo en secuencias, disminuyendo su rendimiento.

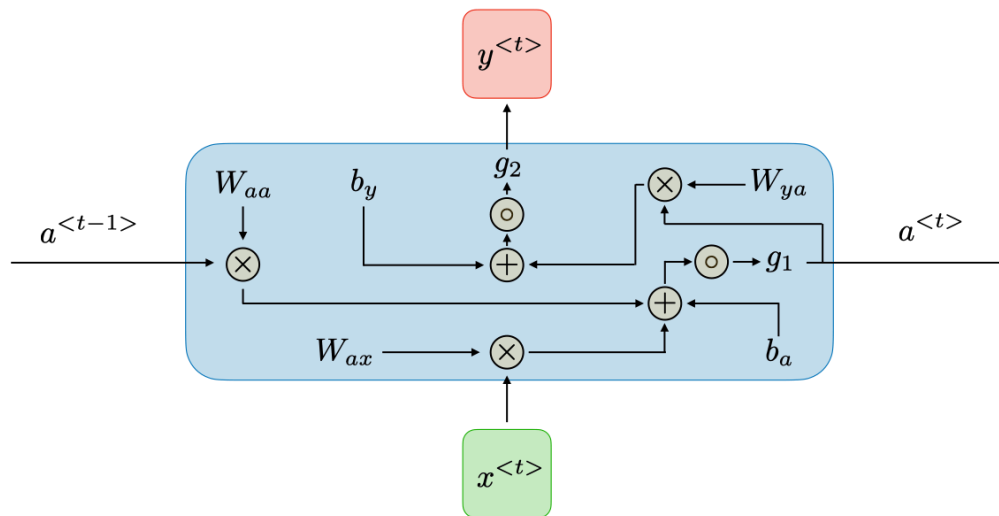


Ilustración 2 Arquitectura de una RNN estándar <https://images.app.goo.gl/U1G1vqMxh6cv25F1A>

2.2.3. Backpropagation Through Time (BPTT)

Los principios de BPTT son los mismos que de *backpropagation*, donde el modelo se entrena por medio de calcular errores de su capa de salida hacia su capa de entrada.

Estos cálculos permiten ajustar los parámetros del modelo de manera apropiada. La principal diferencia yace en que BPTT hace una sumatoria de los errores a cada paso de tiempo (time step), en contraste con redes *feed-forward*, que no requieren de esta sumatoria por lo que no comparten parámetros a través de cada capa. (*What Are Recurrent Neural Networks?* | IBM, n.d.)

2.2.4. Descenso por Gradiente

“El método de descenso por gradiente, *gradient descent* de ahora en adelante, es uno de los algoritmos de optimización más populares en aprendizaje automático, particularmente por su uso extensivo en el campo de las redes neuronales. *Gradient descent* es un método general de minimización para cualquier función f . A la versión original se le considera lenta pero versátil, sobre todo para casos de que las funciones multidimensionales.” (Ruiz, 2016)

Este algoritmo está igualmente presente en Redes Neuronales regulares, como manera de optimizar su ejecución, por lo que el objetivo de cualquier red es minimizar el error de entrenamiento y prueba. En la red desarrollada en este trabajo se evaluará una métrica relacionada al error, que es la pérdida.

El problema de la desaparición del gradiente es solventado parcialmente al aplicar *time steps*, pero es más un remedio y no una solución, por lo que se sigue sin mantener las dependencias a largo plazo.

2.2.5. Arquitecturas de RNN

2.2.5.1. LSTM

A diferencia de una red neuronal recurrente tradicional, esta incorpora el concepto de celdas de memoria y unidades de compuerta. LSTM está compuesta de un sistema de compuertas de entrada, olvidar y salida para controlar el flujo de información a través de la red y puede capturar dependencias en secuencias a largo plazo, solventando el problema de desaparición del gradiente.

Para construir una arquitectura que permita el flujo de error constante a través de unidades especiales (conectadas a sí mismas) sin las desventajas del método tradicional, se extiende el error a través de un CEC (constant error carousel) por medio de la introducción de características adicionales. Una unidad de entrada (compuerta de entrada) es introducida para proteger el contenido almacenado en la memoria de las neuronas de perturbaciones causadas por entradas irrelevantes. De misma manera se introdujo una compuerta multiplicativa de salida que protege a otras unidades de perturbaciones causadas por contenido irrelevante almacenado en memoria de la neurona. (*Hochreiter & Schmidhuber, 1997*)

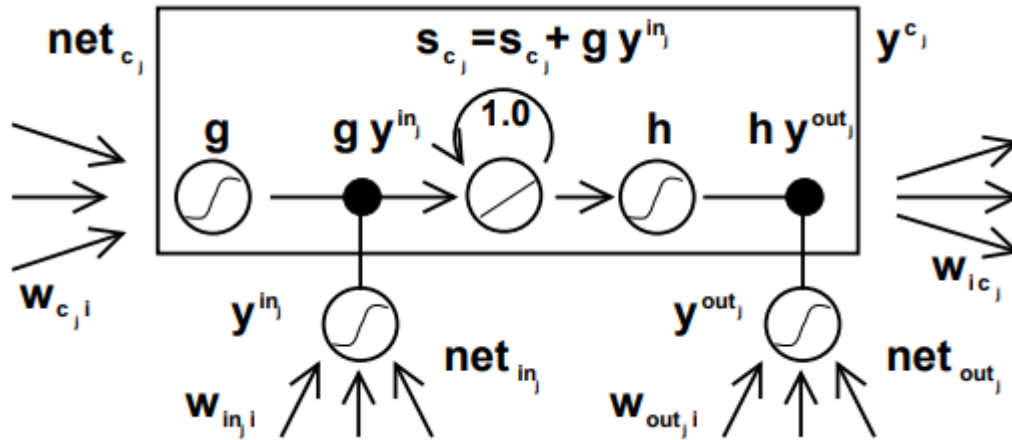


Ilustración 3 Arquitectura de una célula de memoria para LSTM <http://bit.ly/3LNmGJ>

Esta es la arquitectura con la cual se desarrolló el algoritmo de MT con Python. En la sección de metodología se explica a más detalle las razones por las cuales se prefirió LSTM sobre GRU. A pesar de ser propuesta en 1997, LSTM tomó relevancia alrededor del año 2010 y tuvo un auge en el campo de MT; fue igualmente fundamental y ampliamente utilizado hasta que vino GRU, pero aún hay casos particulares donde el rendimiento de LSTM sobrepasa al de GRU, por ejemplo: tareas que requieran de RNN apiladas, mayor control, mayor complejidad de modelado, entre otros.

2.2.5.2. GRU

GRU es una variante similar a LSTM, pero se destaca por tener una arquitectura de célula (o neurona) más simple, y aborda el problema de desaparición del gradiente. "En lugar de utilizar una información de regulación de "estado de celda", utiliza estados ocultos y, en lugar de tres puertas, tiene solo dos: una puerta de reinicio y una puerta

de actualización. Al igual que las puertas dentro de los LSTM, las puertas de reinicio y actualización controlan cuánta y qué información retener.” (*What Are Recurrent Neural Networks?* | IBM, n.d.)

Este tipo de RNN reduce las señales de compuerta a dos en comparación con LSTM, llamadas las compuertas de actualización z_t y la de reinicio r_t . El modelo está presentado en la forma:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

$$\tilde{h}_t = g(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h)$$

(*Gate-variants of Gated Recurrent Unit (GRU) Neural Networks*, 2017)

GRU posee menos señales externas en las compuertas. En otras palabras, GRU es más simple que LSTM y rinde, en la mayoría de los casos, mejor o de manera equivalente a LSTM. (2017) Principalmente es mejor para generalización, aplicaciones en tiempo real, limitaciones de memoria y capacidad de almacenamiento, etc.

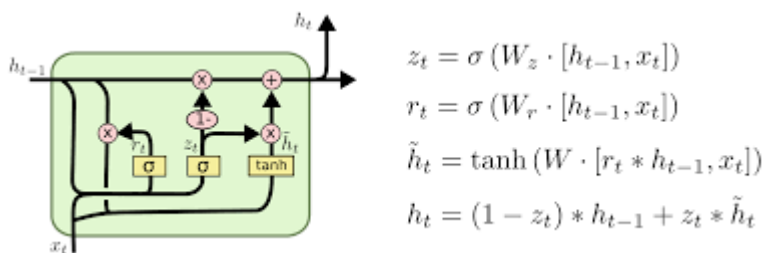


Ilustración 4 Célula de GRU con las fórmulas de las compuertas y su forma <https://images.app.goo.gl/hcv4HHYhw8FCdjW6>

2.2.5.3. Bidireccionales

Es una variante de la arquitectura de RNN tradicional, mientras que RNN unidireccionales “solo pueden extraerse de datos de entrada anteriores para hacer predicciones sobre el estado actual, los RNN bidireccionales extraen datos futuros para mejorar su precisión.” (IBM, *n.d.*)

Para sobrepasar las limitaciones de una RNN tradicional, se propuso una BRNN que puede ser entrenada utilizando toda la información disponible en la entrada en el pasado y futuro de un paso de tiempo específico. Principalmente, la idea yace en dividir las neuronas de estado de una RNN regular en una parte responsable por la dirección positiva en el tiempo (estados futuros) y otra parte para la dirección negativa (estados pasados). Donde las salidas de estados futuros no están conectadas con entradas de estados pasados, y viceversa. (*Bidirectional Recurrent Neural Networks*, 1997)

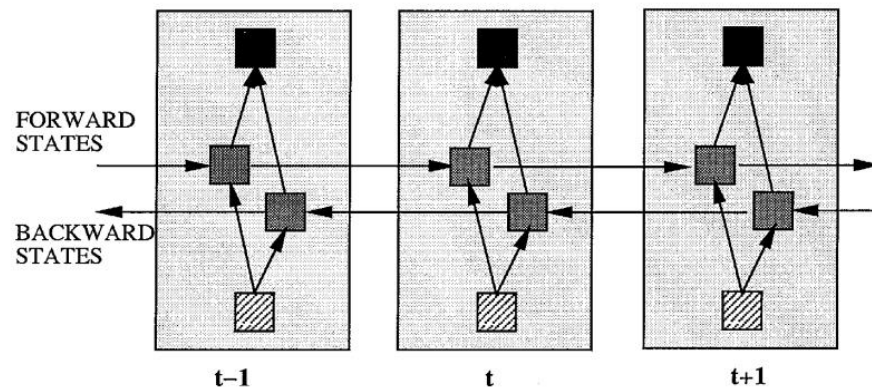


Ilustración 5 Estructura general de una BRNN a través de tres pasos de tiempo <http://bit.ly/3PF0mt7>

2.2.5.4. Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (o CNN por sus siglas en inglés: *Convolutional Neural Networks*), son análogas a redes neuronales tradicionales en el sentido de que están compuestas por neuronas que se optimizan a través del aprendizaje automático. Cada neurona aun recibe una entrada y realizará una operación (como una cantidad escalar por una función no lineal). La principal diferencia notable es que las CNN son utilizadas principalmente en el campo de reconocimiento de patrones con imágenes. Lo que permite codificar características específicas a imágenes en la arquitectura, haciendo que esta se ajuste más a tareas relacionadas y enfocadas en imágenes, mientras se reducen la cantidad de parámetros requeridos para iniciar el modelo. Redes tradicionales tienden a tener dificultades con la complejidad computacional requerida para computar datos relacionados a imágenes. (O'Shea, 2015)

En este trabajo, dado que tiene como enfoque MT, no es apropiado utilizar una CNN para abordar este problema, dada a su alta especialización en tratar con datos y metadata relacionada con imágenes. Las CNN son aplicables a casos que requieran de visión de máquina.

2.2.6. Redes Adversarias Generativas

“Las redes adversarias generativas (GANs) son un método basado en el entrenamiento de dos redes neuronales, una denominada generadora y otra discriminadora, compitiendo entre sí para generar nuevas instancias que se asemejen

a las de la distribución de probabilidad de los datos de entrenamiento. Las GANs tienen una amplia gama de aplicaciones en campos como la visión por computadora, la segmentación semántica, la síntesis de series temporales, la edición de imagen, el procesamiento del lenguaje natural y la generación de imagen a partir de texto, entre otros.” (La Torre Jordi, 2023)

En sí, su arquitectura se conforma de una NN discriminadora (D) y otra generadora (G). Mientras “(...) la red G se encarga de generar nuevas instancias del mismo dominio que el del conjunto de datos de origen. La red D se encarga de discriminar si los datos de entrada son reales (...)” o son datos ficticios generados por la red G. Ambas redes compiten, con la red G buscando no ser detectada y la red D buscando mejorar sus capacidades de discernir; compitiendo en un juego de suma 0 hasta que lleguen a un denominado equilibrio de Nash. (2023)

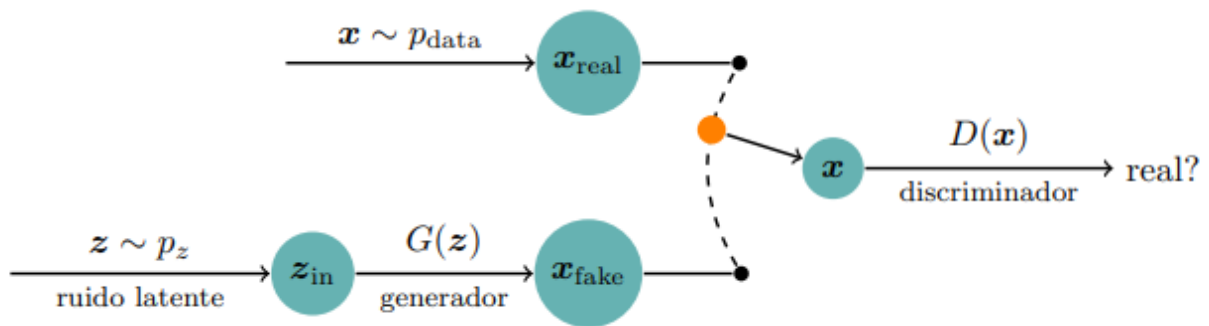


Ilustración 6 Diagrama del proceso de entrenamiento de una GAN <https://bit.ly/3LRmSzv>

2.2.7. Redes Neuronales con Transformadores

Estas redes se basan en la propuesta de Transformadores, que se fundan netamente sobre mecanismos de atención, dispensando resultados con recurrencia y

convoluciones. Igualmente funcionan con una estructura de codificador-decodificador, en donde el codificador mapea como entrada una secuencia de representaciones (x_1, \dots, x_n) a una secuencia de representaciones continuas $\mathbf{z} = (z_1, \dots, z_n)$. Dado \mathbf{z} , el decodificador genera una secuencia de salida (y_1, \dots, y_m) de símbolos de elemento en elemento. En cada paso el modelo es autorregresivo, consumiendo los símbolos generados previamente como entradas adicionales mientras genera la siguiente secuencia. Como tal, el Transformador sigue esta arquitectura utilizando capas de autoatención (*self-attention*) y punto a punto (*point-wise*) apiladas para tanto el codificador como el decodificador. (Vaswani, 2017)

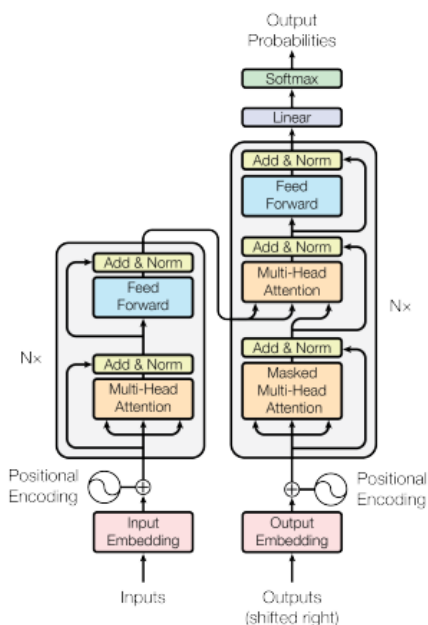


Ilustración 7 Modelo de la arquitectura del Transformador
<https://bit.ly/3LR0B4N>

Un transformador está compuesto, como se puede apreciar en la ilustración 7, por múltiples bloques transformadores, conocidos también como capas. Por ejemplo, un transformador tiene capas de *self-attention* (auto-atención), *feed-forward layers* (capas

de avance) y capas de normalización que trabajan en conjunto para descifrar una entrada para predecir flujos de salidas como inferencias. (*What Are Large Language Models? | NVIDIA Glossary, 2023*)

RNN con transformadores son en sí, RNN tradicionales con una capa de transformadores añadida para generar un output. Redes con transformadores en sí causaron un gran impacto, llegando a obtener un récord en la calificación BLEU en el dataset de WMT 2014 de traducción de inglés al alemán. Igualmente ha llegado a impactar los campos de Visión por Computadora, NLP como mencionado antes, análisis de series de tiempo, sistemas de recomendación, reconocimiento de voz, etc.

Principalmente, la razón por la cual no se utilizó Transformadores para realizar este trabajo es por el hecho de que requieren de un gran volumen de datos para entrenarse de manera apropiada y tienen una alta complejidad computacional. Como tal, no se dispone de dichos recursos para realizar este estudio con Transformadores, por lo que se optó por utilizar Chat GPT (por lo que es un LLM), que funciona actualmente con GPT3.5 de manera gratuita.

2.2.7.1. Mecanismo de atención

El mecanismo de atención, tal como otros métodos basados en redes neuronales, intenta imitar al cerebro humano y a su visión para procesar datos. La visión humana no procesa toda la imagen que observa, sino se enfoca en ciertas partes. Con esto, las partes a las cuales se enfoca nuestra visión son percibidas con más “alta resolución” mientras que lo de alrededor se percibe en “baja resolución”. En otras palabras, da un

peso más alto a las partes relevantes mientras minimiza las que no considera relevantes; permitiendo que se procesen y enfoquen en las partes más importantes de los datos en vez de procesarlo todo. Fue propuesto inicialmente en 2014 para aplicaciones de NLP. (Ghaffarian et al., 2021)

2.2.7.2. Arquitectura de Codificador-Decodificador

Generalmente, en problemas de secuencia a secuencia como en MT, las entradas y salidas son de largos variables que no son alineados, por lo que generalmente a este tipo de problemas se los solventa por medio de esta arquitectura. Consiste en dos componentes principales, un codificador que toma una secuencia de largo variable como entrada y un decodificador que actúa como un modelo condicional de lenguaje. Toma la entrada codificada y el contexto de la secuencia objetivo y predice el token siguiente en la secuencia objetivo.

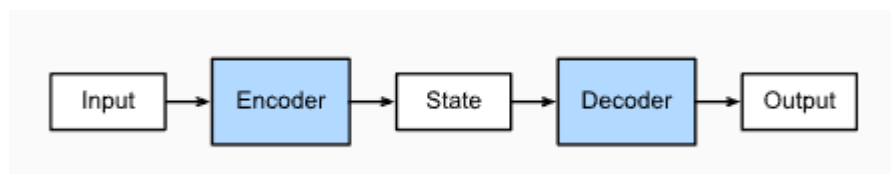


Ilustración 8 Arquitectura estándar de Codificador-Decodificador <https://bit.ly/46i98WN>

Esta es la arquitectura utilizada para crear el algoritmo de NMT con PyTorch, en conjunto con LSTM. La mayoría de los algoritmos de NMT posee esta arquitectura y al ser especializada en tratar con datos de entrada y salida de largos variables, es ideal para el algoritmo desarrollado en este trabajo.

2.3. GPT-3

GPT-3 es un modelo de lenguaje autorregresivo con 175 billones de parámetros, llamado Transformador Generativo Pre-entrenado o GPT-3 por sus siglas en inglés (*Generative Pre-Trained Transformer*). Es un LLM diseñado para ser agnóstico a la tarea. (Brown, 2020) Al utilizar un modelo de aprendizaje no supervisado, utilizó inputs generados por usuarios y BD y capacidad de computación de Azure para su entrenamiento y desarrollo. En cuanto a arquitectura, no es muy diferente a la de GPT-2, la diferencia principal siendo que utiliza patrones de atención densos alternativos y vendados localmente en las capas del Transformador (2020). Probaron con modelos de 8 diferentes cantidades de parámetros para evaluar su rendimiento a cada variación, el modelo más grande es el que finalmente fue nombrado GPT-3.

Se evaluó igualmente la capacidad de traducción de GPT-3. En contraste con GPT-2, se añadieron los idiomas alemán y rumano. Los investigadores encontraron que logra una calificación BLEU competitiva con estudios anteriores o incluso los sobrepasa, pero netamente en traducción de inglés a francés, alemán o rumano. Al hacer traducción inversa encontraron que el rendimiento bajaba de la media. (2020) Vale recalcar que los datos con los que se entrenaron GPT-3 son mayoritariamente en inglés (93%).

OpenAI ha logrado mejorar esto aún más gracias a las entradas de miles de millones de usuarios, que utilizan la IA con varios idiomas más. En este estudio se utilizará la API integrada de Chat GPT (que funciona actualmente con GPT-3 de manera gratuita) en el navegador Opera GX para realizar la comparativa de traducciones. Es importante

destacar que este es un modelo de lenguaje que no fue entrenado específicamente para MT, pero que llega a ser competitivo con IA especialmente desarrolladas para tratar con traducción, por lo cual se consideró incluirlo en este estudio para tener una perspectiva de cuán bien adaptado está un modelo de lenguaje popular para la tarea de traducción.

2.4. Machine Translation en Google y con Python

2.4.1. Google Translate (GNMT)

Google antes trabajaba con un algoritmo llamado Phase-Based Machine Translation (MT basado en fases) para su aplicación de Google Translate. Pero gracias a los avances en MT, Google lanzó su propia versión de NMT llamado GNMT. NMT es un enfoque de aprendizaje de extremo a extremo para traducción automática. Tiene el potencial de superar muchas debilidades de los sistemas convencionales de traducción basados en frases. Pero este modelo tiene sus desventajas, es lento para entrenar y tiene baja velocidad de inferencia, es inefectiva al tratar con palabras raras y falla a veces para traducir todas las palabras en una oración. (Wu, 2016)

Por lo cual, Google desarrolló su propio modelo, el cual resuelve esos problemas presentados y más. En su implementación, utilizaron RNN con LSTM. Su red posee 8 capas con conexiones residuales entre capas para motivar el flujo del gradiente. Para paralelismo, conectaron el mecanismo de atención desde lo más bajo de la capa de la red del decodificador y utilizan Unidades de Procesamiento Tensorial (TPU por sus

siglas en inglés: *Tensor Processing Units*) especializados de ellos para entrenar el modelo. En cuanto a su arquitectura, consta de tres capas: una red de codificador, una de decodificación y una de atención. Transforma los inputs en una lista de vectores, uno por símbolo de entrada. (2016)

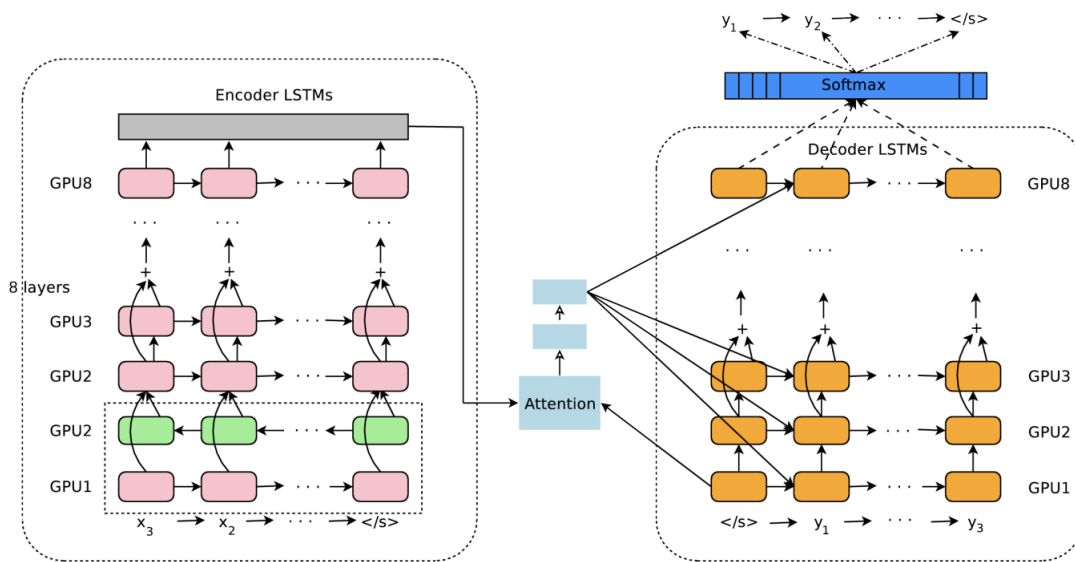


Ilustración 9 Modelo de la arquitectura de GNMT <https://bit.ly/45mNX4D>

En sí, el modelo utilizado en la RNN desarrollada para este trabajo es similar, solo que consta de una sola capa de RNN con LSTM para codificar y decodificar. En este caso, sería una comparación directa con un modelo a gran escala de lo que se desarrolló.

2.4.2. PyTorch torch.nn

Es un módulo en PyTorch, un popular framework de aprendizaje profundo de código abierto, que proporciona clases y funciones para construir y entrenar redes neuronales. Este módulo es una parte fundamental de PyTorch y se utiliza ampliamente para

definir, entrenar y ajustar diferentes tipos de arquitecturas de redes neuronales. Algunas de las redes que incluye son: redes neuronales de avance (también conocidas como perceptrones multicapa), redes neuronales convolucionales (CNN), redes neuronales recurrentes (RNN) y más. Incluso puede trabajar con células de LSTM y GRU mencionadas previamente y ajustar las funciones de activación, capas de padding, pooling, convolución, entre otros. (*torch.nn — PyTorch 2.1 Documentation*, n.d.)

Esta es la librería de torch que se utilizó principalmente para alzar el modelo, y con células de LSTM para imitar la arquitectura de GNMT. Explorar este algoritmo con células GRU podría considerarse para exploraciones futuras, pero por restricciones de tiempo y practicidad, se decidió mantener las células con LSTM para tener una comparación directa con la red de Google Translate.

2.5. Métricas de Evaluación

2.5.1. Cross-Entropy Loss

“Para dos variables aleatorias discretas p y q , la entropía cruzada se define como:

$$H(p, q) = - \sum_x p(x) \log q(x)$$

Esta definición no es simétrica. P se entiende como la distribución “verdadera”, solo parcialmente observada, mientras que Q se entiende como la distribución “no natural” obtenida a partir de un modelo estadístico construido.

En la teoría de la información, la entropía cruzada se puede interpretar como la longitud esperada en bits para codificar mensajes cuando se utiliza Q en lugar de P .”
(*Definición De Cross-Entropy*, n.d.)

En esencia, lo que hace esta función de pérdida es sumar sobre las probabilidades logarítmicas que el modelo predice como la palabra correcta en cada posición en la oración de output. Dado que la función logarítmica negativa tiene un valor de 0 cuando la entrada es 1 e incrementa exponencialmente mientras la entrada se acerca a 0, esto significa que mientras más se acerque la probabilidad que el modelo de la palabra correcta en cada punto a 100%, menor será la pérdida. (*Lanners*, 2022)

Cross-Entropy Loss

$$-\sum_{w=1}^{|S|} \sum_{e=1}^{|V|} y_{w,e} \log(\hat{y}_{w,e})$$

$|S|$ = Length of Sentence

$|V|$ = Length of Vocabulary

$\hat{y}_{w,e}$ = predicted probability of vocab entry e on word w .

$y_{w,e} = 1$ when the vocabulary entry is the correct word

$y_{w,e} = 0$ when the vocabulary entry is not the correct word

Ilustración 10 Ecuación de la pérdida de entropía cruzada explicada en el tutorial <https://bit.ly/3ZQDayk>

2.5.2. Métrica BLEU

La métrica BLEU varía en rango de 0 a 1. Pocas traducciones obtendrán una calificación de 1 a menos que sean idénticas a la traducción de referencia. Por esta

razón, hasta traductores humanos no obtendrán una calificación de 1. La idea tras crear esta métrica fue crear una manera de evaluar el rendimiento de MT, que mientras más se acerque a la traducción de un profesional, mejor será. Esta calificación calcula las precisiones de n -gramos a lo largo con la media geométrica y un factor de penalización por brevedad de la oración. (*Papineni, 2002*) La fórmula de dicha métrica es la siguiente:

$$\log\text{Bleu} = \min\left(1 - \frac{r}{c}, 0\right) + \sum_{n=1}^N w_n \log p_n$$

(2002)

Esta métrica se hizo bastante popular entre investigadores y se hizo estándar implementarla para tareas de MT, por lo cual, además de la pérdida, se computó también la métrica BLEU para la comparativa entre modelos. Vale recalcar que el BLEU se puede igualmente medir de 0 a 100, algo que aparece muy frecuentemente en estudios que tratan de MT.

CAPÍTULO III: METODOLOGÍA

3. Metodología de desarrollo del plan de trabajo de titulación

3.1. Investigación Cualitativa

Este tipo de investigación se realizó principalmente para tratar con el impacto que tienen los modelos de MT y consta de un reporte de los resultados de una entrevista (adjunta en anexos) a los traductores oficiales de alemán de la Embajada de la República Federal de Alemania y a algunos estudiantes de la carrera de Lingüística de la Facultad de Comunicación, Lingüística y Literatura de la PUCE (FCLL), quienes ejercen la traducción por su alto nivel del idioma alemán.

Como tal, en el cuestionario para la entrevista se evalúa el impacto que han tenido software de traducción con AI en el campo de traducción, desde el nivel de estudiante a profesional. Igualmente se preguntó la opinión de los entrevistados para identificar los usos, opiniones y pronósticos que tienen acerca de su campo. Para ello, se les proveyó en la entrevista de ejemplos de frases traducidas por Chat GPT, Google Translate y en el modelo con PyTorch.

3.1.1. Diseño de la investigación

La entrevista consta de 3 apartados, 7 preguntas y de unas tablas comparativas para que el/la entrevistado/a dé una corta evaluación hacia la fidelidad de la traducción presentada con la frase original, calificando de 1 a 5 la semántica, sintáctica y pragmática. El resto son preguntas de naturaleza abierta, del cual se realizó un análisis

de sentimientos y un resumen de las respuestas por categoría y del total. Las categorías consideradas son: Profesional y Estudiante.

En cuanto a la evaluación de los modelos, se realizó una tabla comparativa entre ellos evaluando los siguientes parámetros para cada algoritmo que utilizan las aplicaciones de Chat GPT, Google Translate y el modelo creado con PyTorch:

- Tipo de modelo
- Arquitectura de Célula
- Nº de capas usadas para entrenar
- Nº de parámetros usados para entrenar
- Procesamiento utilizado para entrenar y decodificar
- Tiempo total de Computación (h)
- BLEU

Luego se hizo una calificación preliminar de la semántica y pragmática; en otras palabras, la fidelidad de la traducción realizada, de las frases extraídas de los documentos previamente mencionados en el apartado 1.5; dando una calificación baremo de 1 a 5 (1 siendo nada fiel y 5 fiel al original), done al final se sumarán las calificaciones. Se tomaron 10 frases de las fuentes indicadas, las cuales fueron traducidas por cada modelo y posteriormente evaluadas por una docente lingüista de la FCLL.

Esta tabla consta de los siguientes campos:

- Frase Original
- Frase Traducida
- Frase Esperada
- Calificación

Para estas tablas lo que se calificó es la fidelidad de la traducción. Según la Asociación de Traductores Americanos (ATA) (2022a), la fidelidad es la medida por la cual el trabajo de un traductor puede ser juzgado, refiriéndose a la medida exacta de cómo se ajusta o concuerda con un documento traducido con respecto a su fuente o idioma original.

Los fragmentos de la tabla previamente mencionada fueron los que se incluyeron en la entrevista realizada, sin su respectiva calificación dada por la experta.

Para evaluar a profundidad la calidad de las traducciones hechas con los modelos estudiados se adaptaron parámetros tomados de Munday (2012) para una tabla adicional, en la cual se evaluaron las siguientes métricas (igualmente con una calificación baremo sobre 5, con 5 siendo la mayor calificación):

- Fidelidad: la función ideológica/ideacional direccionada a buscar o representar a la verdad (desde la perspectiva del hablante). (Munday, 2012) En otras palabras, se evaluó si se mantiene el sentido que el autor de la oración quiso darle antes de ser traducido, que tan cercanamente se pudo representar la '*verdad*'.

- Sensibilidad: construye o negocia comunidad o solidaridad y da al texto su 'orientación de valor'. (Munday, 2012) Esta métrica evalúa, en esencia, si la traducción

es sensible al contexto del que proviene o solidario con él, en corto, su cercanía al texto original.

- Contenido: si la traducción se orienta adecuadamente al contenido. (Munday, 2012)
- Directo: cuán directa resultó ser la traducción, sin sacrificar la fidelidad o sensibilidad. (Munday, 2012)
- Tenor: negociación, a través de la función del habla (escritura en este caso) e intercambio. (Munday, 2012) Evaluando cuán acertadamente refleja el contexto social e interpersonal, o la relación entre el escritor y su audiencia.
- Poder: valoración, a través de afecto, juzgamiento, apreciación, graduación y comprometimiento. En otras palabras, esta métrica evalúa la actitud, la cual es la forma más básica de evaluación. (Munday, 2012) Evalúa si se mantuvo en general la emoción atrás del texto original.
- Solidaridad: involucramiento, a través del nombramiento, tecnicidad, abstracción, etc. (Munday, 2012) Esta métrica evalúa si se utilizó apropiadamente palabras técnicas en el contexto específico del texto presentado para traducirse.

Para realizar esta rúbrica de calificación se tomó particularmente el apartado de la teoría de valoración y el concepto de evaluación en la traducción de Munday (2012). Esta rúbrica fue corroborada y utilizada para evaluar una muestra de las traducciones obtenidas de las aplicaciones por un experto en el Centro de Traducciones de la PUCE.

3.1.2. Método de recopilación de datos cualitativos

- Entrevista cualitativa.

3.2. Enfoque Metodológico

El enfoque que se decidió tomar, tras haber hecho una revisión extensa de la literatura, es de un ciclo estándar del ciclo de vida de Ciencia de Datos reducido junto con una evaluación cualitativa que tomó la forma de las encuestas mencionadas previamente.

3.2.1. Metodología de Ciencia de Datos

La metodología reducida consta de los pasos de:

- Entendimiento de los Datos
- Preparación de los Datos
- Modelado
- Evaluación del Modelo

Debido a que el uso de este modelo es limitado a la evaluación de las capacidades que tienen las librerías de Python para la traducción, no se lo implementará, por lo que las fases de entendimiento del negocio, despliegue y retroalimentación no son necesarias.

CAPÍTULO IV:

COMPATIVA DE TRADUCTORES EN LÍNEA CON UNA RED PARA NEURAL MACHINE TRANSLATION EN PYTHON.

4.1. Comparativa de Metodologías de IA para NLP en línea

Principalmente, en esta tabla se compara las metodologías de las diferentes aplicaciones que se consideró para este estudio y algunos parámetros generales para tener una idea acerca de su rendimiento y tamaño. Para poder observar la capacidad que tiene cada modelo y comprender las capacidades que tienen los mismos para traducir.

Es importante tomar en cuenta que los datos de la siguiente tabla son los de los modelos de prueba o utilizados para investigación y desarrollo, más no de los que se encuentran en producción. Los datos de GNMT y GPT-3.5 de producción son secretos guardados por las respectivas empresas que las desarrollaron, por lo que no es información disponible en el dominio público.

Tabla 1 Comparativa entre las aplicaciones de MT

<i>Aplicación de Traducción</i>	<i>Tipo de modelo</i>	<i>Célula</i>	<i>Nº de capas</i>	<i>Nº de parámetros</i>	<i>Procesamiento</i>	<i>Tiempo total de Computación (h)</i>	<i>BLEU</i>
<i>Google Translate</i>	Google Neural Machine Translation (GNMT)	LSTM	8	36M WMT En→Fr ¹ 5M WMT En→De ²	8 GPU TPU	100.37h (WPM-32K en WMT En→De)	32.2

¹ WMT En→Fr: WMT'14, dataset de desarrollo de inglés a francés (Wu, 2016)

<i>Chat GPT</i>	Large Language Model (Modelo de Lenguaje Grande)	GPT-3	96	175B	V100 GPU (no mencionan cantidad)	277.97h	29.7
<i>Modelo personal con PyTorch</i>	Neural Machine Translation (NMT)	LSTM	2	261K	2 GPU	15h	14.35

Tabla comparativa de los tres modelos a evaluar, información obtenida de Wu(2016) y Brown(2020)

OpenAI utilizó los recursos de Microsoft para el entrenamiento de su modelo, y Google los recursos in-situ. Experimentaron igualmente con un tipo especial de GPU desarrollados por ellos llamado TPU (Tensor Processing Unit). De los tres modelos, el más grande es el de Chat GPT, poseyendo la mayor cantidad de capas, datos de entrenamiento y mayor tiempo de ejecución. Es importante destacar que GPT-3 es un modelo de lenguaje y no uno de MT, por lo que no se especializa en la traducción, pero es capaz de ella.

Tanto GPT-3 como GNMT se entrenaron también con datasets de inglés a francés, mientras que el modelo generado con PyTorch sólo se entrenó con un dataset pequeño de inglés a alemán. Interesantemente, tanto en el modelo de Google como en el de OpenAI, sus datos en alemán eran significativamente menores a los que tenían en inglés. GPT-3 se entrenó con un 96% de datos en inglés, mientras que para GNMT el dataset de alemán es un séptimo del tamaño del dataset francés. Esto demuestra que hay una escasez de datos para traducción a alemán, lo cual impacta de gran manera a este tipo de modelos en general, por lo que tanto para modelos de lenguaje como para

² WMT En→De: WMT'14, dataset de desarrollo de inglés a alemán

modelos de NMT se requiere una cantidad significativa de datos para lograr un buen entrenamiento.

Todos los modelos trabajan con mecanismos de atención, pero solo GPT-3 trabaja con transformadores, lo que le da una ligera ventaja sobre LSTM en términos de eficiencia, utilización de recursos, entre otros. Pero, al no ser un modelo dedicado a la traducción, se puede observar que su BLUE es menor al de GNMT. Como el modelo desarrollado personalmente para esta investigación es del mismo tipo que el de Google, si se lo escalase se podría esperar resultados similares a los de GNMT. Lo cual implicaría escalar igualmente la infraestructura dedicada a ello, algo que no es viable en el estudio actual.

Por lo tanto, al entrenar modelos de lenguaje se recalca la necesidad de una gran cantidad de recursos de cómputo y de una cantidad considerable de datos para obtener los mejores resultados. A estas escalas se trata con Big Data y clústeres de cómputo de alto rendimiento, lo cual demuestra la complejidad de modelar un aspecto del lenguaje que frecuentemente tomamos a la ligera, la traducción. En sí, los resultados obtenidos con el modelo personal son una buena prueba del concepto y da lugar para la experimentación y el entendimiento acerca del funcionamiento y procesamiento necesarios para entrenar modelos que usamos a diario.

4.1.1. Tabla comparativa de traducciones

4.1.1.1. Google Translate

Tabla 2 Traducciones de frases propuestas con Google Translate (GNMT)

Nº	Frase Original	Frase Traducida	Frase Esperada	Calificación (/5)
#1	Dear Sixth Imo, Grandfather's youngest daughter,	Liebe Sechste Imo, Großvaters jüngste Tochter,	Liebe Sechste Imo, Großvaters jüngste Tochter,	5
#2	Nobody asked where her money came from. They knew where youngest daughters disappeared to	Niemand fragte, woher ihr Geld kam aus. Sie wussten, wohin die jüngsten Töchter verschwanden	Niemand fragte, woher sie das Geld hatte. Sie wussten, wohin die jüngsten Töchter verschwanden	3
#3	The first time I heard the rumor of you it was a mistake to ask your name	Das erste Mal, dass ich Ich habe das Gerücht gehört, dass es ein Fehler war, nach Ihrem Namen zu fragen	Das erste Mal, als ich das Gerücht über dich hörte, war es falsch nach deinem Namen zu fragen	3
#4	There was once a young man named Jack. He lived with his mother on a small farm at the foot of the Foggy Mountains.	Es war einmal ein junger Mann namens Jack. Er lebte mit seiner Mutter auf einer kleinen Farm am Fuße der Foggy Mountains.	Es war einmal ein junger Mann namens Jack. Er lebte mit seiner Mutter auf einem kleinen Hof am Fuß der Nebelberge.	4
#5	"Jack, wake up! You must go to the market today and sell our cow, Bess!" "But why?" asked Jack, yawning.	„Jack, wach auf! Du musst heute auf den Markt gehen und unsere Kuh verkaufen, Bess!“ "Aber warum?" fragte Jack gähmend.	„Jack, wach auf! Du musst heute zum Markt gehen und unsere Kuh Bess verkaufen.“ „Aber wieso?“, fragte Jack und gähnte.	3
#6	Jack packed his bag and fetched Bess from the barn. As he was walking out the gate, he heard his mother shout: "She is worth at least five gold coins! Don't sell her for anything less!"	Jack packte seine Tasche und holte Bess aus der Scheune. Als er aus dem Tor ging, hörte er seine Mutter rufen: „Sie ist mindestens fünf Goldmünzen wert! Verkaufen Sie sie nicht für weniger!“	Jack packte seinen Beutel und holte Bess aus dem Stall. Als er aus der Tür ging, hörte er seine Mutter rufen: „Sie ist mindestens fünf Goldmünzen Wert! Verkaufe sie nicht für weniger!“	2
#7	He ran home and showed his mother the beans. But she was not happy, instead she was furious. "You foolish boy! You traded our only cow for a handful of beans!"	Er rannte nach Hause und zeigte seiner Mutter die Bohnen. Aber sie war nicht glücklich, sondern wütend. „Du dummer Junge! Du hast unsere einzige Kuh gegen eine	Er rannte nach Hause und zeigte seiner Mutter die Bohnen. Aber sie war nicht glücklich, sondern wütend. „Du dummer Junge! Du hast unsere einzige Kuh für eine Handvoll Bohnen getauscht!“	2

		Handvoll Bohnen eingetauscht!“		
#8	The entire European Union wants to be climate-neutral by 2050, and Germany has set a national goal of 2045. By then, energy supplies the world over need to be transitioned to green, renewable sources.	Die gesamte Europäische Union will bis 2050 klimaneutral sein, Deutschland hat sich als nationales Ziel das Jahr 2045 gesetzt. Bis dahin muss die Energieversorgung weltweit auf grüne, erneuerbare Quellen umgestellt werden.	Die gesamte Europäische Union will bis spätestens 2050 klimaneutral werden, Deutschland bereits bis 2045. Bis dahin muss die Energieversorgung weltweit durch grüne, erneuerbare Energien gesichert sein.	3
#9	The fact that Germany and France do not agree on all aspects of energy-related issues – for example the role of nuclear energy – does not prevent them from engaging in a close dialogue.	Dass sich Deutschland und Frankreich nicht in allen Aspekten der Energiefragen einig sind – beispielsweise über die Rolle der Kernenergie –, hindert sie nicht daran, einen engen Dialog zu führen.	Dass Deutschland und Frankreich dabei in Energiefragen nicht bei allen Aspekten einer Meinung sind, zum Beispiel mit Blick auf die Nuklearenergie, steht einem engen Dialog in diesen Fragen nicht entgegen – im Gegenteil.	3
#10	Germany is determined to work together with France and other partners to drive forward the development of hydrogen. Together, we can actively shape the international ramp-up of the hydrogen market, and thus advance climate protection and seize economic opportunities at the same time.	Deutschland ist entschlossen, gemeinsam mit Frankreich und anderen Partnern die Entwicklung von Wasserstoff voranzutreiben. Gemeinsam können wir den internationalen Hochlauf des Wasserstoffmarktes aktiv mitgestalten und so den Klimaschutz vorantreiben und gleichzeitig wirtschaftliche Chancen nutzen.	Deutschland ist entschlossen, gemeinsam mit Frankreich und weiteren Partner*innen die Entwicklung von Wasserstoff voranzutreiben. Gemeinsam haben wir die Möglichkeit, den Markthochlauf des globalen Wasserstoffmarktes aktiv zu gestalten und so Klimaschutz und ökonomische Chancen zu verbinden.	3
Total:				31

Nota: Tabla con oraciones ejemplares de las fuentes (Frase Original) mencionadas a continuación; incluye su traducción oficial al alemán (Frase Esperada), y la traducción (Frase Traducida) con Google Translate. Incluida la calificación dada por una experta.

Fragmentos obtenidos de:

- Artículo sobre energía sostenible: <https://www.auswaertiges-amt.de/en/aussenpolitik/-/2621982>
- *Note Left at a U.S. Camptown Brothel for My Missing Imo*: <https://www.lyrikline.org/de/uebersetzerinnen/details/2760/irina-bondas>
- *Jack and the Beanstalk* (Jack y las habichuelas mágicas): <https://www.thefablecottage.com/fables/jack-und-die-bohnenranke>

Traducido con: <https://translate.google.com>

4.1.1.2. Chat GPT

Tabla 3 Traducciones de frases propuestas con Chat GPT (GPT-3.5)

Nº	Frase Original	Frase Traducida	Frase Esperada	Calificación (/5)
#1	Dear Sixth Imo, Grandfather's youngest daughter,	Liebe Sechste Imo, jüngste Tochter des Großvaters,	Liebe Sechste Imo, Großvaters jüngste Tochter,	4
#2	Nobody asked where her money came from. They knew where youngest daughters disappeared to	Niemand fragte, woher ihr Geld kam. Sie wussten, wohin die jüngsten Töchter verschwanden.	Niemand fragte, woher sie das Geld hatte. Sie wussten, wohin die jüngsten Töchter verschwanden	4
#3	The first time I heard the rumor of you it was a mistake to ask your name	Als ich zum ersten Mal das Gerücht über dich hörte, war es ein Fehler, nach deinem Namen zu fragen.	Das erste Mal, als ich das Gerücht über dich hörte, war es falsch nach deinem Namen zu fragen	4
#4	There was once a young man named Jack. He lived with his mother on a small farm at the foot of the Foggy Mountains.	Es war einmal ein junger Mann namens Jack. Er lebte mit seiner Mutter auf einem kleinen Bauernhof am Fuße der Nebelberge.	Es war einmal ein junger Mann namens Jack. Er lebte mit seiner Mutter auf einem kleinen Hof am Fuß der Nebelberge.	4
#5	"Jack, wake up! You must go to the market today and sell our cow, Bess!" "But why?" asked Jack, yawning.	"Jack, wach auf! Du musst heute auf den Markt gehen und unsere Kuh Bess verkaufen!" "Aber warum?" fragte Jack gähmend.	„Jack, wach auf! Du musst heute zum Markt gehen und unsere Kuh Bess verkaufen.“ „Aber wieso?“, fragte Jack und gähnte.	4

#6	Jack packed his bag and fetched Bess from the barn. As he was walking out the gate, he heard his mother shout: "She is worth at least five gold coins! Don't sell her for anything less!"	Jack packte seine Tasche und holte Bess aus dem Stall. Als er durch das Tor ging, hörte er seine Mutter rufen: "Sie ist mindestens fünf Goldmünzen wert! Verkaufe sie nicht für weniger!"	Jack packte seinen Beutel und holte Bess aus dem Stall. Als er aus der Tür ging, hörte er seine Mutter rufen: „Sie ist mindestens fünf Goldmünzen Wert! Verkaufe sie nicht für weniger!“	4
#7	He ran home and showed his mother the beans. But she was not happy, instead she was furious. "You foolish boy! You traded our only cow for a handful of beans!"	Er rannte nach Hause und zeigte seiner Mutter die Bohnen. Doch sie war nicht glücklich, stattdessen war sie wütend. "Du törichter Junge! Du hast unsere einzige Kuh gegen eine Handvoll Bohnen getauscht!"	Er rannte nach Hause und zeigte seiner Mutter die Bohnen. Aber sie war nicht glücklich, sondern wütend. „Du dummer Junge! Du hast unsere einzige Kuh für eine Handvoll Bohnen getauscht!“	3
#8	The entire European Union wants to be climate-neutral by 2050, and Germany has set a national goal of 2045. By then, energy supplies the world over need to be transitioned to green, renewable sources.	Die gesamte Europäische Union strebt an, bis 2050 klimaneutral zu sein, und Deutschland hat sich das nationale Ziel von 2045 gesetzt. Bis dahin müssen die Energieversorgungen weltweit auf grüne, erneuerbare Quellen umgestellt werden.	Die gesamte Europäische Union will bis spätestens 2050 klimaneutral werden, Deutschland bereits bis 2045. Bis dahin muss die Energieversorgung weltweit durch grüne, erneuerbare Energien gesichert sein.	4
#9	The fact that Germany and France do not agree on all aspects of energy-related issues – for example the role of nuclear energy – does not prevent them from engaging in a close dialogue.	Die Tatsache, dass Deutschland und Frankreich nicht in allen Aspekten energiebezogener Fragen übereinstimmen - beispielsweise in Bezug auf die Rolle der Kernenergie - hindert sie nicht daran, einen engen Dialog zu führen.	Dass Deutschland und Frankreich dabei in Energiefragen nicht bei allen Aspekten einer Meinung sind, zum Beispiel mit Blick auf die Nuklearenergie, steht einem engen Dialog in diesen Fragen nicht entgegen – im Gegenteil.	3
#10	Germany is determined to work together with France and other partners to drive forward the development of hydrogen. Together, we can actively shape the international	Deutschland ist entschlossen, mit Frankreich und anderen Partnern zusammenzuarbeiten, um die Entwicklung von Wasserstoff	Deutschland ist entschlossen, gemeinsam mit Frankreich und weiteren Partner*innen die Entwicklung von Wasserstoff voranzutreiben. Gemeinsam haben wir die Möglichkeit, den Markthochlauf des globalen	3

ramp-up of the hydrogen market, and thus advance climate protection and seize economic opportunities at the same time.	voranzutreiben. Gemeinsam können wir aktiv den internationalen Hochlauf des Wasserstoffmarktes gestalten und damit den Klimaschutz vorantreiben und gleichzeitig wirtschaftliche Chancen nutzen.	Wasserstoffmarktes aktiv zu gestalten und so Klimaschutz und ökonomische Chancen zu verbinden.	
Total:			37

Nota: Tabla con oraciones ejemplares de las fuentes (Frase Original) mencionadas a continuación; incluye su traducción oficial al alemán (Frase Esperada), y la traducción (Frase Traducida) con Chat GPT. Incluida la calificación dada por una experta.

Fragmentos obtenidos de:

- Artículo sobre energía sostenible: <https://www.auswaertiges-amt.de/en/aussenpolitik/-/2621982>
- *Note Left at a U.S. Camptown Brothel for My Missing Imo*: <https://www.lyrikline.org/de/uebersetzerinnen/details/2760/irina-bondas>
- *Jack and the Beanstalk* (Jack y las habichuelas mágicas): <https://www.thefablecottage.com/fables/jack-und-die-bohnenranke>

Traducido con: <https://chat.openai.com/share/da1e1af6-b239-449a-b813-42dcada4a553>

4.1.1.3. Modelo de NMT con PyTorch

Tabla 4 Traducciones de frases propuestas con el modelo credo con PyTorch (NMT)

Nº	Frase Original	Frase Traducida	Frase Esperada	Calificación (/5)
#1	Dear Sixth Imo, Grandfather's youngest daughter,	Die <UNK> Vaters <UNK> größte Tochter.	Liebe Sechste Imo, Großvaters jüngste Tochter,	1
#2	Nobody asked where her money came from. They knew where	Niemand fragte woher ihr Geld kam. Sie wussten wo man zum	Niemand fragte, woher sie das Geld hatte. Sie wussten, wohin die jüngsten Töchter ver-	2

	youngest daughters disappeared to	<UNK> verschwand.	schwanden	
#3	The first time I heard the rumor of you it was a mistake to ask your name	Das erste Mal liegt bei dir wie einen Fehler von	Das erste Mal, als ich das Gerücht über dich hörte, war es falsch nach deinem Namen zu fragen	2
#4	There was once a young man named Jack. He lived with his mother on a small farm at the foot of the Foggy Mountains.	Vor einem jungen gab es noch einmal ein jungen Mann. er lebte bei einem kleinen Bauernhof am fußen mit	Es war einmal ein junger Mann namens Jack. Er lebte mit seiner Mutter auf einem kleinen Hof am Fuß der Nebelberge.	2
#5	"Jack, wake up! You must go to the market today and sell our cow, Bess!" "But why?" asked Jack, yawning.	Auf den Markt sofort stehen und verkaufe unsere <UNK>!" aber warum? warum fordert das <UNK>.	„Jack, wach auf! Du musst heute zum Markt gehen und unsere Kuh Bess verkaufen.“ „Aber wieso?“, fragte Jack und gähnte.	1
#6	Jack packed his bag and fetched Bess from the barn. As he was walking out the gate, he heard his mother shout: "She is worth at least five gold coins! Don't sell her for anything less!"	<UNK> rührte seine Tasche und seine <UNK> <UNK>. Er ging raus wie er seine Mutter bekam hörte: „Sie ist mindestens fünf <UNK> Münzen nicht für all weniger“	Jack packte seinen Beutel und holte Bess aus dem Stall. Als er aus der Tür ging, hörte er seine Mutter rufen: „Sie ist mindestens fünf Goldmünzen Wert! Verkaufe sie nicht für weniger!“	1
#7	He ran home and showed his mother the beans. But she was not happy, instead she was furious. "You foolish boy! You traded our only cow for a handful of beans!"	Er rannte nach Hause und zeigte seine Mutter die Bohnen. Sie war nicht glücklich weil sie wütend war. Du hast heißt jungen deines einzigen zum Bohnen!	Er rannte nach Hause und zeigte seiner Mutter die Bohnen. Aber sie war nicht glücklich, sondern wütend. „Du dummer Junge! Du hast unsere einzige Kuh für eine Handvoll Bohnen getauscht!“	2
#8	The entire European Union wants to be climate-neutral by 2050, and Germany has set a national goal of 2045. By then, energy supplies the world over need to be transitioned to green, renewable sources.	Das ganze <UNK> mochte <UNK> wie durch Deutschland einen <UNK>. Indem <UNK> die Welt sind um <UNK> <UNK>. Indem er die <UNK> <UNK> muss die <UNK> aus grünen	Die gesamte Europäische Union will bis spätestens 2050 klimaneutral werden, Deutschland bereits bis 2045. Bis dahin muss die Energieversorgung weltweit durch grüne, erneuerbare Energien gesichert sein.	1
#9	The fact that Germany and France do not agree on all aspects of energy-related issues – for example the role of nuclear energy – does not prevent them from	die Tatsache dieses Deutschland sind in allen Punkten der Frankreich - zum Beispiel gibt es die rolle Hauptstadt? - sie verhindern sich nicht	Dass Deutschland und Frankreich dabei in Energiefragen nicht bei allen Aspekten einer Meinung sind, zum Beispiel mit Blick auf die Nuklearenergie, steht einem engen Dialog in diesen Fragen	2

	engaging in a close dialogue.	vom Heben einer <UNK>.	nicht entgegen – im Gegenteil.	
#10	Germany is determined to work together with France and other partners to drive forward the development of hydrogen. Together, we can actively shape the international ramp-up of the hydrogen market, and thus advance climate protection and seize economic opportunities at the same time.	Deutschland ist entschlossen mit Frankreich und <UNK> um die <UNK>. Man kann die internationale Temperaturen auf der <UNK> <UNK> und	Deutschland ist entschlossen, gemeinsam mit Frankreich und weiteren Partner*innen die Entwicklung von Wasserstoff voranzutreiben. Gemeinsam haben wir die Möglichkeit, den Markthochlauf des globalen Wasserstoffmarktes aktiv zu gestalten und so Klimaschutz und ökonomische Chancen zu verbinden.	1
Total:				15

Nota: Tabla con oraciones ejemplares de las fuentes (Frase Original) mencionadas a continuación; incluye su traducción oficial al alemán (Frase Esperada), y la traducción (Frase Traducida) con el modelo hecho con PyTorch. Incluida la calificación dada por una experta.

Fragmentos obtenidos de:

- Artículo sobre energía sostenible: <https://www.auswaertiges-amt.de/en/aussenpolitik/-/2621982>
- *Note Left at a U.S. Camptown Brothel for My Missing Imo*: <https://www.lyrikline.org/de/uebersetzerinnen/details/2760/irina-bondas>
- *Jack and the Beanstalk* (Jack y las habichuelas mágicas): <https://www.thefablecottage.com/fables/jack-und-die-bohnenranke>

4.2. Diseño de la RNN para realizar NMT con PyTorch

4.2.1. Entendimiento de los Datos

Para el entrenamiento del modelo desarrollado para este trabajo de titulación, se siguió la recomendación del tutorial y se tomó el dataset de inglés-alemán de una página que contiene varios dataset de traducción de y a diferentes idiomas (link adjunto

en los anexos); el que se tomó fue el de *deu.txt*. Es un archivo de texto separado por tabulador, el cual consta de las siguientes características:

- Tamaño: 4.064 MB
- Nº de campos: 267186
- Nº de columnas: 3
- Columnas: vale recalcar que, como tal, no tiene una cabecera, pero se puede distinguir las columnas por su contenido.
 - o Oración en inglés
 - o Oración en alemán
 - o Creative Commons

```
user@RHEA: /mnt/c/Users/mgld2/OneDrive/Documents/Octavo_Semestre/Integracion Curricular$ wc -l deu.txt
267186 deu.txt
user@RHEA: /mnt/c/Users/mgld2/OneDrive/Documents/Octavo_Semestre/Integracion Curricular$ stat deu.txt
  File: deu.txt
  Size: 41615081      Blocks: 81280      IO Block: 512     regular file
Device: 49h/73d Inode: 42221246506616459 Links: 1
Access: (0777/-rwxrwxrwx)  Uid: ( 1000/   user)   Gid: ( 1000/   user)
Access: 2023-10-10 08:39:13.712442800 -0500
Modify: 2023-09-13 22:28:14.227987100 -0500
Change: 2023-10-10 08:39:13.712442800 -0500
```

Ilustración 11 Información acerca del archivo deu.txt

```

Go.   Geh.   CC-BY 2.0 (France) Attribution: tatoeba.org #2877272 (CM) & #8597805
(Roujin)
Hi.   Hallo! CC-BY 2.0 (France) Attribution: tatoeba.org #538123 (CM) & #380701
(cburgmer)
Hi.   Grüß Gott! CC-BY 2.0 (France) Attribution: tatoeba.org #538123 (CM) & #
659813 (Esperantostern)
Run!  Lauf!  CC-BY 2.0 (France) Attribution: tatoeba.org #906328 (papabear) & #941078
(Fingerhut)
Run.  Lauf!  CC-BY 2.0 (France) Attribution: tatoeba.org #4008918 (JSakuragi) & #
941078 (Fingerhut)
Wow!  Potzdonner! CC-BY 2.0 (France) Attribution: tatoeba.org #52027 (Zifre) & #
2122382 (Pfirsichbaeumchen)
Wow!  Donnerwetter! CC-BY 2.0 (France) Attribution: tatoeba.org #52027 (Zifre) & #
2122391 (Pfirsichbaeumchen)
Duck! Kopf runter! CC-BY 2.0 (France) Attribution: tatoeba.org #280158 (CM) & #
9968521 (wolfgangth)
Fire! Feuer! CC-BY 2.0 (France) Attribution: tatoeba.org #1829639 (Spamster) & #
1958697 (Tamy)
Help! Hilfe! CC-BY 2.0 (France) Attribution: tatoeba.org #435084 (lukaszpp) & #575889
(MUIRIEL)
Help! Zu Hülf! CC-BY 2.0 (France) Attribution: tatoeba.org #435084 (lukaszpp) &
#2122375 (Pfirsichbaeumchen)
Stay. Bleib! CC-BY 2.0 (France) Attribution: tatoeba.org #8907595 (CK) & #5344007
(wochenweise)
Stop! Stopp! CC-BY 2.0 (France) Attribution: tatoeba.org #448320 (CM) & #626467
(jakov)
Stop! Anhalten! CC-BY 2.0 (France) Attribution: tatoeba.org #448320 (CM) & #
7481623 (Yorwba)
Wait! Warte! CC-BY 2.0 (France) Attribution: tatoeba.org #1744314 (belgavox) & #
2122378 (Pfirsichbaeumchen)
Wait. Warte. CC-BY 2.0 (France) Attribution: tatoeba.org #3048304 (camilozeta) & #
-----
Ln 1, Col 1                               100% | Unix (LF) | UTF-8

```

Ilustración 12 Primeras 16 líneas del archivo deu.txt

4.2.2. Preparación de los Datos

Como tal, no se requirió hacer limpieza o tratamiento de valores nulos o faltantes, se trató pero con expresiones *Regex* los signos de puntuación como: “ , ”, “ ’ ”, y caracteres especiales como “ß”, “ä”, “ö”, entre otros; los cuales son particulares al idioma alemán. Principalmente lo que se hizo para asegurar que el modelo o el computador no se saturen fue limitar el vocabulario para cada idioma a 20000 palabras, se definió que no haya recorte de pares de oraciones más largas de 12 palabras y se trató a las contracciones comunes en inglés como *I am (I’m), you are (you’re), they are (they’re)*, entre otros.

Para la preparación de la data para que lo pueda utilizar la red, se convirtió a las oraciones en vectores One-Hot y separó en baches de 32 pares de oraciones.

4.2.3. Modelado

Se siguió el tutorial (referirse a los anexos) de Towards Data Science, con algunas modificaciones personales para ajustar con los objetivos de esta investigación. Es posible la creación de un archivo de texto que captura el resultado del entrenamiento, pero se puso la opción para no generar el archivo y se optó por guardar la ejecución como pdf.

Se aplicó un modelo de Codificador-Decodificador con 2 capas por cada nivel (incluyendo la capa escondida) con células LSTM y una tasa de aprendizaje de 0.8 por 100 épocas. El criterio de pérdida utilizado fue el de la pérdida de entropía cruzada (Cross-Entropy Loss). El siguiente gráfico del tutorial con la oración de ejemplo *El gato le gusta comer pizza* describe perfectamente la arquitectura utilizada. (Lanners, 2022)

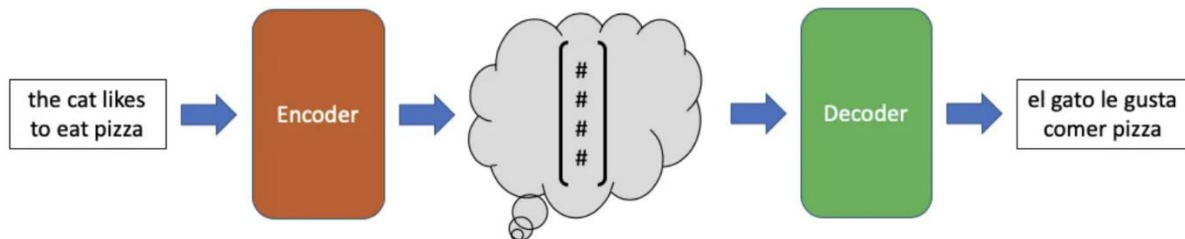


Ilustración 13 Arquitectura del modelo de MT desarrollado

Las librerías necesarias para el modelo son las siguientes:

```

import unicodedata
import re
import math
import psutil
import time
import datetime
from io import open
import random
from random import shuffle
import argparse
import numpy as np
import matplotlib.pyplot as plt
import nltk
from nltk.translate.bleu_score import sentence_bleu

import torch
from torch.autograd import Variable
import torch.nn as nn
import torch.nn.functional as F
from torch import optim
import torch.cuda

"""this line clears sys to allow for argparse to work as gradient clipper"""
import sys; sys.argv=['']; del sys

```

Ilustración 14 Librerías usadas para la generación del modelo

CUDA es una librería esencial para el cómputo del entrenamiento y pruebas, por lo que es necesario verificar su funcionalidad. El código se puede igualmente ejecutar en CPU, pero es recomendable ejecutar en GPU por los beneficios que trae en cuanto a tiempo de ejecución, utilización de recursos, etc.

Luego se crea la clase *Lang*, la cual se encargará de generar el vocabulario para los idiomas de entrada y salida con una función que revisa que ambos idiomas tengan diccionarios de largos apropiados. (Lanners, 2022)

```

"""Lang class, used to store the vocabulary of each language"""
class Lang:
    def __init__(self, language):
        self.language_name = language
        self.word_to_index = {"SOS":SOS_token, "EOS":EOS_token, "<UNK>":UNK_token}
        self.word_to_count = {}
        self.index_to_word = {SOS_token: "SOS", EOS_token: "EOS", UNK_token: "<UNK>"}
        self.vocab_size = 3
        self.cutoff_point = -1

    def countSentence(self, sentence):
        for word in sentence.split(' '):
            self.countWords(word)

    """counts the number of times each word appears in the dataset"""
    def countWords(self, word):
        if word not in self.word_to_count:
            self.word_to_count[word] = 1
        else:
            self.word_to_count[word] += 1

    """if the number of unique words in the dataset is larger than the
    specified max_vocab_size, creates a cutoff point that is used to
    leave infrequent words out of the vocabulary"""
    def createCutoff(self, max_vocab_size):
        word_freqs = list(self.word_to_count.values())
        word_freqs.sort(reverse=True)
        if len(word_freqs) > max_vocab_size:
            self.cutoff_point = word_freqs[max_vocab_size]

    """assigns each unique word in a sentence a unique index"""
    def addSentence(self, sentence):

```

Ilustración 15 Fragmento del código de la clase Lang

Debido a dicha función, el o los documentos de entrada tienen que ser separados por tabulador con un formato específico. Una vez que se lea los archivos, la función *prepareData* se encarga de limpiar y hacer el preprocesamiento para cada idioma, devolviendo ambos idiomas como objetos de clase *Lang* junto con los pares de oraciones de entrenamiento y prueba. (Lanners, 2022)

Una vez creados los diccionarios para cada idioma, se crea una función para transformar los pares de oraciones a vectores One-Hot y leerlos. Luego, para entrenar el modelo se utiliza el descenso de gradiente por minibatches, por lo cual se debe crear una función para separar los vectores obtenidos en dichos batches. Al tener todo el

preprocesamiento listo, se procede con la codificación de las clases de Codificador y Decodificador para crear las bases de las capas que se definirán luego en el algoritmo.

(Lanners, 2022)

Conversión de oraciones a vectores

```
"""converts a sentence to one hot encoding vectors - pytorch allows us to just
use the number corresponding to the unique index for that word,
rather than a complete one hot encoding vector for each word"""
def indexesFromSentence(lang, sentence):
    indexes = []
    for word in sentence.split(' '):
        try:
            indexes.append(lang.word_to_index[word])
        except:
            indexes.append(lang.word_to_index["<UNK>"])
    return indexes

def tensorFromSentence(lang, sentence):
    indexes = indexesFromSentence(lang, sentence)
    indexes.append(EOS_token)
    result = torch.LongTensor(indexes).view(-1)
    if use_cuda:
        return result.cuda()
    else:
        return result

"""converts a pair of sentence (input and target) to a pair of tensors"""
def tensorsFromPair(input_lang, output_lang, pair):
    input_variable = tensorFromSentence(input_lang, pair[0])
    target_variable = tensorFromSentence(output_lang, pair[1])
    return (input_variable, target_variable)
```

Ilustración 16 Funciones para convertir las oraciones a vectores One-Hot por medio de tensores

Separación de datos en batches

```
"""seperates data into batches of size batch_size"""
def batchify(data, input_lang, output_lang, batch_size, shuffle_data=True):
    if shuffle_data == True:
        shuffle(data)
    number_of_batches = len(data) // batch_size
    batches = list(range(number_of_batches))
    longest_elements = list(range(number_of_batches))

    for batch_number in range(number_of_batches):
        longest_input = 0
        longest_target = 0
        input_variables = list(range(batch_size))
        target_variables = list(range(batch_size))
        index = 0
        for pair in range((batch_number*batch_size), ((batch_number+1)*batch_size)):
            input_variables[index], target_variables[index] = tensorsFromPair(input_lang, output_lang, data[pair])
            if len(input_variables[index]) >= longest_input:
                longest_input = len(input_variables[index])
            if len(target_variables[index]) >= longest_target:
                longest_target = len(target_variables[index])
            index += 1
        batches[batch_number] = (input_variables, target_variables)
        longest_elements[batch_number] = (longest_input, longest_target)
    return batches , longest_elements, number_of_batches

"""pads batches to allow for sentences of variable lengths to be computed in parallel"""
def pad_batch(batch):
    padded_inputs = torch.nn.utils.rnn.pad_sequence(batch[0],padding_value=EOS_token)
```

Ilustración 17 Separación de los vectores en batches para el entrenamiento y prueba

Para entrenar y probar el modelo se utiliza la función de *train_batch*, el cual da un paso adelante para el modelo y calcula la pérdida, la cual es propagada hacia atrás y guardada para actualizar todas las matrices de pesos en el Codificador y Decodificador. Con la función *train* iterando para la cantidad de baches creados para completar una época de entrenamiento. (Lanners, 2022)

Clase del Codificador

```
class EncoderRNN(nn.Module):
    def __init__(self, input_size, hidden_size, layers=1, dropout=0.1,
                 bidirectional=True):
        super(EncoderRNN, self).__init__()

        if bidirectional:
            self.directions = 2
        else:
            self.directions = 1
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.num_layers = layers
        self.dropout = dropout
        self.embedder = nn.Embedding(input_size, hidden_size)
        self.dropout = nn.Dropout(dropout)
        self.lstm = nn.LSTM(input_size=hidden_size, hidden_size=hidden_size,
                            num_layers=layers, dropout=dropout,
                            bidirectional=bidirectional, batch_first=False)
        self.fc = nn.Linear(hidden_size*self.directions, hidden_size)

    def forward(self, input_data, h_hidden, c_hidden):
        embedded_data = self.embedder(input_data)
        embedded_data = self.dropout(embedded_data)
        hiddens, outputs = self.lstm(embedded_data, (h_hidden, c_hidden))

        return hiddens, outputs

    """creates initial hidden states for encoder corresponding to batch size"""
    def create_init_hiddens(self, batch_size):
```

Ilustración 18 Fragmento de código de la clase del Codificador

Clase del Decodificador

```
class DecoderAttn(nn.Module):
    def __init__(self, hidden_size, output_size, layers=1, dropout=0.1, bidirectional=True):
        super(DecoderAttn, self).__init__()

        if bidirectional:
            self.directions = 2
        else:
            self.directions = 1
        self.output_size = output_size
        self.hidden_size = hidden_size
        self.num_layers = layers
        self.dropout = dropout
        self.embedder = nn.Embedding(output_size, hidden_size)
        self.dropout = nn.Dropout(dropout)
        self.score_learner = nn.Linear(hidden_size*self.directions,
                                      hidden_size*self.directions)
        self.lstm = nn.LSTM(input_size=hidden_size, hidden_size=hidden_size,
                            num_layers=layers, dropout=dropout,
                            bidirectional=bidirectional, batch_first=False)
        self.context_combiner = nn.Linear((hidden_size*self.directions)
                                         +(hidden_size*self.directions), hidden_size)
        self.tanh = nn.Tanh()
        self.output = nn.Linear(hidden_size, output_size)
        self.soft = nn.Softmax(dim=1)
        self.log_soft = nn.LogSoftmax(dim=1)

    def forward(self, input_data, h_hidden, c_hidden, encoder_hiddens):
```

Ilustración 19 Fragmento de código de la clase de Decodificador

```

'''Performs training on a single batch of training data. Computing the loss
according to the passed loss_criterion and back-propagating on this loss.'''
def train_batch(input_batch, target_batch, encoder, decoder,
               encoder_optimizer, decoder_optimizer, loss_criterion):
    encoder_optimizer.zero_grad()
    decoder_optimizer.zero_grad()
    loss = 0
    enc_h_hidden, enc_c_hidden = encoder.create_init_hiddens(input_batch.shape[1])

    enc_hiddens, enc_outputs = encoder(input_batch, enc_h_hidden, enc_c_hidden)

    decoder_input = Variable(torch.LongTensor(1,input_batch.shape[1]).
                             fill_(output_lang.word_to_index.get("SOS")).cuda()) if use_cuda \
                             else Variable(torch.LongTensor(1,input_batch.shape[1]).
                             fill_(output_lang.word_to_index.get("SOS")))

    dec_h_hidden = enc_outputs[0]
    dec_c_hidden = enc_outputs[1]

    for i in range(target_batch.shape[0]):
        pred, dec_outputs = decoder(decoder_input, dec_h_hidden,
                                   dec_c_hidden, enc_hiddens)

        decoder_input = target_batch[i].view(1,-1)
        dec_h_hidden = dec_outputs[0]
        dec_c_hidden = dec_outputs[1]

        loss += loss_criterion(pred,target_batch[i])

    loss.backward()

```

Ilustración 20 Fragmento del código de entrenamiento por batches

Las funciones de *test_batch* y *test* tienen funciones similares a las anteriores, pero trabajan sólo para computar la pérdida y no la propagan hacia atrás. Esto sirve principalmente para evaluar el modelo con oraciones que no se encontraban en el entrenamiento. Para evaluar oraciones externas al dataset, se creó la función de *evaluate* que toma de entrada una oración y de salida da la traducción predecida. Se creó igualmente la función *evaluate_randomly* para realizar dicha evaluación con las oraciones de entrenamiento y prueba para evaluar al modelo a través de cada época. (Lanners, 2022)

```
'''Evaluates the loss on a single batch of test data. Computing the loss
according to the passed loss_criterion. Does not perform back-prop'''

def test_batch(input_batch, target_batch, encoder, decoder, loss_criterion):

    loss = 0

    #create initial hidde state for encoder
    enc_h_hidden, enc_c_hidden = encoder.create_init_hiddens(input_batch.shape[1])

    enc_hiddens, enc_outputs = encoder(input_batch, enc_h_hidden, enc_c_hidden)

    decoder_input = Variable(torch.LongTensor(1,input_batch.shape[1]).
                              fill_(output_lang.word_to_index.get("SOS")).cuda()) if use_cuda \
                              else Variable(torch.LongTensor(1,input_batch.shape[1]).
                              fill_(output_lang.word_to_index.get("SOS")))
    dec_h_hidden = enc_outputs[0]
    dec_c_hidden = enc_outputs[1]

    for i in range(target_batch.shape[0]):
        pred, dec_outputs = decoder(decoder_input, dec_h_hidden, dec_c_hidden, enc_hiddens)

        topv, topi = pred.topk(1,dim=1)
        ni = topi.view(1,-1)

        decoder_input = ni
        dec_h_hidden = dec_outputs[0]
        dec_c_hidden = dec_outputs[1]

        loss += loss_criterion(pred,target_batch[i])
```

Ilustración 21 Fragmento de código de la función test_batches para la evaluación de los datos de prueba

Función para retornar la traducción predecida de una o varias oraciones

```
'''Returns the predicted translation of a given input sentence. Predicted
translation is trimmed to length of cutoff_length argument'''

def evaluate(encoder, decoder, sentence, cutoff_length):
    with torch.no_grad():
        input_variable = tensorFromSentence(input_lang, sentence)
        input_variable = input_variable.view(-1,1)
        enc_h_hidden, enc_c_hidden = encoder.create_init_hiddens(1)

        enc_hiddens, enc_outputs = encoder(input_variable, enc_h_hidden, enc_c_hidden)

        decoder_input = Variable(torch.LongTensor(1,1).fill_(output_lang.word_to_index.get("SOS")).cuda()) if use_cuda \
            else Variable(torch.LongTensor(1,1).fill_(output_lang.word_to_index.get("SOS")))
        dec_h_hidden = enc_outputs[0]
        dec_c_hidden = enc_outputs[1]

        decoded_words = []

        for di in range(cutoff_length):
            pred, dec_outputs = decoder(decoder_input, dec_h_hidden, dec_c_hidden, enc_hiddens)

            topv, topi = pred.topk(1,dim=1)
            ni = topi.item()
            if ni == output_lang.word_to_index.get("EOS"):
                decoded_words.append('<EOS>')
                break
            else:
```

Ilustración 22 Función para probar el modelo con una o varias oraciones fuera del dataset de entrenamiento y prueba

```
def evaluate_randomly(encoder, decoder, pairs, n=2, trim=100, total_BLEU = 0):

    for i in range(n):
        pair = random.choice(pairs)
        print('>', pair[0])
        print('=' , pair[1])
        output_sentence = evaluate(encoder, decoder, pair[0], cutoff_length=trim)
        print('<', output_sentence)

        reference_tokens.extend([nltk.word_tokenize(pair[0])])
        candidate_tokens=nltk.word_tokenize(output_sentence[:-5])
        # Calculate the BLEU score
        bleu_score = sentence_bleu(reference_tokens, candidate_tokens,
                                   smoothing_function=nltk.translate.bleu_score.SmoothingFunction().method1)
        print(f"BLEU Score: {bleu_score:.4f}")
```

Ilustración 23 Función de evaluación randómica para observar la pérdida y BLEU a través de las distintas épocas con oraciones de los datos de prueba

Todas esas funciones igualmente son coordinadas por la función maestra *train_and_test*, la cual se ejecutó para correr el modelo como tal. Las siguientes secciones del código sirven meramente para definir los hiperparámetros, la generación de un txt que guarde la ejecución, la creación de un gráfico de pérdida vs tiempo. (Lanners, 2022)

```
'''The master function that trains the model. Evlautes progress on the train set
(if present) and also records the progress of training in both a txt file and
a png graph. Also can save the weights of both the Encoder and Decoder
for future use.'''

def train_and_test(epochs, test_eval_every, plot_every, learning_rate,
                  lr_schedule, train_pairs, test_pairs, input_lang,
                  output_lang, batch_size, test_batch_size, encoder, decoder,
                  loss_criterion, trim, save_weights):

    times = []
    losses = {'train set':[], 'test set': []}

    test_batches, longest_seq, n_o_b = batchify(test_pairs, input_lang,
                                                output_lang, test_batch_size,
                                                shuffle_data=False)

    start = time.time()
    for i in range(1,epochs+1):

        '''adjust the learning rate according to the learning rate schedule
        specified in lr_schedule'''
        if i in lr_schedule.keys():
            learning_rate /= lr_schedule.get(i)

        encoder.train()
        decoder.train()

        encoder_optimizer = optim.SGD(encoder.parameters(), lr=learning_rate)
        decoder_optimizer = optim.SGD(decoder.parameters(), lr=learning_rate)
```

Ilustración 24 Fragmento del código de la función máster que coordina al resto de funciones

En el tutorial no se consideró la métrica BLEU para evaluar las oraciones que traduce el modelo, por lo que se añadió esto a la función de evaluación randómica

(*evaluate_randomly*) para que evalúe esta métrica con los ejemplares que presenta de los datos de prueba. La librería utilizada para ello es *nltk* y *sentence_bleu*. Esta librería toma la oración de entrada y la de salida dividida por palabras, a lo largo con una función de suavizamiento logarítmico para presentar la calificación BLEU de una traducción. En las tablas se presenta esta calificación en base 100, por lo que la calificación presentada se debe multiplicar por 100 para obtener los resultados que se observaron en la Tabla 1.

4.2.4. Evaluación del modelo

Principalmente, se utilizó las métricas de Cross-Entropy Loss y el score BLEU. Para cada época se evalúa la pérdida en el entrenamiento y en la prueba. El score BLEU se utiliza para evaluar el modelo como tal y para lograr compararlo con GPT-3 y GNMT.

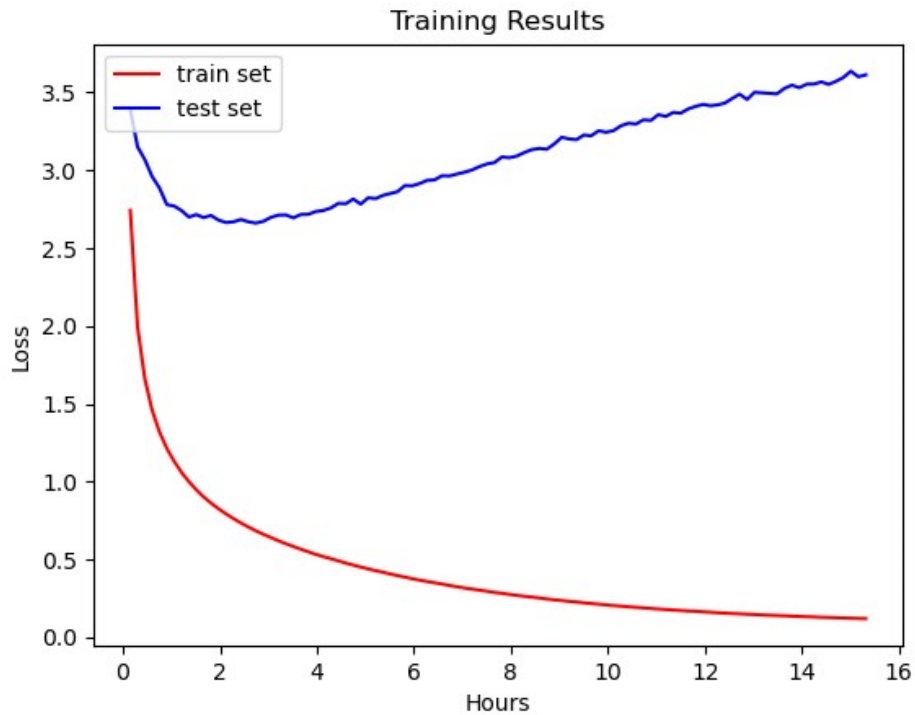


Ilustración 25 Gráfico de Pérdida vs Tiempo de ejecución del modelo (en horas)

```

Iter: 100
Learning Rate: 0.8
Time: 15h 18m 32s
Train Loss: 0.11964737085890882

Test set loss: 3.6135033316817013
> tom took three steps backward .
= tom trat drei schritte zuruck .
< tom hat drei schritte zur ganzen uhr ubernommen . <EOS>
BLEU Score: 0.0316

> could you please park somewhere else ?
= konntest du bitte woanders parken ?
< konntest du bitte woanders parken ? <EOS>
BLEU Score: 0.0408

```

Ilustración 26 Ejemplo del output del modelo con las métricas de pérdida en entrenamiento y prueba y la calificación BLEU de algunas oraciones de prueba

```
> she was born blind .  
= sie kam blind zur welt .  
< sie wurde blind geboren . <EOS>  
BLEU Score: 0.0639  
  
> get lost !  
= zisch ab !  
< hau ! <EOS>  
BLEU Score: 0.1495  
  
> that s the worst joke i ve ever heard .  
= das ist der schlechteste witz den ich je gehört habe .  
< das ist das schlechteste witz das ich je gehört habe . <EOS>  
BLEU Score: 0.0189
```

Ilustración 27 Ejemplos de traducciones de oraciones de prueba con su calificación BLEU

Como mencionado previamente, mientras más baja la pérdida, más acertado el modelo. Se puede observar un sobreajuste del modelo alrededor de la 8va hora de entrenamiento, donde la pérdida se incrementó considerablemente de manera lineal, mientras el error de entrenamiento se acercaba a cero. Desafortunadamente por el tiempo limitado disponible para ejecutar el modelo, la cantidad de datos y debido a la naturaleza de modelos de NMT, un cierto grado de sobreajuste era inevitable en estas circunstancias.

Pero, sorprendentemente, para ciertas traducciones predice muy cercana o exactamente la oración que se esperaba, incluso con oraciones fuera de las de entrenamiento, como se puede observar en la ilustración a continuación:

1.15 Traducción de oraciones personalizadas

```
outside_sent_eval = "take me home now please"
outside_sent = normalizeString(outside_sent_eval)
output = evaluate(encoder, decoder, outside_sent, cutoff_length=10)
print(output)
```

bring mich jetzt bitte nach hause ! <EOS>

Ilustración 28 Ejemplo de traducción con el modelo entrenado

En la ilustración 26 se observa igualmente la mejor calificación BLEU a la cual llegó el modelo, de 14.95. Aunque a lado de las calificaciones y resultados de ChatGPT y Google Translate sea la mitad de su calificación, es impresionante observar la capacidad que tiene un modelo reducido que, aunque aún impreciso, con más datos y escalándolo, podría llegar al rendimiento que reportó Google con su modelo en 2016.

Este modelo reducido, en su estado actual, no resulta competitivo ni equiparable en precisión a los otros dos modelos. Sin embargo, muestra el potencial que tiene PyTorch para el modelado de redes para NMT. Google y OpenAI no especificaron las librerías ni el lenguaje utilizado para programar sus modelos, por lo que son secretos corporativos de los modelos y no abiertos al público; pero sería interesante observar cómo parametrizaron sus redes y los retos que sobrellevaron para evitar problemas de sobreajuste en sus modelos. Aunque no sean muy aplicables a esta escala, tal vez el añadir algunos métodos para el tratamiento del sobreajuste mejoraría a el modelo presentado. Tanto para GNMT como para GPT-3 se realizó pruebas con modelos de diferentes tamaños (principalmente en cantidad de datos). Y en el caso de GNMT se realizó validación cruzada (*cross-validation*), donde expusieron los resultados de sus varios modelos, pero el que se describe en este estudio es el mismo del cual trata la investigación de Wu (2016).

4.3. Comparativas de la calidad de traducción de las IA

Tabla 5 Comparativa detallada de la calidad de las traducciones generadas por las diferentes aplicaciones traductoras calificada por experto #1

Oración Original	Oración Traducida	Aplicación	Sentido	Sensibilidad	Contenido	Directo	Tenor	Poder	Solidaridad	Total (/35)
Nobody asked where her money came from. They knew where youngest daughters disappeared to	Niemand fragte, woher ihr Geld kam aus. Sie wussten, wohin die jüngsten Töchter verschwanden	Google	4.5	4	5	5	5	4	4	31.5
	Niemand fragte, woher ihr Geld kam. Sie wussten, wohin die jüngsten Töchter verschwanden.	Chat GPT	5	5	5	5	5	5	5	35
	Niemand fragte woher ihr Geld kam. Sie wussten wo man zum <UNK> verschwand.	Modelo PyTorch	4	5	3.5	5	4.5	5	4	31
There was once a young man named Jack. He lived with his mother on a small farm at the foot of the Foggy Mountains.	Es war einmal ein junger Mann namens Jack. Er lebte mit seiner Mutter auf einer kleinen Farm am Fuße der Foggy Mountains.	Google	4.5	5	5	5	5	4	4.5	33
	Es war einmal ein junger Mann namens Jack. Er lebte mit seiner Mutter auf einem kleinen Bauernhof am Fuße der Nebelberge.	Chat GPT	5	5	5	5	5	5	5	35
	Vor einem jungen gab es noch einmal ein jungen Mann. er lebte bei einem kleinen Bauernhof am fußen mit	Modelo PyTorch	2	2	2	1	1	1	2	11
The fact that Germany and France do not agree on all	Dass sich Deutschland und Frankreich nicht in allen Aspekten der Energiefragen einig sind –	Google	4.5	5	5	5	5	5	5	34.5

aspects of energy-related issues – for example the role of nuclear energy – does not prevent them from engaging in a close dialogue.	beispielsweise über die Rolle der Kernenergie –, hindert sie nicht daran, einen engen Dialog zu führen.									
	Die Tatsache, dass Deutschland und Frankreich nicht in allen Aspekten energiebezogener Fragen übereinstimmen - beispielsweise in Bezug auf die Rolle der Kernenergie - hindert sie nicht daran, einen engen Dialog zu führen.	Chat GPT	5	5	5	5	5	5	5	35
	die Tatsache dieses Deutschland sind in allen Punkten der Frankreich - zum Beispiel gibt es die rolle Hauptstadt? - sie verhindern sich nicht vom Heben einer <UNK>.	Modelo PyTorch	2	2	1	2	1	2	1	11

Nota: Tabla para evaluar la calidad de traducciones generadas con aplicaciones de traducción; consta de oraciones ejemplares de las fuentes (Frase Original) mencionadas a continuación; incluye su traducción realizada por la aplicación de traducción (Frase Traducida) con Google Translate, Chat-GPT y el modelo hecho con PyTorch. Incluida la calificación dada por un experto.

Fragmentos obtenidos de:

- Artículo sobre energía sostenible: <https://www.auswaertiges-amt.de/en/aussenpolitik/-/2621982>
- *Note Left at a U.S. Camptown Brothel for My Missing Imo:* <https://www.lyrikline.org/de/uebersetzerinnen/details/2760/irina-bondas>
- *Jack and the Beanstalk* (Jack y las habichuelas mágicas): <https://www.thefablecottage.com/fables/jack-und-die-bohnenranke>

Tabla 6 Comparativa detallada de la calidad de las traducciones generadas por las diferentes aplicaciones traductoras calificada por experto #2

Oración Original	Oración Traducida	Aplicación	Sentido	Sensibilidad	Contenido	Directo	Tenor	Poder	Solidaridad	Total (/35)
Nobody asked where her money came from. They knew where youngest daughters disappeared to	Niemand fragte, woher ihr Geld kam aus. Sie wussten, wohin die jüngsten Töchter verschwanden	Google	5	4	5	5	5	5	4	33
	Niemand fragte, woher ihr Geld kam. Sie wussten, wohin die jüngsten Töchter verschwanden.	Chat GPT	5	5	5	5	5	5	5	35
	Niemand fragte woher ihr Geld kam. Sie wussten wo man zum <UNK> verschwand.	Modelo PyTorch	0	1	1	0	1	1	0	4
There was once a young man named Jack. He lived with his mother on a small farm at the foot of the Foggy Mountains.	Es war einmal ein junger Mann namens Jack. Er lebte mit seiner Mutter auf einer kleinen Farm am Fuße der Foggy Mountains.	Google	5	4	5	5	4	5	4	33
	Es war einmal ein junger Mann namens Jack. Er lebte mit seiner Mutter auf einem kleinen Bauernhof am Fuße der Nebelberge.	Chat GPT	5	5	5	4	3	5	5	32
	Vor einem jungen gab es noch einmal ein jungen Mann. er lebte bei einem kleinen Bauernhof am fußten mit	Modelo PyTorch	2	1	1	2	2	0	1	9
The fact that Germany and France do not agree on all aspects of energy-related issues – for example the role of nuclear energy – does not prevent them from engaging in a close	Dass sich Deutschland und Frankreich nicht in allen Aspekten der Energiefragen einig sind – beispielsweise über die Rolle der Kernenergie –, hindert sie nicht daran, einen engen Dialog zu führen.	Google	5	4.5	5	5	5	5	5	34.5
	Die Tatsache, dass Deutschland und Frankreich nicht in allen Aspekten	Chat GPT	5	5	5	5	4	4	4	32

dialogue.	energiebezogener Fragen übereinstimmen - beispielsweise in Bezug auf die Rolle der Kernenergie - hindert sie nicht daran, einen engen Dialog zu führen.									
	die Tatsache dieses Deutschland sind in allen Punkten der Frankreich - zum Beispiel gibt es die rolle Hauptstadt? - sie verhindern sich nicht vom Heben einer <UNK>.	Modelo PyTorch	1	0	2	1	1	0	1	6

Nota: Tabla para evaluar la calidad de traducciones generadas con aplicaciones de traducción; consta de oraciones ejemplares de las fuentes (Frase Original) mencionadas a continuación; incluye su traducción realizada por la aplicación de traducción (Frase Traducida) con Google Translate, Chat-GPT y el modelo hecho con PyTorch. Incluida la calificación dada por un experto.

Fragmentos obtenidos de:

- Artículo sobre energía sostenible: <https://www.auswaertiges-amt.de/en/aussenpolitik/-/2621982>
- *Note Left at a U.S. Camptown Brothel for My Missing Imo:* <https://www.lyrikline.org/de/uebersetzerinnen/details/2760/irina-bondas>
- *Jack and the Beanstalk* (Jack y las habichuelas mágicas): <https://www.thefablecottage.com/fables/jack-und-die-bohnenranke>

Tabla 7 Promedio de las calificaciones dadas en las tablas 5 y 6

<i>Modelo</i>	<i>Promedio</i>
<i>Google</i>	33.25
<i>Chat GPT</i>	34
<i>Modelo PyTorch</i>	12

Al igual que en las tablas 2, 3 y 4, donde se evaluó la fidelidad de varias oraciones traducidas, tras hacer una calificación cuantitativa más detallada acerca de la calidad de las traducciones realizadas por estos modelos, se puede observar resultados similares. El modelo con las mejores calificaciones es el de Chat GPT, seguido por Google y al final el modelo realizado con PyTorch. Esto fundamenta lo que se encontró inicialmente con la evaluación de la fidelidad de las traducciones; que al ser un modelo de lenguaje, Chat GPT posee cierta capacidad de entender contexto y poder empatizar más con el objetivo para quién traduce, lo cual está ausente en modelos de NMT. Aunque los tres modelos trabajen con mecanismos de atención, se puede observar que un modelo a base de transformadores provee traducciones de mejor calidad a través de diferentes niveles de complejidad en las traducciones.

El segundo evaluador fue más riguroso con su calificación, pero ambos concuerdan que Chat GPT en general produce traducciones de mejor calidad. Google obtuvo una calificación menor a la esperada, pero cercana a la de Chat GPT. Interesantemente, para el segundo evaluador Google produjo una mejor traducción para la 2da y 3ra

oración. Principalmente porque calificó más bajo a Chat GPT en los parámetros de Tenor (2da y 3ra Oración), Poder y Solidaridad (3ra Oración) con respecto a Google.

4.4. Análisis del impacto que tienen los softwares de traducción IA sobre la profesión de traducción y sus implicaciones

En total, se entrevistó a 7 personas, de las cuales fueron 3 traductores oficiales y 4 estudiantes de Lingüística con enfoque en traducción. En el apartado de evaluación de las traducciones generadas con los modelos evaluados se puso como primero la traducción generada por el modelo con PyTorch, luego con Chat GPT y finalmente con Google Translate. No se informó al inicio de la entrevista que las traducciones evaluadas fueron hechas con IA, pero varios traductores comentaron que las traducciones eran bastante rígidas, muy literales o hasta artificiales. Generalmente las traducciones peor evaluadas fueron las que estaban generadas con el modelo con PyTorch, obteniendo calificaciones de 1 y 2.

Frase Original: There was once a young man named Jack. He lived with his mother on a small farm at the foot of the Foggy Mountains. Frase Traduci...r lebte bei einem kleinen Bauernhof am fußten mit.

7 responses

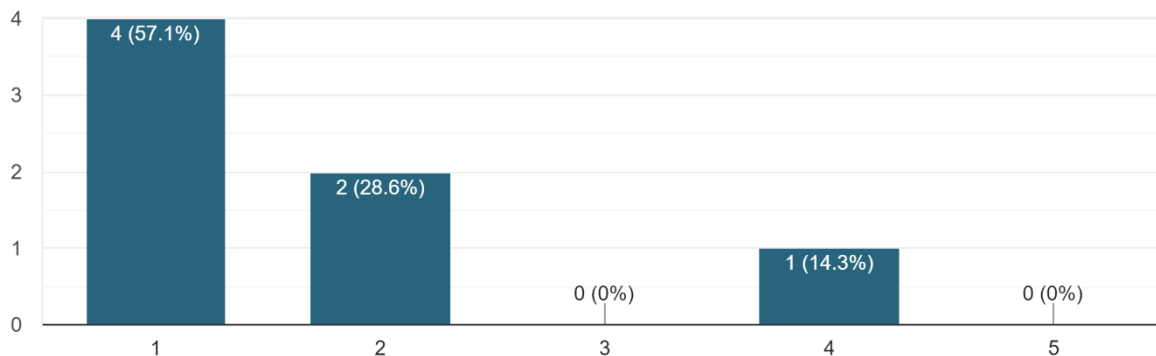


Ilustración 29 Primera pregunta del apartado de evaluación de la traducción presentada, esta fue hecha con el modelo con PyTorch

El modelo que mejor rindió acorde a las entrevistas fue el de Chat GPT, obteniendo mayoritariamente calificaciones de 5.

Frase Original: He ran home and showed his mother the beans. But she was not happy, instead she was furious. "You foolish boy! You traded our only...inzige Kuh gegen eine Handvoll Bohnen getauscht!"

7 responses

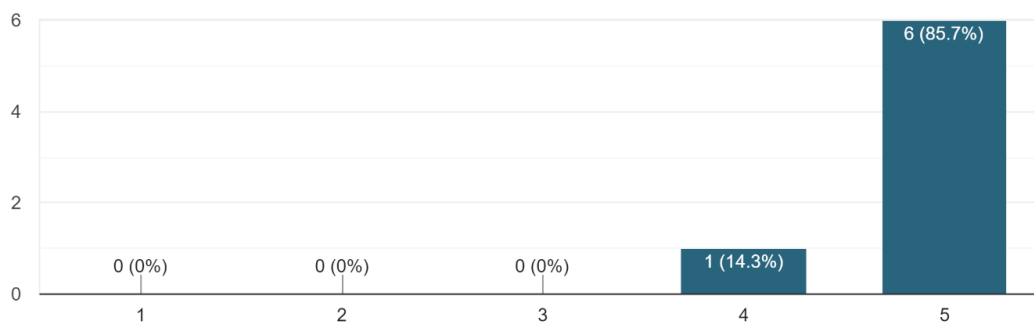


Ilustración 30 Quinta pregunta del apartado de evaluación de las traducciones, esta fue hecha con Chat GPT

Frase Original: Nobody asked where her money came from. They knew where youngest daughters disappeared to Frase Traducida: Niemand fragte...ussten, wohin die jüngsten Töchter verschwanden.

7 responses

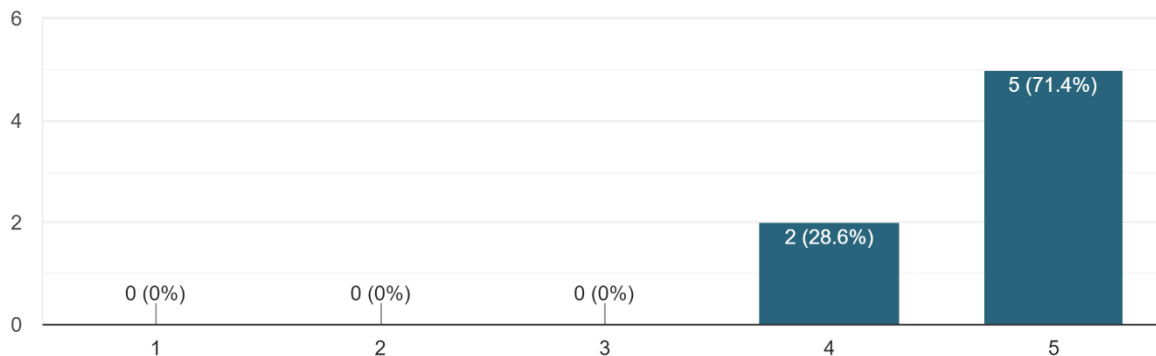


Ilustración 31 Octava pregunta del apartado de evaluación, traducción generada con Chat GPT

En cuanto a Google Translate, las calificaciones fueron más mezcladas. Obtuvo entre 3 y 5, como se observa a continuación:

Frase Original: He ran home and showed his mother the beans. But she was not happy, instead she was furious. "You foolish boy! You traded our only...ige Kuh gegen eine Handvoll Bohnen eingetauscht!"

7 responses

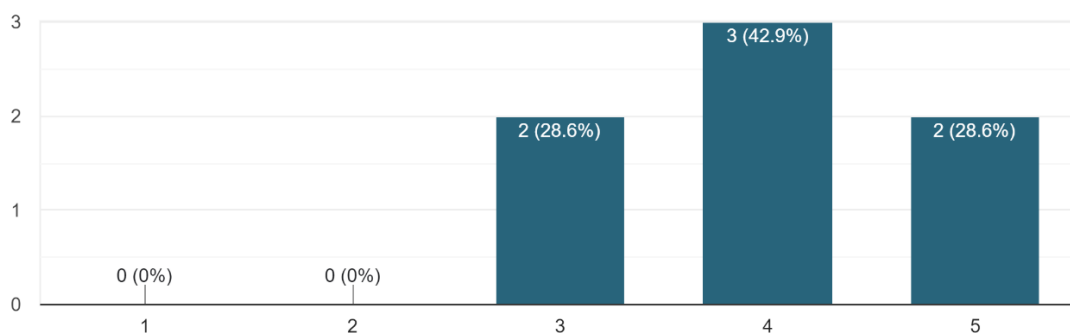


Ilustración 32 Sexta pregunta del apartado de evaluación, traducción hecha por Google Translate

4.4.2. Aceptación social

Al revelar que las traducciones fueron hechas con IA, la recepción fue positiva, sólo en un caso una traductora oficial respondió negativamente. Expresaron que las herramientas de IA para traducción son de gran ayuda para ellos, hasta sorprendiéndose de lo fiel que pueden llegar a ser dichas traducciones al original, pero que no reemplazan a un traductor humano. Un entrevistado mencionó limitaciones de los traductores automáticos, la comprensión de conceptos más abstractos como ironía, cultura, dialectos, entre otros. Otro comentó que una limitación principal es que traducen de manera muy literal, especialmente el traductor de Google.

Opina que el impacto de la IA sobre la traducción es:

7 responses

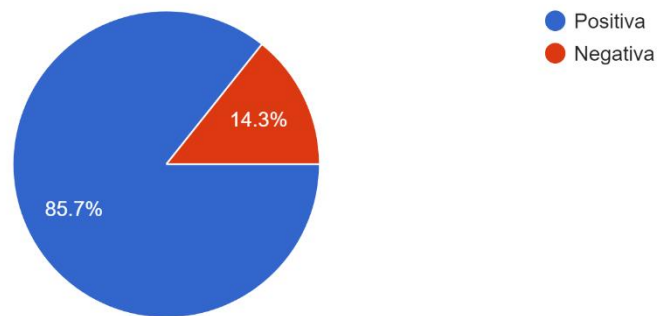


Ilustración 33 Pregunta 5 del tercer apartado

Generalmente, las aplicaciones evaluadas son vistas como herramientas o un apoyo al momento de hacer una traducción, varios entrevistados recalcan que no es un reemplazo para el traductor humano. Comentaron que lo utilizan de manera referencial,

para apoyar con términos más técnicos o especializados, buscar sinonimia, reducir el tiempo que lleva traducir y hasta como diccionario. Como tal, todos los traductores entrevistados concuerdan que estas herramientas han facilitado de una manera u otra su trabajo.

¿ Considera que los traductores en línea han facilitado o dificultado su trabajo de alguna forma?

7 responses

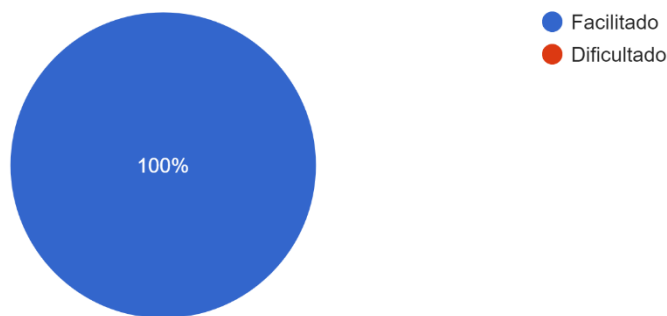


Ilustración 34 Pregunta 7 del tercer apartado

Los beneficios más comunes que comentaron fueron: velocidad de consulta, acelera el trabajo de traducir documentos y economización de tiempo. Aunque sus beneficios no eran muy aparentes al inicio, algunos entrevistados recalcaron que antes les defraudaba o lo vieron como amenaza a su profesión; pero ahora ha ayudado bastante en la comprensión de términos desconocidos de diversos temas y para poder traducir de manera más precisa y más rápido. Interesantemente, un traductor oficial comentó que actualmente las IA en general están sobrevaloradas, y aunque esté a favor de ellas, considera que se debe desarrollar una cultura para su uso adecuado, citando un

caso de abogados estadounidenses que arruinaron sus carreras por usar Chat GPT de manera inapropiada.

Un aspecto interesante para notar es que, a pesar de que Chat GPT haya obtenido la calificación más alta, es la que menos es usada por los traductores entrevistados.

¿Qué traductores en línea utiliza en el día a día para realizar su trabajo?

7 responses

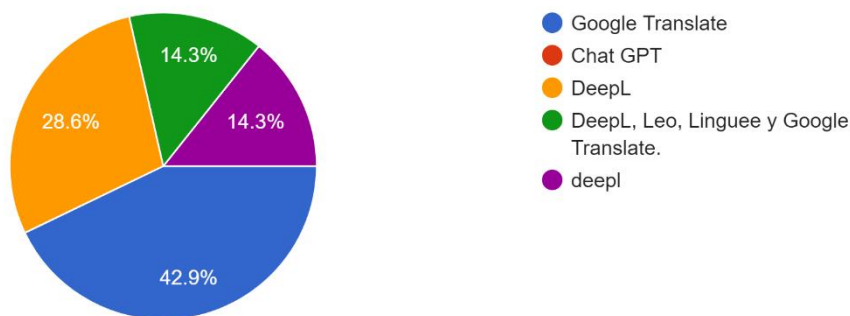


Ilustración 35 Tercera pregunta del tercer apartado

Google Translate y DeepL son las herramientas más populares. DeepL es un diccionario de traducción en línea que presenta traducciones de términos a lado de unos ejemplos de usos, fue igualmente entrenado utilizando Deep Learning, pero no se incluyó en este estudio como modelo al no tener la capacidad de traducir oraciones completas. Tomando en cuenta los usos que dan los traductores a este tipo de herramientas, no es sorprendente encontrar que esta herramienta sea popular, por la facilidad de consultar terminología y usos que tiene la palabra en ambos idiomas. Aunque, es evidente que LLM como Chat GPT no son muy conocidos por la comunidad de traductores; al consultar informalmente con los entrevistados, comentaron que

desconocían de herramientas como Bard y Copilot. Muchos no consideraron utilizar estos LLM como herramientas de traducción.

En cuanto a la validez legal de estas traducciones, un entrevistado que también trabaja como abogado recalcó que traductores tienen una responsabilidad civil; presentar una traducción oficial hecha completamente con AI, y sin firma de responsabilidad humana, no va a ser válida, más aún si esta va a ser presentada ante autoridades en el exterior. Dependiendo del caso, hay que observar también formalidades adicionales. En Ecuador, por ejemplo, las traducciones oficiales requieren por lo general del reconocimiento de firma del traductor ante notario; para trámites en Alemania deben ser realizadas por un Traductor Oficial autorizado por la Embajada, entre otros. Como tal, la credibilidad del traductor es un aspecto muy importante al momento de presentar una traducción oficial, por lo que presentar una traducción de calidad es esencial; es necesario hacer múltiples revisiones y ediciones a los documentos traducidos. Las traducciones actualmente hechas por los modelos evaluados no son 100% acertadas y pueden llevar a costos ingentes para el cliente si se traduce algún término de manera errónea. Por lo que, en el estado actual de dichos modelos, no se aceptaría por parte de las autoridades una traducción hecha netamente con una de estas herramientas.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

En este estudio se encontró que, las traducciones más fieles a las originales resultaron ser las realizadas por Chat GPT con GPT-3.5, teniendo una puntuación de 37 sobre 50 (referirse a la tabla 3 para más detalles). A pesar de no ser especializado para la traducción, este modelo de lenguaje logró superar las expectativas. Esto se debe principalmente a dos aspectos: el hecho que tiene acceso a recursos actuales del internet y el ser un LLM. Al tener los recursos de internet puede referenciar a varias versiones del texto puesto a traducir y/o hasta encontrar las traducciones esperadas, por lo cual tiene una ventaja sobre Google y modelos hechos con la librería de PyTorch (cuyos modelos son NMT) los cuales no se actualizan tan frecuentemente como GPT-3.5 o GPT-4. En cuanto a las ventajas por ser un LLM, Chat-GPT puede (de manera limitada) aplicar la pragmática y sensibilidad a la traducción que realiza, por lo tanto mejorando su calidad como traductor. Tanto Google Translate como el modelo generado con PyTorch no consideran la pragmática ni son sensibles al contexto en el cual traducen, por lo que Chat GPT logró ser mejor, a pesar de tener una calificación BLEU más baja en sus fases de prueba.

En cuanto a la creación de una RNN con PyTorch; se logró realizar NMT con un modelo que esencialmente es una versión reducida del modelo de Google. Aunque ambos utilizaran la misma célula de LSTM, se puede observar claramente una desventaja de utilizarla, que es el hecho de que requiera grandes volúmenes de datos para entrenarse apropiadamente. Se encontró que en el modelo hubo sobreajuste con

231k pares de oraciones, y al tener un vocabulario limitado, no se adaptó bien a traducir oraciones más complejas, al simplemente no tener una referencia para traducir ciertos términos como el de “energía renovable”. Las traducciones resultaron muy literales en ciertos casos, pero acertaron mucho más frecuentemente para oraciones con sintaxis simples. En sí, aunque el modelo con PyTorch rindió peor de lo esperado, fue impresionante probar con las capacidades de las células LSTM y trabajar con lo que hizo en esencia Google. El poner en práctica esta teoría aclaró bastantes dudas personales acerca de las diferencias entre GRU, LSTM y el funcionamiento de RNNs.

En términos de ventajas y desventajas, con las tablas se logró encontrar que efectivamente, una gran desventaja de modelos basados en NMT es la cantidad de datos disponibles, requiriendo un gran volumen de datos para lograr ser competitivo con los otros modelos evaluados. La desventaja que tuvo el modelo realizado con PyTorch en cambio es la ventaja de Chat GPT; su gran cantidad de datos de entrenamiento en conjunto de la capacidad de transformadores permitió que este modelo tenga una capacidad impresionante de traducción que sorprendió incluso a los traductores entrevistados. Aunque, una desventaja clara de estos modelos, que también fue mencionada en las entrevistas, es que estos modelos no son confiables al momento de realizar una traducción completa; es necesario el criterio de un traductor para determinar si la traducción proveída por la herramienta le resulta útil o acertada.

Desde la perspectiva de los usuarios, según lo encontrado en las entrevistas, las principales ventajas de estos modelos yacen en su habilidad de funcionar como diccionarios o corroborar alguna traducción, especialmente en casos donde requieran hacer una traducción con terminología técnica. Tanto estudiantes como profesionales

los utilizan de la misma forma, y ambos concuerdan que una desventaja actual de estos modelos es que sus traducciones son muy literales, no consideran al contexto, ni semántica, etc. Como tal, los usuarios de estas herramientas han visto más ventajas que desventajas en usar estos modelos y han tenido un impacto positivo en su trabajo y estudios, economizando el tiempo que requieren para ello.

Como visto en las evaluaciones de fidelidad y calidad de las traducciones realizadas por los modelos estudiados, Chat GPT produjo las traducciones de mejor calidad, puesto que los expertos calificaron sus traducciones casi con una puntuación perfecta; a pesar de ser la herramienta menos utilizada y conocida, como lo revelaron las entrevistas con los traductores. Muchos de los traductores, de hecho, conocieron y observaron las capacidades de traducción que tiene Chat GPT con este estudio. Google Translate ha consolidado su presencia como herramienta de traducción para los traductores, tanto estudiantes como profesionales, aunque vale recalcar que no desconocían totalmente de herramientas para la traducción potenciadas con modelos de Deep Learning, puesto que la segunda herramienta más usada es Linguee que utiliza DeepL.

No se puede negar el impacto que han tenido las herramientas para traducción basadas en IA en el campo de la Traducción, pero es importante recalcar que toman netamente un rol referencial, más no reemplazan a un traductor humano. Modelos de NMT carecen de la comprensión de semántica, sensibilidad y poder. En contraste, un LLM como Chat GPT demostró una cierta comprensión de estos parámetros y logró dar una traducción equivalente a la de un profesional, pero no totalmente satisfactoria para traductores actuales. La comunidad de traductores ha aceptado estas herramientas con

el tiempo, todos los entrevistados confirmaron que estas herramientas han facilitado su trabajo y aunque ha mostrado cierta resistencia a ella al inicio, rápidamente le dieron el uso que se observa ahora.

Es posible que, con el auge de los LLM, incluso herramientas como Google Translate se vean opacadas o forzadas a cambiar de modelo para satisfacer las necesidades de traductores modernos. Y en cuanto a las implicaciones éticas de estas herramientas, tal como lo mencionó un entrevistado con el caso de los abogados (video del caso disponible en anexos), su uso ético dependerá del uso que lo den los usuarios futuros; si se mantiene netamente como herramienta referencial, no surgirían dilemas éticos como los que se han dado recientemente con gremios de escritores y artistas o en temas de plagio. En otras palabras, si se utiliza de manera responsable la capacidad que tienen las IA para traducción, seguirán facilitando el trabajo de los traductores, mas no complicándolo.

Recomendaciones

En calidad de recomendaciones, tal como lo han recalcado los traductores entrevistados, estas herramientas deberían ser utilizadas de manera referencial y no son un reemplazo para traducciones oficiales o de calidad. El criterio de un profesional es clave para maximizar el uso de dichas herramientas. Para la comunidad de traductores recomendaría explorar los modelos de LLM emergentes e incorporarlos a su trabajo diario como herramienta alternativa (o complementaria) a Google Translate y DeepL.

Igualmente, para estudios futuros en este tema, sería interesante observar como Bard, BERT, CTRL, Bing Chat, Copilot, entre otros rinden en la comparación de uno con otro.

Finalmente, si se desea experimentar con la creación de modelos de NMT personalmente, es de suma importancia considerar el hardware disponible, la compatibilidad de diferentes librerías y el uso de GPU para el entrenamiento de estos modelos. No es recomendable entrenar estos modelos complejos y masivos en CPU, puesto que desperdicia recursos y tiempo. Igualmente es importante asegurarse que sea posible conseguir la mayor cantidad de datos posibles para obtener los mejores resultados.

BIBLIOGRFÍA

10.6. *The Encoder–Decoder Architecture — Dive into Deep Learning 1.0.3 documentation.*

(n.d.). https://d2l.ai/chapter_recurrent-modern/encoder-decoder.html

American Translators Association (ATA). (2022, June 27). *Machine Translation - American*

Translators Association (ATA). <https://www.atanet.org/client-assistance/machine-translation/>

American Translators Association (ATA). (2022a, June 6). *Fidelity In Translation.*

<https://www.atanet.org/translation/fidelity-in-translation/>

Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural machine translation by jointly learning*

to align and translate. *arXiv preprint arXiv:1409.0473.*

Bidirectional recurrent neural networks. (1997, November 1). IEEE Journals & Magazine |

IEEE Xplore. <https://ieeexplore.ieee.org/abstract/document/650093>

Brown, T. B. (2020, May 28). *Language Models are Few-Shot Learners.* arXiv.org.

<https://arxiv.org/abs/2005.14165>

Chowdhary, K. R. (2020). Natural language processing. In *Springer eBooks* (pp. 603–649).

https://doi.org/10.1007/978-81-322-3972-7_19

Crystal, D. (2000). *Diccionario de lingüística y fonética.*

Definición de Cross-Entropy. (n.d.). [https://www.lokad.com/es/definicion-de-entropia-](https://www.lokad.com/es/definicion-de-entropia-cruzada/)

[cruzada/](https://www.lokad.com/es/definicion-de-entropia-cruzada/)

Gate-variants of Gated Recurrent Unit (GRU) neural networks. (2017, August 1). IEEE

Conference Publication | IEEE Xplore.

<https://ieeexplore.ieee.org/abstract/document/8053243>

- Ghaffarian, S., Valente, J., Van Der Voort, M., & Tekinerdogan, B. (2021). Effect of Attention Mechanism in Deep Learning-Based Remote Sensing Image Processing: A Systematic Literature review. *Remote Sensing*, 13(15), 2965. <https://doi.org/10.3390/rs13152965>
- Grossberg, S. (2013). Recurrent neural networks. *Scholarpedia*, 8(2), 1888. <https://doi.org/10.4249/scholarpedia.1888>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Lanners, Q. (2022, March 30). Neural Machine Translation - towards Data science. *Medium*. <https://towardsdatascience.com/neural-machine-translation-15ecf6b0b>
- La Torre Jordi, D. (2023, February 18). *Redes Generativas Adversarias (GAN) Fundamentos Teóricos y Aplicaciones*. arXiv.org. <https://arxiv.org/abs/2302.09346>
- Munday, J. (2012). *Evaluation in translation: Critical Points of Translator Decision-making*. Routledge.
- Neural networks*. (1995). Google Books. <https://books.google.com/books?hl=en&lr=&id=EFUzMYjOXk8C&oi=fnd&pg=PA3&dq=neural+network&ots=WkDvQIOrYB&sig=Bdxs6FGwHma0SyJIBWyeHL3pA3Y>
- O'Shea, K. (2015, November 26). *An introduction to convolutional neural networks*. arXiv.org. <https://arxiv.org/abs/1511.08458>
- Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (pp. 311-318).
- Ruiz, I. V. M. (2016, November 21). *Descenso por gradiente (Gradient descent)*. https://turing.iimas.unam.mx/~ivanvladimir/posts/gradient_descent/

Sánchez, G. B. (2021). La traducción profesional y sus beneficios. *Lionbridge*.

[https://www.lionbridge.com/es/blog/translation-localization/la-traduccion-profesional-y-sus-](https://www.lionbridge.com/es/blog/translation-localization/la-traduccion-profesional-y-sus-beneficios/#:~:text=%C2%BFQu%C3%A9%20es%20la%20traducci%C3%B3n%20profesional,garant%C3%ADa%20de%20satisfacci%C3%B3n%20del%20cliente)

[benefi-](https://www.lionbridge.com/es/blog/translation-localization/la-traduccion-profesional-y-sus-beneficios/#:~:text=%C2%BFQu%C3%A9%20es%20la%20traducci%C3%B3n%20profesional,garant%C3%ADa%20de%20satisfacci%C3%B3n%20del%20cliente)

[cios/#:~:text=%C2%BFQu%C3%A9%20es%20la%20traducci%C3%B3n%20profesional,garant%C3%ADa%20de%20satisfacci%C3%B3n%20del%20cliente](https://www.lionbridge.com/es/blog/translation-localization/la-traduccion-profesional-y-sus-beneficios/#:~:text=%C2%BFQu%C3%A9%20es%20la%20traducci%C3%B3n%20profesional,garant%C3%ADa%20de%20satisfacci%C3%B3n%20del%20cliente).

Vaswani, A. (2017, June 12). *Attention is all you need*. arXiv.org.

<https://arxiv.org/abs/1706.03762>

What are Large Language Models? | NVIDIA Glossary. (2023). NVIDIA.

<https://www.nvidia.com/en-us/glossary/data-science/large-language-models/>

What are Recurrent Neural Networks? | IBM. (n.d.). [https://www.ibm.com/topics/recurrent-](https://www.ibm.com/topics/recurrent-neural-networks)

[neural-networks](https://www.ibm.com/topics/recurrent-neural-networks)

What is Machine Translation? - Everything you need to know - AWS. (n.d.). Amazon Web

Services, Inc. <https://aws.amazon.com/what-is/machine-translation/>

Wu, Y. (2016, September 26). *Google's Neural Machine Translation System: Bridging the*

Gap between Human and Machine Translation. arXiv.org.

<https://arxiv.org/abs/1609.08144>

ANEXOS

6.1. Anexo A:

Tutorial para el desarrollo de un modelo de NMT con PyTorch:

<https://towardsdatascience.com/neural-machine-translation-15ecf6b0b>

6.2. Anexo B:

Ejecución del modelo creado con el tutorial: [rntranslation-v2.pdf](#)

6.3. Anexo C:

Formulario de Google Forms utilizado para realizar la entrevista:

<https://forms.gle/6UVt7CDFynVTA9fY6>

6.4. Anexo D:

Recopilación de las respuestas de los entrevistados:

<https://docs.google.com/spreadsheets/d/1dPlmRra4gMPSemwvIY0Gp14BBE8fqQm-aDEs0lixv38/edit?usp=sharing>

6.5. Anexo E:

Página de donde se descargó el dataset para entrenar la red:

<https://www.manythings.org/anki/>

6.6. Anexo F:

Caso de abogados despedidos por mal uso de Chat GPT:

https://youtu.be/oqSYIjRYDEM?si=wASYQK_2rxtnqXXq